

Giới thiệu về Phát triển phía Máy chủ (Session 1)

Tác giả: Đặng Kim Thi

Buổi học: 1

1. Mục tiêu học tập

Sau buổi học này, sinh viên sẽ có thể:

- Hiểu kiến trúc client-server và các thành phần của nó.
 - Giải thích vai trò của máy chủ DNS và máy chủ Web.
 - Mô tả lập trình phía máy chủ và các ưu điểm của nó.
 - Xác định các ngôn ngữ lập trình phía máy chủ phổ biến.
 - Hiểu những khái niệm cơ bản về Node.js và môi trường thực thi của nó.
 - Giải thích lý do tại sao Node.js được ưa chuộng cho các ứng dụng Web.
 - Sử dụng Visual Studio Code như một IDE cho phát triển Node.js.
-

2. Kiến trúc Client-Server

2.1 Định nghĩa

Kiến trúc client-server là mô hình tính toán trong đó nhiều máy khách gửi yêu cầu đến máy chủ, máy chủ xử lý và trả về phản hồi.

2.2 Các thành phần

- **Máy khách (Client):** Thiết bị hoặc ứng dụng gửi yêu cầu (ví dụ: trình duyệt web, ứng dụng di động).
- **Máy chủ (Server):** Hệ thống xử lý yêu cầu và gửi phản hồi.
- **Mạng (Network):** Phương tiện giúp máy khách và máy chủ giao tiếp với nhau.

2.3 Ví dụ

- Khi một người dùng truy cập <https://example.com>, trình duyệt (máy khách) gửi yêu cầu đến máy chủ lưu trữ trang web và máy chủ phản hồi với nội dung trang.

Câu hỏi kiến tạo

1. Theo bạn, kiến trúc client-server có thể áp dụng vào lĩnh vực nào ngoài ứng dụng web? Tại sao?
2. Nếu không có mô hình client-server, việc truy cập internet sẽ bị ảnh hưởng như thế nào?
3. Làm thế nào để cải thiện hiệu suất của hệ thống client-server?

Bài tập 1

👉 Xác định ba ứng dụng thực tế sử dụng mô hình client-server.

3. Máy chủ DNS và Máy chủ Web

3.1 Máy chủ DNS

Máy chủ DNS (Domain Name System) chuyển đổi tên miền (ví dụ: `google.com`) thành địa chỉ IP mà máy tính có thể hiểu.

3.2 Máy chủ Web

Máy chủ Web lưu trữ và cung cấp nội dung web. Một số máy chủ phổ biến gồm:

- Apache
- Nginx
- Microsoft IIS

3.3 Ví dụ

Khi bạn nhập `www.google.com`, máy chủ DNS chuyển đổi thành địa chỉ IP và hướng trình duyệt đến máy chủ web phù hợp.

Bài tập 2

👉 Sử dụng lệnh `nslookup google.com` (Windows) hoặc `dig google.com` (Linux/macOS) để tìm địa chỉ IP của máy chủ Google.

4. Lập trình phía Máy chủ

4.1 Định nghĩa

Lập trình phía máy chủ là mã được thực thi trên máy chủ thay vì trên máy khách. Nó xử lý dữ liệu, tương tác với cơ sở dữ liệu và tạo nội dung động.

4.2 Ưu điểm

- Giữ logic xử lý ẩn khỏi máy khách.
- Bảo mật và kiểm soát tốt hơn.
- Giảm tải xử lý trên phía máy khách.

4.3 Các ngôn ngữ lập trình phổ biến

- **Java**
- **PHP**
- **Python**
- **Node.js**
- **Ruby**
- **C#**

4.4 Ví dụ

Một đoạn mã PHP xử lý biểu mẫu đăng nhập:

```
<?php
$username = $_POST['username'];
$password = $_POST['password'];
if($username == 'admin' && $password == '1234') {
    echo "Đăng nhập thành công";
} else {
    echo "Thông tin đăng nhập không chính xác";
}
?>
```

Bài tập 3

👉 Xác định các tình huống mà lập trình phía máy chủ phù hợp hơn lập trình phía máy khách.

5. Giới thiệu về Node.js

5.1 Node.js là gì?

Node.js là **môi trường thực thi JavaScript đơn luồng** ngoài trình duyệt.

5.2 Tại sao sử dụng Node.js?

- **Sử dụng JavaScript**, một ngôn ngữ phổ biến.
- **Mở rộng tốt**, cho phép phát triển các ứng dụng hiệu năng cao.
- **Hiệu quả**, sử dụng mô hình bất đồng bộ, không chặn.

5.3 Ví dụ

Một máy chủ HTTP đơn giản trong Node.js:

```
const http = require('http');
http.createServer((req, res) => {
    res.writeHead(200, {'Content-Type': 'text/plain'});
    res.end('Xin chào, thế giới!\n');
}).listen(3000);
console.log('Máy chủ chạy tại http://localhost:3000/');
```

Chạy mã này và truy cập <http://localhost:3000/> trên trình duyệt.

Bài tập 4

👉 Chỉnh sửa mã để trả về một trang HTML thay vì văn bản thuần túy.

6. Sử dụng Visual Studio Code với Node.js

6.1 Tại sao chọn VS Code?

- Nhẹ và mạnh mẽ.
- Hỗ trợ tích hợp JavaScript và Node.js.
- Có hệ sinh thái mở rộng phong phú.

6.2 Các bước cài đặt

1. Cài đặt **Node.js** từ nodejs.org.
2. Cài đặt **Visual Studio Code** từ code.visualstudio.com.
3. Mở VS Code và cài đặt **Node.js Extension Pack**.
4. Tạo một tập tin JavaScript và chạy với terminal bằng lệnh `node filename.js`.

Bài tập 5

👉 Viết một chương trình Node.js đọc và hiển thị nội dung của một tệp tin.

7. Tổng kết

- **Kiến trúc client-server** giúp nhiều máy khách giao tiếp với một máy chủ.
 - **Máy chủ DNS và web** đóng vai trò quan trọng trong duyệt web.
 - **Lập trình phía máy chủ** cho phép xử lý dữ liệu an toàn và động.
 - **Node.js** là một môi trường JavaScript mạnh mẽ cho phát triển web.
 - **VS Code** là một IDE phổ biến cho lập trình Node.js.
-

8. Bài tập về nhà

Hoàn thành các bài tập trong từng phần. Buổi học tiếp theo, chúng ta sẽ tìm hiểu **các module trong Node.js và quản lý gói với NPM**.
