**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**

**KHOA CÔNG NGHỆ THÔNG TIN**

-------------------

**BÁO CÁO ĐỒ ÁN**

**MẠNG MÁY TÍNH**

**Đề tài: Lập trình socket và bắt gói tin bằng wireshark.**

**Giáo viên lý thuyết: Lê Giang Thanh**

**Giáo viên thực hành: Nguyễn Thị Thanh Huyền**

**Thành viên nhóm**

**Nguyễn Tấn Tài – 19127265**

**Nguyễn Quang Huy – 19127161**

**Nguyễn Thái Bình - 19127103**

**Lớp : 19CLC4**

TP. Hồ Chí Minh, ngày 10 tháng 5 năm 2021

**Mục lục**

Phần 1: Cách xây dựng protocol ……………………………………………...3

* Kịch bản chương trình ………………………………………………3
* Cấu trúc chương trình ……………………………………………….5
* Giao thức thực hiện …………………………………………...6
* Các hàm chính ………………………………………………...6
* Các kiểu cấu trúc ……………………………………………...7

Phần 2: Bắt và phân tích gói tin ………………………………………………8

**Phần 1: Cách xây dưng protocol**

● **Kịch bản chương trình**:

Chương trình cho phép người dùng đăng nhập, đăng kí tài khoản. Sau khi đã đăng nhập người dùng có thể thay đổi mật khẩu, kiểm tra thông tin, sửa đổi thông tin, tải lên (tải xuống) các tài liệu lên server và có thể tạo các nhóm chat với người dùng khác.

**◦ Chức năng đăng nhập:**

**+ Phía Client**: Cho phép người dùng nhập tên đăng nhập, gửi tên đăng nhập lên server và server gửi về yêu cầu nhập mật khẩu và hỏi người dùng có mã hoá hay không, sau đó yêu cầu người dụng nhập mật khẩu và tiếp tục gửi lên server.

**+ Phía Server**: Tiến hành nhận thông tin đăng nhập từ phía client và kiểm tra.

-> Kiểm tra mật khẩu có mã hoá không (nếu có thì giải mã)

-> Kiểm tra người dùng đã đăng nhập hay chưa (nếu đã đăng nhập thì thông báo về phía client và không thực hiện gì thêm)

-> Kiểm tra trong cơ sở dữ liệu của server, kiểm tra tên đăng nhập và mật khẩu (nếu đúng thì xác nhận người dùng đã đăng nhập thành công và thông báo về phía client, nếu sai thì thông báo về phía client việc đăng nhập không thành công)

**◦ Chức năng đăng kí:**

**+ Phía Client**: Cho phép người dùng nhập tên đăng nhập, gửi tên đăng nhập lên server và server gửi về yêu cầu nhập mật khẩu và hỏi người dùng có mã hoá hay không, gửi mật khẩu lên server và server sẽ yêu cầu nhập các thông tin thêm như ngày sinh, chú thích, nickname (khi nhập xong 1 thông tin thì gửi lên server và server sẽ yêu cầu nhập thông tin tiếp theo)

**+ Phía Server**: Tiến hành nhận thông tin đăng kí từ phía client và kiểm tra.

-> Kiểm tra mật khẩu có mã không (nếu có thì giải mã)

-> Kiểm tra người dùng đã đăng nhập hay chưa (nếu đã đăng nhập thì thông báo về phía client và không thực hiên gì thêm)

-> Kiểm tra trong cơ sở dữ liệu của server, nếu tên đăng nhập chưa tồn tại, tiến hành tạo mới người dùng trong cơ sở dữ liệu và thông báo về phía client (nếu tên đăng nhập đã tồn tại thì thông báo về phía client đăng kí không thành công)

**◦ Chức năng thay đổi mật khẩu:**

**+ Phía Client**: Cho phép người dùng nhập mật khẩu cũ, sau đó gửi mật khẩu lên server và server sẽ yêu cầu nhập mật khẩu mới. Cho phép người dùng nhập mật khẩu mới và gửi lên server. (hỏi người dùng có muốn mã hoá mật khẩu không (cả mật khẩu cũ và mới))

**+ Phía Server**: Nhận thông tin đổi mật khẩu từ client và tiến hành kiểm tra

-> Kiểm tra mật khẩu cũ và mật khẩu mới có mã hoá hay không (nếu có tiến hành giải mã)

-> Kiểm tra trong cơ sở dữ liệu của server, nếu tên đăng nhập và mật khẩu cũ hợp lệ thì tiến hành cập nhật mật khẩu mới và thông báo về phía client (nếu không hợp lệ thì thông báo về phía client là không thành công và không chỉnh sửa cơ sở dữ liệu)

**◦ Chức năng kiểm tra thông tin người dùng:**

**+ Phía Client**: Cho phép người dùng nhập username và các trường muốn kiểm tra thông tin, gửi lên server

**+Phía Server**: Nhận tên đăng nhập và các trường từ phía client và tiến hành kiểm tra thông tin.

-> Kiểm tra tên đăng nhập có trong cơ sở dữ liệu hay không (nếu không thì gửi thông báo về phía client và không làm gì thêm)

-> Nếu tên đăng nhập hợp lí thì lấy các thông tin tương ứng trong cơ sở dữ liệu và gửi về phía client

**◦ Chức năng thay đổi thông tin người dùng:**

**+ Phía Client:** Cho phép người dùng nhập trường thông tin muốn thay đổi và giá trị mới của trường đó, sau đó gửi thông tin lên server

**+ Phía Server**: Nhận thông tin trường thay đổi và giá trị mới của trường đó từ client và tiến hành cập nhật lại thông tin trong cơ sở dữ liệu.

