

# ĐỒ ÁN THỰC HÀNH 1 – LẬP TRÌNH SOCKET

## MÔN MẠNG MÁY TÍNH

### 1. Quy định chung

- Đồ án được làm theo nhóm: mỗi nhóm tối đa 3 sinh viên, tối thiểu 2 sinh viên (trong trường hợp sĩ số lớp lẻ), sinh viên tự chọn nhóm (sử dụng nhóm thực hành đã đăng ký nếu có).
- **Các bài làm giống nhau sẽ đều bị điểm 0 toàn bộ phần thực hành tất cả các nhóm liên quan (dù có điểm các bài tập, đồ án thực hành khác).**
- Môi trường lập trình: Tự do lựa chọn ngôn ngữ lập trình, tự do lựa chọn môi trường hệ điều hành: Windows, Unix/Linux, macOS
- Ngôn ngữ lập trình GV có thể hỗ trợ: C/C++, C#, Java, Python
- Thư viện hỗ trợ lập trình socket cho phép sử dụng: Socket, CSocket, winsock. Tức là chỉ sử dụng các thư viện Socket do ngôn ngữ lập trình cung cấp. **Không phải lập trình website.**

### 2. Cách thức nộp bài

- **Nộp bài trực tiếp trên Website môn học, không chấp nhận nộp bài qua email hay hình thức khác.**
- Tên file: **MSSV1\_MSSV2\_MSSV3.zip** (Với MSSV1 < MSSV2 < MSSV3)

Ví dụ: Nhóm gồm 2 sinh viên: 2012001, 2012002, và 2012003, tên file nộp: **2012001\_2012002\_2012003.zip**

**Cấu trúc file nộp gồm thư mục MSSV1\_MSSV2\_MSSV3 chứa các file:**

1. **Report.pdf:** chứa báo cáo về bài làm
2. **Release:** thư mục chứa file thực thi của chương trình, **nếu có** (\*.exe/ ...)
3. **Source:** thư mục chứa source code của chương trình, yêu cầu nộp cả project đã xóa bỏ thư mục Debug và các file không cần thiết khác.. **Nhóm nào chỉ nộp file \*.cpp và \*.h và không biên dịch được thì bị 0 điểm.**

**Lưu ý: Cần thực hiện đúng các yêu cầu trên, nếu không, bài làm sẽ không được chấm.**

### 3. Hình thức chấm bài

Chấm vấn đáp vào thời điểm kết thúc phần thực hành.

### 4. Tiêu chí đánh giá

#### Về chương trình:

- Mục tiêu của đồ án này tập trung chủ yếu vào 2 vấn đề: lập trình socket, xây dựng giao thức trao đổi giữa client và server. Do đó các tiêu chí đánh giá dựa vào các chức năng chính được liệt kê trong yêu cầu của chương trình (có ghi chú thang điểm cho từng chức năng)

#### Về báo cáo:

- Thông tin của nhóm.
- Đánh giá mức độ hoàn thành từ 0 – 100% (Chú thích rõ những mục làm được, chưa làm được và còn bị lỗi)
- Kịch bản giao tiếp của chương trình: Giao thức trao đổi giữa client và server, cấu trúc thông điệp, kiểu dữ liệu của thông điệp, cách tổ chức cơ sở dữ liệu (nếu có).
- Môi trường lập trình và các framework hỗ trợ để thực thi ứng dụng.
- Hướng dẫn sử dụng các tính năng chương trình.
- Bảng phân công công việc và cho biết rõ ràng ai làm việc gì cách rõ ràng. Không chia đều công việc hay cùng làm mọi việc.
- Các nguồn tài liệu tham khảo.

**Lưu ý:** Trong báo cáo không dán các đoạn source code của chương trình. Mã chương trình chỉ trình bày nếu thật sự cần thiết và nếu cần minh họa cho các mô hình cài đặt hay các cơ chế đồng bộ (minh họa dạng mã giả, prototype hàm).

#### Về vấn đáp:

- Chuẩn bị thiết bị, chương trình, báo cáo đầy đủ (không cần in).
- Trả lời các câu hỏi từ GV
- Trường hợp trả lời sai hoặc không trả lời được sẽ trừ trực tiếp điểm vào tổng điểm đồ án.

**Lưu ý:** Tất cả thành viên của nhóm phải tham gia buổi vấn đáp. Thành viên vắng mặt sẽ xử lý theo quy định sau:

- Có phép (gửi email xin phép trước buổi vấn đáp): trừ điểm vấn đáp trực tiếp
- Không phép: 0 điểm toàn đồ án.

