

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN SOCKET

BỘ MÔN MẠNG MÁY TÍNH

Sinh viên thực hiện :

- **NGUYỄN THÀNH SỬU**
MSSV: 1712736

Lớp: 19CTT3

GVHD: LÊ HÀ MINH

Tp. Hồ Chí Minh, tháng 12 năm 2020.

MỤC LỤC

MỤC LỤC.....	1
PHẦN I. THÔNG TIN MÔN HỌC.....	2
PHẦN II. MÔ TẢ PROJECT VÀ CÁC HÀM XỬ LÝ CHÍNH.....	2
1. Mô tả project.....	2
2. Hướng giải quyết.....	2
3. Các hàm xử lý chính.....	2
PHẦN III. CÁCH CHẠY CHƯƠNG TRÌNH VÀ KẾT QUẢ CHẠY ĐƯỢC	5
PHẦN IV. BẢNG ĐÁNH GIÁ MỨC ĐỘ HOÀN THÀNH.....	9
PHẦN V. TÀI LIỆU THAM KHẢO.....	10

PHẦN I. THÔNG TIN MÔN HỌC

- **MÔN HỌC:** MẠNG MÁY TÍNH
- **GIẢNG VIÊN HƯỚNG DẪN:** LÊ HÀ MINH
- **SINH VIÊN THỰC HIỆN:**
 1. NGUYỄN THÀNH SỬU_MSSV: 1712736
- **LỚP:** 19CTT3
- **ĐỒ ÁN1:** ĐỒ ÁN SOCKET

PHẦN II. MÔ TẢ PROJECT VÀ CÁC HÀM XỬ LÝ CHÍNH

1. Mô tả project

Viết một chương trình Proxy Server sử dụng cho HTTP thỏa mãn những yêu cầu sau

1. Chương trình cho phép Client truy cập website thông qua Proxy Server (GET, POST)
2. Chương trình chỉ cần hỗ trợ http, không cần https.
3. Proxy Server phải xử lý đồng thời được các request từ client.
4. Proxy Server sẽ chạy trên port 8888.
5. Proxy Server cấu hình 1 file blacklist.conf, mỗi dòng chứa các domain website cấm Client truy cập. Proxy Server phải chặn tất cả các truy cập trùng với các domain trong file.

Ví dụ:

abc.com.vn

www.thongtin.net

...

2. Hướng giải quyết

- Project được thực hiện với ngôn ngữ C++ trên Microsoft Visual Studio 2017.
- Sử dụng thư viện "afxsock.h"
- Tạo 1 proxy socket như yêu cầu của đề án để can thiệp giữa Client và Server .
- Ý tưởng tạo 2 socket , 1 cái giao tiếp với client (có thể dùng nó làm client luôn) và 1 cái giao tiếp với Server (gọi là serverSocket)

3. Các hàm xử lý chính

*Các cấu trúc dữ liệu

```
class httpRequest
{
public:
    char method[4];
    char url[1024];
    char host[1024];
    char cookie[1024 * 10];
    httpRequest()
    {
        ZeroMemory(this, sizeof(httpRequest));
    }
};
```

class httpRequest: thông tin gói tin request từ client gửi đến proxy

với method: GET/POST

url : địa chỉ url yêu cầu

host : tên máy chủ

cookie

```
struct cache_httpHeaderRequest
{
    char method[4];
    char url[1024];
    char host[1024];
    cache_httpHeaderRequest()
    {
        ZeroMemory(this, sizeof(cache_httpHeaderRequest));
    }
};
```

Struct để lưu thông tin request trong cache

```
struct CACHE
{
    cache_httpHeaderRequest cache_req ;
    char buffer[MAXSIZE];
    char dateTime[40];
    CACHE()
    {
        ZeroMemory(this->buffer, MAXSIZE);
        ZeroMemory(this->dateTime, sizeof(dateTime));
    }
};
```

struct CACHE với các thuộc tính

cache_req: thông tin gói tin request

buffer: gói tin response từ server

dateTime: If-Modified-Since: dateTime

```
struct ProxyParam {
    SOCKET clientSocket;
    SOCKET serverSocket;
};
```

cấu trúc 1 proxy server, gồm 2 socket đóng vai trò client và server

*Các hàm hỗ trợ:

```
bool initialSocket()
```

- **Tham số:** không có tham số
- **Chức năng:** khởi tạo socket(ProxyServer với port 8888)
- **Trả về:** true/false

void parseHttpRequest(**char** *buffer, **HttpRequest** * httpRequest)

