

Ôn thi giữa kỳ môn Phát triển phần mềm mã nguồn mở (Buổi 3 và Buổi 4-5)

Phần 1: Lập trình Shell Script (Buổi 3)

Kiến thức lý thuyết cần nắm về Shell Script

Shell Script là gì và vai trò của nó: Shell script là một tệp văn bản chứa một chuỗi các lệnh mà hệ điều hành Unix/Linux có thể thực thi tuần tự. Thay vì nhập từng lệnh thủ công, ta có thể viết các lệnh này vào một tệp (script) và chỉ cần chạy tệp đó để tự động hóa các tác vụ lặp lại ¹. Shell script giúp tiết kiệm thời gian và công sức khi quản trị hệ thống, hỗ trợ tự động hóa nhiều việc như quản lý tệp tin, sao lưu dữ liệu, kiểm tra hệ thống, v.v. Ngoài ra, shell script có thể kết hợp với các công cụ dòng lệnh mạnh mẽ như `grep`, `awk`, `sed` để xử lý văn bản và dữ liệu hiệu quả ².

Cấu trúc một chương trình shell script cơ bản: Một file shell script thông thường có cấu trúc như sau:

- Dòng đầu tiên là **shebang** để chỉ định trình thông dịch sẽ thực thi script. Ví dụ: `#!/bin/bash` (khi script chạy với Bash shell). Đây là bắt buộc để hệ thống biết script sẽ chạy bằng shell nào ³.
- Phần thân script gồm các lệnh shell được viết tuần tự. Mỗi lệnh như `gõ` trực tiếp trong terminal, nhưng khi đặt trong script sẽ thực hiện lần lượt theo kịch bản.
- Có thể bao gồm **comment** (chú thích) bắt đầu bằng `#` để giải thích code (trừ dòng shebang đặc biệt).
- Cuối cùng, cần cấp quyền thực thi cho file script bằng lệnh `chmod +x tên_file.sh` để có thể chạy nó như một chương trình ⁴. Sau đó có thể thực thi script bằng cách gọi đường dẫn tới file (ví dụ: `./ten_file.sh` nếu đang ở cùng thư mục).

Biến và cách sử dụng biến: Shell script cho phép định nghĩa biến để lưu trữ dữ liệu và tái sử dụng trong các lệnh sau đó. Khi khai báo biến trong shell script, ta gán giá trị mà **không có khoảng trắng** quanh dấu `=`. Ví dụ: `name="Alice"`. Khi sử dụng biến, cần đặt ký hiệu `$` trước tên biến, ví dụ: `echo "Hello, $name"` ⁵ ⁶. Biến có thể lưu trữ nhiều loại dữ liệu (chuỗi, số, kết quả lệnh khác...).

Trong Linux có hai loại biến chính:

- **Biến hệ thống:** Những biến do hệ thống hoặc shell định nghĩa sẵn, thường viết hoa (ví dụ: `PATH`, `HOME`, `USER`). Chúng chứa thông tin về hệ thống như đường dẫn thực thi, thư mục home, tên user hiện tại... ⁷ ⁸.
- **Biến do người dùng định nghĩa:** Là biến do người viết script tạo ra để phục vụ mục đích của script. Biến này thường viết thường hoặc viết hoa tùy ý (tránh trùng tên biến hệ thống). Ví dụ: `who="World"` rồi sử dụng `echo "Hello, $who!"` ⁵ ⁶ sẽ in ra `Hello, World!`.

Các câu lệnh thường dùng trong shell script:

- Lệnh `echo`: In thông tin ra màn hình. Ví dụ: `echo "Welcome"` sẽ xuất chuỗi **Welcome** ra `stdout`. Lệnh này thường dùng để hiển thị thông báo, kết quả tính toán, v.v.
- Lệnh `read`: Đọc input từ người dùng và gán vào biến. Ví dụ:

```

echo "Nhập tên của bạn:"
read name
echo "Chào bạn, $name"

```

Lệnh trên sẽ nhắc người dùng nhập tên, lưu vào biến `name` và sau đó chào tên đó ⁹.

- Các lệnh thao tác với **biến số học**: Shell cho phép tính toán số đơn giản, ví dụ: `x=$((a + b))` để tính tổng, hoặc dùng lệnh `expr`. Tuy nhiên, cần chú ý mọi thứ ban đầu đều là chuỗi nên khi so sánh số nên dùng toán tử số học (như `-eq`, `-gt` thay vì `=`).
- **Lệnh test** (hoặc cú pháp `[...]`): Được dùng để kiểm tra điều kiện trong câu lệnh if. Ví dụ: `["$age" -gt 20]` để kiểm tra biến `age` lớn hơn 20 hay không ¹⁰.
- Các lệnh quản lý user trên Linux: `useradd` (thêm user mới), `userdel` (xóa user), `passwd` (đặt mật khẩu cho user), `id` (kiểm tra thông tin user). Các lệnh này hữu ích trong kịch bản quản lý tài khoản người dùng.
- **Lệnh apt** (Advanced Package Tool): dùng trên Ubuntu/Debian để quản lý gói cài đặt. Cụ thể thường dùng `sudo apt update` để cập nhật danh sách gói, `sudo apt install <tên_gói>` để cài đặt gói phần mềm, `sudo apt remove <tên_gói>` để gỡ bỏ gói, v.v. Ví dụ: `sudo apt install -y apache2 php mysql-server` cài đặt Apache2, PHP và MySQL không cần xác nhận lại ¹¹. (Giải thích thêm: tùy chọn `-y` để tự động trả lời "yes" khi cài đặt). Lệnh apt sẽ tự động xử lý phụ thuộc và cài đặt các gói liên quan ¹².

Cấu trúc điều kiện (if...then...else): Cho phép thực thi lệnh có điều kiện. Cú pháp chung:

```

if [ <điều kiện> ]; then
    <lệnh khi điều kiện true>
else
    <lệnh khi điều kiện false>
fi

```

Trong đó `fi` kết thúc cấu trúc if. Điều kiện thường đặt trong dấu `[]` (thực chất là lệnh `test`). Ví dụ:

```

if [ "$age" -ge 18 ]; then
    echo "You are an adult."
else
    echo "You are a minor."
fi

```

Script trên kiểm tra giá trị của biến `age`. Nếu `age >= 18` thì in ra "You are an adult", ngược lại in "You are a minor" ¹³. Lưu ý khi so sánh chuỗi dùng `=` hoặc `!=`, khi so sánh số dùng các toán tử dạng `-eq` (bằng), `-ne` (không bằng), `-lt` (nhỏ hơn), `-gt` (lớn hơn), `-ge` (\geq), `-le` (\leq).

Vòng lặp (loops): Cho phép thực thi lặp lại một nhóm lệnh nhiều lần. Có các loại vòng lặp như `for`, `while`, `until`. Ví dụ vòng lặp `for`:

```
for i in 1 2 3 4 5; do
    echo "Number $i"
done
```

Script trên sẽ in ra các số từ 1 đến 5; mỗi lần lặp, giá trị của `i` thay đổi và lệnh `echo` sẽ in ra "Number 1", "Number 2", ... ¹⁴. Shell script cũng hỗ trợ vòng lặp `while` (lặp khi điều kiện đúng) và `until` (lặp đến khi điều kiện đúng) ¹⁵ ¹⁶. Vòng lặp thường được dùng kết hợp với đọc file hoặc danh sách để tự động xử lý nhiều mục.

