

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN



# BÁO CÁO ĐỒ ÁN

Môn: Mạng máy tính

ĐỀ TÀI: Proxy Server

Thành viên thực hiện:

- |                     |         |
|---------------------|---------|
| 1. Nguyễn Tấn Tài   | 1712742 |
| 2. Hoàng Quốc Thịnh | 1712790 |
| 3. Mai Huy Thông    | 1712800 |
| 4. Lê Trung Tiến    | 1712811 |

TP Hồ Chí Minh, tháng 6 năm 2019

# MỤC LỤC:

PHẦN 1: PHÂN CÔNG CÔNG VIỆC .....	3
PHẦN 2: SỰ CẦN THIẾT CỦA PROXY SERVER .....	4
PHẦN 3: MÔ TẢ CÁC HÀM CHÍNH .....	5
PHẦN 4: GÓI TIN TỪ CLIENT - SERVER VÀ BẮT GÓI TIN ....	9
PHẦN 5: CÁCH CHẠY CHƯƠNG TRÌNH VÀ KẾT QUẢ ....	10
PHẦN 6: TÀI LIỆU THAM KHẢO .....	16

## I. Phân công công việc:

Trong suốt quá trình làm đồ án, nhóm tập hợp lại ngồi chung với nhau là chủ yếu, các công việc dưới đây là các công việc được phân công cho từng thành viên của nhóm. Tuy nhiên trong quá trình làm việc, không thể tránh khỏi việc có những thành viên bế tắc trong việc tìm lời giải cho công việc của mình được giao, những lúc như vậy, các thành viên khác đã giúp đỡ, cùng nhau giải quyết vấn đề, mặc dù đó không phải là công việc của mình được giao. Vậy nên xét cho cùng, hầu như tất cả các công việc đều có sự tham gia của nhiều thành viên trong nhóm chứ không phải chỉ có một mình người được chỉ định.

### Tiến:

Xử lý GET request (trường hợp load trang lần đầu, chưa có cachedFiles trên Server):  
hàm `sendNonCachedToClient(String urlString)`

Xử lý việc cho phép thêm trang bị chặn, xem các trang bị chặn ngay trong khi Server đang chạy: Hàm `run()` trong class Proxy

Xử lý việc lưu trữ cả thông tin về các trang bị chặn và các thành phần đã được cache khi đóng Server: hàm `closeServer()` trong class Proxy

Tổng hợp và viết hàm chạy chương trình: hàm `main(String args[])`

### Tài:

Xử lý GET request cùng với Thông (khi bị block): Hàm `blockSiteRequested(String urlString)`

Xử lý đa luồng đồng thời cho các requests từ các Clients: Hàm `listen()`

Thực hiện tổng hợp các hàm để viết hàm `run()` để vận hành cho class RequestHandler:  
hàm `run()` trong class RequestHandler

Xử lý yêu cầu unblock trang: chức năng này được viết thêm trong hàm `run()` của class Proxy

Tổng hợp và chỉnh sửa báo cáo

### Thịnh:

Xử lý GET request (trường hợp có file cache trên Proxy Server): Hàm  
`sendCachedPageToClient(File f)`

Build java project thành file exe

Thực hiện bắt gói tin và mô phỏng quá trình gửi/nhận gói tin thông qua ProxyServer trên Wireshark.

Xử lý yêu cầu xóa toàn bộ cachedFiles: chức năng này được viết thêm trong hàm `run()` của class Proxy

## Thông:

Xử lý POST request

Xử lý GET request cùng với Tài (trường hợp trang bị block): Hàm

`blockSiteRequested(String urlString)`

Xử lý việc tạo file `blacklist.conf`, load các file cache được lưu trong đĩa cũng như những trang web bị block trước đó khi khởi động Server: hàm constructor `Proxy(int port)`

Các hàm hỗ trợ cho các chức năng khác: `isBlocked(String urlString)`,

`addCachedPage(String urlString, File fileToCache)`, `getCachedPage(String url)`,

`stripUnwantedHeader()`.