- **Tham số:** buffer (là chuỗi (request) nhận được từ client), httpRequest
- **Chức năng:** phân tích buffer thành httpRequest với httpRequest là một class đã được định nghĩa trong source
- **Trả về:** không trả về

bool lsequal(**cache_httpHeaderRequest** cache_rq, **HttpRequest** rq)

- **Tham số:** cache_rq (được lưu trong mảng cache), rq (request từ client)
- **Chức năng:** so sánh rq có giống cache_rq, kiểm tra request đã được yêu cầu trước đó và lưu trong bộ đệm chưa
- **Trả về:** true nếu request đã được yêu cầu trước đó, ngược lại trả về false

string message_403()

- **Tham số:** không có tham số
- **Chức năng:** tạo chuỗi html xuất thông báo HTTP response 403 (Forbidden) Khi Client truy cập các trang web bị cấm
- **Trả về:** chuỗi html(**string**)

bool ConnectToServer(**SOCKET** *serverSocket, **char** *host)

- **Tham số:** serverSocket, host
- **Chức năng:** tạo kết nối từ serverSocket tới Server với host truyền vào
- **Trả về:** true nếu kết nối thành công, ngược lại trả về false

bool goodWeb(**char*** host, **ifstream** &fIn)

- **Tham số:** host, fIn
- **Chức năng:** xác định xem host có tồn tại trong file blacklist.conf không?
- **Trả về:** true nếu trang web yêu cầu không có trong blacklist và khi không có file blacklist. Ngược lại trả về false khi trang web yêu cầu trong blacklist

string add_IfModifiedSince(**string** buffer, **string** dateTime)

- **Tham số:** buffer, dateTime
- **Chức năng:** thêm dòng If-Modified-Since: dateTime vào chuỗi buffer(request)
- **Trả về:** chuỗi đã thêm

string getDateFromLastModified(**string** buffer)

- **Tham số:** buffer
- **Chức năng:** lấy dateTime từ dòng LastModified: của response
- **Trả về:** chuỗi (dateTime)

void insert(**CACHE** C)

- **Tham số:** bộ đệm C
- **Chức năng:** insert 1 CACHE C vào mảng cache(toàn cục) (CACHE là một struct đã được định nghĩa trong source)
- **Trả về:** không trả về

int Cache_find(**CACHE** *cache, **HttpRequest** rq)

- **Tham số:** mảng cache kiểu CACHE tập hợp các yêu cầu trước đó của client, rq gói http request hiện tại.
- **Chức năng:** tìm xem rq có nằm trong mảng cache(toàn cục) không?
- **Trả về:** vị trí của rq trong mảng cache

unsigned int __stdcall ProxyThread(void* lpParameter)