Chạy lệnh trong script và kết hợp lệnh: Ta có thể sử dụng phép toán logic `&&` (AND), `||` (OR) để kết hợp lệnh, hoặc dùng dấu `|` (pipe) để chuyển kết quả lệnh bên trái thành đầu vào cho lệnh bên phải. Ngoài ra có thể dùng cú pháp ``lệnh`` hoặc `$(lệnh)` để chèn kết quả của một lệnh này vào lệnh khác. Ví dụ: `today=$(date)` gán biến `today` bằng chuỗi ngày giờ hiện tại trả về từ lệnh `date`.

Hướng dẫn tạo và chạy shell script trên máy cá nhân

Để thực hành shell script trên máy cá nhân (giả sử dùng Ubuntu hoặc Linux tương tự), sinh viên cần làm các bước sau:

- Tạo file script:** Mở trình soạn thảo (có thể dùng nano, vi, gedit, VS Code...) và tạo file với phần mở rộng `.sh` (không bắt buộc nhưng nên dùng để dễ nhận biết). Ví dụ tạo file `welcome.sh`.
- Viết nội dung script:** Bắt đầu file bằng dòng shebang `#!/bin/bash`. Sau đó viết các lệnh shell tuân tự như mong muốn. Ví dụ đơn giản:

```
#!/bin/bash
echo "Welcome"
```

(Script này chỉ in ra dòng "Welcome"). Ngoài ra có thể viết sẵn một số kịch bản như yêu cầu của bài tập.

- Lưu file và cấp quyền thực thi:** Lưu file lại. Mở Terminal, điều hướng đến thư mục chứa file và chạy lệnh `chmod +x welcome.sh` để cấp quyền chạy.
- Chạy thử script:** Có thể thực thi script bằng nhiều cách:
 - Cách 1: Gõ `./welcome.sh` (nếu thư mục hiện tại đã nằm trong PATH hoặc ta dùng `./` chỉ rõ đường dẫn hiện tại).
 - Cách 2: Gọi shell và truyền file vào: `bash welcome.sh` hoặc `sh welcome.sh`.
 - Cách 3: Di chuyển file vào thư mục trong PATH (vd: `/usr/local/bin`) để có thể gọi tên trực tiếp ở mọi nơi.

Khi chạy, nếu script chưa có quyền thực thi hoặc không có shebang, có thể gặp lỗi "command not found" hoặc "Permission denied". Hãy đảm bảo đã thêm shebang và cấp quyền đúng như bước trên ³ ¹⁷.

Môi trường thử nghiệm: Để thuận tiện, sinh viên có thể sử dụng ngay môi trường Ubuntu đã cài (buổi 1 đã làm quen Ubuntu). Nếu dùng Windows, có thể cài **WSL2** (Windows Subsystem for Linux) hoặc dùng máy ảo Ubuntu để chạy các shell script.

Các ví dụ thực hành trọng tâm và câu lệnh quan trọng

Dưới đây là một số bài tập và ví dụ **trọng tâm** liên quan đến shell script mà sinh viên cần nắm, vì rất có thể sẽ xuất hiện trong bài kiểm tra:

- **In ra câu chào "Welcome":** Đây là bài cơ bản nhất. Chỉ cần dùng lệnh `echo "Welcome"`. Mục tiêu: kiểm tra sinh viên biết cách tạo script và xuất chuỗi ra màn hình.
- **Đọc tên người dùng và xuất ra màn hình:** Sử dụng lệnh `read` để nhập họ tên, sau đó dùng `echo` để in ra. Ví dụ:

```
#!/bin/bash
echo "Nhập họ tên của bạn:"
read fullname
echo "Họ tên vừa nhập: $fullname"
```

Khi chạy, script sẽ dừng để nhận input, nhập xong ấn Enter thì sẽ hiển thị lại tên. Bài này kiểm tra khả năng tương tác nhập/xuất của script.

- **Giải phương trình bậc nhất $ax + b = 0$:** Sinh viên cần viết script yêu cầu nhập vào hệ số `a` và `b`, sau đó tính nghiệm `x`. Lưu ý trường hợp đặc biệt:
 - Nếu `a = 0` và `b = 0`: vô số nghiệm.
 - Nếu `a = 0` và `b ≠ 0`: vô nghiệm.
 - Nếu `a ≠ 0`: nghiệm duy nhất `x = -b/a`.

Gợi ý: sử dụng toán tử so sánh `==` trong câu lệnh `if` để xử lý các trường hợp trên. Phép toán chia có thể thực hiện bằng cách dùng lệnh `bc` (Basic Calculator) cho kết quả số thực, hoặc trong bash có thể thực hiện số học dấu phẩy động bằng cách sử dụng `awk` hoặc `python` nếu muốn. Tuy nhiên, trong phạm vi bài cơ bản, có thể chấp nhận in dưới dạng phân số nếu cần đơn giản (ví dụ: `echo "Nghiệm x = -$b/$a"`).

- **Đọc file chứa danh sách user/password và tự động tạo user:** Bài này quan trọng vì tích hợp kiến thức về đọc file, xử lý chuỗi và lệnh quản lý user. Đề bài: Cho một file văn bản (ví dụ `users.txt`) chứa nhiều dòng, mỗi dòng có username và password tương ứng, cách nhau bởi khoảng trắng, ví dụ:

```
user1 pass1
user2 pass2
user3 pass3
```

Viết script đọc lần lượt từng dòng, kiểm tra xem user đó đã tồn tại trên hệ thống chưa, nếu chưa thì tạo user với password tương ứng.

Hướng dẫn: Có thể dùng vòng lặp `while` để đọc file:

```
while read user pass; do
    # Kiểm tra sự tồn tại của user
```

```

if id "$user" &>/dev/null; then
    echo "User $user đã tồn tại, bỏ qua."
else
    echo "Tạo user $user"
    sudo useradd "$user"
    echo "$user:$pass" | sudo chpasswd
fi
done < users.txt

```

Giải thích: lệnh `id "$user"` kiểm tra user có trên hệ thống (exit code 0 nếu có), `chpasswd` được dùng để đặt password cho user một cách non-interactive (ở đây echo đầu vào theo format `user :password`)¹⁸. Cần chạy script này với quyền root (hoặc dùng `sudo` trước các lệnh thêm user, đặt mật khẩu) thì mới thực hiện được trên hệ thống thật. Bài này kiểm tra khả năng dùng vòng lặp đọc file và gọi lệnh hệ thống (`useradd`, `chpasswd`) trong script. Lưu ý tránh lạm dụng vòng lặp `for` đọc file vì sẽ gặp vấn đề với khoảng trắng; dùng `while read` như trên sẽ an toàn¹⁹²⁰.

- **Cài đặt các phần mềm văn bản bằng shell script:** "Phần mềm văn bản" có thể hiểu là các ứng dụng soạn thảo tài liệu như LibreOffice, hoặc các trình soạn thảo văn bản đơn giản. Bài tập yêu cầu viết các lệnh (script) để cài những phần mềm này. Thực chất chỉ cần vận dụng lệnh `apt`. Ví dụ: cài LibreOffice và Gedit:

```

sudo apt update
sudo apt install -y libreoffice gedit

```

Hoặc cài các công cụ soạn thảo khác như Vim, Nano: `sudo apt install -y vim nano`. Bài này nhằm kiểm tra kỹ năng sử dụng lệnh `apt` để cài đặt phần mềm và có thể lồng ghép trong script.

