

## BÀI TẬP TUẦN 5 – IT4060

### Cách thức nộp mã nguồn:

VS2015Project > HomeworkXX > TaskXXX			
Name	Date modified	Type	Size
Debug	01-Feb-18 15:34	File folder	
ReadMe.txt	01-Feb-18 15:34	Text Document	2 KB
stdafx.cpp	01-Feb-18 15:34	C++ Source File	1 KB
stdafx.h	01-Feb-18 15:34	C Header File	1 KB
targetver.h	01-Feb-18 15:34	C Header File	1 KB
TaskXXX.cpp	01-Feb-18 15:34	C++ Source File	1 KB
TaskXXX.vcxproj	01-Feb-18 15:34	VC++ Project	8 KB
TaskXXX.vcxproj.filters	01-Feb-18 15:34	VC++ Project Filte...	2 KB

Chỉ nộp các file  
mã nguồn (.c,  
.cpp, .h)

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\_HW04.zip.

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

`Server.exe PortNumber` . Ví dụ: `Server.exe 5500`

- 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: `Client.exe 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

### Yêu cầu môi trường:

- Công cụ phát triển ứng dụng: Microsoft Visual Studio 2015 Community
- Tên solution: Homework05
- Tên project: Task1\_Server và Task1\_Client

**Bài 2.** 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 Caesar 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

Gợi ý: File tạm trên server nên có tên ngẫu nhiên.

- 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: Client lưu lại file với tên là tên file ban đầu gắn thêm phần mở rộng .enc. Ví dụ file gửi lên có tên là test.jpg thì file kết quả có tên là test.jpg.enc
- B6: 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.exe PortNumber` . Ví dụ: `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:

`Client.exe ServerIP ServerPort` . Ví dụ: `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.
- 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: Homework03
- Tên project: Task2\_Server và Task2\_Client

**\* Thuật toán mã hóa Ceasar:**

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

- Mã hóa:  $c = (m + k) \bmod 256$

- Giải mã:  $m = (c - k) \bmod 256$

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)