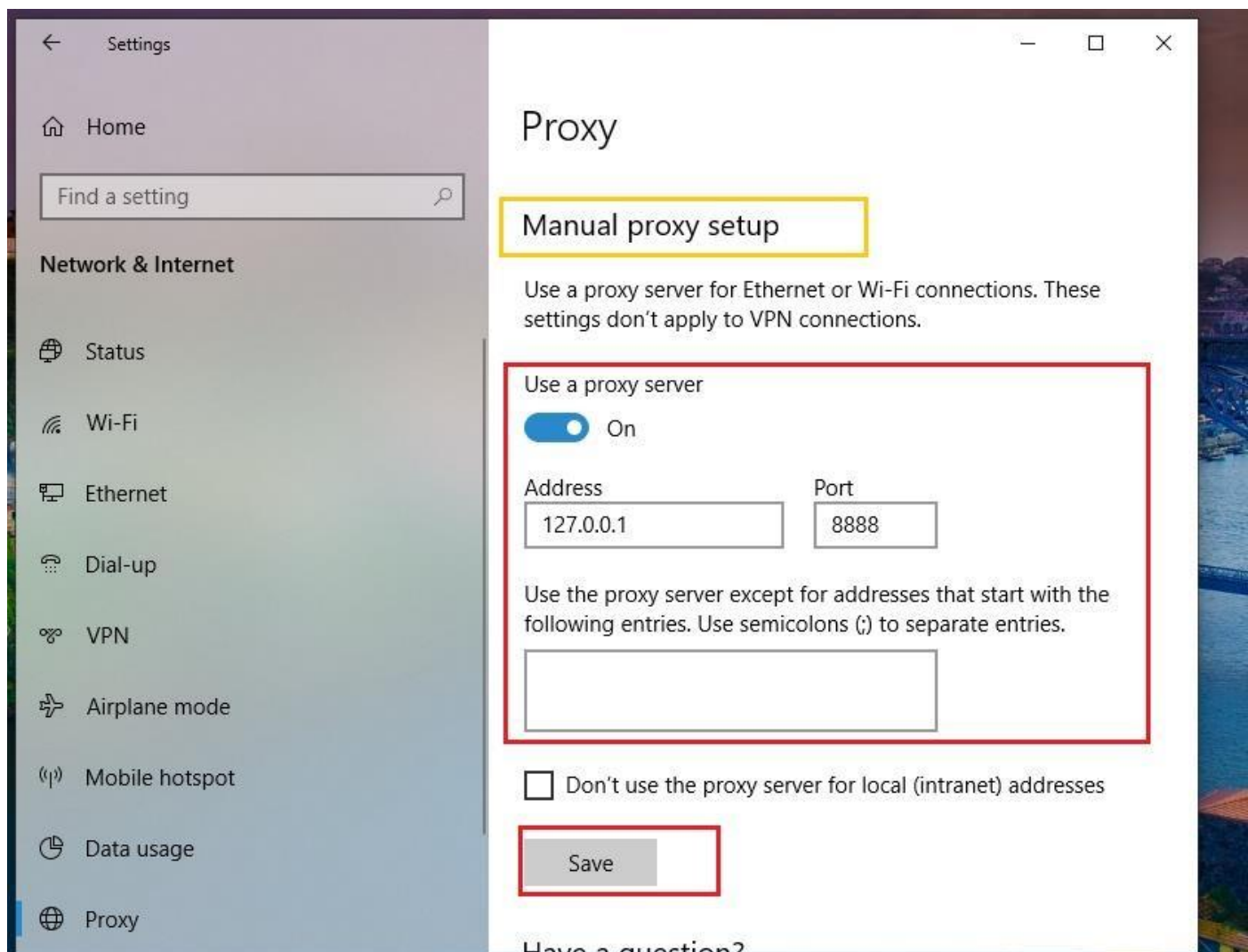
- **Tham số:** lpParameter-2 socket, 1 cái giao tiếp với client (có thể dùng nó làm client luôn) và 1 cái giao tiếp với Server (gọi là serverSocket)
- **Chức năng:** tạo ra 1 luồng xử lý từ lúc nhận request từ client , thông qua proxy , xong trả response về client.
- **Trả về:** 0

