**THỰC HÀNH BUỔI 1**

# Mục tiêu

* Cài đặt hệ điều hành Ubuntu lên máy ảo VirtualBox
* Làm quen và tìm hiểu cách sử dụng một số ứng dụng trên Ubuntu
* Thực hành một số lệnh cơ bản của shell
* Thiết lập cấu trúc dự án, môi trường phát triển, git repo
* Kết nối môi trường thông qua SSH

# Yêu cầu chuẩn bị

* Cài đặt phần mềm tạo máy ảo VirtualBox
* Tải và cài đặt các phần mềm cần thiết: Visual Studio Code,
* Cài đặt hệ điều hành Ubuntu trên phần mềm VirtualBox theo hướng dẫn trong phần thực hành
* Download Ubuntu Desktop ISO tại link:

<https://ubuntu.com/download/desktop/thank-you?version=24.04.3&architecture=amd64&lts=true>

# Định nghĩa và Cấu trúc Hệ Điều Hành Linux

## Linux là gì?

* Linux là một hệ điều hành mã nguồn mở, được phát triển dựa trên nền tảng Unix.
* Được sử dụng rộng rãi trong máy chủ, lập trình, hệ thống nhúng, điện thoại, và nghiên cứu khoa học.
* Một số bản phân phối phổ biến: **Ubuntu, Debian, CentOS, Fedora**

## Cấu trúc hệ điều hành Linux

* **Kernel** (nhân hệ điều hành): Quản lý tài nguyên phần cứng (CPU, RAM, ổ đĩa, thiết bị ngoại vi…). Là “trái tim” của hệ điều hành.
* **Shell**: Giao tiếp giữa người dùng và hệ thống, chuyển đổi các lệnh sang ngôn ngữ máy cho kernel thực hiện. Ví dụ: bash, zsh, sh.
* **File System**: Toàn bộ dữ liệu trên Linux được quản lý dưới dạng cây thư mục. Mọi thứ đều là file (thiết bị, cài đặt, user, ứng dụng…).
* **System Libraries**: Các thư viện giúp ứng dụng truy cập tài nguyên hệ thống.
* **User Space**: Các ứng dụng, tiện ích, desktop environment… mà người dùng nhìn thấy. (/home/[user])
* **Daemons/Services**: Các tiến trình chạy ngầm phục vụ hệ thống.

## Phân biệt 2 dạng Ubuntu

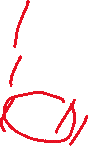
* **Ubuntu Desktop**
* Là phiên bản dành cho người dùng máy tính cá nhân. Đi kèm giao diện đồ họa (GUI) thân thiện như GNOME, dễ thao tác bằng chuột, thích hợp học tập, văn phòng, lập trình giao diện và sử dụng thường ngày.
* **Ubuntu Server**
* Là phiên bản được thiết kế cho máy chủ ("server"). Không có giao diện đồ họa mặc định, chủ yếu thao tác qua dòng lệnh (terminal), tập trung tối ưu tài nguyên cho các dịch vụ mạng như web server, database, cloud, container...

**So sánh chi tiết**

| **Tiêu chí** | **Ubuntu Desktop** | **Ubuntu Server** |
| --- | --- | --- |
| Giao diện | Có GUI (GNOME, KDE, ...), thao tác trực quan | Không có GUI, chỉ dòng lệnh (CLI) |
| Tài nguyên sử dụng | Tiêu tốn RAM + CPU nhiều hơn | Nhẹ hơn, tối ưu cho hiệu suất server |
| **Cài đặt ứng dụng** | Cài qua **giao diện** hoặc **dòng lệnh** | Chỉ qua **dòng lệnh** |
| Mục tiêu sử dụng | Người dùng cá nhân, lập trình, văn phòng | Máy chủ web, lưu trữ, cloud, dịch vụ mạng |
| Phần mềm mặc định | Có nhiều phần mềm văn phòng, chat, trình duyệt | Chỉ có các dịch vụ server quan trọng |
| Update/Upgrade | Tự động cập nhật đồ họa và hệ thống | Tập trung cập nhật bảo mật, ổn định hệ thống |
| Quản trị | Quản lý dễ dàng đối với người mới | Thích hợp cho admin/server chuyên nghiệp |

A diagram of a computer program

AI-generated content may be incorrect.



# Hướng dẫn thực hành

## Cài đặt phần mềm tạo máy ảo VirtualBox

* Bước 1: Mở file cài đặt VitualBox và Next



* Bước 2: Chọn thư mục cài đặt sau đó chọn Next 2 lần trong 2 cửa sổ tiếp theo

A screenshot of a computer

AI-generated content may be incorrect.

* Bước 3: Chọn thư mục cài đặt sau đó chọn Next 2 lần trong 2 cửa sổ tiếp theo

A screenshot of a computer

AI-generated content may be incorrect.

* Bước 4: Nhấn Install.

A screenshot of a computer

AI-generated content may be incorrect.

* Bước 5: Chờ một thời gian khi chương trình cài đặt xong, và chọn Finish để hoàn thành cài đặt
* Bước 6: Chương trình sau khi cài đặt, chúng ta có thể sử dụng toàn bộ tính năng của chương trình

A screenshot of a computer

AI-generated content may be incorrect.