## II. Tại sao proxy server là cần thiết:

Về cơ bản, một máy chủ proxy hoạt động như một người trung gian giữa máy chủ và máy khách phục vụ máy tính

Nó cho phép các máy tính khách thực hiện các kết nối mạng gián tiếp đến các dịch vụ mạng khác. Nếu sử dụng máy chủ proxy, trước tiên máy tính khách sẽ kết nối với máy chủ proxy, yêu cầu một số tài nguyên như trang web, trò chơi, video, mp3, sách điện tử, bất kỳ tài nguyên nào khác có sẵn từ các máy chủ khác nhau qua Internet. Ngay khi nhận được yêu cầu như vậy, máy chủ proxy sẽ tìm kiếm các tài nguyên từ bộ đệm trong đĩa cứng cục bộ của nó. Nếu các tài nguyên đã được lưu trữ trước đó, máy chủ proxy sẽ trả lại chúng cho các máy khách. Nếu không được lưu trữ, nó sẽ kết nối với các máy chủ có liên quan và yêu cầu tài nguyên thay mặt cho các máy khách. Sau đó, nó 'lưu trữ' tài nguyên từ các máy chủ từ xa và trả lại các yêu cầu tiếp theo cho cùng một nội dung.

Ngày nay, chúng ta sử dụng máy chủ proxy cho nhiều mục đích khác nhau như chia sẻ kết nối internet trên mạng cục bộ, ẩn địa chỉ IP của chúng ta, thực hiện kiểm soát truy cập internet, truy cập các trang web bị chặn, v.v. Dưới đây là một số lợi ích tại sao mọi người sử dụng máy chủ proxy:

- Để chia sẻ kết nối Internet trên mạng LAN. Một số doanh nghiệp và gia đình nhỏ có nhiều máy tính nhưng chỉ với một kết nối Internet, họ có thể chia sẻ kết nối Internet cho các máy tính khác trên mạng LAN với máy chủ proxy.
- Để tăng tốc độ lướt Internet. Nếu sử dụng máy chủ proxy, ban đầu tất cả các yêu cầu từ máy khách sẽ đến máy chủ proxy, nếu máy chủ proxy đã lưu trữ các tài nguyên được yêu cầu trong ổ cứng cục bộ trước đó với chức năng bộ đệm web, khách hàng sẽ nhận được phản hồi trực tiếp từ máy chủ proxy, nó sẽ nhanh hơn truy cập trực tiếp.

- Để ẩn địa chỉ IP của máy khách để nó có thể lướt ẩn danh, điều này chủ yếu là vì lý do bảo mật. Một máy chủ proxy có thể hoạt động như một trung gian giữa máy tính của người dùng và Internet để ngăn chặn sự tấn công và truy cập bất ngờ.
- Để thực hiện kiểm soát truy cập Internet như xác thực cho kết nối Internet, kiểm soát băng thông, kiểm soát thời gian trực tuyến, bộ lọc web internet và bộ lọc nội dung, v.v...
- Để bỏ qua các hạn chế bảo mật và bộ lọc. Ví dụ: nhiều văn phòng làm việc đã chặn facebook và myspace, tuy nhiên, bạn có thể sử dụng máy chủ proxy để vượt qua các hạn chế đó và dễ dàng truy cập các trang web bị chặn.
- Để quét nội dung ra bên ngoài, ví dụ, để bảo vệ rò rỉ dữ liệu.
- Để phá vỡ các hạn chế khu vực. Ví dụ: một máy chủ sử dụng định vị địa lý dựa trên IP để hạn chế dịch vụ của mình ở một quốc gia nhất định có thể được truy cập bằng proxy được đặt tại quốc gia đó để truy cập dịch vụ.

### III. Mô tả các hàm chính:

#### File Proxy.java

##### ➤ Phương thức main:

- Tên phương thức: ***public static void main(String[] args)***
- Chức năng: Khởi tạo proxy với port theo yêu cầu (8888) và lắng nghe các yêu cầu từ Client gửi đến để xử lý.
- Kết quả khi gọi phương thức proxy sẽ bắt đầu hoạt động.

##### ➤ Phương thức khởi tạo Proxy:

- Tên phương thức: ***public Proxy(int port)***
- Đối số truyền vào: một số nguyên là số port cần khởi tạo proxy kết nối đến đó.
- Chức năng: Khởi tạo các HashMap cache và blockSites để lưu nhưng thông tin được cache/block trước đó. Nếu như thông tin đã tồn tại trước đó thì tải thông tin lên và lưu vào HashMap. Phương thức cũng khởi tạo danh sách quản lý các luồng vào bắt đánh dấu bắt đầu luồng để xử lý.