PHẦN III. CÁCH CHẠY CHƯƠNG TRÌNH VÀ KẾT QUẢ CHẠY ĐƯỢC

1. Cách chạy chương trình

Bước 1: Mở và thiết lập cài đặt proxy cho các trình duyệt

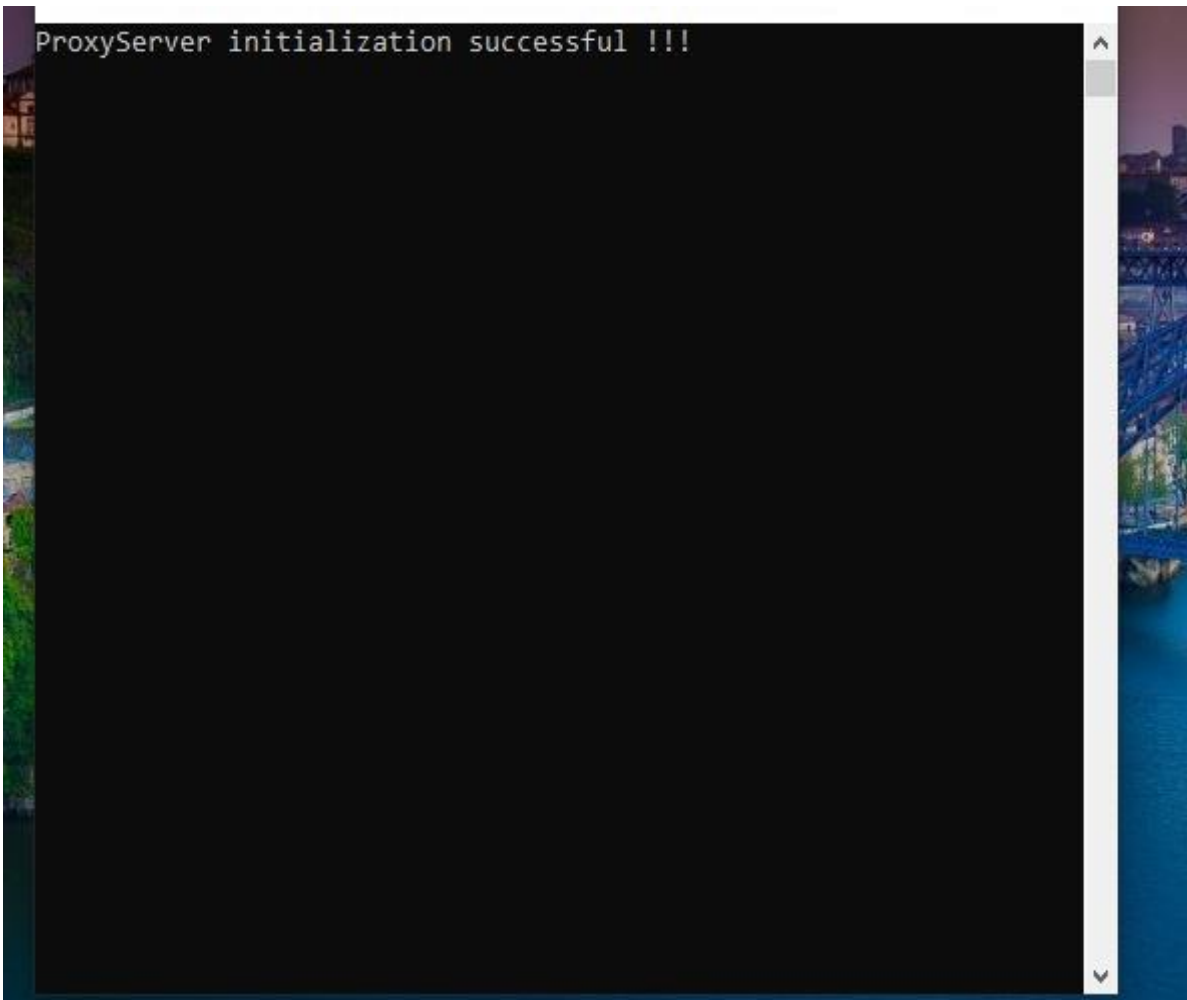
Vào Setting – Network & Internet – Proxy – Manage proxy setup



Cấu hình các thông số như trên hình. Với Address là 127.0.0.1 là địa chỉ loopback, port 8888 là port của proxy

Bước 2: mở file exe trong thư mục release của source code và chạy, để file balcklist.conf chung thư mục với release, nếu không có blacklist.conf sẽ mặc định cho truy cập tất cả trang web

Để khởi tạo proxy thành công và đợi các kết nối từ client



Bước 3: Mở trình duyệt web (Firefox hoặc Google Chrome) và nhập trang web cần truy cập vào thanh địa chỉ, lưu ý chỉ những trang HTTP, và những trang HTTP này không yêu cầu truy cập đến HTTPS. Xóa hết bộ đệm duyệt web trước khi truy cập.

