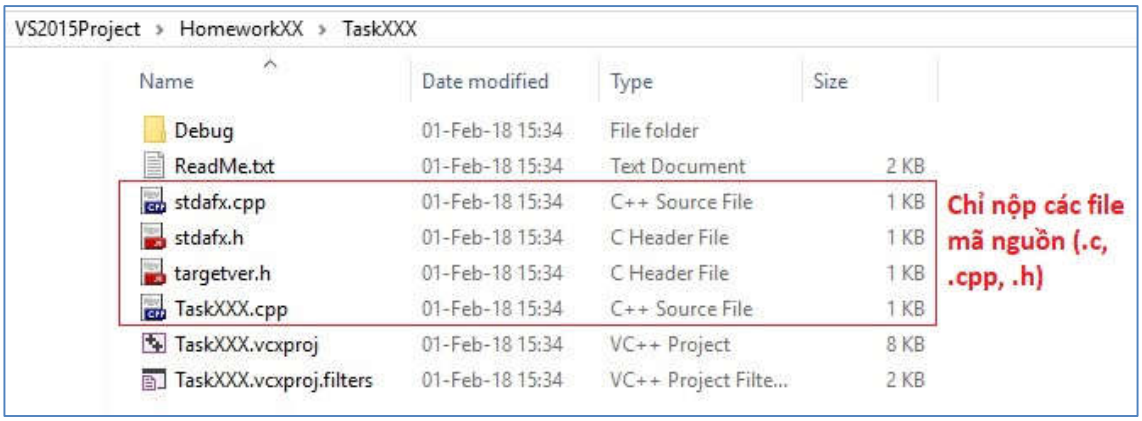
BÀI TẬP TUẦN 6 – IT4060

Cách thức nộp mã nguồn:



Đặt mã nguồn(.c, .cpp, .h) của mỗi Project vào thư mục riêng rẽ có tên thư mục là tên Project. Đóng gói các thư mục này vào file nén có tên theo định dạng HotenSV_MSSV_HW06.zip.

Bài 1*(70%)*. Sử dụng TCP socket và kỹ thuật vào ra theo sự kiện để xây dựng ứng dụng đăng bài cho người dùng.

- Server khởi động với địa chỉ là 127.0.0.1 và số hiệu cổng là 6000
- Client khởi động với địa chỉ server là các giá trị truyền qua tham số dòng lệnh có cú pháp như sau:

  `Client.exe ServerIP ServerPort.`

  Ví dụ: `Task1_Client.exe 10.0.0.1 6000`

Yêu cầu:

- Trên server, tài khoản người dùng lưu trong file văn bản account.txt, mỗi dòng một tài khoản dạng(xem file ví dụ):

  `username status`

  Trong đó giá trị status là 1 nếu tài khoản bị khóa, là 0 nếu tài khoản hoạt động.
- Người dùng cần phải đăng nhập bằng tên tài khoản đã có trên hệ thống trước khi gửi bài đăng. Người dùng có thể gửi nhiều bài đăng trong một phiên.
- Người dùng có thể kết thúc phiên bằng cách gửi yêu cầu đăng xuất hoặc tắt chương trình client.
- Trên mỗi cửa sổ chương trình client, người dùng chỉ đăng nhập được 1 tài khoản
- Mỗi tài khoản có thể được đăng nhập trên nhiều client. *Lưu ý: Khi xử lý yêu cầu đăng xuất từ một client nào, chỉ kết thúc phiên tại client đó.*
- Chương trình client cần phải có giao diện. Không nên để người dùng phải nhập thông điệp yêu cầu dạng thô.

Server cần ghi lại cần ghi lại nhật ký hoạt động vào file có tên theo định dạng log_MSSV.txt (Ví dụ log_20181234.txt). Mỗi dòng có cấu trúc như sau:

ClientIP:ClientPort$ [dd/mm/yyyy hh:mm:ss] $ Thông điệp yêu cầu của client $ Mã kết quả
Trong đó:

dd/mm/yyyy: Định dạng ngày nhận yêu cầu

hh:mm:ss: Định dạng thời điểm nhận yêu cầu

Mã kết quả: Do sinh viên tự định nghĩa. Ví dụ: +OK, -ERR, 200, 404…

Ví dụ:

127.0.0.1:10000 [31/03/2021 14:42:24] $ USER binhnv $ 11
127.0.0.1:10000 [31/03/2021 14:46:24] $ USER tungbt $ 10
127.0.0.1:10000 [31/03/2021 14:48:12] $ USER levn $ 12
127.0.0.1:10000 [31/03/2021 14:48:12] $ POST Hello World $ 20
127.0.0.1:10000 [31/03/2021 14:48:35] $ POST I am superman $ 20
127.0.0.1:10001 [31/03/2021 14:50:01] $ USER tungbt $ 10
127.0.0.1:10000 [31/03/2021 14:50:47] $ QUIT $ 30
127.0.0.1:10000 [31/03/2021 14:51:57] $ QUIT $ 31
127.0.0.1:10000 [31/03/2021 14:52:10] $ USER levn $ 10

Yêu cầu môi trường:

- Công cụ phát triển ứng dụng: Microsoft Visual Studio 2015 Community
- Tên solution: Homework06
- Tên project: Task1_Server và Task2_Client

Gợi ý:

- Xây dựng giao thức với các gợi ý trong bài giảng trên lớp.
- Nên sử dụng thông điệp dạng xâu ký tự. Cần sử dụng dấu báo kết thúc thông điệp và xử lý vấn đề truyền dòng
- Trạng thái đăng nhập trong phiên cần phải được quản lý tại server. Mỗi phiên cung cấp dịch vụ cho 1 client trên server nên được lưu trữ dưới dạng cấu trúc. Cấu trúc nên có các trường thông tin sau*(Lưu ý: Sinh viên có thể lựa chọn cách thiết kế khác)*:
  - Định danh socket mà server sử dụng để kết nối với client của phiên đó
  - Địa chỉ của client
  - Tên tài khoản đã đăng nhập tại client
  - Trạng thái đăng nhập

Tên tài khoản và trạng thái đăng nhập cần được cập nhật khi xử lý thành công các yêu cầu đăng nhập và đăng xuất.