- **Cài đặt Apache, PHP, MySQL bằng shell script:** Tương tự, yêu cầu liệt kê các lệnh cài đặt môi trường LAMP (web server + ngôn ngữ server + cơ sở dữ liệu) trên Ubuntu. Các lệnh chính:

```

sudo apt update
sudo apt install -y apache2 php mysql-server

```

Thực tế, để chạy PHP mượt mà còn cần một số module PHP và thư viện bổ trợ (sẽ trình bày chi tiết ở phần sau), nhưng về căn bản câu lệnh `apt` cài 3 thành phần chính như trên là đúng mục tiêu²¹. Sau khi cài, có thể cần khởi động dịch vụ: `sudo systemctl start apache2` (mặc dù thường dịch vụ tự khởi chạy) và thiết lập MySQL (ví dụ chạy `sudo mysql_secure_installation` để đặt mật khẩu cho tài khoản root của MySQL, v.v.). Bài tập này kiểm tra khả năng nhớ và viết đúng lệnh cài đặt các thành phần LAMP.

Các tình huống có thể gặp trong bài kiểm tra (Shell Script)

Trong bài thi giữa kỳ, phần shell script (3 điểm) có thể được kiểm tra dưới nhiều hình thức. Sinh viên nên chuẩn bị cho các tình huống sau:

- **Chỉnh sửa một shell script có sẵn:** Giảng viên có thể cho sẵn một đoạn shell script có lỗi hoặc thiếu phần hoàn chỉnh, yêu cầu sinh viên tìm và sửa lỗi hoặc bổ sung cho đúng yêu cầu. Ví dụ:

một script thiếu dòng shebang gây lỗi chạy, hoặc sai cú pháp if/for, hoặc chưa xử lý hết trường hợp đặc biệt. Sinh viên cần nắm vững cú pháp để nhận ra lỗi (như thiếu `then/fi`, thiếu dấu `;` trước `do` trong for, dùng `=` thay vì `-eq` cho số, v.v.) và nhanh chóng sửa đúng.

- **Viết mới hoặc hoàn thiện một shell script nhỏ:** Có thể yêu cầu tương tự các bài tập đã nêu: ví dụ viết script tính toán đơn giản, script tạo user từ file, v.v. Do thời gian có hạn, đề thi có thể tập trung vào một chức năng chính (như đọc input và quyết định kết quả). Hãy đảm bảo thực hành trước các bài mẫu để quen tay.
- **Hiểu và giải thích output của script:** Một dạng câu hỏi lý thuyết: cho một đoạn mã shell và hỏi kết quả in ra là gì, hoặc giải thích vai trò của một đoạn lệnh (ví dụ `2>>` là gì, `&>/dev/null` nghĩa là gì,...). Sinh viên cần nắm các ký hiệu chuyển hướng (`>`, `>>`, `2>`, `&>`, `|`) và ý nghĩa của chúng.
- **Kết hợp lệnh Linux trong script:** Có thể có tình huống script kết hợp nhiều lệnh Unix cơ bản (tạo thư mục, sao chép file, tìm kiếm chuỗi trong file, v.v.). Hãy sẵn sàng ôn lại các lệnh cơ bản (cp, mv, grep, chmod, etc) và cách dùng chúng trong script.

Tóm lại, phần shell script đòi hỏi sinh viên hiểu rõ cú pháp và ý nghĩa các thành phần (biến, vòng lặp, điều kiện) cũng như biết áp dụng lệnh Linux để giải quyết bài toán cụ thể. Cách ôn tập hiệu quả là tự tay viết lại các ví dụ đã học, chạy thử trên máy và sửa lỗi nếu có. Shell scripting là công cụ mạnh mẽ, nắm vững phần này không chỉ giúp qua bài thi mà còn hữu ích cho việc quản trị hệ thống sau này ²².

Phần 2: LAMP và Cài đặt CMS (Buổi 4-5)

Tổng quan lý thuyết về LAMP và các thành phần ứng dụng web

Kiến trúc LAMP: "LAMP" là viết tắt của **Linux**, **Apache**, **MySQL**, **PHP** – một tập hợp công nghệ mã nguồn mở thường được sử dụng cùng nhau để triển khai máy chủ web cho ứng dụng web động ²³. Cụ thể: - **Linux**: Hệ điều hành máy chủ (ở đây là nhân Linux, ví dụ Ubuntu). Linux đóng vai trò nền tảng cho các thành phần khác chạy trên nó ²⁴. - **Apache**: Phần mềm **web server** (máy chủ HTTP) phổ biến nhất. Apache chịu trách nhiệm lắng nghe các yêu cầu HTTP từ client (trình duyệt) và phản hồi. Nó phục vụ nội dung tĩnh (HTML, hình ảnh, file) và chuyển tiếp các yêu cầu động tới PHP để xử lý ²⁴. Apache rất linh hoạt, hỗ trợ nhiều module mở rộng (PHP, Perl, Python, SSL, etc.) ²⁵. - **MySQL** (hoặc MariaDB): Hệ quản trị **cơ sở dữ liệu (database)** quan hệ, dùng để lưu trữ dữ liệu cho các ứng dụng web. Mọi thông tin như tài khoản người dùng, nội dung bài viết, sản phẩm, v.v. đều được lưu trong database và truy xuất thông qua truy vấn SQL ²⁶. (MariaDB là nhánh fork của MySQL, thường thay thế tương đương trong hệ thống LAMP hiện đại). - **PHP** (hoặc ngôn ngữ kịch bản máy chủ tương đương như Python, Perl): Ngôn ngữ lập trình phía **server-side**. Mã PHP được nhúng trong trang web để tạo nội dung động. Apache khi nhận yêu cầu tới trang PHP sẽ gọi trình thông dịch PHP để xử lý logic (kết nối database, truy vấn dữ liệu, xử lý form,...) rồi sinh ra HTML kết quả. Apache sau đó trả HTML này về cho client ²⁶. PHP dễ học, phổ biến trong phát triển web mã nguồn mở (WordPress, Joomla, Moodle đều viết bằng PHP).

Tóm lại, mô hình hoạt động: **Client (browser)** gửi request -> **Apache web server** nhận request: - Nếu là file tĩnh, Apache trả về trực tiếp. - Nếu là trang động (PHP), Apache chuyển cho **PHP** xử lý. PHP có thể tương tác với **MySQL database** để lấy dữ liệu, sau đó trả kết quả HTML cho Apache. -> **Apache** gửi phản hồi HTML về cho client.

Chu trình này tạo nên một hệ thống web hoàn chỉnh trên nền LAMP. Ưu điểm của LAMP: hoàn toàn

miễn phí, mã nguồn mở, cộng đồng hỗ trợ rộng, tính tương thích cao và đã trở thành giải pháp máy chủ web mặc định trong nhiều năm ²⁷ ²³.