**◦ Chức năng tải lên tài liệu:**

**+ Phía Client**: Cho phép người dùng nhập tên file (1 hoặc nhiều file), và theo tác với file (nếu là đổi tên file thì sẽ yêu cầu nhập tên mới của file), tiến hành gửi thông tin lên cho server và hỏi người dùng có muốn mã hoá dữ liệu khi gửi file hay không, , nếu yêu cầu là gửi 1 file lên server thì client sẽ tiến hành đọc mỗi 2048 bytes của file và gửi lên server (nếu có mã hoá thì sẽ mã hoá dữ liệu sau đó gửi đi), nếu yêu cầu là gửi nhiều file lên server, client sẽ tiến hành đọc mỗi 2048 bytes của 1 file và gửi lên server, sau đó client sẽ chờ tín hiệu từ server xác nhận đã nhận xong 1 file thì sẽ gửi file tiếp theo đi cho đến khi hết.

**+ Phía Server**: Nhận thông tin tên file và thao tác từ phía client và tiến hành xử lí.

-> Nếu thao tác là đổi tên file: tiến hành cập nhật lại tên file trong cở sở dữ liệu và thông báo về phía client.

-> Nếu là tải lên nhiều file: nhận tên các file và tiến hành nhận dữ liệu của các file từ client. Kiểm tra xem dữ liệu có mã hoá hay không (nếu có thì giải mã). Sau khi nhận hết dữ liệu của 1 file, server sẽ tiến hành tạo file và ghi dữ liệu vào file. Sau khi ghi dữ liệu hoàn thành, sẽ gửi thông báo về phía client là đã nhận thành công 1 file và sẵn sàng nhận file tiếp theo. Sau khi nhận thành công tất cả các file, server sẽ gửi thông báo về phía client là đã nhận thành công toàn bộ.

**◦ Chức năng tải về tài liệu:**

**+ Phía Client**: Cho phép người dùng nhập các tên file cần tải về va hỏi người dùng có mã hoá không, sau đó gửi dữ liệu lên server. Tiến hành nhận dữ liệu của file từ server gửi về (nếu có mã hoá thì sẽ giải mã), sau khi nhận xong 1 file sẽ thông báo lên server để nhận file tiếp theo cho đến khi hết

**+ Phía Server**: Nhận các tên file từ client. Mở các file tương ứng ở server và tiến hành gửi nội dung file về phía client (nếu có mã hoá thì mã hoá sau đó gửi dữ liệu đi), sau khi gửi xong 1 file sẽ chờ tín hiệu thông báo từ phía client là đã nhận thành công và tiến hành gửi file tiếp theo. Sau khi gửi hết tất cả file sẽ thông báo về client là đã tải xuống thành công.

**◦ Chức năng phòng chat:**

**+ Phía Client**: cho phép người dùng nhập dữ liệu (kiểm tra người dùng trực tuyến hoặc tạo phòng chat), sau đó gửi dữ liệu lên server, chờ thông điệp gửi về từ server. Nếu là thông điệp tạo phòng chat, sẽ tiến hành xoá màn hình console để tham gia vào phòng chat

**+ Phía Server**: nhận dữ liệu từ phía client và xử lí:

-> Nếu là kiểm tra người dùng trực tuyến thì truy xuất lấy các người dùng đang trực tuyến và trả về phía client

-> Nếu là tạo room chat sẽ gửi thông điệp thông báo tham gia phòng chat tới tất cả các thành viên trong phòng

**● Cấu trúc chương trinh:**

**Giao thức thực hiện: TCP**

**Các hàm chính:**



Dùng để gửi thông tin đến tất cả các người dùng đang kết nối đến server

Tham số đầu vào là tin nhắn của người dùng và người dùng đang gửi tin nhắn đó



Dùng để gửi tin nhắn đến tất cả các người dùng trong phòng chat

Tham số đầu vào là tất cả người dùng trong phòng chat, tin nhắn của người gửi và người gửi tin nhắn đó



Dùng để xác thực người dùng

Tham số đầu vào là tên đăng nhập và mật khẩu

Kết quả trả về là:

- 0 nếu tên đăng nhập và mật khẩu chính xác

- 1 nếu đúng tên đăng nhập và sai mật khẩu

- 2 nếu tên đăng nhập không có trong cơ sở dữ liệu



Dùng để kiểm tra người dùng có trong cơ sở dữ liệu hay không

Tham số đầu vào là tên đăng nhập

Kết quả trả về True nếu người dùng có trong cơ sở dữ liệu, ngược lại trả về False