Một số trang HTTP

<http://www.lotteria.vn/>

http://wiki.ross-tech.com/wiki/index.php/Main_Page

<http://www.vietcombank.com.vn/>

<http://tracuunnt.gdt.gov.vn/tcnnt/mstdn.jsp>

<http://www.noip.gov.vn/>

<http://www.hoangvancong.com/>

<http://wiki.ros.org/>

<http://www.partow.net/programming/>

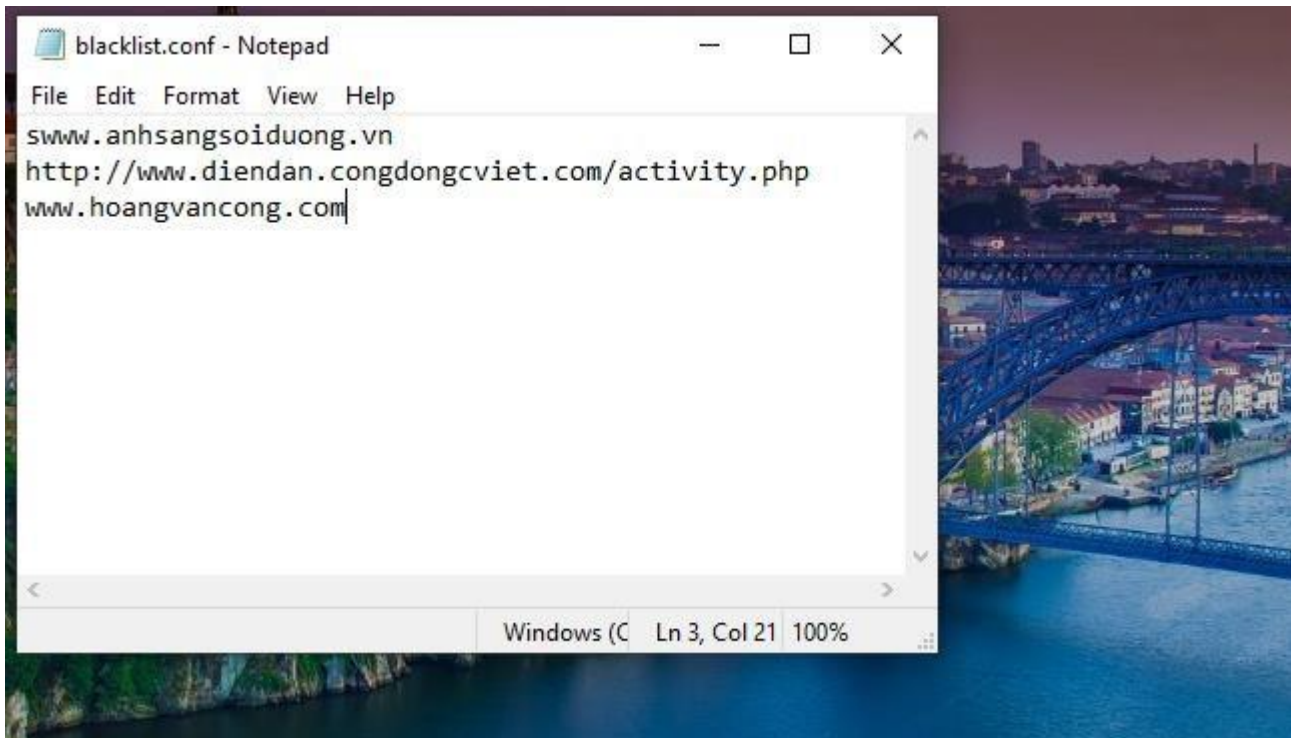
<http://thisinh.thithptquocgia.edu.vn/Account/Login?ReturnUrl=%2f>