##### ➤ Phương thức listen:

- Tên phương thức: ***public void listen()***
- Phương thức không có đối số truyền vào
- Chức năng: Lắng nghe các Client gửi yêu cầu đến từ port (8888) và tạo các luồng xử lý các yêu cầu đó.

- Kết quả sau khi gọi phương thức sẽ thực hiện việc nghe các yêu cầu từ Client và tạo các luồng xử lý liên tục đến khi có yêu cầu dừng (close) từ người dùng proxy, cờ hiệu sẽ được bật false và dừng phương thức.

➤ Phương thức closeServer:

- Tên phương thức: ***private void closeServer()***
- Phương thức không có đối số truyền vào.
- Chức năng: bật cờ hiệu thông báo dừng phương thức listen để đóng proxy. Phương thức còn lưu lại những trang đã bị chặn/đã cache lưu trong trong HashMap vào file tương ứng để cho lần hoạt động sau.
- Kết quả sau khi gọi phương thức, cờ hiệu running được bật false để thông báo đóng proxy và các HashMap cache, blockedSites được lưu vào file tương ứng.

➤ Override phương thức run

- Tên phương thức: ***public void run()***
- Phương thức không có đối số truyền vào.
- Chức năng: Phương thức sẽ nhận các yêu cầu từ người dùng proxy nhập vào gồm có:
  - ❖ blocked: cho người dùng proxy biết các domain đã bị chặn
  - ❖ close: thực hiện lệnh đóng proxy (gọi phương thức closeServer)
  - ❖ Domain/URL : nhận chuỗi từ người dùng và tách lấy domain(nếu cần) và thêm domain vào danh sách trang bị chặn.
  - ❖ Unblock + Domain/URL: bỏ chặn Domain đó.
- Kết quả: tùy vào người dùng nhập vào mà xử lý cách yêu cầu đó cho phù hợp và đúng đắn.

➤ Phương thức getCachedPage

- Tên phương thức: ***public static File getCachedPage(String url)***
- Chức năng: Phương thức sẽ kiểm tra URL truyền vào có được cache trong HashMap cache không và trả về kết quả tương ứng.
- Kết quả trả về là File cache nếu như URL đó đã được cache, ngược lại trả về null.

➤ Phương thức addCachedPage

- Tên phương thức: ***public static File addCachedPage (String urlString, File fileToCache)***

- Chức năng: Phương thức sẽ thêm file cache vào HashMap cache với key là urlString và value là file fileToCache được truyền vào.
- Phương thức không có kết quả trả về.

## File RequestHandler.java

### ➤ Override phương thức Run

- Tên phương thức: *public void run()*
- Phương thức không có tham số truyền vào
- Chức năng: Xử lý chính trong file RequestHandler.java, hàm là một luồng nhận yêu cầu từ Client gửi đến Server. Hàm sẽ xử lý chuỗi yêu cầu từ Client gửi đến, kiểm tra yêu cầu là GET/POST, kiểm tra domain có bị chặn hay không, kiểm tra url đã đã cache hay chưa. Tùy vào trường hợp mà gọi các hàm tương ứng để xử lý.
- Sau khi phương thức này kết thúc thì luồng chạy đến phương thức mặc định exit và kết thúc luồng

### ➤ Phương thức sendCachedPageToClient

- Tên phương thức: ***private void sendCachedPageToClient(File cachedFile)***
- Tham số truyền vào: File cache
- Chức năng: Phương thức sẽ mở file đã cache và gửi phản hồi về Client ứng với yêu cầu từ Client gửi đến Proxy. Phương thức được gọi trong phương thức run() nếu kiểm tra URL trong yêu cầu mà Client gửi lên đã cache và được lưu trong thư mục cached
- Kết quả sau khi gọi phương thức sẽ trả về nội dung của yêu cầu gửi đến từ Client (URL)

### ➤ Phương thức sendNonCachedToClient:

- Tên phương thức: ***private void sendNonCachedPageToClient(String urlString)***
- Tham số truyền vào: Chuỗi url trong yêu cầu Client gửi đến
- Chức năng: Phương thức sẽ mở một HttpURLConnection kết nối đến Server của URL và gửi nội dung từ URL trả về đến Client. Trong quá trình trả về, phương thức sẽ tạo một file cache tương ứng (có sửa đổi) với URL đó và lưu lại nội dung vào file cache. Phương thức được gọi trong phương thức run() nếu kiểm tra URL trong yêu cầu mà Client gửi đến chưa được lưu trong cache