Dùng để lấy ngày sinh của người dùng

Tham số đầu vào là tên đăng nhập

Kết quả trả về là ngày sinh của người dùng (dạng chuỗi)





Tương tự như hàm getDateOfBirth()



Dùng để ghi thông tin của người dùng vào cơ sở dữ liệu

Tham số nhận vào là tên đăng nhập, mật khẩu, ngày sinh, chú thích và biệt danh của người dùng



Dùng để thay đổi mật khẩu người dùng

Tham số nhận vào là tên đăng nhập, mật khẩu cũ và mật khẩu mới

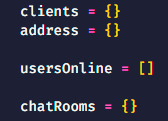






Tương tự như hàm changePassword()

**Các kiểu cấu trúc:**



* Clients là 1 dictionary dùng để lưu tên đăng nhập của các kết nối đến server

+ Key là kết nối đang kết nối đến server

+ Value là tên đăng nhập của kết nối đó

* Address là 1 dictionary dùng để lưu địa chỉ của các client kết nối đến server

+ Key là kết nối đang kết nối đến server

+ Value là địa chỉ IP của client đang kết nối đến

* usersOnline dùng để lưu tên đăng nhập của tất cả người dùng đang kết nối đến server
* chatRoom là 1 dictionary để lưu thông tin của tất cả các phòng chat

+ Key là mã phòng chat

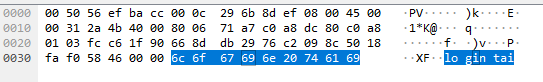
+ Value là 1 mảng chứa tất cả các tên đăng nhập của người dùng đang trong phòng chat đó

**Phần 2: Bắt và phân tích gói in**

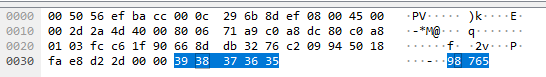
**Chức năng login:**

**Chưa mã hoá:**

Username

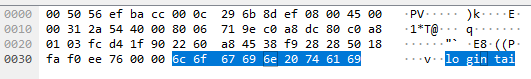


Password

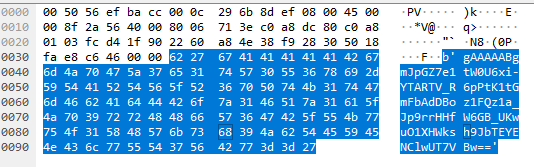


**Đã mã hoá:**

Username



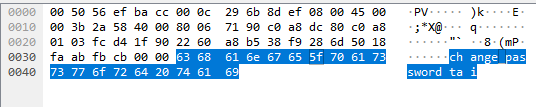
Password



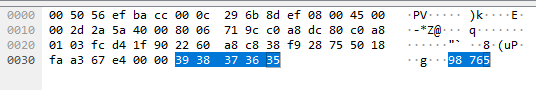
**Chức năng change\_password:**

**Chưa mã hoá**

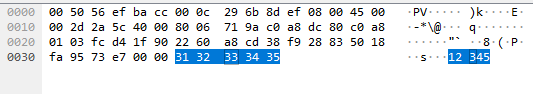
Username:



Password cũ

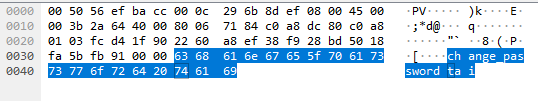


Password mới

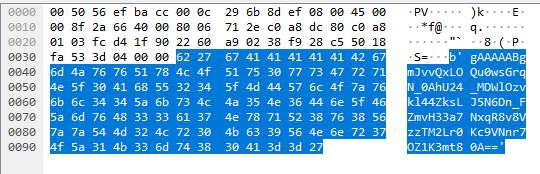


**Đã mã hoá**

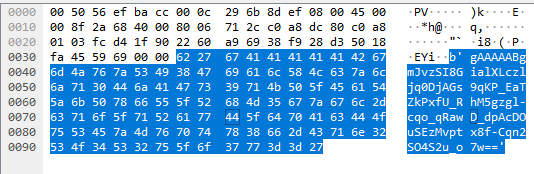
Username



Password cũ



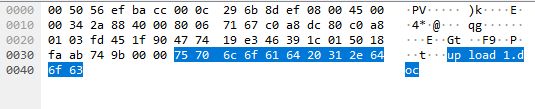
Password mới



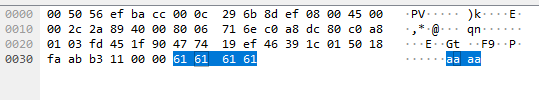
**Chức năng upload file**

**Chưa mã hoá**

Tên file

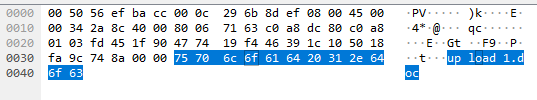


Nội dung



**Đã mã hoá**

Tên file



Nội dung