Các thành phần khác và công cụ hỗ trợ trong buổi 4-5: - **phpMyAdmin:** Ứng dụng web (PHP) cung cấp giao diện quản trị MySQL qua trình duyệt. Cài phpMyAdmin giúp sinh viên dễ dàng tạo cơ sở dữ liệu, bảng, xem/sửa dữ liệu mà không cần gõ lệnh SQL thủ công. Đây không bắt buộc cho ứng dụng chạy, nhưng hữu ích cho việc quản lý database trực quan. - **Visual Studio Code, Sublime Text:** Các trình soạn thảo mã nguồn/IDE hỗ trợ viết code PHP, HTML thuận tiện. Cài đặt chúng giúp sinh viên có môi trường phát triển tốt (nhưng trong bối cảnh thi, quan trọng là đã có các file cấu hình và mã nguồn đúng, việc code PHP từ đầu có thể không nhiều). - **Khái niệm CMS (Content Management System) và các phần mềm web mã nguồn mở:** CMS là hệ quản trị nội dung, cho phép tạo website mà không cần lập trình nhiều, thông qua giao diện quản lý và các mô-đun có sẵn. Trong buổi 4-5, sinh viên được yêu cầu tìm hiểu, cài đặt và sử dụng cơ bản 2 ứng dụng web mã nguồn mở. Danh sách gợi ý gồm: Joomla, osCommerce, OpenCart, Tomato (có thể là TomatoCart), Moodle, Django-Oscar,... Trong đó: - **Joomla:** một CMS (hệ thống quản trị nội dung) mã nguồn mở rất phổ biến, dùng để xây dựng website, cổng thông tin, blog... Joomla viết bằng PHP và sử dụng MySQL/MariaDB làm database ²⁸. Joomla dễ sử dụng, thiết kế module hóa, hỗ trợ MVC, nhiều tính năng và giao diện đẹp, phù hợp để xây dựng các trang web chuyên nghiệp ²⁹. - **Moodle:** một LMS (Learning Management System) mã nguồn mở dành cho giáo dục, cho phép tạo các khóa học trực tuyến, quản lý học viên, bài giảng. Moodle hoàn toàn miễn phí, viết bằng PHP, nhằm giúp giáo viên và trường học tạo cộng đồng học tập trực tuyến hiệu quả ³⁰. Moodle có đầy đủ tính năng quản lý khóa học, bài tập, diễn đàn thảo luận, v.v., thường được dùng trong các trường đại học và tổ chức đào tạo. - (Các phần mềm khác: osCommerce, OpenCart là về thương mại điện tử; Django-Oscar là framework Python cho e-commerce; TomatoCart là giỏ hàng online, v.v. Tùy sinh viên lựa chọn, nhưng trong phạm vi thi, Joomla và Moodle là đại diện tiêu biểu cho CMS và LMS.)

Tại sao cần hiểu thành phần web: Trong bài thi, giảng viên có thể kiểm tra khái niệm như *web server là gì?, PHP chạy ở đâu?, database dùng để làm gì?*. Sinh viên cần nắm: web server (Apache) phục vụ nội dung web; PHP xử lý logic động phía server; database lưu trữ dữ liệu nền cho trang web. Hiểu mối quan hệ này giúp thực hiện đúng các bước cài đặt và khắc phục sự cố (ví dụ: trang trắng do PHP lỗi, không kết nối DB do sai cấu hình, v.v.).

Hướng dẫn cài đặt và cấu hình LAMP trên máy cá nhân (Linux)

Dưới đây là hướng dẫn từng bước để cài đặt **LAMP stack** (Apache, MySQL/MariaDB, PHP) trên máy tính cá nhân chạy Ubuntu (hoặc tương tự). Các bước này giúp chuẩn bị môi trường để cài các ứng dụng web mã nguồn mở như Joomla, Moodle.

1. Cập nhật hệ thống: Mở Terminal và chạy:

```
sudo apt update && sudo apt upgrade -y
```

Lệnh này cập nhật danh sách gói và nâng cấp các phần mềm đã cài lên phiên bản mới nhất, đảm bảo hệ thống sẵn sàng.

2. Cài đặt Apache: Chạy lệnh:

```
sudo apt install -y apache2
```

Hệ thống sẽ cài Apache2 (web server). Sau khi cài, Apache thường tự khởi động. Kiểm tra bằng cách mở trình duyệt truy cập <http://localhost>. Nếu thấy trang "Apache2 Ubuntu Default Page" nghĩa là Apache đã chạy thành công.

3. Cài đặt MySQL/MariaDB: Chạy:

```
sudo apt install -y mariadb-server
```

(Ubuntu 20.04+ dùng MariaDB thay cho MySQL nhưng tương đương chức năng). Sau khi cài, nên chạy lệnh bảo mật:

```
sudo mysql_secure_installation
```

Lệnh này sẽ hướng dẫn đặt mật khẩu cho user `root` của MySQL, xóa bớt user ẩn danh, cấm remote root login, và xóa database test. Hãy đặt mật khẩu root (nhớ mật khẩu này để dùng quản trị DB hoặc cho ứng dụng web kết nối) và chấp nhận các tùy chọn an toàn (gõ `Y`) ³¹.

4. Cài đặt PHP và các module cần thiết: Chạy lệnh (ví dụ trên Ubuntu 20.04 cài PHP 7.4):

```
sudo apt install -y php libapache2-mod-php php-mysql
```

Lệnh trên cài PHP và module kết nối Apache-PHP, module PHP cho MySQL. Ngoài ra, tùy ứng dụng mà cần thêm các extension khác. Ví dụ Joomla/Moodle thường cần các module: `php-xml`, `php-json`, `php-gd`, `php-curl`, `php-intl`, `php-mbstring`, `php-zip`, v.v. Ta có thể cài tất cả một lần:

```
sudo apt install -y php-xml php-json php-gd php-curl php-intl php-mbstring php-zip
```

(Nếu dùng PHP7.4 có thể cần ghi phiên bản, nhưng với lệnh trên apt tự chọn phiên bản phù hợp). Trong tài liệu cài Joomla trên Ubuntu, người ta ví dụ cài PHP7.4 cùng các module: `php7.4-cli`, `php7.4-mysql`, `php7.4-xml`, `php7.4-mbstring`, ... ²¹. Tương tự, Moodle yêu cầu một số extension như `php-gd`, `php-intl`, `php-soap`, `php-xmlrpc`,... ta cũng cài qua apt ³². Sau khi cài, cần **khởi động lại Apache** để nạp module PHP:

```
sudo systemctl restart apache2
```

Kiểm tra PHP bằng cách tạo file `/var/www/html/info.php` với nội dung `<?php
phpinfo(); ?>` và mở <http://localhost/info.php> xem trang thông tin PHP hiện ra là thành công.

5. Cài đặt phpMyAdmin (tùy chọn): Nếu muốn có công cụ quản lý MySQL qua giao diện web, cài:

```
sudo apt install -y phpmyadmin
```

Trong quá trình cài, trình cài đặt sẽ hỏi chọn web server cho phpMyAdmin (hãy chọn apache2, nhấn phím **Space** để đánh dấu rồi Enter). Sau đó hỏi cấu hình database cho phpMyAdmin, chọn "Yes" và nhập mật khẩu MySQL root khi được yêu cầu. Kết thúc cài đặt, phpMyAdmin sẽ được cấu hình sẵn với Apache. Truy cập <http://localhost/phpmyadmin>, đăng nhập bằng user "root" và mật khẩu MySQL đã đặt để sử dụng.

Tới đây, môi trường LAMP của bạn đã sẵn sàng. Tóm tắt: Apache chạy trên port 80, MySQL/MariaDB chạy background (port 3306), PHP chạy như module của Apache. Các file website đặt trong thư mục </var/www/html> (mặc định webroot của Apache).

Cài đặt Joomla trên LAMP (trên Ubuntu)

Sau khi có LAMP, chúng ta tiến hành cài đặt **Joomla** – một CMS mã nguồn mở phổ biến. Các bước chính: tạo cơ sở dữ liệu cho Joomla, tải mã nguồn Joomla, cấu hình Apache (nếu cần), chạy trình cài đặt Joomla qua trình duyệt.