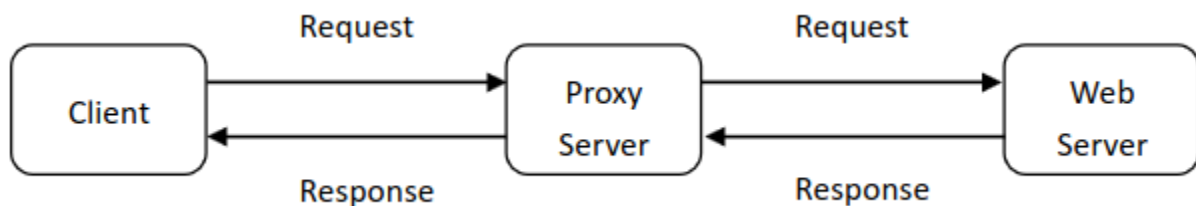
## 5.Nội dung:

### HTTP Web Proxy Server

Trong đồ án này, bạn sẽ tìm hiểu cách hoạt động của máy chủ web proxy và một trong những chức năng cơ bản của chúng “**Caching**”.

Nhiệm vụ của bạn là phát triển một máy chủ web proxy cơ bản, có khả năng lưu trữ các trang web vào bộ đệm (cache). Đây là một web proxy đơn giản có chức năng như: hiểu các yêu cầu (http request) nội dung trang web bằng phương thức GET và các phương thức khác (POST, PUT), có thể xử lý tất cả các loại đối tượng: trang HTML, hình ảnh, video, audio...

Khi web client đưa ra yêu cầu (http request), yêu cầu đó sẽ được gửi đến web server. Sau đó, web server sẽ xử lý yêu cầu và gửi lại thông báo phản hồi (http response) cho web client. Để cải thiện hiệu suất, chúng ta tạo một máy chủ proxy nằm giữa web client và web server. Các yêu cầu do máy web client gửi và thông báo phản hồi từ webserver gửi đều đi qua web proxy. Nói cách khác, web proxy chính là máy trung gian đại diện các web client trong một hệ thống mạng gửi yêu cầu đến web server, đồng thời nhận phản hồi ngược lại từ web server và chuyển tiếp đến web client. **Bạn có thể sử dụng "Connection: Close" trong header của Request gửi từ "Proxy Sever" đến "Web Server" để xử lý nhận dữ liệu và khi "Web Server" gửi dữ liệu xong thì "Web Server" sẽ tự đóng kết nối để Proxy Server biết được đã nhận đủ dữ liệu.**



#### Lưu ý:

Bạn phải sử dụng socket để gửi và nhận dữ liệu giữa web client và Proxy, và giữa Proxy với web server, bạn không được phép sử dụng bất kỳ thư viện hoặc đối tượng hiện có nào được ngôn ngữ lập trình hỗ trợ sẵn ngoài socket (Ví dụ, sử dụng python thì là “import socket”). Còn các thư viện khác như xử lý URL, ghi file .. thì được sử dụng bình thường.

Chỉ sử dụng socket đơn thuần, **không** dùng đối tượng HTTP socket.

Ví dụ các thư viện **không** được phép sử dụng:

- C/C++: boost::network, boost::network::http, cURL, libcurl, cpp-httplib, ...
- Python: request, httplib2, http.client, urllib, ....
- Java: Apache HttpClient, OkHttp, Java HttpURLConnection, Spring WebClient, ...

- Go: net/http, Go HTTP Client, GRequests, Gentleman ...
- C#: HttpClient, ...
- Node: axios, node-fetch, request, ...

### 5.1. Code tham khảo:

Dựa vào khung mã chương trình đã được cung cấp trong file **proxy.py** (viết bằng ngôn ngữ Python), các bạn có thể bổ sung các chức năng phù hợp với yêu cầu đề bài. Hướng dẫn bổ sung code ở các vị trí giữa cặp từ sau: “**#Fill in start..... #Fill in end**”. Sinh viên có thể sử dụng ngôn ngữ khác để phát triển chương trình web proxy server.

Link:

<https://drive.google.com/file/d/1vvslv7PpoAHfX9rFHg3pSYl4JE7m2HOx/view?usp=sharing>

### 5.2. Cách sử dụng web proxy:

Khởi chạy chương trình proxy server bằng dòng lệnh. Sử dụng web client (trình duyệt web) để gửi yêu cầu đến webserver. Chuyển hướng yêu cầu đến webproxy bằng địa chỉ IP cục bộ và port hoạt động của proxy server.