- Sử dụng mảng cấu trúc đã gợi ý *(Lưu ý: Sinh viên có thể lựa chọn cách thiết kế khác)*

Bài 2*(30%)*. Sử dụng TCP socket và kỹ thuật vào ra theo sự kiện để xây dựng ứng dụng mã hóa giải mã file đơn giản sử dụng thuật toán mã hóa dịch vòng 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 tính theo đơn vị byte
  - 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. Khi hoàn tất việc gửi file, client gửi thông điệp với Opcode = 2 và Length = 0.
  - B3: Server lưu dữ liệu nhận được từ client vào file tạm(*Lưu ý: Tên file cần đặt sao cho không bị trùng với file khác*). Server thực hiện mã hóa/giải mã file tạm theo yêu cầu. Nếu có lỗi gửi lại thông điệp với Opcode = 3

  Gợi ý: File tạm trên server nên có tên ngẫu nhiên.
  - B4: Server gửi dữ liệu của file kết quả cho client. Khi hoàn tất việc gửi file, server gửi thông điệp với Opcode = 2 và Length = 0. Server xóa file tạm và file kết quả.
  - B5: Khi nhận file kết quả, nếu người dùng yêu cầu mã hóa thì tên file có thêm đuôi là .enc, ví dụ file image.jpg được mã hóa thành image.jpg.enc. Ngược lại, nếu người dùng yêu cầu giải mã thì tên file có thêm đuôi là .dec.

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:

    `Task2_Server.exe PortNumber`. Ví dụ: `Task2_Server.exe 5500`
  - Sử dụng một thư mục chung để lưu file của người dùng gửi lên

- **Client**:
  - Khởi động với địa chỉ server là các giá trị truyền qua tham số dòng lệnh:
    `Task2_Client.exe ServerIP ServerPort.`
    Ví dụ: `Task2_Client.exe 10.0.0.1 5500`
  - Nhận đường dẫn file do người dùng nhập từ bàn phím. File có kích thước không giới hạn và định dạng bất kỳ.
  - Gửi file lên server
  - Chức năng lặp lại cho tới khi người dùng nhập vào đường dẫn file là xâu rỗng

Yêu cầu môi trường:

- Công cụ phát triển ứng dụng: Microsoft Visual Studio 2015 Community
- Tên solution: Homework06
- Tên project: Task2_Server và Task2_Client

* Thuật toán mã hóa dịch vòng:

- Thực hiện mã hóa/giải mã theo từng byte dữ liệu

- Mã hóa: c = (m + k) mod 256

- Giải mã: m = (c – k) mod 256

m: byte dữ liệu gốc(số nguyên không dấu)

c: byte mã (số nguyên không dấu)

k: khóa mã hóa/giải mã (số nguyên không dấu)

Gợi ý:

- Khi truyền file, đọc file theo từng khối dữ liệu để truyền đi. Kích thước khối nên đủ lớn (Ví dụ tối thiểu là 10KB) để hiệu năng truyền tốt.
- Server cần phải nhận đầy đủ file mới thực hiện mã hóa và gửi lại file kết quả.
- Với mỗi client, server chỉ nhận từng khối dữ liệu, sau đó cần chuyển sang bước lặp mới để có thể phục vụ các client một cách đồng đều.
- Một kịch bản kiểm thử có thể như sau:

Bước	Cửa sổ Client 1	Cửa sổ Client 2	Cửa sổ Client 3
1	Khởi động client 1		
2		Khởi động client 2	
3			Khởi động client 3

4			Gửi file nén có kích thước khoảng 20MB để mã hóa với giá trị khóa nào đó
5		Gửi file ảnh có kích thước khoảng 10MB để mã hóa với giá trị khóa nào đó	
6	Gửi file .txt có kích thước khoảng 200KB để mã hóa với giá trị khóa nào đó		
7	Chờ hoàn tất Kết quả: Thành công. Kiểm tra nội dung file mã hóa.	Chờ hoàn tất Kết quả: Thành công. Kiểm tra nội dung file mã hóa.	Chờ hoàn tất Kết quả: Thành công. Kiểm tra nội dung file mã hóa.
8			Gửi file nén có kích thước khoảng 20MB để giải mã
9		Gửi file ảnh có kích thước khoảng 10MB để giải mã	
10	Gửi file .txt có kích thước khoảng 200KB để giải mã với khóa khác khóa mã hóa		
11	Chờ hoàn tất Kết quả: Thành công nhưng không thể đọc được file văn bản	Chờ hoàn tất Kết quả: Thành công, có thể xem được ảnh	Chờ hoàn tất Kết quả: Thành công, có thể giải nén thành công
12			Gửi file nén có kích thước khoảng 20MB để giải mã. Đóng cửa sổ client khi đang chờ kết quả.
13		Gửi file ảnh có kích thước khoảng 10MB để giải mã với khóa khác khóa mã hóa	
14	Gửi file .txt có kích thước khoảng 200KB để giải mã		
15	Chờ hoàn tất Kết quả: Thành công, có thể đọc được file văn bản	Chờ hoàn tất Kết quả: Thành công nhưng không thể xem được ảnh	