Bước 1: Tạo cơ sở dữ liệu và user cho Joomla: Mở MySQL shell:

```
sudo mysql -u root -p
```

(nhập mật khẩu root MySQL đã đặt). Sau đó trong MySQL prompt, chạy các lệnh SQL:

```
CREATE DATABASE joomladb CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;
GRANT ALL ON joomladb.* TO 'usuariojoomla'@'localhost' IDENTIFIED BY
'123password';
FLUSH PRIVILEGES;
EXIT;
```

Lệnh trên tạo một database tên **joomladb** với mã hóa UTF8MB4 (để hỗ trợ cả emoji, tiếng Việt đầy đủ) ³³. Sau đó tạo user **usuariojoomla** (ví dụ đặt tên cho dễ nhớ) với mật khẩu **123password**, và cấp toàn quyền trên database joomladb cho user này ³⁴. (Ghi chú: nên dùng mật khẩu mạnh thực tế). Cuối cùng **FLUSH PRIVILEGES** để áp dụng quyền ngay, rồi **EXIT** thoát MySQL.

Mẹo: Tên database, tên user, mật khẩu này hãy ghi lại, lát nữa cấu hình Joomla sẽ cần.

Bước 2: Tải xuống và cài đặt mã nguồn Joomla: Truy cập trang chủ Joomla để lấy link tải phiên bản mới nhất (file nén .zip hoặc .tar.gz). Giả sử tải bản ổn định mới nhất (vd Joomla 3.9.19) bằng lệnh wget:

```
cd /tmp
wget https://downloads.joomla.org/cms/joomla3/3-9-19/Joomla_3.9.19-Stable-
Full_Package.zip
```

Sau khi tải xong, giải nén bộ cài Joomla vào thư mục web (ví dụ tạo thư mục </var/www/html/joomla> cho website Joomla):

```
sudo mkdir -p /var/www/html/joomla  
sudo unzip Joomla_3.9.19-Stable-Full_Package.zip -d /var/www/html/joomla
```

Lệnh trên sẽ giải nén toàn bộ mã nguồn Joomla vào `/var/www/html/joomla` ³⁵. Tiếp theo, **phân quyền** thư mục này cho webserver:

```
sudo chown -R www-data:www-data /var/www/html/joomla
```

(`www-data` là user chạy Apache trên Ubuntu) ³⁶. Bước phân quyền rất quan trọng, nếu không Joomla có thể không ghi được cấu hình hoặc tải lên tệp.

Bước 3: Cấu hình Apache cho Joomla (tạo Virtual Host): Nếu trên máy chỉ chạy một site Joomla thì có thể sử dụng luôn thư mục trên với URL dạng `http://localhost/joomla`. Tuy nhiên để chuyên nghiệp và để cài nhiều site song song (vd Joomla và Moodle cùng chạy), ta nên thiết lập VirtualHost riêng cho Joomla trên Apache:

- Tạo file cấu hình `/etc/apache2/sites-available/joomla.conf`:

```
sudo nano /etc/apache2/sites-available/joomla.conf
```

Điền nội dung VirtualHost, ví dụ:

```
<VirtualHost *:80>  
    ServerName joomla.local  
    DocumentRoot /var/www/html/joomla  
    DirectoryIndex index.php index.html  
    ErrorLog ${APACHE_LOG_DIR}/joomla-error.log  
    CustomLog ${APACHE_LOG_DIR}/joomla-access.log combined  
  
    <Directory /var/www/html/joomla>  
        Options FollowSymLinks  
        AllowOverride All  
        Require all granted  
    </Directory>  
</VirtualHost>
```

Ở đây, ta sử dụng một tên miền giả là **joomla.local** trả về site Joomla (có thể chọn tên khác tùy ý) ³⁷ ³⁸. `AllowOverride All` cho phép Joomla sử dụng file .htaccess để cấu hình URL, v.v.

- Kích hoạt cấu hình này và tắt site mặc định (nếu cần):

```
sudo a2ensite joomla.conf  
sudo a2dissite 000-default.conf # (tắt site mặc định nếu xung đột)  
sudo systemctl reload apache2
```

Lệnh `a2ensite` tạo liên kết kích hoạt VirtualHost Joomla, rồi reload Apache để áp dụng ³⁹.

- Thêm tên miền vào file hosts của máy để `joomla.local` trả về localhost: mở file hosts:

```
sudo nano /etc/hosts
```

Thêm dòng:

```
127.0.0.1 joomla.local
```

Lưu lại. Bước này giúp máy bạn hiểu `joomla.local` là địa chỉ cục bộ. (Nếu không làm bước này, trình duyệt sẽ không tìm thấy tên miền tùy chỉnh) ⁴⁰.

Bước 4: Chạy trình cài đặt Joomla qua giao diện web: Mở trình duyệt, vào địa chỉ `http://joomla.local/` (hoặc `http://localhost/joomla` nếu không dùng virtual host). Bạn sẽ thấy giao diện **Joomla Web Installer**. Các bước cài đặt Joomla thường gồm 3 trang chính: 1. **Cấu hình chính:** Nhập tên site, mô tả (có thể bỏ qua), email quản trị, tên user admin, mật khẩu admin. Ghi nhớ user/password admin này để đăng nhập sau. Nhấn "Tiếp theo". 2. **Cấu hình cơ sở dữ liệu:** Chọn loại cơ sở dữ liệu MySQL (MySQLi), host (localhost), tên database (`joomladb`), tên user database (`usuariojoomla`), mật khẩu database (`123password`), tiền tố bảng (mặc định hoặc tùy chỉnh) ⁴¹. Nhấn "Tiếp theo". 3. **Cài đặt cuối:** Joomla hỏi cài dữ liệu mẫu hay không (có thể chọn None hoặc sample data tùy mục đích). Kiểm tra thông tin tóm tắt, sau đó nhấn "Install".

Quá trình cài đặt sẽ chạy trong vài chục giây. Nếu mọi cấu hình đúng, sẽ hiện thông báo **cài đặt hoàn thành**. Joomla yêu cầu xóa thư mục `installation` vì lý do bảo mật – chỉ cần nhấn nút "Remove installation folder" (Joomla sẽ tự xóa) ⁴².

Sau đó, bạn có thể nhấn vào nút **Administrator** để truy cập trang quản trị Joomla, hoặc **Site** để vào trang chủ. Trang quản trị Joomla (backend) có đường dẫn `http://joomla.local/administrator`, đăng nhập bằng tài khoản admin đã tạo và bạn sẽ thấy bảng điều khiển Joomla ⁴³. Trang front-end (frontend) là `http://joomla.local/` sẽ hiển thị giao diện website Joomla mặc định ⁴⁴.

Đến đây, Joomla đã được cài đặt thành công trên máy cá nhân. Bạn có thể thử tạo một bài viết mới trong admin, thay đổi giao diện, cài thêm plugin nếu muốn tìm hiểu sâu hơn. Tuy nhiên, trong phạm vi ôn thi, quan trọng là biết các bước cài và các file cấu hình liên quan (file cấu hình Apache cho Joomla, cơ sở dữ liệu Joomla kết nối,...). Joomla lưu cấu hình kết nối DB và site trong file `configuration.php` tại thư mục web của Joomla – có thể mở file này để thấy các thông số đã khai báo (không cần chỉnh nếu cài đặt thuận lợi).