- Kết quả sau khi gọi phương thức sẽ trả về nội dung của yêu cầu gửi đến từ Client và tạo ra một file tương ứng với nội dung được lưu trong thư mục cached và lưu lại file đã cache trong HashMap cache của Proxy.java

➤ Phương thức blockedSiteRequested:

- Tên phương thức: ***private void blockedSiteRequested()***
- Phương thức không có tham số truyền vào.
- Chức năng: Gửi mã lỗi **403 Forbidden** trả về Client nếu yêu cầu nhập vào là domain bị chặn (được gọi trong phương thức run() khi kiểm tra domain đó đã bị chặn) và đọc file *forbiddenPageNoti.html* để thông báo cho người dùng đã truy cập vào trang bị chặn.
- Kết quả khi gọi phương thức sẽ trả về mã lỗi 403 Forbidden ngăn chặn truy cập vào trang web (theo domain) và thông báo cho người dùng trên Browser.

➤ Phương thức checkClientServerStream:

- Tên phương thức: ***private void checkClientServerStream()***
- Phương thức không có tham số truyền vào.
- Chức năng: Kiểm tra tình trạng đóng các chiều nhận/gửi (input/output stream) của socket tại client và tại server proxy
- Kết quả sau khi gọi phương thức: Tại cả client và server, nếu chiều gửi đã bị đóng thì tạo một output stream để gửi dữ liệu và tương tự nếu chiều nhận đã bị đóng thì tạo một input stream để nhận dữ liệu.

➤ Phương thức getUri:

- Tên phương thức: ***private void getUri()***
- Phương thức không có tham số truyền vào.
- Chức năng: Lấy URI của đối tượng trên mạng thông qua URL
- Kết quả sau khi gọi phương thức: URI được lưu bởi thuộc tính uri để đưa vào requestStr trong hàm postHandle

➤ Phương thức stripUnwantedHeaders:

- Tên phương thức: ***private void stripUnwantedHeaders()***
- Phương thức không có tham số truyền vào.
- Chức năng: loại bỏ những header không cần thiết ra khỏi hashmap lưu trữ.
- Kết quả sau khi gọi phương thức: chỉ những header mong muốn mới được giữ lại.



➤ Phương thức `postHandle`:

- Tên phương thức: ***private void postHandle()***
- Phương thức không có tham số truyền vào.
- Chức năng: Xử lý request có `requestType` là POST
- Kết quả sau khi gọi phương thức: Proxy lấy dữ liệu trả về cho client kết quả truy xuất được.

## IV. Mô tả gói tin chạy từ client đến server và bắt gói tin bằng Wireshark

### 1. Trường hợp nếu Proxy không lưu trữ cache

- Khi **Client** nhập URL vào Browser (Browser sẽ trở thành **Client**) để lấy thông tin từ trang web HTTP mà **Client** muốn, ví dụ <http://www.hcmut.edu.vn/vi> thì Browser sẽ chuyển URL thành dạng tin nhắn *request* gửi đến **Proxy**.
- **Proxy** sẽ thông qua yêu cầu đó và gửi *request* lên **Server**, sau đó **Server** phản hồi lại với **Proxy** bằng một tin nhắn *reponse*, tin *reponse* có thể là thông tin từ **Server** mà **Client** đã yêu cầu hoặc một thông báo lỗi.
- Sau đó **Proxy** gửi tất cả *reponse* về cho **Client**.
- **Client** tiếp tục gửi lên các yêu cầu cho **Proxy** để lấy các thông tin còn lại từ **Server**.

### 2. Trường hợp nếu Proxy có lưu trữ cache:

- Sau khi **Client** gửi *request* lên **Proxy**, **Proxy** nhận thấy có cache liên quan đến *request* mà **Client** gửi lên, **Proxy** sẽ không gửi *request* lên **Server** mà sẽ load các file cache mà **Proxy** đã lưu trong đĩa lên cho **Client**.

### 3. Trường hợp Proxy chặn trang web:

- Sau khi **Client** gửi *request* lên **Proxy**, **Proxy** nhận thấy trang web này nằm trong danh sách chặn, **Proxy** sẽ gửi *reponse* báo lỗi về cho **Client**, không cho phép truy cập lên **Server**.