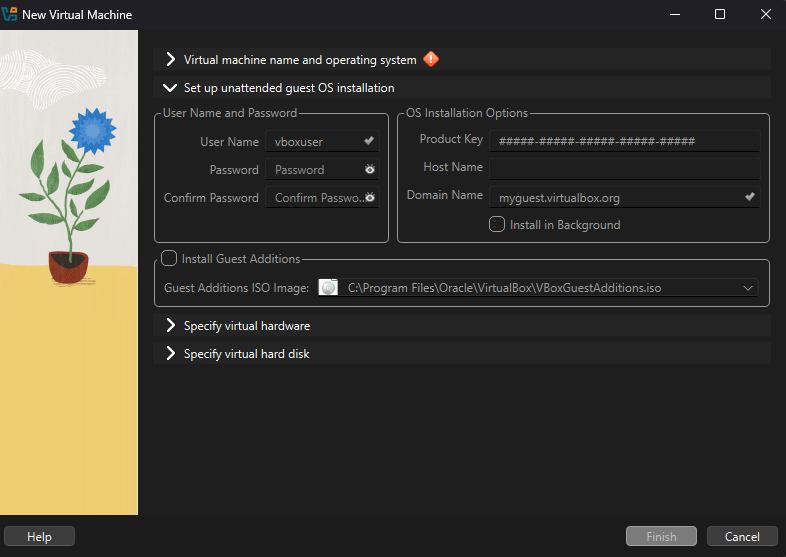
## Cài đặt hệ điều hành Ubuntu Desktop trên phần mềm VirtualBox.

* Bước 1: Chọn biểu tượng Create a New Virtual Machine (chọn New trong giao diện của VirtualBox)
* Bước 2: Đặt tên cho máy ảo, chọn loại hệ điều hành và phiên bản hệ điều hành dự kiến cài cho máy ảo
* A screenshot of a computer

  AI-generated content may be incorrect.

Phần ISO Image: Chọn file ISO đã download để cài đặt Ubuntu Desktop.

* Phần Setup unattended guest OS installation: Dùng để thiết lập tài khoản riêng (nếu cần) và Key cài đặt nếu sử dụng các OS có license.



Phần Specify Virtual Hardware: Dùng để phân bổ tài nguyên RAM và Số nhân CPU cho Ubuntu

A screenshot of a computer

AI-generated content may be incorrect.

Phần Create a New Virtual Hard Disk: Dùng để phân bổ ổ đĩa cho Ubuntu và định dạng loại ổ đĩa: VDI (Ổ đĩa được định dạng bởi VirtualBox)

A screenshot of a computer

AI-generated content may be incorrect.

* Bước 3: Cài đặt hệ điều hành Ubuntu: Bấm Install Ubuntu

A screenshot of a computer

AI-generated content may be incorrect.

Ngoài ra trong cấu hình của VitualBox cho phép lựa chọn nhiều dạng cấu hình **Network** khác, hiện tại cấu hình mặc định là **NAT**, tham khảo chi tiết tại: <https://www.nakivo.com/blog/virtualbox-network-setting-guide/>

## Thực hành các lệnh cơ bản của shell Linux.

### Định nghĩa shell

Máy tính chỉ có thể hiểu ngôn ngữ nhị phân 0 và 1, để giao tiếp với máy tính chúng ta cần đưa vào các lệnh dưới dạng chuỗi nhị phân này, điều đó khó có thể hiện thực. Để làm được điều này, hệ điều hành có một chương trình đặc biệt gọi là shell để chuyển các lệnh về ngôn ngữ máy để máy có thể hiểu được. Shell là chương trình giao tiếp với người dùng, chấp nhận các chỉ dẫn hoặc các lệnh bằng tiếng Anh và chuyển các lệnh đó tới kernel.

Trong phần này chúng ta sẽ làm việc với BASH shell, đây là một loại shell thông dụng nhất trong hệ điều hành Linux. Trong

Ubuntu để khởi động shell chúng ta bấm tổ hợp phím Ctrl + Alt + T.

Một vài lệnh liên quan đến shell:

 Liệt kê tất cả các loại shell có thể có trong hệ thống

$cat /etc/shells

 Kiểm tra loại shell đang được sử dụng

$echo $SHELL

### Lệnh -ls

Lệnh ls được dùng để liệt kê dữ liệu có trong một thư mục hay thư mục hiện thời.

ls [options] [folder]

Ví dụ minh họa:

ls

ls -laht

ls -laht /usr/local

Có thể nghiên cứu thêm các options của câu lệnh này với cú pháp “ls –-help”

### Lệnh – cd

$cd path

Lệnh cd là lệnh cơ bản nhất nếu bạn di chuyển thư mục hiện thời đến thư mục mà bạn muốn.

A screenshot of a computer program

AI-generated content may be incorrect.

Ví dụ minh họa:

cd /usr/local/lib/

cd home/admin/ cd ..

cd ../../../

cd ../home/admin/

Trong đó:

 cd /usr/local/lib/ Di chuyển tới thư mục /usr/local/lib bắt đầu từ thư mục gốc / của Linux/Unix.

