

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN THÀNH PHỐ HỒ CHÍ MINH**



ĐỒ ÁN MẠNG MÁY TÍNH

Môn: Mạng máy tính

LẬP TRÌNH SOCKET

MÃ MÔN HỌC
CSC100003

Thành phố Hồ Chí Minh - 2020

Mục Lục

| | |
|--|----------|
| 1. THÔNG TIN: | 3 |
| 2. THỰC HIỆN | 4 |
| 2.1 Xử lý Web server khi Client truy cập lên server (cụ thể đây là Index.html) | 4 |
| 2.2 Xử lý đăng nhập trên Web Client: POST Username và Password lên Server: | 5 |
| 2.3 Download File : | 7 |
| 3. REFERENCE: | 8 |

1. THÔNG TIN:

Giảng viên

- Đỗ Hoàng Cường
- Huỳnh Thị Bảo Trân

Sinh viên:

| MSSV | Tên |
|----------|-------------------|
| 19127422 | Nguyễn Đức Huy |
| 19127525 | Nguyễn Thanh Quân |

2. THỰC HIỆN

2.1 Xử lý Web server khi Client truy cập lên server (cụ thể đây là Index.html)

Khi Client truy cập lên: [Http://127.0.0.1:8080/index.html](http://127.0.0.1:8080/index.html)

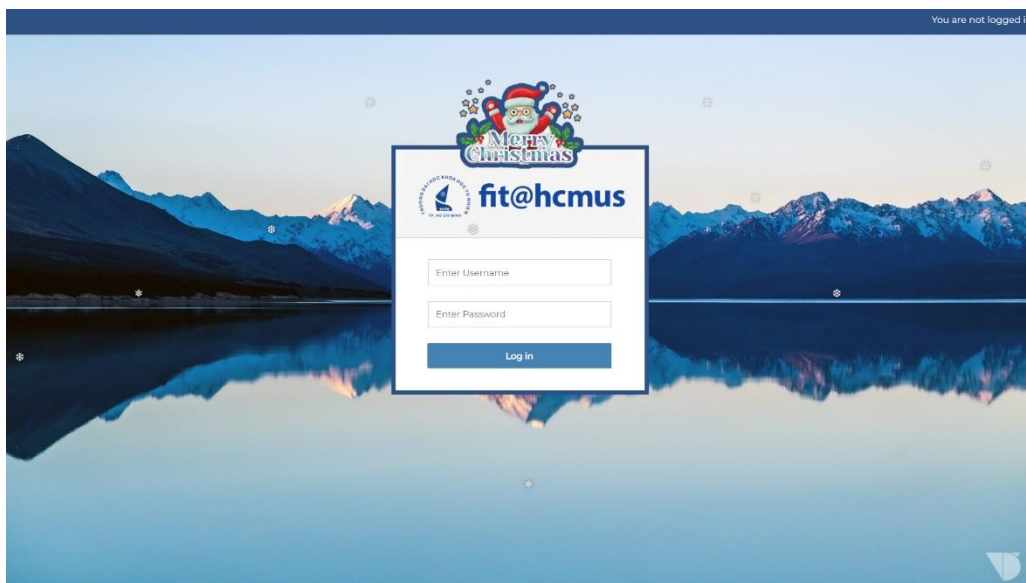
Thì khi đó request (yêu cầu) của Client gửi lên server có dạng là:

```
'GET /index.html HTTP/1.1'
```

Xử lý yêu cầu của Client tại Server: Gửi response (đáp lại) bằng mã trạng thái **200 OK** tức là phản hồi tiêu chuẩn cho các yêu cầu Http của Client đã được xử lý thành công, đồng thời khi ấy Server cũng gửi toàn bộ data (dữ liệu) của file index.html cho client

```
'HTTP/1.1 200 OK'
```

Khi đó giao diện trên browser web sẽ hiển thị như hình bên dưới:



Bonus thêm: thay vì code html thuần túy nhóm em sử dụng thêm css để tạo hiệu ứng tuyệt vời cho các trang html

Truy cập sai địa chỉ: thay vì truy cập lên [Http://127.0.0.1:8080/index.html](http://127.0.0.1:8080/index.html), nhập sai địa chỉ khác với địa chỉ trên.

