

BÀI TẬP THỰC HÀNH

MÔN HỌC: HỆ PHÂN TÁN

CHƯƠNG 1: TỔNG QUAN HỆ PHÂN TÁN

HỌ TÊN SV: Mạc Quang Huy

MSSV: 20173169

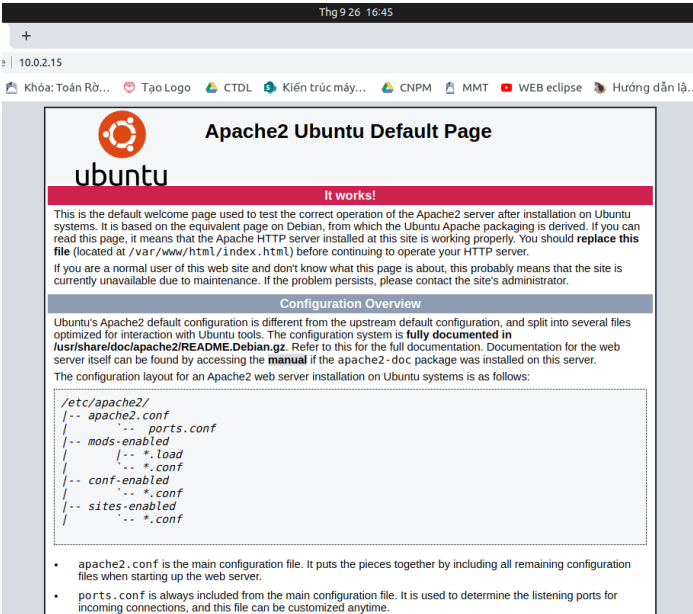
MÃ LỚP: 118636

MÃ HỌC PHẦN: IT4611

Câu hỏi 1: Đường dẫn đến file html chứa nội dung mặc định của trang web các bạn vừa xem là gì?

Trả lời:

Đường dẫn: /var/www/html/index.html



Thg 9 26 16:45

10.0.2.15

Khóa: Toán Rờ... Tạo Logo CTDL Kiến trúc máy... CNPM MMT WEB eclipse Hướng dẫn lậ..

Apache2 Ubuntu Default Page

ubuntu

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in** `/usr/share/doc/apache2/README.Debian.gz`. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|   |-- mods-enabled
|   |-- *.load
|   |-- *.conf
|-- conf-enabled
|-- *.conf
|-- sites-enabled
|-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.

Câu hỏi 2: Cổng mặc định của dịch vụ www là gì?

Trả lời:

Cổng 80

```
huymacquang@huymacquang-VirtualBox:~$ sudo netstat -ntlp | grep LISTEN
tcp        0      0 127.0.0.53:53         0.0.0.0:*           LISTEN      536/systemd-r
tcp        0      0 127.0.0.1:631        0.0.0.0:*           LISTEN      547/cupsd
tcp6       0      0 :::80               :::*                LISTEN      3768/apache2
tcp6       0      0 :::1:631            :::*                LISTEN      547/cupsd
```

Câu hỏi 3: Hãy giải thích quyền mang số 755 là gì?

Trả lời:

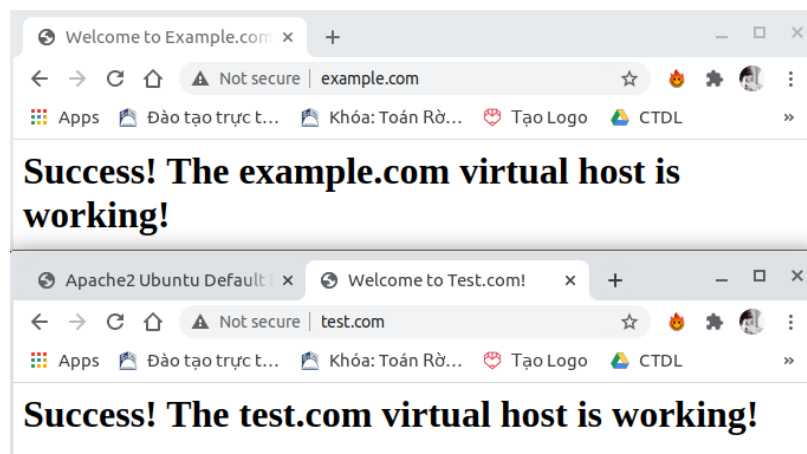
755

- Quyền của owner: $7 = 4(\text{read}) + 2(\text{write}) + 1(\text{excute}) = \text{rwn}$
- Quyền của group: $5 = 4(\text{read}) + 1(\text{excute}) = \text{re}$
- Quyền của other: $5 = 4(\text{read}) + 1(\text{excute}) = \text{re}$

Câu hỏi 4: Bạn quan sát thấy nội dung gì sau khi gõ 2 địa chỉ trên? Giải thích.