* cd home/admin/ : Di chuyển đến thư mục home/admin/ từ thư mục hiện thời.
* cd .. / : Di chuyển về thư mục gốc của thư mục hiện thời.
* cd ../../../ Di chuyển về hướng thư mục gốc 3 lần. (Bạn đang ở /usr/local/lib/ sẽ về thư mục / là thư mục gốc của Linux/Unix).
* cd ../home/admin kết hợp 2 cấu trúc trên để di chuyển về thư mục gốc rồi lại truy cập đến thư mục home/admin.

### Đường dẫn trong hệ điều hành Linux

Đường dẫn (path) dùng để định vị một tài nguyên nào đó (thông thường là file) trong hệ thống cây thư mục. Có hai loại đường dẫn thường được sử dụng là: đường dẫn tuyệt đối và đường dẫn tương đối.

#### Đường dẫn tuyệt đối

Xác định đầy đủ toàn bộ đường dẫn từ thư mục gốc (ký hiệu :/) đến nơi cần định vị.

Ví dụ:

*/dev*

*/usr/bin*

*/usr/local/bin*

Để chuyển vào thư mục /usr/local/bin dùng lệnh như sau:

*$cd /usr/local/bin*

#### Đường dẫn tương đối

Là đường dẫn được tính từ vị trí hiện hành đến vị trí cần định vị. Một đường dẫn tương đối không bao giờ bắt đầu từ ký tự ‘/’.

Ví dụ: Giả sử chúng ta đang ở thư mục hiện hành là /usr. Khi đó để chuyển đến thư mục /usr/local/bin chúng ta sử dụng lệnh: *$cd local/bin*

### Lệnh – pwd

Lệnh pwd sẽ trả về cho bạn thư mục hiện thời bạn đang ở tại đó.

pwd

### Lệnh – touch

Lệnh touch là lệnh cho phép người dùng tạo một tệp tin mới dạng clear text. Nội dung trong tệp tin do người dùng quyết định.

*touch [options] [file name]*

*touch bash.sh*

Ngoài ra bạn có thể tham khảo các [options] qua câu “touch h”

### Lệnh – vi – nano

Lệnh vi và lệnh nano là hai lệnh cơ bản để đọc và chỉnh sửa văn bản dạng text trên Linux/Unix.

Trong đó, chủ yếu dùng vi, còn nano thì đã có một số thông báo và hướng dẫn khi dùng, chỉ cần chú ý đọc là bạn có thể sử dụng.

Khi dùng vi đọc một tệp tin, ví dụ như: info.php thì bạn đang ở chế độ “read” nghĩa là chỉ đọc văn bản. Muốn vào chế độ ghi, chỉnh sửa văn bản, bạn ấn “i” hoặc “Insert”. Để hủy chế độ ghi “INSERT” hãy ấn “Esc”.

Ví dụ: Nếu bạn muốn tìm từ “phpinfo” thì hãy ấn “Esc” để về chế độ đọc, rồi ấn “/phpinfo” /, đây chính là ký hiệu để yêu cầu tìm ký tự đằng sau mà bạn muốn, sau đó nhấn “Enter” để quá trình tìm kiếm được thực hiện.

Nếu đó vẫn chưa phải từ bạn muốn tìm, hãy ấn phím “n”, máy sẽ tự động tìm tiếp ký tự tiếp theo mà bạn đang muốn tìm kiếm trong đoạn văn bản.

Thoát khỏi quá trình soạn thảo của vi trở về chế độ đọc bằng phím “Esc” rồi gõ “:q” hoặc “:q!” để thoát. Khi bạn vô tình chỉnh sửa trong tệp tin nhưng không muốn lưu lại thì dùng “:q!” để thoát, còn “:wq” được dùng khi bạn muốn ghi sự thay đổi vào tệp tin đó. Ngoài ra, đối với trường hợp các tệp tin được đặt ở chế độ “read-only”, bạn bắt buộc phải sử dụng cú pháp “:wq!” để ghi vào tệp tin.

Lệnh nano đơn giản hơn vì đã có hướng dẫn sử dụng. Ngay sau khi thực thi lệnh nano, bạn đã có thể ghi và đọc tệp tin theo các hướng dẫn được sắp xếp ở dưới. Dấu ^ trong các chức năng như “^X” để Exit hay “^W” để Where is (tìm kiếm ký tự) chính là phím Ctrl.

### Lệnh – mkdir

Lệnh mkdir là mẫu lệnh giống trên Windows để tạo một thư mục.

*mkdir [options] [folder name]*

Ví dụ minh họa: *mkdir user mkdir -p /usr/local/src/ mkdir -m 777 /home/user/*

Trong đó:  mkdir user/ Sẽ tạo ra một thư mục user ngay tại thư mục hiện thời.

 mkdir -p /usr/local/src/ Tạo ra thư mục src tại đường dẫn /usr/local/ sẽ không báo lỗi nếu thư mục này đã tồn tại sẵn.

 mkdir -m 777 /home/user/ Tạo một thư mục user với đường dẫn /home/ và chmod 777 cho thư mục này luôn.