Cài đặt Moodle trên LAMP (trên Ubuntu)

Tương tự Joomla, ta sẽ cài **Moodle** – hệ quản lý học tập. Moodle phức tạp hơn một chút do cần thêm thư mục lưu dữ liệu riêng và nhiều extension PHP, nhưng nhìn chung cài đặt không quá khó. Giả sử ta cài Moodle 4.x trên cùng môi trường Ubuntu:

Bước 1: Tạo cơ sở dữ liệu và user cho Moodle: Thực hiện trong MySQL shell (giống như với Joomla nhưng tên khác):

```
CREATE DATABASE moodle DEFAULT CHARACTER SET utf8mb4 COLLATE
utf8mb4_unicode_ci;
```

```
CREATE USER 'usuariomoodle'@'localhost' IDENTIFIED BY 'password123';
GRANT ALL PRIVILEGES ON moodle.* TO 'usuariomoodle'@'localhost';
QUIT;
```

Lệnh trên tạo database **moodle** và user **usuariomoodle** (mật khẩu **password123**) rồi cấp quyền [45](#) [46](#). (Lưu ý: Moodle khuyến cáo dùng `utf8mb4_unicode_ci` cho tương thích tốt). Đảm bảo nhớ các thông số này.

Bước 2: Tải xuống và triển khai mã nguồn Moodle: Vào trang download Moodle để lấy link. Ví dụ tải Moodle phiên bản ổn định mới nhất (giả sử 3.9 hoặc 4.x) bằng wget:

```
cd /tmp
wget https://download.moodle.org/download.php/direct/stable39/moodle-
latest-39.tgz
```

Sau đó giải nén vào thư mục web:

```
sudo tar -xzf moodle-latest-39.tgz -C /var/www/html/
```

Lệnh trên sẽ tạo thư mục `/var/www/html/moodle` chứa mã nguồn Moodle [47](#). Phân quyền thư mục Moodle:

```
sudo chown -R www-data:www-data /var/www/html/moodle/
```

(do Moodle cần quyền ghi trong thư mục cài đặt của chính nó) [48](#) [49](#).

Bước 3: Tạo thư mục dữ liệu (moodledata): Moodle yêu cầu một thư mục riêng để lưu các file upload, nội dung khóa học, v.v., *tách biệt* khỏi mã nguồn (để tăng bảo mật). Ta tạo:

```
sudo mkdir /var/www/moodledata
sudo chown -R www-data:www-data /var/www/moodledata
```

Thư mục `/var/www/moodledata` này nằm ngoài `/var/www/html` nên sẽ không truy cập trực tiếp qua web được (điều này tốt, tránh lộ dữ liệu) [50](#) [51](#).

Bước 4: Cấu hình Apache cho Moodle (tuỳ chọn): Nếu chạy song song với Joomla, ta có thể tạo VirtualHost cho Moodle. Tương tự như Joomla: - Tạo file `/etc/apache2/sites-available/moodle.conf` với nội dung:

```
<VirtualHost *:80>
    ServerName moodle.local
    DocumentRoot /var/www/html/moodle
    DirectoryIndex index.php index.html
    <Directory /var/www/html/moodle>
        Options FollowSymLinks
```

```
        Require all granted
    </Directory>
</VirtualHost>
```

(Moodle thường dùng URL có `index.php` xử lý chính, không cần `.htaccess` phức tạp như Joomla trừ khi cài thêm plugin). Lưu file. - Kích hoạt site: `sudo a2ensite moodle.conf`, sửa file hosts (`127.0.0.1 moodle.local`), reload Apache như đã làm với Joomla.

Nếu không muốn làm VirtualHost, có thể truy cập Moodle qua `http://localhost/moodle` cũng được (khi đó đảm bảo Apache cho phép `DirectoryIndex index.php` trong thư mục Moodle).

Bước 5: Chạy trình cài đặt Moodle qua web: Mở trình duyệt đến `http://moodle.local/` (hoặc `localhost/moodle`). Trình cài đặt Moodle sẽ bắt đầu: - Chọn ngôn ngữ cài đặt (có tiếng Việt nếu cần, hoặc tiếng Anh). Tiếp tục. - Xác nhận đường dẫn cài đặt và đường dẫn thư mục dữ liệu. Moodle sẽ tự nhận biết đường dẫn đến mã nguồn (web) và đề xuất đường dẫn mặc định cho thư mục dữ liệu (thường là `/var/www/moodledata`). Vì ta đã tạo đúng thư mục và phân quyền, không cần thay đổi, cứ tiếp tục ⁵². - Chọn loại cơ sở dữ liệu: chọn MariaDB (hoặc Improved MySQLi). Tiếp tục ⁵³. - Cung cấp thông tin kết nối database: host (localhost), tên database (**moodle**), user (**usuariomoodle**), mật khẩu (**password123**). Nhấn Next. Moodle sẽ kiểm tra kết nối DB ⁵⁴. - Chấp nhận điều khoản GNU GPL của Moodle (Yes/Continue) ⁵⁵. - Kiểm tra môi trường: Moodle sẽ liệt kê danh sách các yêu cầu (PHP extension, phiên bản PHP, database, thư viện GD, etc.) ⁵⁶. Nếu ta đã cài đủ PHP extensions ở bước trước thì các mục sẽ hiện dấu "OK". Bất cứ mục nào không đạt sẽ hiện cảnh báo đỏ, cần cài bổ sung trước khi tiếp tục. Giả sử mọi thứ đều xanh, bấm tiếp tục. - Quá trình cài đặt sẽ chạy (Moodle tạo các bảng trong database, có thể mất vài phút) ⁵⁷. - Sau khi cài xong database, Moodle sẽ yêu cầu thiết lập **tài khoản quản trị**: bạn điền username (mặc định "admin"), mật khẩu (phải thỏa mãn độ mạnh: gồm chữ hoa, thường, số, ký tự đặc biệt, tối thiểu 8 ký tự), email quản trị, v.v. ⁵⁸. Điền xong, lưu lại. - Điền thông tin trang Moodle: tên trang, v.v. (có thể chỉnh sau). Hoàn tất.

Nếu thành công, Moodle sẽ chuyển đến trang chủ hoặc trang cá nhân của người quản trị ⁵⁹. Bạn có thể duyệt qua giao diện quản trị của Moodle (nhiều mục như quản lý khóa học, người dùng, v.v. khá phức tạp so với Joomla). Nhưng tối thiểu, đảm bảo rằng bạn có thể đăng nhập bằng tài khoản admin và thấy được bảng điều khiển.

Lưu ý: Moodle có nhiều yêu cầu tài nguyên hơn Joomla, cài trên máy yếu có thể hơi chậm. Nhưng để trình bày trong bài kiểm tra, chỉ cần chạy được cǎn bản, không cần nhập nhiều dữ liệu.

Các lệnh và thao tác quan trọng trong cài đặt CMS

Khi cài đặt Joomla, Moodle hay các web app khác, sinh viên cần thành thạo một số lệnh và thao tác sau, vì rất có thể giảng viên sẽ hỏi hoặc yêu cầu thực hiện trong bài kiểm tra:

- **Sử dụng lệnh apt để cài đặt gói phần mềm:** Đã đề cập ở trên, nhưng nhấn mạnh: cài đúng gói tên gì. Ví dụ nhiều bạn nhầm `mysql-server` và `mariadb-server` - trên Ubuntu 20.04, cài mariadb-server sẽ tự cung cấp MySQL (MariaDB) server. Cài `php` đôi khi chưa đủ module, cần nhớ cài thêm `php-mysql` để PHP kết nối MySQL. Hãy liệt kê các gói đã cài (có thể dùng `apt list --installed | grep php` để kiểm tra module PHP đã có). Nếu thiếu, dùng apt cài bổ sung ⁶⁰.

