

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN  
KHOA CÔNG NGHỆ THÔNG TIN



---

# BÁO CÁO ĐỒ ÁN LẬP TRÌNH SOCKET

## MẠNG MÁY TÍNH

---

TRIỆU NHẬT MINH - 21127112  
BÙI ĐỖ DUY QUÂN - 21127141  
HOÀNG ĐỨC VIỆT - 21127203

**Giảng viên hướng dẫn:**

Lê Hà Minh

Lê Ngọc Sơn

Nguyễn Thanh Quân

26th December 2022

# Contents

1	Thông tin nhóm . . . . .	2
1.1	Thành viên nhóm . . . . .	2
1.2	Đề tài . . . . .	2
2	Đánh giá mức độ hoàn thành . . . . .	2
3	Kịch bản giao tiếp của chương trình . . . . .	2
4	Cấu trúc thông điệp . . . . .	3
4.1	Nhận (Từ Client đến Server) . . . . .	3
4.2	Gửi (Từ Server đến Client) . . . . .	3
4.3	Kiểu dữ liệu của thông điệp . . . . .	4
5	Môi trường lập trình và các framework hỗ trợ . . . . .	4
6	Hướng dẫn sử dụng các tính năng chương trình . . . . .	4
7	Bảng phân công công việc . . . . .	5
8	Tài liệu tham khảo . . . . .	5

## 1 Thông tin nhóm

### 1.1 Thành viên nhóm

- 21127112 - Triệu Nhật Minh
- 21127141 - Bùi Đỗ Duy Quân
- 21127203 - Hoàng Đức Việt

### 1.2 Đề tài

- Mã đề:  $(2112711\underline{2} + 2112714\underline{1} + 2112720\underline{3}) \% 2 + 1 = 1$

## 2 Đánh giá mức độ hoàn thành

STT	Chức năng	Mức độ hoàn thành	Note
1	Kết nối	100%	
2	Quản lý kết nối	100%	
3	Tải được page index.html	100%	
4	Đăng nhập	100%	
5	Lỗi page	100%	
6	Multiple requests	100%	
7	Multiple connection	100%	
8	Report	100%	

## 3 Kịch bản giao tiếp của chương trình

- Thực hiện các bước khởi động server.
- Kết nối: Client kết nối đến server thông qua kết nối TCP bằng cách nhập địa chỉ IPv4 của máy server vào trình duyệt với điều kiện là phải sử dụng chung đường mạng với máy server.
- Nếu client mất kết nối thì server vẫn tiếp tục chạy mà không có lỗi.
- Nếu client kết nối thành công thì:
  - Nếu trang được yêu cầu không tồn tại trong server thì sẽ trả về **trang 404** cho client.
  - Nếu trang tồn tại thì:
    - Nếu client muốn truy cập trực tiếp vào trang */images.html* thì sẽ trả về **trang 401** cho client.

- Nếu client muốn truy cập vào những trang còn lại (ngoại trừ trang */images.html*) thì server sẽ trả về trang được yêu cầu cho client. Đặc biệt khi client yêu cầu truy cập trang */* thì sẽ trả về trang */index.html*.
- Nếu muốn truy cập vào trang */images.html* thì phải nhập đúng username và password, nếu không đúng thì sẽ trả về **trang 401**, và ngược lại thì sẽ truy cập được trang */images.html*.

## 4 Cấu trúc thông điệp

### 4.1 Nhận (Từ Client đến Server)

```
1 POST /images.html HTTP/1.1
2 Host: 10.123.0.247:8080
3 Connection: keep-alive
4 Content-Length: 34
5 Cache-Control: max-age=0
6 Upgrade-Insecure-Requests: 1
7 Origin: http://10.123.0.247:8080
8 Content-Type: application/x-www-form-urlencoded
9 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537.36 Edg/108.0.1462.54
10 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
11 Referer: http://10.123.0.247:8080/
12 Accept-Encoding: gzip, deflate
13 Accept-Language: en-US,en;q=0.9
14
15 uname=admin&psw=123456&remember=on
```

- Dòng đầu: Cho biết thông tin yêu cầu: GET hoặc POST, tên file cần lấy và phiên bản HTTP (thường là 1.1)
- Phần Header: Nó là một tập hợp các dòng chứa thông tin về HTTP Message, thông tin về phần body.
- Dòng trắng: Tách phần Header và Body (Nếu có Body)
- Body: Chứa dữ liệu đi kèm với request (Cụ thể là như trong hình là tên đăng nhập và mật khẩu)

### 4.2 Gửi (Từ Server đến Client)

```
1 HTTP/1.1 200 OK\r\n
2 [—]Content-Type: <Type>\r\n
3 [—]Content-Length: <Length>\r\n
4 [—]\r\n
5 [—]<Data>
```

- Dòng đầu: Phiên bản HTTP (1.1), mã trả về (200, 404, 401), Đoạn text ngắn giải thích mã ví dụ (OK, NOT FOUND, UNAUTHORIZED).
- Header: Tương tự như Header request. Ngoài ra phải có Content-Type và Content-Length để có thể sử dụng kết nối Persistent.
- Dòng trắng: Tách Header và Body (`\r\n`)
- Body: Chứa dữ liệu của Response.

#### 4.3 Kiểu dữ liệu của thông điệp

- Thông điệp được gửi từ Client đến Server được lưu trữ bằng mảng *char*.
- Để tiện lợi cho việc xử lý, trong quá trình này, mảng *char* dùng để lưu trữ thông điệp sẽ được chuyển sang *string*.
- Thông điệp trả về của Server cho Client là *string*.

### 5 Môi trường lập trình và các framework hỗ trợ

- Hệ điều hành: Windows 10 Home/Professional 64-bit
- Ngôn ngữ lập trình: C++
- IDE: Visual Studio 2022
- Thư viện hỗ trợ: CSocket cùng các thư viện cơ bản của C++
- Trình duyệt: Safari (iOS), Microsoft Edge (Chromium), Google Chrome

### 6 Hướng dẫn sử dụng các tính năng chương trình

1. Cài đặt Visual Studio 2022 cũng như thư viện MFC
2. Mở source code và chỉnh sửa Web Source ở file *Core.cpp* biến *webSrc* nếu cần thiết
  - Đường dẫn mặc định: */web\_src/*
3. Nhấn F5 hoặc click vào **Local Windows Debugger** để khởi chạy server
4. Truy cập `http://localhost:8080/` hoặc `http://<ip>:8080/` với **<ip>** là địa chỉ IP của server.
5. Nhấn nút **Login**
6. Nhập tên tài khoản: admin, password: 123456 và nhấn **Login**

## 7 Bảng phân công công việc

STT	MSSV	Công việc phân công
1	21127112	Hàm processGET, Mục 4, 5 báo cáo L <sup>A</sup> T <sub>E</sub> X
2	21127141	Hàm getResponseMess, processPOST, Mục 3 báo cáo L <sup>A</sup> T <sub>E</sub> X
3	21127203	Hàm GetHeader, responseThread, Mục 6 báo cáo L <sup>A</sup> T <sub>E</sub> X

## 8 Tài liệu tham khảo

- Trang bìa L<sup>A</sup>T<sub>E</sub>X
- Tham khảo code Server
- Multithread cơ bản bằng MFC trong C++
- *send* và *recv* trong C++
- Windows Socket sử dụng CAsyncSocket
- Kiểm tra sub-string có tồn tại trong string hay không?
- Đọc file bằng string C++
- [21127223 - Nguyễn Tiến Bách] Sửa lỗi Multithread