Thì khi đó request (yêu cầu) của Client gửi lên Server sẽ có dạng: GET /... HTTP/1.1 (... ở đây là tùy ý)

Cách xác định khi request sai yêu cầu: Thay vì dùng method bắt exception, nhóm em sẽ dùng 1 cách khác đơn giản hơn và dễ hiểu hơn đó là tận dụng method split trong python để tách nhỏ yêu cầu của Client

```

Client, Request = ReadHTTPRequest(Server)
print("-----HTTP request: ")
print(Request)
# 4. Send HTTP Response + 5. Close Server
temp = Request
string_list = temp.split(" ")
file = string_list[1].lstrip("/")

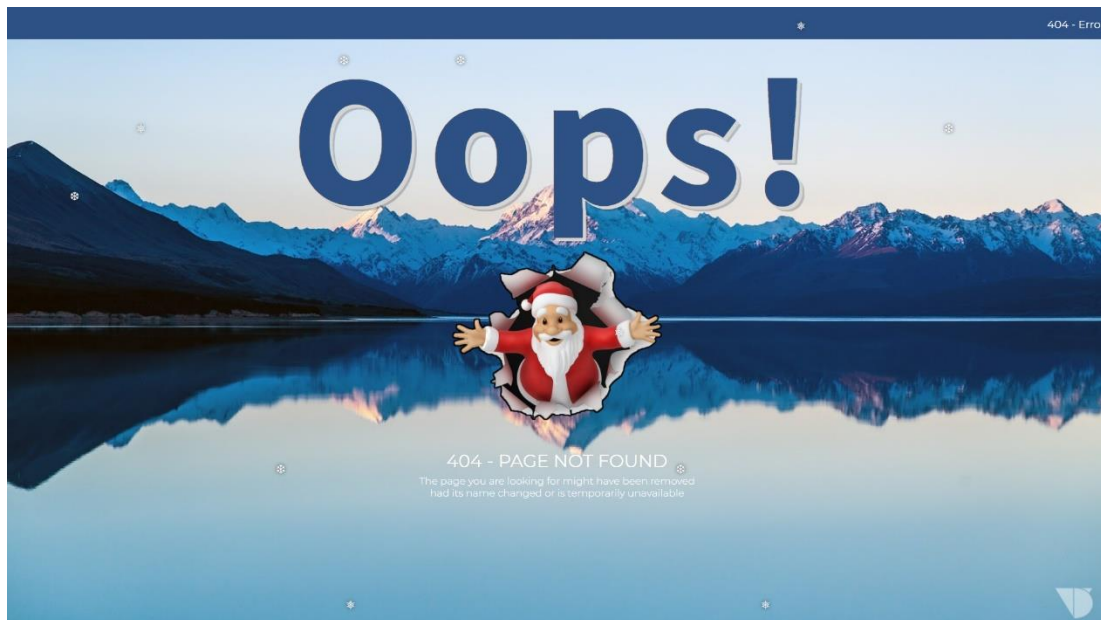
```

Khi ấy ta sẽ tách được yêu cầu sau dấu / trong request để xác định địa chỉ có sai.

Nếu địa chỉ khác với index.html thì trên Server sẽ xử lý: Gửi response (đáp lại) bằng mã trạng thái **301 Moved Permanently** tức là chuyển hướng Client đến một vị trí mới của file cụ thể ở đây là file 404.html

'HTTP/1.1 301 Moved Permanently'

Khi ấy hiển thị tại web browser:



2.2 Xử lý đăng nhập trên Web Client: POST Username và Password lên Server:

Khi Client nhập Username và Password trên trang đăng nhập index.html thì tại file index.html chúng ta sẽ dùng method POST để Server có thể nhận được yêu cầu đăng nhập từ Client, như ảnh dưới thông tin Client nhập vào sẽ được post trong port 10000,

```

<form action="http://127.0.0.1:10000" method="post">
<tr>
  <td style="text-align: center;">
    <input type="text" placeholder="Enter Username" name="Username" style="margin-top: 30px;">
  </td>
</tr>
<tr>

```

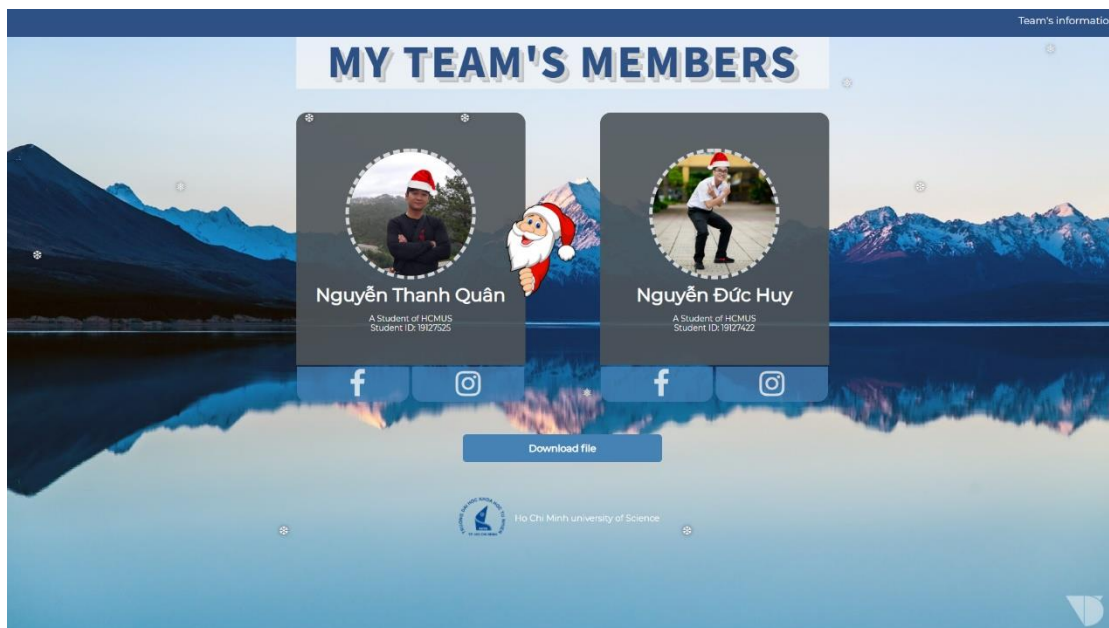
Khi ấy request nhận được từ Client sẽ có dạng: Lưu ý: ... ở đây là tùy ý vào Client

```
"POST / HTTP/1.1"
"Username=...&Password=..."
```