- **Lệnh quản lý dịch vụ:** `sudo systemctl start/stop/restart apache2` (bắt đầu/dừng/khởi động lại Apache), tương tự cho mysql (dịch vụ mysql hoặc mariadb). Nếu trong bài thi, site không chạy, có thể cần kiểm tra xem Apache/MySQL đã khởi động chưa.
- **Cấu hình VirtualHost Apache:** Biết vị trí file cấu hình (thư mục `/etc/apache2/sites-available/`), lệnh `a2ensite` và `a2dissite` để bật/tắt site, lệnh `a2enmod rewrite` (khi cần bật mod_rewrite cho hỗ trợ URL đẹp, Joomla cần cái này nếu dùng URL thân thiện). Biết lệnh `reload` Apache sau khi thay đổi cấu hình. Biết chỉnh file `/etc/hosts` khi dùng tên miền ảo cục bộ.
- **Thiết lập cơ sở dữ liệu qua MySQL CLI:** Cú pháp lệnh `CREATE DATABASE`, `CREATE USER`, `GRANT ALL PRIVILEGES` như đã trình bày. Đôi khi giảng viên có thể hỏi: *Làm thế nào tạo database và user MySQL?* – bạn có thể trả lời bằng cách liệt kê các câu lệnh SQL (như ví dụ trên) hoặc nói "dùng phpMyAdmin tạo bằng giao diện" (nhưng viết ra lệnh SQL sẽ chứng tỏ bạn nắm chắc hơn).
- **Quyền file và thư mục trên Linux:** Hiểu rằng web server (Apache) chỉ ghi được vào thư mục nếu quyền cho phép. Thư mục web thường thuộc về user `www-data`. Lệnh `chown -R www-data:www-data <thư_mục>` rất quan trọng để tránh lỗi cài đặt (nếu quên, Joomla/Moodle thường báo lỗi không tạo được file cấu hình hoặc không ghi được cache). Ngoài ra, `chmod` đôi khi cần thiết (vd `chmod 777` tạm thời cho thư mục dữ liệu nếu có vấn đề, nhưng 777 là không an toàn, nên dùng `chown` về đúng user thì hơn).
- **Cấu hình PHP (`php.ini`):** Trong một số trường hợp, để Joomla/Moodle chạy mượt, ta cần chỉnh vài thông số PHP. Ví dụ Joomla khuyên tăng `memory_limit` lên 512M, `upload_max_filesize` và `post_max_size` lên 256M, tăng `max_execution_time` nếu import dữ liệu lớn ⁶¹. Moodle cũng cần bật `intl`, `soap`, etc. Kiểm tra nhanh: mở file cấu hình PHP (ví dụ `/etc/php/7.4/apache2/php.ini` trên Ubuntu 20.04) và sửa các giá trị cần thiết, sau đó restart Apache ⁶². Trong thi, có thể không bắt buộc làm việc này, nhưng nếu đề bài yêu cầu upload file lớn chẳng hạn, mà PHP limit thấp, bạn nên biết cách tăng.

Sử dụng cơ bản CMS và các tình huống kiểm tra (phần LAMP & CMS)

Phần kiểm tra ứng dụng web mã nguồn mở thường chiếm 7 điểm. Sinh viên có thể được yêu cầu **trình bày kết quả cài đặt và sử dụng** ngay trong buổi 5 (như đề cương: cho SV về cài đặt trước, lên kiểm tra chạy) ⁶³. Một số tình huống và yêu cầu cụ thể có thể có thể có thể có:

- **Kiểm tra đã cài đặt thành công:** Giảng viên có thể yêu cầu mở trang web ứng dụng đã cài trên máy bạn. Ví dụ: truy cập `http://joomla.local` và `http://moodle.local` (nếu bạn cài cả hai) để xem trang chủ có hoạt động không. Nếu không dùng virtual host, có thể là `http://localhost/joomla`. Hãy đảm bảo trước buổi kiểm tra rằng các dịch vụ đang chạy và trang có thể truy cập được. Điểm 5/7 điểm giữa kỳ được dành cho việc "cài đặt chạy được và hiểu các file config" ⁶⁴, nghĩa là bạn phải chứng minh site hoạt động và bạn hiểu cấu hình (có thể bị hỏi: file cấu hình Apache tên gì, nằm ở đâu? Cách bạn tạo database? File cấu hình của ứng dụng ở đâu?).
- **Trình bày cấu hình đã thực hiện:** Ví dụ, giảng viên có thể yêu cầu: *Hãy mở file cấu hình virtual host bạn đã tạo hoặc file cấu hình kết nối database của Joomla/Moodle.* Bạn nên chuẩn bị để chỉ ra:
- File Apache virtual host (ví dụ `joomla.conf` như ở trên). Nắm nội dung chính: `ServerName`, `DocumentRoot` trả đúng, các thư mục cho phép.

- File hosts (nếu có chỉnh).
- Đối với Joomla, file `configuration.php` trong thư mục Joomla chứa thông tin như `$host`, `$user`, `$password`, `$db` (các biến PHP lưu host DB, user, pass, tên DB). Đối với Moodle, file `config.php` trong thư mục Moodle (sau khi cài xong sẽ được tạo) chứa cấu hình tương tự: `$CFG->dbuser`, `$CFG->dbname`, `$CFG->dataroot` (chỉ đến thư mục moodledata), v.v. Hiểu những điều này cho thấy bạn "nắm các file config".
- Nếu sử dụng Windows (XAMPP chẳng hạn), cấu hình sẽ khác (file `php.ini`, `my.ini`...). Nhưng nhìn chung khuyến khích dùng Linux để được điểm cao hơn (7 điểm thay vì 6)⁶⁵.
- **Chức năng quản trị cơ bản:** Phần "Sử dụng" chiếm 2 điểm⁶⁴. Điều này có thể đòi hỏi bạn thao tác một vài chức năng trong trang admin của ứng dụng. Ví dụ:
 - Trong **Joomla**: thêm một User mới (Users -> Manage -> New), hoặc tạo một Category và Article mới rồi xuất bản, hoặc vào Global Configuration và thiết lập gửi mail (ví dụ chuyển phương thức gửi mail sang SMTP và nhập thử thông tin SMTP... tất nhiên nếu không có SMTP thật thì khó kiểm tra gửi được, nhưng ít nhất biết chỗ cấu hình)⁶⁴. Bạn có thể được hỏi: làm thế nào để cho phép đăng ký người dùng, làm thế nào cấu hình trang web ở chế độ offline,... Những thứ này đều làm trong admin Joomla.
 - Trong **Moodle**: tạo một khóa học mới, thêm người dùng mới (vai trò học viên), hay đơn giản là bật chế độ chỉnh sửa trên trang course. Moodle cũng có thiết lập gửi email (để thông báo cho học viên), bạn nên biết chỗ cấu hình (Site administration -> Server -> Email). Nếu giảng viên yêu cầu "gửi email cấu hình..." thì có thể họ muốn thấy bạn biết vào cấu hình SMTP hoặc thử chức năng gửi thư thông báo⁶⁴. Dĩ nhiên, Moodle rất rộng, nên khả năng cao chỉ cần chứng minh bạn biết tạo nội dung cơ bản.
 - Với **phpMyAdmin** (nếu có): có thể không hỏi trực tiếp, nhưng nếu Joomla gặp lỗi kết nối DB, bạn có thể nhanh chóng mở phpMyAdmin để kiểm tra database và user tồn tại hay chưa. Biết sử dụng phpMyAdmin là lợi thế nhưng không phải mục tiêu chính.
- **Cài đặt hai ứng dụng web trở lên:** Một yêu cầu có thể có: cài 2 CMS khác nhau (ví dụ Joomla + Moodle) và trình bày được cả hai. Điều này thử thách khả năng cấu hình nhiều site trên cùng server. Nếu bạn đã làm theo hướng dẫn trên (dùng virtual hosts `joomla.local`, `moodle.local`) thì việc chạy đồng thời rất gọn gàng. Trong kiểm tra, hãy mở lần lượt cả hai trang để cho thấy chúng độc lập. Bạn có thể giải thích: "Em đã cấu hình 2 VirtualHost, mỗi cái trỏ đến một thư mục và database riêng, nên 2 hệ thống chạy song song không xung đột." Điều này sẽ gây ấn tượng tốt.
- **Xử lý sự cố:** Có thể giảng viên sẽ hỏi tình huống "nếu làm bước X bị lỗi Y thì sao?". Ví dụ:
 - *Lỗi port 80 bị chiếm, Apache không chạy được:* Có thể do dịch vụ khác (IIS, Nginx) chạy trước đó. Giải quyết: dừng dịch vụ kia hoặc đổi Apache sang port khác.
 - *Truy cập `joomla.local` không được:* Kiểm tra file hosts, kiểm tra đã a2ensite chưa, Apache đã reload chưa.
 - *Trang Joomla báo không kết nối được MySQL:* Có thể do sai mật khẩu DB trong khi cài, hoặc MySQL chưa chạy. Kiểm tra dịch vụ, hoặc mở file `configuration.php` để sửa lại thông tin kết nối (nếu cần, Joomla cài xong mới có file này).
 - *Trang Moodle báo thiếu extension PHP:* Cài bổ sung bằng apt rồi restart Apache, reload trang cài đặt để tiếp tục.
 - *Không đăng nhập được admin:* Kiểm tra đã gõ đúng pass, capslock... Nếu quên mật khẩu admin Joomla/Moodle, biết cách reset (vd Joomla có thể đổi trong database user table, Moodle có thể dùng lệnh PHP script dòng lệnh). Nhưng hy vọng không cần đi xa vậy trong bài thi.