#### 4. Trường hợp Client truy xuất dữ liệu lên Server:

- Khi **Client** nhập vào thanh tìm kiếm và tìm thông tin tức là gửi yêu cầu POST lên **Proxy** và được **Proxy** gửi lên **Server**, khi được chấp thuận **Proxy** sẽ lấy dữ liệu thông tin từ **Server** để trả về cho **Client**.

\*Vì Wireshark không hỗ trợ bắt gói tin của localhost traffic nên không có ảnh minh hoạ.

## V. Cách chạy chương trình và kết quả:

### 1. Chưa có dữ liệu cache:

- Thực hiện truy cập trang web <http://hcmut.edu.vn/>

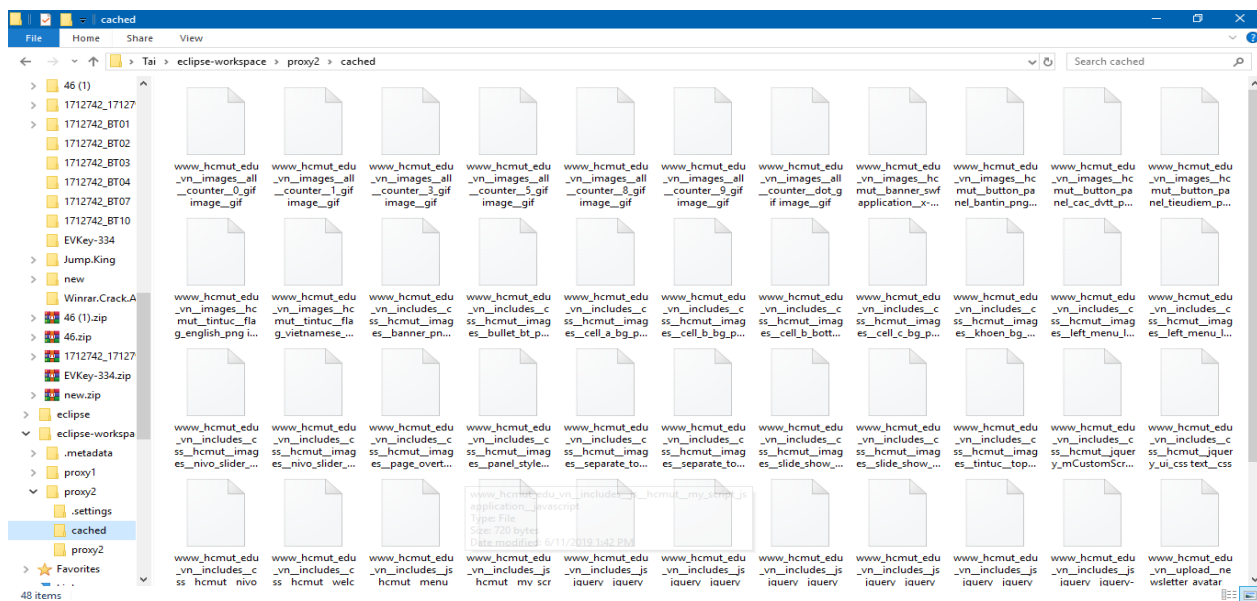
```
Proxy [Java Application] C:\Program Files\Java\jdk-12.0.1\bin\javaw.exe (Jun 11, 2019, 12:07:46 PM)
Create new request for GET http://www.hcmut.edu.vn/images/all/counter/dot.gif HTTP/1.1
0
Create new request for GET http://www.hcmut.edu.vn/images/all/counter/8.gif HTTP/1.1
Create new request for GET http://www.hcmut.edu.vn/images/all/counter/8.gif HTTP/1.1
Create new request for GET http://www.hcmut.edu.vn/images/all/counter/8.gif HTTP/1.1
0
Create new request for GET http://www.hcmut.edu.vn/images/all/counter/5.gif HTTP/1.1
Create new request for GET http://www.hcmut.edu.vn/images/all/counter/5.gif HTTP/1.1
Create new request for GET http://www.hcmut.edu.vn/images/all/counter/5.gif HTTP/1.1
0
Create new request for GET http://www.hcmut.edu.vn/images/all/counter/6.gif HTTP/1.1
Create new request for GET http://www.hcmut.edu.vn/images/all/counter/6.gif HTTP/1.1
Create new request for GET http://www.hcmut.edu.vn/images/all/counter/6.gif HTTP/1.1
0
Create new request for GET http://www.hcmut.edu.vn/includes/css/hcmut/images/tintuc/top_menu_bg.png HTTP/1.1
Create new request for GET http://www.hcmut.edu.vn/includes/css/hcmut/images/tintuc/top_menu_bg.png HTTP/1.1
Create new request for GET http://www.hcmut.edu.vn/includes/css/hcmut/images/tintuc/top_menu_bg.png HTTP/1.1
0
Create new request for GET http://www.hcmut.edu.vn/includes/css/hcmut/images/nivo_slider/arrows.png HTTP/1.1
Create new request for GET http://www.hcmut.edu.vn/includes/css/hcmut/images/nivo_slider/arrows.png HTTP/1.1
Create new request for GET http://www.hcmut.edu.vn/includes/css/hcmut/images/nivo_slider/arrows.png HTTP/1.1
0
Create new request for GET http://www.hcmut.edu.vn/includes/css/hcmut/images/page_overturn.png HTTP/1.1
Create new request for GET http://www.hcmut.edu.vn/includes/css/hcmut/images/page_overturn.png HTTP/1.1
Create new request for GET http://www.hcmut.edu.vn/includes/css/hcmut/images/page_overturn.png HTTP/1.1
0
Create new request for GET http://www.hcmut.edu.vn/includes/css/hcmut/images/cell_c_bg.png HTTP/1.1
Create new request for GET http://www.hcmut.edu.vn/includes/css/hcmut/images/cell_c_bg.png HTTP/1.1
Create new request for GET http://www.hcmut.edu.vn/includes/css/hcmut/images/cell_c_bg.png HTTP/1.1
0
Create new request for GET http://www.hcmut.edu.vn/includes/css/hcmut/images/cell_b_bottom_bg.png HTTP/1.1
Create new request for GET http://www.hcmut.edu.vn/includes/css/hcmut/images/cell_b_bottom_bg.png HTTP/1.1
Create new request for GET http://www.hcmut.edu.vn/includes/css/hcmut/images/cell_b_bottom_bg.png HTTP/1.1
0
Create new request for GET http://www.hcmut.edu.vn/favicon.ico HTTP/1.1
Create new request for GET http://www.hcmut.edu.vn/favicon.ico HTTP/1.1
Create new request for GET http://www.hcmut.edu.vn/favicon.ico HTTP/1.1
0
```