Vd: khi client cần truy cập trang [www.google.com](http://www.google.com), proxy server hoạt động trên máy localhost với port 8888. Để chuyển hướng truy cập đến proxy, ta nhập URL sau vào web client (trình duyệt web): <http://localhost:8888/www.google.com>

Để sử dụng máy chủ proxy chạy trên các máy tính riêng biệt với trình web client, bạn cần thay localhost trong URL bằng địa chỉ IP mà máy chủ proxy của bạn đang chạy trên đó. Bạn cần thay thế số hiệu port “8888” bằng số hiệu port lắng nghe mà bạn đã sử dụng trong mã chương trình của proxy server

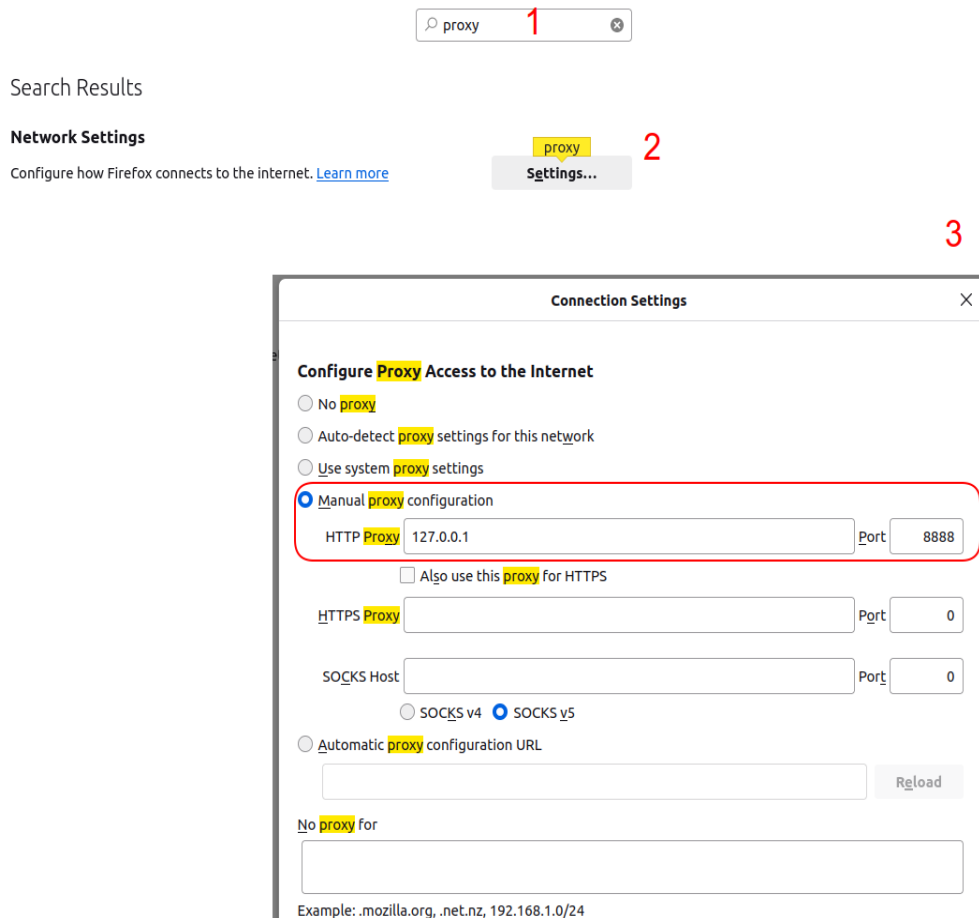
### 5.3. Cấu hình web client (trình duyệt web)

Ngoài cách sử dụng URL như mục 5.2, bạn có thể cấu hình trình duyệt web sử dụng web proxy bằng cách thay đổi các thiết lập của trình duyệt web. Bạn có thể tham khảo cách thiết lập proxy cho một số trình duyệt web theo link sau:

<https://fptshop.com.vn/hoi-dap/thu-thuat-huong-dan/huong-dan-thay-doi-thiet-lap-proxy-tren-nhi-eu-trinh-duyet-12405>

Với hướng tiếp cận này, bạn chỉ cần nhập địa chỉ trang web mà client muốn truy cập vào trình duyệt web như: [www.google.com](http://www.google.com)

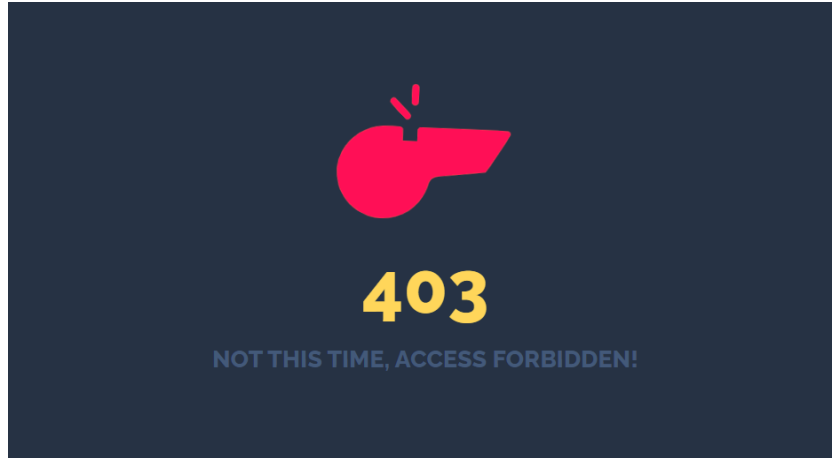
Ví dụ cách thay đổi cấu hình chuyển hướng truy cập đến proxy trên trình duyệt Firefox:



