

BÀI TẬP TUẦN 6 – IT4062

Bài 1. Sử dụng TCP socket và kỹ thuật đa luồng để viết ứng dụng mã hóa giải mã file đơn giản sử dụng thuật toán mã hóa Ceasar giữa client và server. Giao thức được mô tả như sau:

- Khuôn dạng thông điệp của ứng dụng:

Opcode	Length	Payload
--------	--------	---------

- Opcode (1 byte): Mã thao tác:

- 0: Mã hóa
- 1: Giải mã
- 2: Truyền dữ liệu của file
- 3: Báo lỗi

- Length(2 byte): kích thước của dữ liệu trong Payload

- Payload: Dữ liệu truyền đi.

- Nếu Opcode là 0 và 1 thì Payload chứa giá trị khóa
- Nếu Opcode = 2 và Length > 0 thì Payload chứa dữ liệu của file
- Nếu Opcode = 2 và Length = 0 thì quá trình truyền file hoàn tất
- Nếu Opcode = 3 thì Payload không mang dữ liệu

- Hoạt động của giao thức:

- B1: Client gửi thông điệp yêu cầu mã hóa/giải mã với khóa kèm theo cho server
- B2: Client gửi dữ liệu của file lên server. Nếu quá trình gửi hoàn tất, client gửi thông điệp với Opcode = 2 và Length = 0
- B3: Server lưu dữ liệu nhận được từ client vào file tạm. Server thực hiện mã hóa/giải mã file tạm theo yêu cầu. Nếu có lỗi gửi lại thông điệp với Opcode = 3
- B4: Server gửi dữ liệu của file kết quả cho client. Nếu quá trình gửi hoàn tất, server gửi thông điệp với Opcode = 2 và Length = 0.
- B5: Server xóa file tạm và file kết quả
- *Lưu ý: Để đơn giản quá trình cài đặt giao thức, coi rằng tại mỗi thời điểm, client chỉ gửi 1 file yêu cầu và chờ nhận file kết quả.*

Yêu cầu:

- Server:

- Khởi động với số hiệu cổng là giá trị truyền qua tham số dòng lệnh:

`$/server Port_Number` (Ví dụ: `$/server 5500`)

- Sử dụng một thư mục chung để lưu file của người dùng gửi lên. Lưu ý, người dùng có thể gửi file có tên trùng nhau. Do đó file tạm trên server không nhất thiết phải có tên
- Client:
 - Khởi động với địa chỉ server là các giá trị truyền qua tham số dòng lệnh:


```
$/client IP_Addr Port_Number (Ví dụ: $./client 10.0.0.1 5500)
```
 - Nhận yêu cầu lựa chọn chức năng mã hóa/giải mã, nhập giá trị khóa và tên file từ bàn phím của người dùng
 - Nhận file kết quả trả về từ server

Lưu ý: Tạo Makefile với tên file thực thi sau khi biên dịch là *server* và *client*

Bài 2. Sử dụng TCP socket và kỹ thuật dùng hàm `fork()` để xây dựng ứng dụng đăng nhập, đăng xuất cho người dùng.

- Server khởi động với số hiệu cổng là giá trị truyền qua tham số dòng lệnh:


```
$/server Port_Number (Ví dụ: $./server 5500)
```
- Client khởi động với địa chỉ server là các giá trị truyền qua tham số dòng lệnh có cú pháp như sau:


```
$/client IP_Addr Port_Number (Ví dụ: $./client 10.0.0.1 5500)
```
- Yêu cầu:
 - Mỗi cửa sổ client chỉ đăng nhập được 1 tài khoản
 - Mỗi tài khoản có thể đăng nhập đồng thời trên nhiều cửa sổ
 - Nếu đăng nhập sai quá 3 lần, tài khoản bị khóa
 - Tài khoản người dùng lưu trên file văn bản account.txt, mỗi dòng một tài khoản dạng(xem file ví dụ):

UserID Password Status

Trong đó Status có giá trị 0: Tài khoản bị khóa, 1: Tài khoản hoạt động

Lưu ý: Tạo Makefile với tên file thực thi sau khi biên dịch là *server* và *client*