- Trên console xuất hiện các request GET cho các phần chưa cache của trang web

- Và kết quả thu được:



- Kiểm tra trong thư mục cached



## 2. Đã có dữ liệu cache:

- Vẫn là trang web trên nhưng sau khi đã cache thì khi F5 proxy sẽ không gửi lại yêu cầu mà chỉ lấy từ cache đưa lên. Do đó tốc độ load gần bằng tốc độ ánh sáng.

## 3. Truy xuất dữ liệu:

- Để thử truy xuất dữ liệu, ta truy cập vào một trang web có phần tìm kiếm, cụ thể ở đây minh họa sẽ là trang web <http://hcmute.edu.vn/>
- Nhập từ khoá, chẳng hạn như “Đăng”



- Nhấn Tìm và đây là kết quả trong giao diện console

```
0
Create new request for GET http://hcmute.edu.vn/Resources/Imagephoto/logo.png HTTP/1.1
Create new request for GET http://hcmute.edu.vn/Resources/Imagephoto/logo.png HTTP/1.1
Create new request for GET http://hcmute.edu.vn/Resources/Imagephoto/logo.png HTTP/1.1
0
Create new request for CONNECT beacons3.gvt2.com:443 HTTP/1.1
0
Create new request for POST http://hcmute.edu.vn/Services/CMS.aspx/SearchAndPaging HTTP/1.1
```

- Tức thì, bên trình duyệt đã hiện kết quả truy xuất cho chúng ta



#### 4. Thực hiện chặn domain:

- Chúng ta không muốn người dùng cấp dưới truy cập trang web nào đó, ví dụ như <http://kenh14.vn/> thì chỉ việc copy paste URL hoặc domain vào console và enter

```
Proxy [Java Application] C:\Program Files\Java\jdk-12.0.1\bin\javaw.exe (Jun 11, 2019, 12:56:00 PM)
http://kenh14.vn/
|
kenh14.vn blocked successfully

Enter new site to block, or type "blocked" to see blocked sites or "close" to
close server or "delete" to delete all caches, or "unblock + domain/url" to
unblock.
```