Nếu trang web cần truy cập nằm trong blacklist.conf thì sẽ không cho truy cập

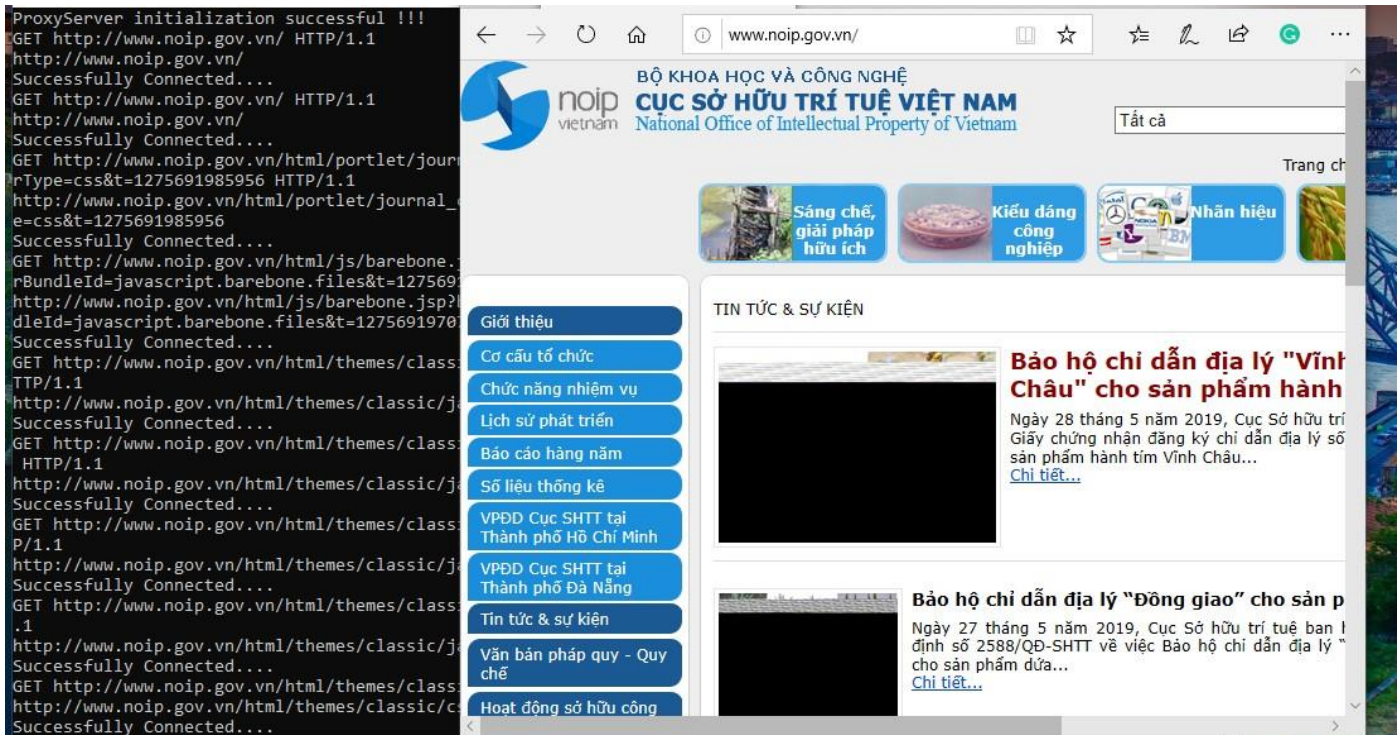
2. Kết quả đạt được

Dưới đây là minh họa 1 số trường hợp

File blacklist.conf chứa các trang web bị cấm như sau:

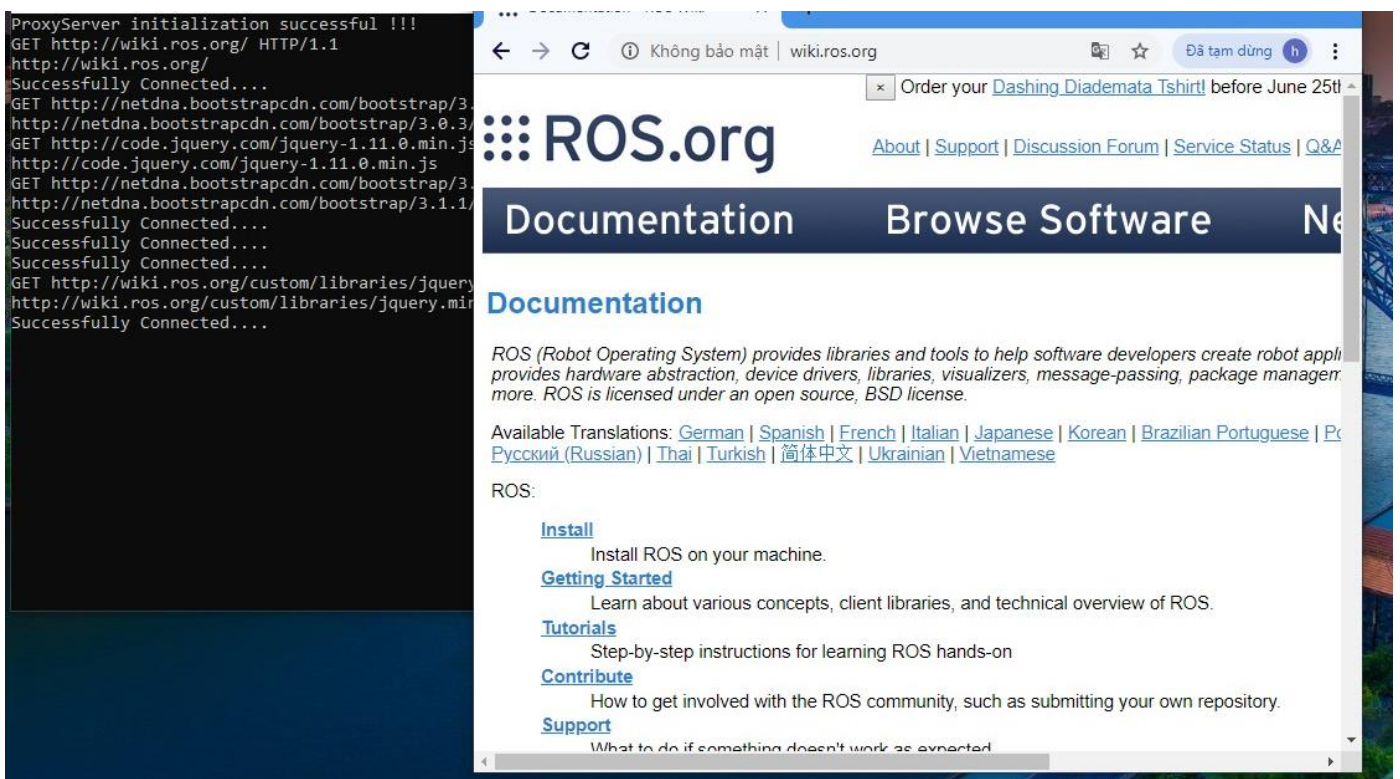


- Truy cập trang <http://www.noip.gov.vn/>

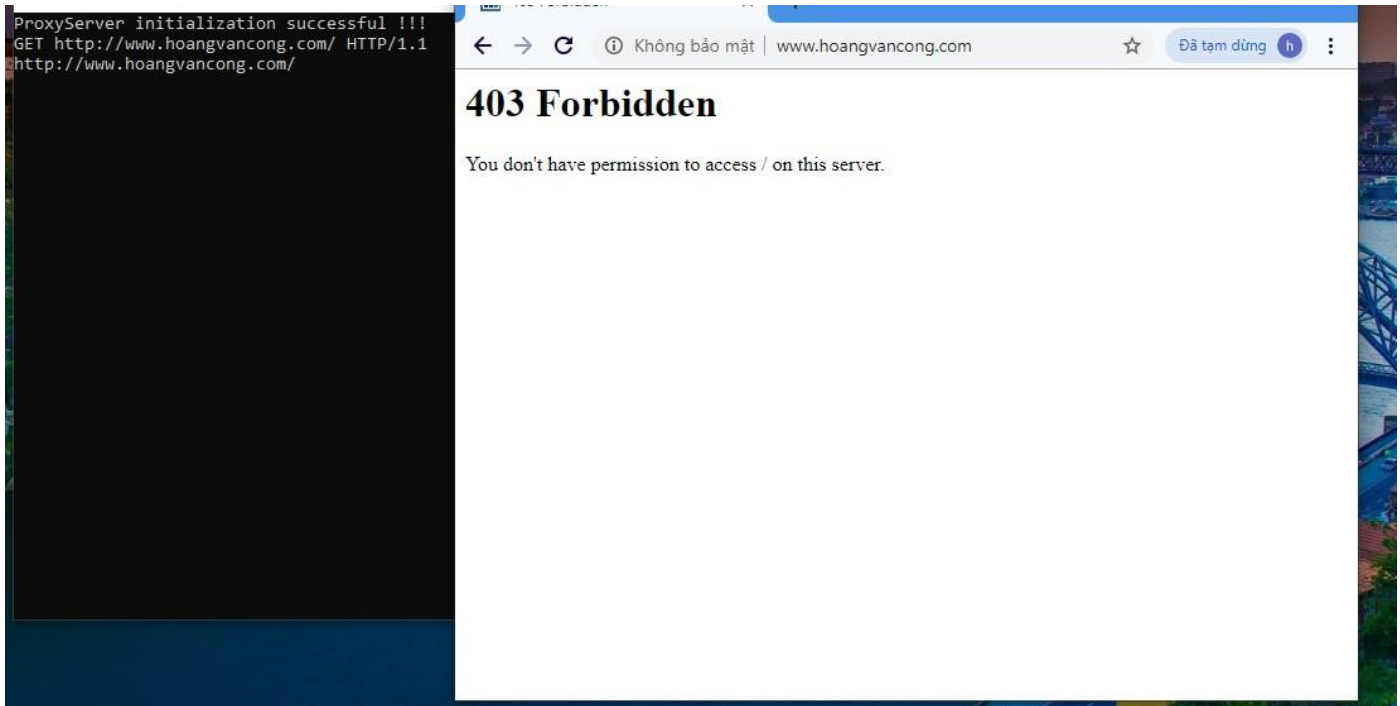


Truy cập thành công.

- Truy cập trang <http://wiki.ros.org/> bằng Google Chrome



- Truy cập trang trong blacklist www.hoangvancong.com



PHẦN IV. BẢNG ĐÁNH GIÁ MỨC ĐỘ HOÀN THÀNH

YÊU CẦU	NỘI DUNG	MỨC ĐỘ HOÀN THÀNH (%)	GHI CHÚ
1. VIẾT MỘT CHƯƠNG TRÌNH PROXY SERVER SỬ DỤNG CHO HTTP	1. cho phép Client truy cập website thông qua Proxy Server (GET, POST)	100	
	2. Hỗ trợ http, không cần https.	100	
	3. Proxy Server phải xử lý đồng thời được các request từ client	100	
	4. Proxy Server sẽ chạy trên port 8888	100	
	5. Proxy Server cấu hình 1 file blacklist.conf	100	
	TỔNG HOÀN THÀNH	100	
2. TEST CHƯƠNG TRÌNH	1. Sử dụng web browser truy cập và hiển thị được nội dung trang web	100	
	2. Khi Client truy cập các trang bị cấm, Client nhận được kết quả 403	100	

	TỔNG HOÀN THÀNH	100	
3.BÁO CÁO	Viết báo cáo hoàn thành các yêu cầu	100	

Nhận xét mức độ hoàn thành đồ án: 100%