## 5.4. Yêu cầu (HTTP, not HTTPS)

Các chức năng cần triển khai trên web proxy server:

1. Hỗ trợ các phương thức GET, POST, HEAD. Nếu web request sử dụng những phương thức khác như PUT, PATCH, DELETE, OPTIONS... proxy cần trả về mã phản hồi 403 và nội dung trang 403 do sinh viên tự thiết kế (có thể dùng HTML code đơn giản)  
Bạn có thể cài phần mềm "Postman" để test các method như HEAD, PUT, PATCH ... trên Windows/Linux/MacOS, hoặc có thể dùng command "curl" để test.  
Ví dụ nội dung HTML trả về phản hồi 403 như mẫu của trang:  
<https://devbeep.com/403-forbidden-html-templates/>



Ví dụ trang web trả về nội dung 404: <https://courses.fit.hcmus.edu.vn/asdfs>

2. Hỗ trợ lưu cache các đối tượng ảnh (hỗ trợ tất cả các định dạng ảnh) trong **15 phút** (*cấu hình thời gian lưu cache trong file config, mã chương trình proxy server sẽ đọc từ file này khi khởi chạy*). Yêu cầu từ các client khác sẽ sử dụng bản cache từ proxy server thay vì nạp từ webserver. Sau mỗi **15 phút**, proxy server cần cập nhật lại dữ liệu (nạp lại ảnh từ web server) khi thời gian cache đã hết hạn  
**Note:** ảnh “abc.png” của website “hello.com” khác ảnh “abc.png” của website “olleh.com”. Bạn cần tổ chức lưu trữ dữ liệu trên proxy server sao cho quản lý được dữ liệu được cache của các site khác nhau
3. Tạo trang web danh sách trắng “whitelisting” (lưu trong file config của proxy server), máy chủ proxy chỉ cho phép máy khách truy cập trang web này, các trang web khác sẽ được phản hồi mã 403 với trang nội dung do bạn tự viết  
Vd: cho phép web client truy cập tên miền **abc.com**, nghĩa là cho phép các URL dạng **\*.abc.com/\*** như: [www.abc.com](http://www.abc.com), [www.abc.com/index.html](http://www.abc.com/index.html), [www.abc.com/image.jpg](http://www.abc.com/image.jpg)
4. Giới hạn thời gian truy cập proxy server. Vd chỉ cho phép truy cập từ 8AM-8PM, ngoài khung giờ này các truy cập từ web client sẽ được proxy server phản hồi mã 403 lỗi
5. Hỗ trợ xử lý đồng thời nhiều truy vấn từ các web client

#### Ví dụ về file config:

```
cache_time = 900 # seconds
whitelisting=oosc.online, example.com, ...
time=8-20
```

### 5.5. Các trang web có thể sử dụng để kiểm tra chương trình:

1. <http://oosc.online/>
2. <http://example.com/>

3. <http://www.google.com/>
4. <http://www.bing.com/>
5. <http://testphp.vulnweb.com/login.php> (test POST, user/pass: test/test)
6. <http://vbsca.ca/login/login.asp> (test POST, user/pass: pmcmahon/somepass hoặc jbloggs/anotherpass)
7. <http://vbsca.ca/login/>
8. <http://vbsca.ca/login/LoginsAndPermissions3.htm>
9. ...

## 6.Thang điểm

STT	Yêu cầu	Điểm
1	Cho phép xử lý các phương thức: GET, POST, HEAD	3
2	Không cho phép: PUT, PATCH, DELETE, OPTIONS ...	0,5
3	Phản hồi mã 403 và nội dung trang 403 ở mục 5.1, 5.3, 5.4	1
4	Cache images	2
5	Whitelisting	1
6	Giới hạn truy cập theo thời gian	0,5
7	Xử lý yêu cầu đồng thời	1
8	Config file	0,5
9	Báo cáo	0,5
10	Sử dụng Connection là: Keep-alive để xử lý trường hợp Content-Length và "Transfer-Encoding: chunked"	1