#### 5. Proxy chặn domain của trang web truy cập:

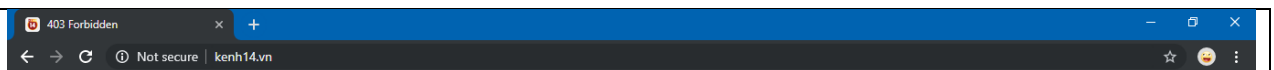
- Tiếp theo ta thử truy cập vào trang web vừa mới chặn thì giao diện console cho kết quả

```
Proxy [Java Application] C:\Program Files\Java\jdk-12.0.1\bin\javaw.exe (Jun 11, 2019, 12:56:00 PM)
http://kenh14.vn/

kenh14.vn blocked successfully

Enter new site to block, or type "blocked" to see blocked sites or "close" to
close server or "delete" to delete all caches, or "unblock + domain/url" to
unblock.
0
Create new request for CONNECT www.google.com:443 HTTP/1.1
0
Create new request for GET http://kenh14.vn/ HTTP/1.1
Create new request for GET http://kenh14.vn/ HTTP/1.1
Blocked site requested : http://kenh14.vn/
```

- Và ngay tức thì bên trình duyệt trả về



403 (Forbidden) HTTPs response

## 6. Xem danh sách domain đã chặn:

- Để xem danh sách các domain đã chặn thì ta chỉ việc gõ lệnh “blocked” ở giao diện console

```
Enter new site to block, or type "blocked" to see blocked sites or "close" to  
close server or "delete" to delete all caches, or "unblock + domain/url" to  
unblock.
```

```
blocked
```

```
Currently Blocked Sites
```

```
ngoctrinh.vip  
chinhphu.vn  
www.congtymmt.com  
g2esport.com  
superproxyever.net  
how.homosapiens.com  
kenh14.vn  
laptrinhitmobile.xyz
```

## 7. Bỏ chặn:

- Ngoài ra ta có thể bỏ chặn một domain nào đó

```
Enter new site to block, or type "blocked" to see blocked sites or "close" to  
close server or "delete" to delete all caches, or "unblock + domain/url" to  
unblock.
```

```
unblock laptrinhitmobile.xyz
```

```
laptrinhitmobile.xyz unblocked successfully
```

- Kiểm tra lại trong danh sách chặn

```
blocked
```

```
Currently Blocked Sites
```

```
ngoctrinh.vip  
chinhphu.vn  
www.congtymmt.com  
g2esport.com  
superproxyever.net  
how.homosapiens.com  
kenh14.vn
```

## 8. Xóa dữ liệu cache

- Dùng lệnh “delete” để xóa tất cả dữ liệu đã cache về

```
Enter new site to block, or type "blocked" to see blocked sites or "close" to close server or "delete" to delete all caches, or
delete
File : clientservices_googleapis_com_chrome-variations__seed__osname=win&channel=stable&milestone=74 application__octet-stream
Has been deleted successfully!!
File : www_hcmut_edu_vn_images__all__counter__0_gif image__gif
Has been deleted successfully!!
File : www_hcmut_edu_vn_images__all__counter__1_gif image__gif
Has been deleted successfully!!
```

## 9. Đóng server, ghi dữ liệu chặn/cache xuống file:

- Dùng lệnh close

```
close

Closing Server..
Cached Sites written
Blocked Site list saved
Terminating Connection
Server closed
```

# TÀI LIỆU THAM KHẢO

1. Tài liệu của giáo viên và bộ môn
2. <https://github.com/wundulo/proxy>
3. <https://github.com/stefano-lupo/Java-Proxy-Server>
4. [https://www.ntu.edu.sg/home/ehchua/programming/webprogramming/HTTP\\_Basics.html](https://www.ntu.edu.sg/home/ehchua/programming/webprogramming/HTTP_Basics.html)
5. <https://www.w3.org/Protocols/rfc2616/rfc2616.txt>
6. <https://www.w3.org/Protocols/rfc2616/rfc2616-sec6.html>
7. <https://www.mkyong.com/java/how-to-send-http-request-getpost-in-java/>