

## BÀI TẬP TUẦN 7 – 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	Chỉ nộp các file mã nguồn (.c, .cpp, .h)
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	

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\_HW07.zip.

**Bài tập.** Sử dụng TCP socket và kỹ thuật vào ra Overlapped sử dụng thủ tục hoàn thành(completion routine) 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(4 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 theo 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 địa chỉ là 127.0.0.1 và số hiệu cổng mặc định là 6600:
- 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:  
`TCP_Client.exe ServerIP ServerPort.`  
Ví dụ: `TCP_Client.exe 127.0.0.1 5500`
- Nhận yêu cầu mã hóa/giải mã của người dùng
- 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

#### **Thang điểm:**

- Sử dụng được kỹ thuật vào ra theo thông báo: 2 điểm
- Xử lý truyền dòng: 1 điểm
- Thực hiện đúng và đủ các chức năng: 4 điểm
- Tổ chức và trình bày mã nguồn: 3 điểm
- Một số lỗi sau đây bị trừ điểm:
  - Lỗi runtime error khiến client kết thúc: -1 điểm
  - Các tình huống khiến server kết thúc: -3 điểm
  - Các tình huống khiến server không thể xử lý thêm các yêu cầu: -3 điểm
  - Không có giao diện phù hợp, thông báo để giao tiếp với người dùng: -1 điểm
  - Không sử dụng kỹ thuật vào ra theo thông báo và giao thức đã mô tả: Bài làm không có điểm
  - Các lỗi khác: trừ điểm tùy theo mức độ nghiêm trọng của lỗi

**Yêu cầu môi trường:**

- Công cụ phát triển ứng dụng: Microsoft Visual Studio 2015 Community
- Tên solution: Homework06
- Tên project: TCP\_Server và TCP\_Client

\* Thuật toán mã hóa dịch vòng:

- 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)

**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/giải mã và gửi lại file kết quả.
- Với mỗi client, server chỉ nhận từng khối dữ liệu, sau đó cần chuyển sang bước lập 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	Chờ hoàn tất	Chờ hoàn tất

	Kết quả: Thành công. Kiểm tra nội dung file mã hóa.	Kết quả: Thành công. Kiểm tra nội dung file mã hóa.	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	

Thay đổi các hằng số và thông điệp cho phù hợp với giao thức mà sinh viên đã thiết kế.