Nếu Client nhập đúng cả Username và Password (tức là admin) thì tại Server sẽ xử lý như sau: Server sẽ response (đáp lại) Client bằng mã trạng thái **301 Moved Permanently** tức là chuyển hướng Client đến một vị trí mới của file cụ thể ở đây là file info.html

```
header = ""HTTP/1.1 301 Moved Permanently
Location: http://127.0.0.1:9090/info.html
```

Đồng thời, Server cũng gửi cho Client data (dữ liệu) của file info.html
Khi ấy trên web browser sẽ hiển thị như sau:



Nếu nhập sai một trong hai hoặc cả hai Username hoặc Password thì tại Server sẽ xử lý như sau: Server sẽ response (đáp lại) Client bằng mã trạng thái 301 Moved Permanently tức là chuyển hướng Client đến một vị trí mới của file cụ thể ở đây là file 404.html

```
header = ""HTTP/1.1 301 Moved Permanently
Location: http://127.0.0.1:7070/404.html
```

Đồng thời Server cũng gửi cho Client data (dữ liệu) của file 404.html
Khi ấy trên web browser sẽ hiển thị như sau:



2.3 Download File :

Lưu ý: ở đây nhóm chúng em không thể cài đặt download bằng Transfer encoding: Chunk nên nhóm em sẽ thực hiện download bằng Content Length

Khi ta chọn file để download,
Client sẽ gửi yêu cầu tải file được chọn

```
GET /?submit=Lorem+text.txt HTTP/1.1
```

ở ví dụ này, client gửi yêu cầu muốn tải file Lorem text.txt

Server sẽ kiểm tra xem Client yêu cầu gửi file nào trong 6 file

```
def Download(Server, Client, Request):  
    file = ["Lorem_text.txt", "background.jpg", "Project_Socket_Programming_112020.pdf",  
           "01_Ethernet_Fundamentals.ppt", "HoaNoKhongMauAcoustic-HoaiLam.mp3", "Hoa_No_Khong_Mau.mp4"]  
    for f in file:  
        if f in Request:  
            SendFile(Client, "./Download/" + f)  
            return True
```

Sau đó Server sẽ chấp nhận yêu cầu và gửi phản hồi qua function SendFile

Đây là format cho header cần gửi của content-length format. Đồng thời với header còn có dữ liệu (data) của file cần download về cho Client.

```
def SendFile(Client, FileName): # server send file for client by Content Length
    with open(FileName, 'rb') as f:
        L = f.read()

        header = "HTTP/1.1 200 OK"
        header += "\nContent-Disposition: attachment; filename=" + \
            os.path.basename(FileName)
        header += "\nAccept-Ranges: bytes"
        header += "\nContent-Length: %d" % len(L)
        header += "\nConnection: close"
        header += "\n\n"
        PrintRequest_Response(header, "Response")
        header = bytes(header, 'utf-8') + L
        Client.send(header)
```

Đây là header response Server gửi cho Client

```
-----HTTP Response :
HTTP/1.1 200 OK
Content-Disposition: attachment; filename=Lorem_text.txt
Content-Length: 10240
Connection: close
```

Trong hình bên trên,

HTTP/1.1 200 OK : phản hồi yêu cầu HTTP thành công

Content-Dispsition: attachment; filename=Lorem text.txt: download file Lorem_text.txt

Content-Length: 10240 độ dài của file gửi (bytes)

Connection: close: đóng kết nối lại

Có nghĩa là Server đã gửi thành công file Lorem_text.txt có độ dài 10240 cho Client download sau khi download xong sẽ đóng kết nối

3. REFERENCE:

- [1]. [List of HTTP header fields - Wikipedia](#)
- [2]. [Network socket - Wikipedia](#)
- [3]. [TcpCommunication - Python Wiki](#)
- [4]. [Python 3 - simple HTTP Request with the Socket module | HackLAB \(geeks3d.com\)](#)
- [5]. [HowTo/Sockets - Python Wiki](#)