(chmod sẽ được giới thiệu ngay sau).

Ngoài ra các bạn có thể tìm hiểu sâu hơn các [options] nâng cao với cú pháp “mkdir –help”.

### Các lệnh về quản lý user trên Linux

User là người có thể truy cập đến hệ thống. User có username và password. Có hai loại user là: super user và regular user.

User – group () - others

Mỗi user có một định danh riêng gọi là UID.

 Tạo user Cú pháp:

$useradd [option] <username>

Ví dụ:

$useradd –c “He dieu hanh – Admin”adminhdh

 Thay đổi thông tin cá nhân Cú pháp:

$usermod [option] <username>

 Xóa người dùng Cú pháp:

$userdel [option] <username>

### Lệnh – chmod

Lệnh **chmod** là mẫu lệnh dùng để thiết lập quyền (xem, sửa và thực thi) trên các tệp tin và thư mục. Ngoài ra còn có những cách chmod dùng cho những tệp tin đặc biệt.

Đây là câu lệnh giúp sửa những lỗi liên quan đến thiết lập quyền khi chạy các tệp tin và thư mục.

chmod [options] [xxxxxx] [file or folder]

Mặc định khi tạo thư mục hay tệp tin trên Linux/Unix với dạng *umask 022* thì thư mục sẽ được chmod 755 và tệp tin được chmod 644.

CHMOD chính là thao tác thay đổi các quyền sau:

* “Read” (Đọc): viết tắt là “**r**”, và được biểu diễn bằng số **4**
* “Write” (Ghi / Chỉnh sửa): viết tắt là “**w**”, và được biểu diễn bằng số **2**
* “Execute” (Thực thi): viết tắt là “**x**”, và được biểu diễn bằng số **1**
* CHMOD cùng lúc thay đổi quyền hạn trên các file/thư mục với các đối tượng sau:
* “Owner” – chủ sở hữu của tệp tin/thư mục
* “Group” – nhóm mà Owner là thành viên,
* “Public/Others/Everybody” - những người còn lại.
* -rwxrwxr-x

chmod 764 bdu-oss.txt

A screenshot of a computer program

AI-generated content may be incorrect.

CHMOD = 755 cho các thư mục có nghĩa là:

 7 = 4 + 2 + 1 : Người sở hữu thư mục có quyền đọc thư mục (read); chỉnh sửa thư mục (write); liệt kê các thư mục và tệp tin bên trong (execute).

 5 = 4 + 0 + 1 : Những người cùng nhóm chỉ có quyền đọc thư mục (read); liệt kê các thư mục và tệp tin bên trong (execute).

 5 = 4 + 0 + 1 : Những người còn lại chỉ có quyền đọc thư mục (read); liệt kê các thư mục và tệp tin bên trong (execute).

Giống như vậy tệp tin được CHMOD = 644 có nghĩa là:  6 = 4 + 2 + 0 : Người sở hữu thư mục có quyền đọc tệp tin (read); chỉnh sửa tệp tin (write); không được thực thi (chạy) tệp tin này (execute).

 4 = 4 + 0 + 0 : Những người cùng nhóm chỉ có quyền đọc tệp tin (read).

 4 = 4 + 0 + 0 : Những người còn lại chỉ có quyền đọc tệp tin (read).

#### Chế độ tượng trưng

Lệnh **chmod** cũng cho phép sử dụng các ký hiệu tượng trưng để điều chỉnh một chế độ cụ thể mà không ảnh hưởng tới các chế độ khác. Chế độ tượng trưng gồm 3 thành phần, kết hợp lại thành một chuỗi:

chmod [tham chiếu][toán tử][chế độ][tệp tin hoặc thư mục]

Tham chiếu (hay lớp) dùng để xác định những người dùng nào sẽ được điều chỉnh quyền truy cập. Tham chiếu mặc định sử dụng là “all” cho tất cả các nhóm nếu không có tham chiếu được chỉ định. Tuy nhiên việc thay đổi các quyền truy cập còn phụ thuộc vào thiết lập umask. Tham chiếu được biểu diễn bởi các ký tự sau:

|  |  |  |
| --- | --- | --- |
| Tham chiếu | Nhóm | Mô tả |
| *u* | người dùng | chủ sở hữu tệp tin |
| *g* | nhóm | các người dùng thuộc nhóm của tệp tin |
| *o* | khác | các người dùng không thuộc nhóm cũng không phải chủ sở hữu |
| *a* | tất cả | tất cả ba lớp trên, tương đương với ugo |

Chương trình **chmod** sử dụng một toán tử để quy định cách thay đổi chế độ của tệp tin:

|  |  |
| --- | --- |
| Toán tử | Mô tả |
| *+* | Thêm chế độ sau đó vào tệp tin |
| *-* | Xóa chế độ theo sau trong tệp tin (nếu có) |
| *=* | Quy định chính xác chế độ cho tệp tin |