PHẦN V. TÀI LIỆU THAM KHẢO

I. SÁCH:

- [1]. Slide Mạng máy tính, Khoa Công Nghệ Thông Tin, Trường ĐH Khoa Học Tự Nhiên
- [2]. *Giáo trình Mạng máy tính*, NXB Khoa học và Kỹ Thuật, 2016

II. INTERNET:

- [1]. <http://tinhocvankhanhsv.blogspot.com/2015/06/lap-trinh-mang-voi-thu-vien-winsock.html>
- [2]. <http://www.partow.net/programming/tcpproxy/index.html>
- [3]. <https://viblo.asia/p/hieu-hon-ve-trinh-duyet-web-aWj53vko16m>
- [4]. <https://kipalog.com/posts/Tim-hieu-ve-lap-trinh-socket---Buoc-dau-lam-quen>
- [5]. <https://sites.google.com/site/embedded247/npcourse/lap-trinh-c-socket>
- [6]. <https://viblo.asia/p/phuong-thuc-get-va-post-aWj53VB16m>
- [7]. <https://www.slideshare.net/LeeNamNguyen/bai-4-lap-trnh-phia-client>
- [8]. <https://quantrimang.com/ly-thuyet-proxy-la-gi-117220>
- [9]. https://vi.wikipedia.org/wiki/M%C3%A1y_ch%E1%BB%A7_proxy
- [10]. https://developer.mozilla.org/en-US/docs/Web/HTTP/Caching?fbclid=IwAR17Jv1q1dFZWShG06ZA627XQDivs4uM3gfoKk6I_NpjC_j3h6AKv4kdIGc