Trả lời:

- Hai địa chỉ test.com và example.com đều gọi đến 2 trang web tương ứng



- Giải thích: Hai địa chỉ test.com và example.com chính là địa chỉ chỉ tới 2 máy ảo tương ứng trên apache2. Khi gõ địa chỉ, hiểu là ip của máy hiện tại, dựa vào các file config, tìm đến file .html tương ứng

<https://viblo.asia/p/serie-he-phan-tan-1-tong-quan-ve-he-phan-tan-distributed-systems-924IJxAXKPM>

Câu hỏi 5: Thử truy cập từ các máy tính khác trong cùng mạng LAN vào 2 trang web đó

Trả lời:

Kết quả thực hiện tương tự ở câu 4 khi cổng mặc định của http và webserver giống nhau

Câu hỏi 6: Hãy tự viết một đoạn code để thực hiện 1 vòng lặp while sao cho nó sẽ nhận các số mà người dùng gõ và gửi về server, cho đến khi nào người dùng gõ ký tự rỗng rồi ấn enter.

Trả lời:

```
Scanner scanner = new Scanner(System.in);
while(1){
    String mess = scanner.nextLine();
    System.out.println(mess);
    if(mess.equals("")){
        socket.close();
        scanner.close();
        break;
    }
    String input = in.readLine();
    System.out.println(input)
}
```

Câu hỏi 7: Vai trò của phương thức run là gì? Khi nào nó được gọi?

Trả lời:

Method run: Chạy khi một client kết nối thành công tới server
Vai trò: Nhận data từ client gửi đến, xử lý và gửi trả lại client

Triển khai các giải thuật sắp xếp khác:

- Bubble sort

```
public class BubbleSort
{
    public void bubbleSort(int arr[])
    {
        int n = arr.length;
        for (int i = 0; i < n-1; i++)
            for (int j = 0; j < n-i-1; j++)
                if (arr[j] > arr[j+1])
                {
                    // swap arr[j+1] and arr[i]
                    int temp = arr[j];
                    arr[j] = arr[j+1];
                    arr[j+1] = temp;
                }
    }
}
```

- Insertion sort

```
public class InsertionSort
{
    public void insertionSort(int arr[])
    {
        int n = arr.length;
        int i, key, j;
        for (i = 1; i < n; i++)
        {
            key = arr[i];
            j = i - 1;

            /* Move elements of arr[0..i-1], that are
            greater than key, to one position ahead
            of their current position */
            while (j >= 0 && arr[j] > key)
            {
                arr[j + 1] = arr[j];
                j = j - 1;
            }
            arr[j + 1] = key;
        }
    }
}
```

- Shell sort

```
public class ShellSort
{
    int sort(int arr[])
    {
        int n = arr.length;

        // Start with a big gap, then reduce the gap
        for (int gap = n/2; gap > 0; gap /= 2)
        {
            // Do a gapped insertion sort for this gap size.
            // The first gap elements a[0..gap-1] are already
            // in gapped order keep adding one more element
            // until the entire array is gap sorted
            for (int i = gap; i < n; i += 1)
            {
                // add a[i] to the elements that have been gap
                // sorted save a[i] in temp and make a hole at
                // position i
                int temp = arr[i];

                // shift earlier gap-sorted elements up until
                // the correct location for a[i] is found
                int j;
                for (j = i; j >= gap && arr[j - gap] > temp; j -= gap)
                    arr[j] = arr[j - gap];

                // put temp (the original a[i]) in its correct
            }
        }
    }
}
```

```
        // location
        arr[j] = temp;
    }
    }
    return 0;
}
```