Một ký tự sẽ biểu diễn chế độ được thêm/xóa trên tệp tin. Có ba chế độ cơ bản như sau:

|  |  |  |
| --- | --- | --- |
| Chế độ | Tên | Mô tả |
| r | read | Đọc tệp tin/ liệt kê các tệp tin trong một thư mục |
| w | write | Ghi vào tệp tin/ tạo tệp tin mới trong thư mục |
| x | execute | Thực thi tệp tin/ xem một cây thư mục |
| X | special execute | Không hẳn là một quyền truy cập nhưng có thể sử dụng thay cho x. X đặt quyền thực thi cho thư mục bất chấp các quyền hiện có và đặt quyền thực thi có tệp tin đã có ít nhất 1 quyền thực thi đã thiết lập (chủ sở hữu, nhóm, khác). Lệnh chỉ hữu dụng khi sử dụng kèm với toán tử ‘+’ và thường sử dụng cùng với tùy chọn -R để đặt quyền truy cập cho nhóm hay “người dùng khác” trên một thư mục lớn mà không cần thiết lập quyền thực thi trên các tệp tin thông thường (ví dụ tệp tin văn bản), ví dụ khi dùng lệnh “chmod -R |
|  |  | a+rx”, tất cả tệp tin trong cây thư mục đều được gắn chế độ x trong khi lệnh “chmod -R a+rX ” chỉ gắn cờ cho các tệp tin có thể thực thi. |
| s | setuid/gid | Chế độ đặc biệt |
| t | sticky | Chế độ đặc biệt |

Dùng dấu phẩy để cách các chế độ tượng trưng nếu muốn thực hiện nhiều thay đổi trong một lệnh.

Minh họa cú pháp và giải thích:

|  |  |
| --- | --- |
| Cú pháp | Giải thích |
| chmod a+r file | Thêm chế độ read cho tất cả người dùng. |
| chmod a+x file | Thêm chế độ execute cho tất cả người dùng. |
| chmod a+rw file | Thêm chế độ read và write cho tất cả người dùng. |
| chmod +rwx file | Nếu như bạn viết câu lệnh như thế này, tệp tin sẽ được chuyển về chế độ 755 (-rwxr-xrx). |
| *Chmod u=rw,go=r file* | Chuyển chính xác quyền *read* và *write* cho người sở hữu (u), người dùng trong cùng nhóm người sở hữu và tất cả người dùng khác sẽ bị vô hiệu tất cả các quyền thiết lập.  Tương tự như 600 (-rw——-) |
| *chmod -R u+w, go-w directory* | Thêm chế độ *write* cho người sở hữu cho tất cả tệp tin và thư mục trong thư mục, đồng thời xóa toàn bộ quyền *write* với những người còn lại. |
| *chmod 777 file* | Cấp tất cả các quyền *read*, *write* và *execute* trên tệp tin cho tất cả các người sử dụng. |
| *chmod 664 file* | Loại bỏ tất cả các quyền *execute* của toàn bộ người dùng trên tệp tin và xóa quyền *write* trên tệp tin của những người dùng không phải chủ sở hữu hay cùng nhóm. |
| *chmod 0755 file* | Tương tự như 755. Số 0 được coi như không có chế độ đặc biệt. |
| *chmod 4755 file* | Đặt chế độ đặc biệt cho người dùng (4) và quyền 755 trên tệp tin. |
| *chmod 2755 file* | Đặt chế độ đặc biệt cho nhóm người dùng (2) và quyền 755 trên tệp tin. |
| *chmod -R u+rwX,g-rwx,orwx directory* | Đặt tất cả các thư mục con về rwx, các tệp tin về rw đối với người sở hữu và xóa tất cả các quyền đối với những người dùng khác. |
| *chmod -R a-x+X directory* | Loại bỏ toàn bộ quyền *execute* trên các tệp  tin trong thư mục. Thêm tất cả các quyền *execute* trên các thư mục cho người dùng. |

### Lệnh – rm

Lệnh rm là mẫu lệnh dùng để xóa dữ liệu trên Linux/Unix.

rm [options] [file or folder]

Lệnh này thường xuyên được sử dụng và có khá ít options.

Ví dụ minh họa:

rm -f /etc/rc.d/init.d/httpd

rm -rf /usr/local/

rm -vrf /usr/local/

 rm -f /etc/rc.d/init.d/httpd sẽ xóa tệp tin httpd trong thư mục /etc/rc.d/init.d/, máy tính sẽ không báo lỗi kể cả khi tệp tin này không tồn tại.

 rm -rf /usr/local/ Xóa toàn bộ thư mục /usr/local/ và toàn bộ dữ liệu chứa trong đó.

 rm -vrf /usr/local/ Giống với rm -rf nhưng khi xóa xong chúng sẽ thông báo đã xóa đi những dữ liệu tên gì trong thư mục này.

Các bạn có thể phối hợp các options khi cần thiết với cú pháp “rm –help”.

Ngoài ra còn có thể sử dụng lệnh rmdir để xóa thư mục.