Nhìn chung, phần này kiểm tra **thực hành** là chính. Do đó cách ôn tốt nhất là thực sự tự cài ít nhất một lần **Joomla và Moodle** (hoặc Joomla + một ứng dụng khác bạn chọn) trên máy mình, ghi chú các bước và vấn đề gặp phải. Khi đã làm được, lúc kiểm tra bạn chỉ việc lặp lại và giải thích những gì mình đã làm.

Tài liệu tham khảo và trích dẫn

Trong quá trình học và cài đặt, sinh viên có thể tham khảo các nguồn tài liệu chính thống để hiểu sâu hơn:

- Tài liệu hướng dẫn shell script căn bản trên Ubuntu [3](#) [17](#).
- Bài viết tổng quan về shell script (ITviec) – giải thích khái niệm và cú pháp shell script một cách chi tiết [1](#) [13](#).
- Tài liệu thực hành Phát triển PM mã nguồn mở (nội dung buổi 3,4,5) do giảng viên cung cấp [66](#) [67](#).
- Hướng dẫn cài LAMP stack trên Ubuntu và cài Joomla, Moodle:
 - Cài đặt Joomla trên Ubuntu 20.04 (Ubunlog) [68](#) [21](#).
 - Cài đặt Moodle trên Ubuntu 20.04 (Ubunlog) [30](#) [49](#).
- Tài liệu chính thức Joomla, Moodle (trên trang chủ của dự án) để tra cứu thêm cấu hình nâng cao.
- Các lệnh Linux và apt (như blog VinaHost) để nắm vững quản lý gói trên Ubuntu [11](#).

Mục tiêu cuối cùng: Sau khi ôn tập những nội dung trên, sinh viên cần tự tin rằng mình có thể: - Viết được một shell script đơn giản, hiểu và sửa được các lỗi cơ bản trong shell script. - Thiết lập được môi trường LAMP và cài đặt ít nhất một ứng dụng web mã nguồn mở (ưu tiên Joomla, Moodle) thành công trên máy cá nhân. - Giải thích được các bước đã làm và thực hiện một số thao tác quản trị cơ bản trên ứng dụng (thêm nội dung, user, cấu hình thông số chính). - Với kiến thức và kỹ năng đó, sinh viên hoàn toàn có thể vượt qua bài kiểm tra giữa kỳ, đạt điểm 5 trở lên như mục tiêu đề ra, thậm chí cao hơn nếu trình bày rõ ràng, hệ thống và thể hiện được sự hiểu biết sâu về vấn đề.

Chúc các bạn ôn tập hiệu quả và đạt kết quả tốt trong kỳ kiểm tra! Các nội dung trên không chỉ giúp qua môn mà còn là nền tảng hữu ích cho việc phát triển và triển khai phần mềm mã nguồn mở thực tế.

[1](#) [2](#) [7](#) [8](#) [13](#) [14](#) [15](#) [16](#) [22](#) Shell script là gì: Từ A-Z về công cụ dòng lệnh cho lập trình viên - ITviec Blog

<https://itviec.com/blog/shell-script-la-gi/>

[3](#) [4](#) [5](#) [6](#) [9](#) [10](#) [17](#) Cách Tạo Và Sử Dụng Bash Script Cơ Bản Trong Ubuntu
<https://viblo.asia/p/cach-tao-va-su-dung-bash-script-co-ban-trong-ubuntu-Qbq5QpAJD8>

[11](#) [12](#) [60](#) Hướng Dẫn Cài Phần Mềm Trong Linux (APT) - VinaHost
<https://blog.vinahost.vn/cai-phan-mem-trong-linux-apt/>

[18](#) Automating User Account Creation with Password in Shell | Baeldung on Linux
<https://www.baeldung.com/linux/automate-user-account-creation>

[19](#) [20](#) bash - Linux Shell script to add a user with a password from a list - Stack Overflow
<https://stackoverflow.com/questions/39728702/linux-shell-script-to-add-a-user-with-a-password-from-a-list>

21 28 29 31 33 34 35 36 37 38 39 40 41 42 43 44 61 62 68 Joomla với Apache2, cài đặt cục bộ
trên Ubuntu 20.04

<https://vi.ubunlog.com/joomla-v%E1%BB%9Bi-apache2-c%C3%A0i-%C4%91%E1%BA%B7t-c%E1%BB%A5c-b%E1%BB%99-ubuntu-20-04/>

23 25 27 LAMP – Wikipedia tiếng Việt

<https://vi.wikipedia.org/wiki/LAMP>

24 26 LAMP là gì? Tổng quan về LAMP/LEMP stack | TopDev

<https://topdev.vn/blog/lamp-la-gi/>

30 32 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 Moodle, một hệ thống quản lý học tập
trong Ubuntu 20.04 LTS

<https://vi.ubunlog.com/h%E1%BB%87-th%E1%BB%91ng-qu%E1%BA%A3n-l%C3%BD-h%E1%BB%8Dc-t%E1%BA%ADp-moodle-ubuntu-20-04-lts/>

63 64 65 66 67 noi-dung-thuc-hanh.pdf

file:///file_00000008788720688c79abe38b391a0