

BÀI TẬP TUẦN 3 – 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_HW03.zip.

Bài tập. Sử dụng TCP socket và kỹ thuật đa luồng để xây dựng ứng dụng đăng bài cho người dùng.

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

```
Server.exe PortNumber .
```

Ví dụ: Task1_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: Task1_Client.exe 10.0.0.1 5500

Yêu cầu:

- Sử dụng giao thức được mô tả ở mục sau đây.
- 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 0 nếu tài khoản hoạt động, là 1 nếu tài khoản bị khóa.

- 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 để người dùng lựa chọn chức năng.

Giao thức của ứng dụng:

Client và server trao đổi các thông điệp như sau:

Chức năng	Thông điệp yêu cầu	Thông điệp trả lời
Đăng nhập	USER <i>username</i> Trong đó <i>username</i> là tên tài khoản người dùng nhập từ bàn phím	10: nếu tài khoản tồn tại và hoạt động 11: nếu tài khoản bị khóa 12: nếu tài khoản không tồn tại 99: nếu không xác định được kiểu thông điệp yêu cầu
Đăng bài	POST <i>article</i> Trong đó <i>article</i> là nội dung bài đăng người dùng nhập từ bàn phím.	20: nếu đăng bài thành công 21: nếu chưa đăng nhập 99: nếu không xác định được kiểu thông điệp yêu cầu
Đăng xuất	BYE	30: nếu đăng xuất thành công 21: nếu chưa đăng nhập 99: nếu không xác định được kiểu thông điệp yêu cầu

Lưu ý: Sinh viên tự quyết định thông điệp định dạng(theo chuỗi byte hay chuỗi ký tự) và kỹ thuật xử lý truyền dòng, nhưng phải giữ nguyên tên các kiểu thông điệp yêu cầu là USER, POST, EXIT như đã mô tả.

Yêu cầu nâng cao (Cộng 2 điểm khi thực hiện được yêu cầu này): Server có khả năng xử lý được số kết nối >1024 tùy ý.

Yêu cầu môi trường:

- Công cụ phát triển ứng dụng: Microsoft Visual Studio 2015 Community
- Tên solution: Homework03
- Tên project: Server và Client

Thang điểm:

- Sử dụng được kỹ thuật thăm dò: 2 điểm
- Xử lý truyền dòng: 2 điểm
- Thực hiện đúng và đủ các chức năng: 3 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 thăm dò và giao thức đã mô tả: Bài làm không có điểm

- Các lỗi khác: trừ điểm tùy theo mức độ nghiêm trọng của lỗi

Gợi ý:

- 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:
 - Định danh socket trên server
 - 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 để quản lý tất cả các phiên (*Lưu ý: Sinh viên có thể lựa chọn cách thiết kế khác*)
- Khi xử lý yêu cầu nâng cao, cần phải đồng bộ luồng
- Một kịch bản kiểm thử các chức năng có thể như sau(không bắt buộc tuân theo):

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			(1) Yêu cầu: Đăng nhập với tài khoản tungbt Kết quả: Thành công (2) Yêu cầu: Đăng nhập với tài khoản admin Kết quả: Thất bại (3) Yêu cầu: Đăng bài với số lần tùy ý Kết quả: Thành công (4) Yêu cầu: Đăng xuất Kết quả: Thành công (5) Yêu cầu: Đăng nhập với tài khoản tungbt Kết quả: Thành công

			(6) Đóng cửa sổ
5		(1) Yêu cầu: Đăng bài Kết quả: Thất bại (2) Yêu cầu: Đăng xuất Kết quả: Thất bại (3) Yêu cầu: Đăng nhập với tài khoản tungbt Kết quả: Thành công	
6	(1) Yêu cầu: Đăng nhập với tài khoản ductq Kết quả: Thất bại (2)Yêu cầu: Đăng nhập với tài khoản admin Kết quả: Thành công (3) Yêu cầu: Đăng bài với số lần tùy ý Kết quả: Thành công		
7		(1) Yêu cầu: Đăng bài với số lần tùy ý Kết quả: Thành công	
8	(1) Yêu cầu: Đăng xuất Kết quả: Thành công		
9		(1) Yêu cầu: Đăng xuất Kết quả: Thành công	

Sử dụng chương trình kiểm thử

Chương trình kiểm thử cho trường hợp sử dụng định dạng theo chuỗi ký tự và xử lý truyền dòng bằng mẫu kết thúc “\r\n”. Sinh viên thay đổi cho phù hợp với kỹ thuật đã sử dụng nếu cần.