### Lệnh – cp

Lệnh cp là lệnh dùng để sao chép dữ liệu.

cp [options] [link] [link]

Ví dụ minh họa:

cp httpd /etc/init.d/

cp httpd http

cp httpd /etc/init.d/http

cp -prf /etc/init.d/httpd/ /usr/local/http/

cp -vrf /etc/init.d/httpd/ /usr/local/

Trong đó:

 cp httpd /etc/init.d/ Sao chép tệp tin httpd ở thư mục hiện thời đến thư mục /etc/init.d/

 cp httpd http/ Sao chép tệp tin httpd thành một tệp tin khác có tên http ở cùng thư mục hiện thời.

 cp httpd /etc/init.d/http/ Sao chép tệp tin httpd ở thư mục hiện thời đến thư mục /etc/init.d/ và đổi tên nó thành http.  cp -prf /etc/init.d/httpd/ /usr/local/http/ Sao chép thư mục httpd ở thư mục /etc/init.d/ đến thư mục /usr/local/ và đổi tên thư mục ấy thành http (nếu không có thư mục nào tên như vậy ở /usr/local/). Nếu tồn tại một thư mục http thì thư mục sao chép sẽ được sao chép đến thư mục

/usr/local/http/. (Không báo lỗi, sao chép đè lên tệp tin nếu trùng tên).

 cp -vrf /etc/init.d/httpd/ /usr/local/ Sao chép thư mục httpd ở đường dẫn /etc/init.d/ đến thư mục /usr/local/ và thông báo đã sao chép những dữ liệu gì.

Nghiên cứu thêm các options khác dùng cú pháp “cp –help”.

### Lệnh – mv

Lệnh mv được dùng để di chuyển và đổi tên thư mục, tệp tin.

mv [options] [link] [link]

Lệnh này tương tự lệnh cp và các options cũng gần giống, nhưng mv không phải sao chép mà là di chuyển. Có thể hiểu nôm na cp và mv gần giống Copy và Cut trên Windows.

### Lệnh – echo

Lệnh echo trả cho ta một giá trị ra màn hình hoặc vào tệp tin. Lệnh này được sử dụng để hiển thị các dòng thông báo hoặc chèn vào một tệp tin.

echo [value]

echo [value] > [file]

echo [value] >> [file]

Lệnh echo có 3 cách sử dụng thông dụng nhất:

echo 1234

echo “include /usr/local/src/” > text.txt

echo “include /usr/local/src/” >> /usr/local/src/text.txt

 echo 1234/ Sẽ in ra màn hình dãy số “1234”

 echo “include /usr/local/src/” > text.txt/ Sẽ in dòng “include /usr/local/src/” trong dấu (“”) vào tệp tin text.txt ở thư mục hiện thời mà không giữ lại dữ liệu gì trong tệp tin.

 echo “include /usr/local/src/” >> /usr/local/src/text.txt/

Tương tự cú pháp trên giữ tất cả các dữ liệu trong tệp tin và dữ liệu thêm vào cuối tệp tin.

### Lệnh – cat

Lệnh cat giúp ta in ra màn hình hoặc vào tệp tin những dữ liệu từ một tệp tin nào đó.

cat [options] [file]

cat [options] [file] > [file]

cat [options] [file] >> [file]

Tương tự lệnh echo. Chúng ta có thể dùng các ký hiệu (>) (>>) để hỗ trợ cho việc in giá trị vào tệp tin. Những options cần thiết các bạn có thể tham khảo thêm với cú pháp “cat –help”.

### Các lệnh về network

Để xác định địa chỉ IP và các network interface ta dùng command:

$ ifconfig –a

Để xem những thiết bị được kết nối vào computer từ IRQ 1 - IRQ 15 :

$ cat /proc/interrupts

Thiết lập địa chỉ IP cho một card mạng ta dùng command sau :

$ ifconfig eth0 192.168.1.5 netmask 255.255.255.0 up

# Chuẩn bị môi trường phát triển

## Môi trường ảo (venv) trong Python

Môi trường ảo giúp bạn tạo một không gian độc lập để cài đặt các thư viện, tránh ảnh hưởng lẫn nhau giữa các project trên máy. Đây là kỹ năng căn bản mà mọi lập trình viên Python đều cần.

* **Lý do sử dụng:** Khi nhiều dự án dùng các phiên bản thư viện khác nhau (ví dụ, FastAPI bản 0.85 ở project A, nhưng 0.100 ở project B), nếu cài trực tiếp lên máy sẽ gây lỗi xung đột. Môi trường ảo giải quyết vấn đề đó bằng cách chỉ cài vào “folder riêng” của từng project.
* **Yêu cầu:** Python cài sẵn trên máy.

Lệnh

python3 -m venv venv

* *python3* gọi chương trình Python phiên bản 3.
* *-m venv* là tham số cho phép chạy module venv (môi trường ảo).
* *venv* cuối cùng là tên thư mục chứa môi trường ảo sẽ được tạo.

Sau khi chạy lệnh này, thư mục venv sẽ chứa bản sao Python và các công cụ quản lý gói, bạn có thể **kích hoạt** môi trường ảo để cài đặt thư viện chỉ dùng cho dự án đó

## requirements.txt là gì?

**requirements.txt** là file liệt kê tên và phiên bản các package mà dự án cần. Nhờ đó, bạn (hoặc bất kỳ thành viên nào) chỉ cần chạy một lệnh để cài đúng bộ package cần thiết cho dự án, không thiếu hoặc thừa.

* Ví dụ nội dung:

# Backend dependencies

fastapi==0.104.1

uvicorn[standard]==0.24.0

sqlalchemy==2.0.23

alembic==1.13.1

pydantic==2.5.0

python-multipart==0.0.6

# Testing tools

pytest==7.4.3

httpx==0.25.2

pytest-asyncio==0.21.1

# Development utilities

python-dotenv==1.0.0

## SSH là gì?

SSH (Secure Shell) là giao thức mạng bảo mật cho phép người dùng kết nối và quản lý máy chủ từ xa thông qua Internet hoặc mạng LAN. SSH thay thế các phương thức cũ không mã hóa như Telnet, giúp truyền dữ liệu, kiểm soát hệ thống an toàn bằng kỹ thuật mã hóa mạnh, ngăn chặn nguy cơ bị đánh cắp thông tin. Port mặc định của SSH là port **22**

### Cách thức hoạt động của SSH

SSH hoạt động theo mô hình Client–Server:

1. Máy client gửi yêu cầu kết nối đến server (thường là máy chủ Linux/Unix).
2. Quá trình xác thực: Sử dụng mật khẩu hoặc cặp khóa công khai/riêng tư.
3. Tạo phiên làm việc mã hóa, bảo vệ mọi dữ liệu truyền nhận.
4. Sau khi xác thực thành công, client có thể thực hiện lệnh, tải file, hoặc bật dịch vụ trên server từ xa.

### Các cách xác thực SSH phổ biến

* **Bằng mật khẩu**: Nhập user và password của server khi kết nối.
* **Bằng cặp Public Key / Private Key**: Cấu hình cho phép đăng nhập không cần mật khẩu, an toàn và automation hơn.

### Khi nào dùng SSH?

* Quản lý, cấu hình, kiểm soát server từ xa (Linux VPS, máy chủ web, dịch vụ trên cloud...)
* Chuyển file/thực hiện backup dữ liệu an toàn
* Tạo kênh truyền tải bảo mật cho các dịch vụ khác: SFTP, port forwarding

## Các bước thiết lập

**Bước 1: Mở Terminal:** Nhấn Ctrl + Alt + T hoặc tìm "Terminal" trong menu ứng dụng.

**Bước 2: Kiểm tra/cài đặt Python:**

Gõ

python3 --version #để kiểm tra phiên bản Python.

Nếu chưa có, cài đặt bằng lệnh:

sudo apt update

sudo apt install python3 python3-pip

**Bước 3: Tạo môi trường ảo Python (venv):**

Gõ:

python3 -m venv [tên]

source venv/bin/activate

pip install -r requirements.txt

Khi môi trường ảo được kích hoạt, dấu hiệu là đầu dòng terminal sẽ hiện (venv).

**Bước 4: Cài đặt Git:**

Kiểm tra bằng

git --version

Nếu chưa có, cài đặt:

sudo apt install git

**Bước 5: Cài đặt OpenSSH server**

sudo apt install openssh-server

**Bước 6: Kiểm tra trạng thái SSH service**

sudo service ssh status

Nếu hiện "active (running)" là SSH server đã hoạt động.

|  |
| --- |
| ***Lưu ý****: Do sử dụng chế độ config network của VirtualBox là NAT, nên cần 1 bước thiết lập port forwarding trên VirtualBox như sau:*  Mở VirtualBox 🡪 Machine 🡪 Setting 🡪 Network 🡪 Adapter 1  *Chọn Port Forwarding (mục đích: thay đổi chỉ định port SSH mặc định (22) tại máy windows sang port mới (2222) và sẽ được forward vào port SSH trên ubuntu (22)*  Name Rule: SSH fw; Host Port: 2222; Guest Port 22  Trên máy Windows, bạn có thể kest kết nối tới server qua SSH theo hướng dẫn bên dưới (Terminal Windows 11 /Power Shell trên Windows 10, tuy nhiên bạn sẽ được hướng dẫn SSH nâng cao qua VSCode tại phần 6) |

**Kết nối tới server qua SSH**

Bước 1: Xác định máy chủ cần truy cập (địa chỉ IP hoặc tên miền). Sau khi biết user và IP, gõ lệnh:

ssh user@ip\_address

**Ví dụ:**

ssh student@192.168.1.123

Sau khi nhập, máy sẽ yêu cầu mật khẩu của user đó. Nếu đúng, bạn sẽ kết nối tới terminal của server từ xa.

**Mẹo: Nếu cần chỉ rõ port khác mặc định (port 22), dùng:**

ssh -p 2222 user@ip\_address

**Bước 7: Tổ chức thư mục dự án chuyên nghiệp**

Một dự án Python nên được tổ chức theo nguyên tắc sau để ai cũng dễ hiểu, bảo trì và phát triển:

**Cấu trúc mẫu:**

/home/admin/kanban-todo-fullstack/

│

├── README.md # Tổng quan dự án và hướng dẫn khởi tạo

├── .gitignore # Quy tắc Git ignore

├── requirements.txt # Dependencies Python backend

├── package.json # Dependencies frontend (nếu cần)

│

├── backend/ # Phần backend (FastAPI + SQLAlchemy)

│ ├── README.md # Hướng dẫn khởi động backend

│ ├── .env.example # Mẫu biến môi trường

│ ├── main.py # Entry point FastAPI

│ ├── requirements.txt # Dependencies riêng backend

│ │

│ ├── app/ # Mã nguồn FastAPI

│ │ ├── \_\_init\_\_.py

│ │ ├── models/ # Pydantic schemas & SQLAlchemy models

│ │ │ ├── schemas.py

│ │ │ └── task\_models.py

│ │ ├── api/ # Định nghĩa các router

│ │ │ ├── \_\_init\_\_.py

│ │ │ ├── tasks.py

│ │ │ └── board.py

│ │ ├── services/ # Business logic

│ │ │ ├── \_\_init\_\_.py

│ │ │ └── task\_service.py

│ │ └── database/ # Database connection & migrations

│ │ ├── connection.py

│ │ ├── models.py

│ │ └── repositories.py

│ │

│ ├── migrations/ # Alembic migrations

│ │ ├── env.py

│ │ └── versions/

│ │

│ └── tests/ # Backend tests (pytest)

│ ├── \_\_init\_\_.py

│ ├── test\_api.py

│ └── test\_services.py

│

├── frontend/ # Phần frontend (Vanilla JS + SortableJS)

│ ├── README.md # Hướng dẫn khởi động frontend

│ ├── index.html # Entry point HTML

│ ├── vercel.json # (Nếu dùng Vercel) cấu hình deploy

│ │

│ ├── src/ # Source code frontend

│ │ ├── js/ # JavaScript modules

│ │ │ ├── api.js # HTTP client

│ │ │ ├── app.js # Main app logic

│ │ │ ├── kanban-board.js # Load & render board

│ │ │ ├── task-card.js # Component task card

│ │ │ ├── modal.js # Task modal dialog

│ │ │ ├── advanced-features.js # (Buổi 7+) multi-select, undo/redo...

│ │ │ └── performance.js # (Buổi 7+) caching, virtual scroll...

│ │ │

│ │ ├── css/ # Stylesheets

│ │ │ ├── reset.css

│ │ │ ├── kanban.css

│ │ │ └── responsive.css

│ │ │

│ │ └── assets/ # Hình ảnh, icon

│ │ ├── icons/

│ │ └── images/

│ │

│ └── tests/ # Frontend tests (Jest)

│ ├── kanban.test.js

│ └── api.test.js

│

└── deployment/ # Script và cấu hình deploy

├── deploy\_backend.py # Script deploy backend (Railway CLI, etc.)

├── deploy\_frontend.py # Script deploy frontend (Vercel CLI)

├── health\_monitor.py # Health check & alerting

└── ci\_cd.yml # (GitHub Actions) CI/CD pipeline

# Môi trường phát triển nâng cao (SSH thông qua VSCode + Extension)

Bước 1: Mở VSCode, và chọn **Open a Remote Window**

A screenshot of a computer

AI-generated content may be incorrect.

Chọn Connect to Host

A screenshot of a computer

AI-generated content may be incorrect.

Chọn Add New SSH Host

A screenshot of a computer

AI-generated content may be incorrect.

Gõ lệnh ssh vào máy ảo trên VirtualBox, ví dụ

ssh vboxuser@localhost -p 2222

A screenshot of a computer

AI-generated content may be incorrect.

Chọn nơi lưu cấu hình ssh connection cho những lần sau, chọn [user]/.ssh/config

A screenshot of a computer

AI-generated content may be incorrect.

Chọn Connect:

A screenshot of a computer

AI-generated content may be incorrect.

Lúc này yêu cầu nhập password để truy cập vào máy ảo

A screenshot of a computer

AI-generated content may be incorrect.

Sau khi thành công, nhìn góc trái trạng thái SSH vào máy ảo.

A screenshot of a computer

AI-generated content may be incorrect.

Vào Exporer 🡪 Chọn Open Folder 🡪 Thư mục TODO\_Project đã thiết lập ở phần trước 🡪 Nhấn OK 🡪 Gõ password để cho phép access.

A screenshot of a computer

AI-generated content may be incorrect.

# Viết và chạy chương trình Python đầu tiên

1. **Tạo file mới:** Trong VS Code, chọn "New File", đặt tên: hello.py.
2. **Viết mã nguồn**:

p rint("Hello, Student!")

1. Mở terminal (Ctrl+Alt+`), khởi động môi trường ảo python:

python3 -m venv venv

source venv/bin/activate

1. Gõ lệnh:

python3 hello.py

1. Nếu thấy dòng chữ "Hello, Student!", bạn đã **HOÀN THÀNH** bước khởi tạo môi trường.

# Bài tập

Sử dụng các câu lệnh đã học, khởi tạo các thư mục và file theo đúng cấu trúc project tại **Bước 7**

Mỗi bạn tự tạo file Python nhỏ, in ra thông tin cá nhân (Tên, Mã SV, sở thích,…) bằng câu lệnh print().