



CLOUD

TÌM HIỂU VỀ LINUX

Phan Thị Hồng Huế - Week 2



PHẦN 1: TỔNG QUAN VỀ LINUX

1. LINUX LÀ GÌ?

a. Khái niệm

- Linux là một hệ điều hành (OS) mã nguồn mở, do Linus Torvalds tạo ra vào năm 1991 và được cộng đồng phát triển thành một trong những OS phổ biến nhất hiện nay. Được tạo bởi Linus Torvalds vào năm 2005.

b. Phân biệt giữa Linux kernel và các bản phân phối (Ubuntu, CentOS, Debian...).

- Linux distro – hay bản phân phối Linux là các hệ điều hành được phát triển dựa trên nhân Linux. Nó có vai trò đóng gói nhân Linux cùng với các phần mềm và tiện ích cần thiết, tạo thành một hệ điều hành hoàn chỉnh mà người dùng có thể cài đặt và khởi chạy.



PHẦN 1: TỔNG QUAN VỀ LINUX

1. LINUX LÀ GÌ?

Tên	Khái niệm	Ưu điểm	Nhược điểm
Linux kernel	Lõi của hệ điều hành Linux, quản lý phần cứng, bộ nhớ, tiến trình và hệ thống tập tin.	Nhỏ, hiệu suất cao, ổn định, có thể tùy chỉnh.	Không có giao diện người dùng, cần các bản phân phối để sử dụng đầy đủ.
Ubuntu	Bản phân phối phổ biến, thân thiện với người mới, dựa trên Debian	Giao diện thân thiện, dễ sử dụng, hỗ trợ phần cứng tốt, cộng đồng lớn.	Tích hợp nhiều gói phần mềm mặc định khiến hệ thống trở nên nặng nề với cấu hình thấp.
CentOS	Bản phân phối ổn định, miễn phí, gần giống Red Hat. Ổn định, thích hợp cho server, chu kỳ nâng cấp dài.	Ổn định, thích hợp cho server, chu kỳ nâng cấp dài.	Một số phiên bản đã EOL, hỗ trợ phần mềm mới hạn chế.

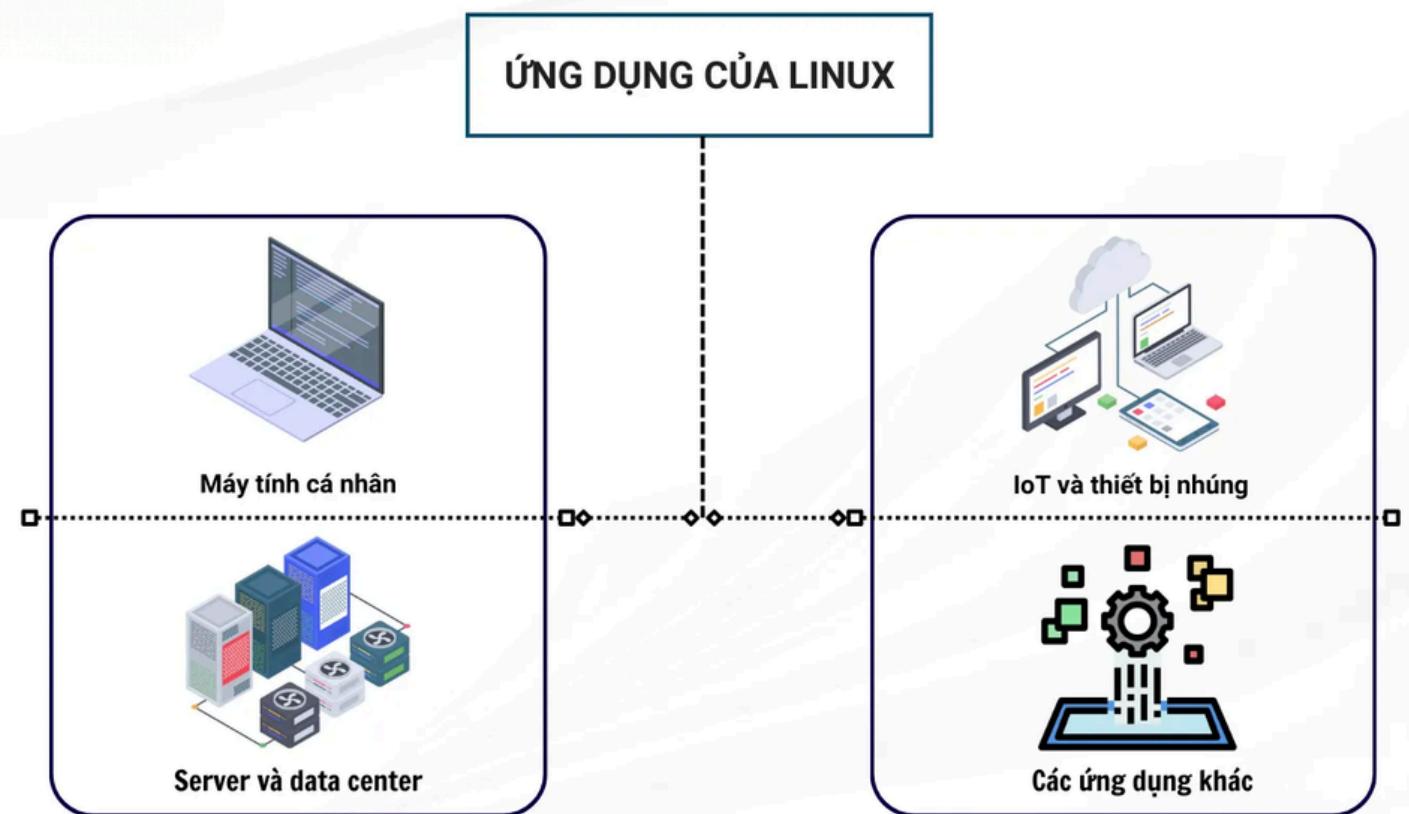
=> Kết luận:

- Linux Kernel: Lõi của hệ điều hành, quản lý phần cứng và tài nguyên.
- Bản phân phối (Distro): Ubuntu, CentOS, Debian... cung cấp giao diện, trình quản lý gói, phần mềm sẵn có.

PHẦN 1: TỔNG QUAN VỀ LINUX

2. TẠI SAO NÊN HỌC LINUX

- Ứng dụng rộng rãi trong server, cloud, DevOps, AI, lập trình hệ thống.



- Ưu điểm:**

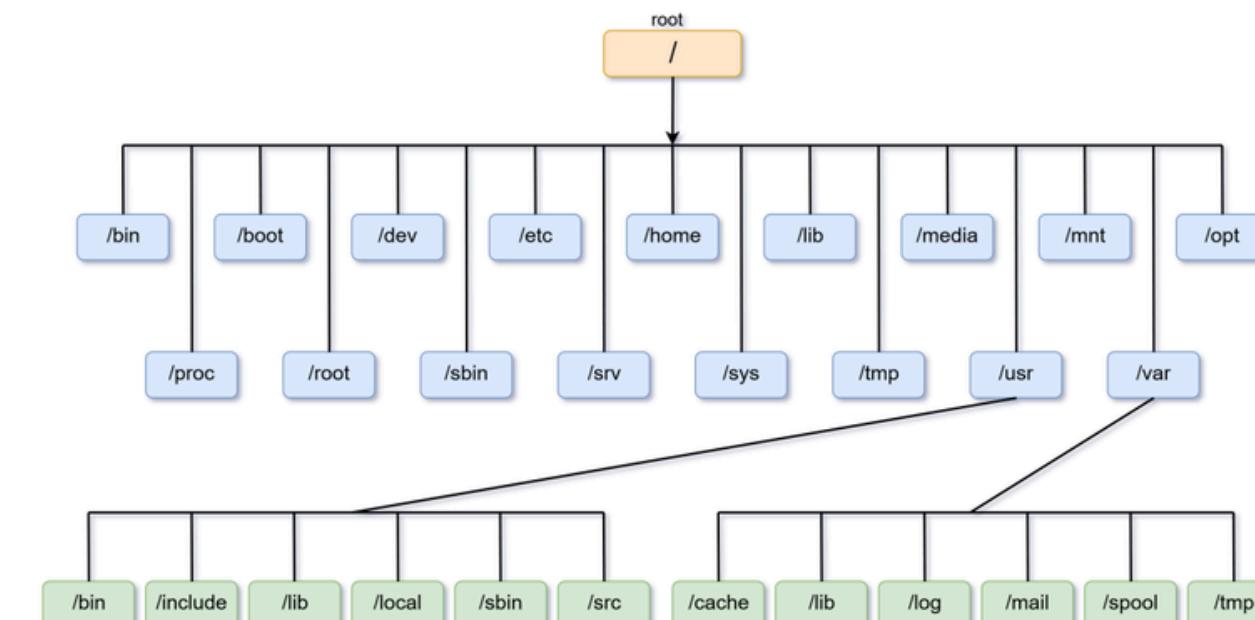
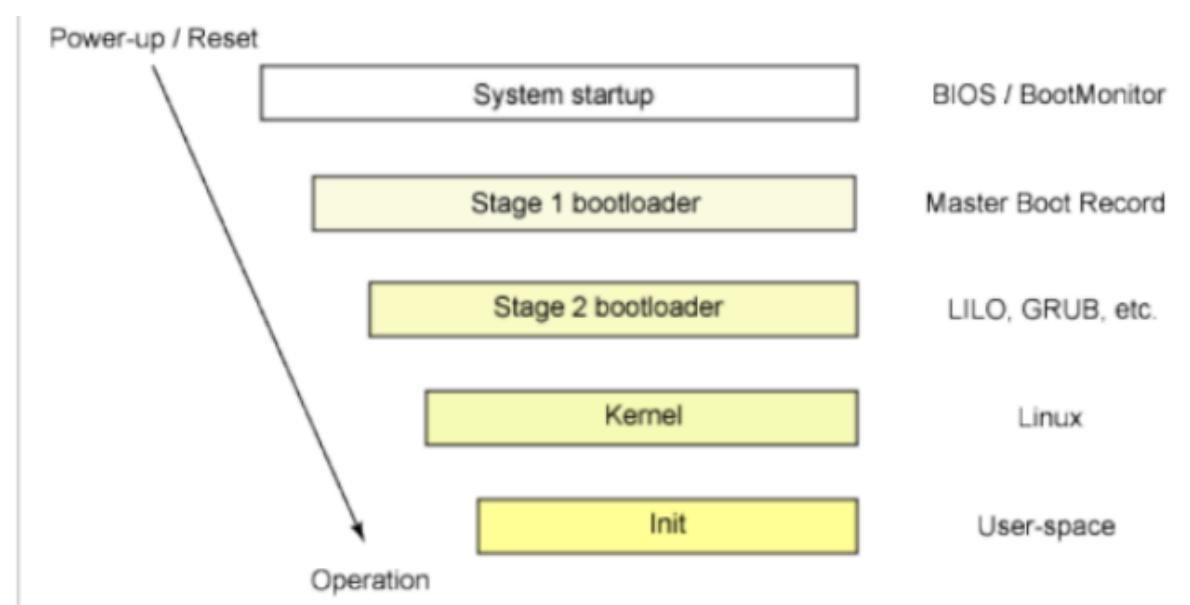
- Mã nguồn mở và miễn phí:
- Khả năng tùy chỉnh và linh hoạt:
- Bảo mật cao:
- Độ ổn định và tin cậy:
- Hỗ trợ cho máy cấu hình yếu:
- Hiệu suất cao:

PHẦN 1: TỔNG QUAN VỀ LINUX

3. KIẾN TRÚC HỆ THỐNG LINUX

- **Cấu trúc:** 3 thành phần chính:

- **Kernel (Nhân):** phần quan trọng nhất trong Linux
 - => quản lý các tài nguyên của phần cứng như bộ nhớ, module, bộ vi xử lý... và giúp phần cứng có thể giao tiếp với các ứng dụng trên hệ điều hành.
- **Shell:** chứa và thực thi các dòng lệnh được yêu cầu đến cho Kernel xử lý từ người dùng hoặc ứng dụng.
 - => cầu nối giữa Application và Kernel.
- **Application:** các ứng dụng hoặc tiện ích được người dùng cài đặt và chạy trên server
 - => phục vụ nhu cầu sử dụng (Ví dụ: Proxy, Samba và FTP...).



PHẦN 2: LÀM QUEN VỚI TERMINAL VÀ SHELL

4. TERMINAL & SHELL LÀ GÌ

- **Terminal:** giao diện dòng lệnh (CLI) – nơi nhập và hiển thị kết quả lệnh. → Không xử lý lệnh, chỉ chuyển lệnh đến Shell.
- **Shell:** là chương trình xử lý lệnh – nhận lệnh từ Terminal → thông dịch → gửi cho Kernel thực thi. → Sau đó nhận kết quả từ Kernel và trả về Terminal.

5. HIỂU CẤU TRÚC ĐƯỜNG DẪN

- **Đường dẫn tuyệt đối:** là đường dẫn đầy đủ bắt đầu từ thư mục gốc, không phụ thuộc vào vị trí hiện tại.
- **Đường dẫn tương đối:** là đường dẫn tính từ thư mục hiện tại, không bắt đầu bằng /
- **Dấu ~, ., .. ý nghĩa và cách dùng.**
 - ~: thư mục home của người dùng hiện tại
 - .: thư mục hiện tại
 - ..: thư mục cha

PHẦN 2: LÀM QUEN VỚI TERMINAL VÀ SHELL

6. LỆNH CƠ BẢN TRONG LINUX

- **pwd:** in ra
- **ls:** liệt kê nội dung các thư mục
 - **ls -l /:** hiển thị nội dung dạng danh sách
 - **ls -a /**: hiển thị các file thư mục ẩn
 - **ls -t /**: sắp xếp lại theo thứ tự mới nhất đến cũ nhất
 - **ls -lta:** gộp 3 câu lệnh trên
- **cd:** di chuyển giữa các thư mục
- **clear:** xóa sạch màn hình terminal
- **history:** hiển thị lịch sử các lệnh đã dùng
- **Tab:** tự động hoàn thành tên lệnh, tên file, thư mục..
- **Ctrl + C:** hủy tiến trình đang chạy
- **Ctrl + D:** đóng terminal hoặc thoát cell

```
huepth@lab-server:~$ pwd
/home/huepth
huepth@lab-server:~$ ls -l
total 29836
drwxrwxr-x 2 huepth huepth 4096 Oct 30 09:59 data
-rw-rw-r-- 1 huepth huepth 10240 Oct 30 10:03 data.tar
-rw-rw-r-- 1 huepth huepth 26650385 Oct 9 13:44 shoeshop-eCommerce.zip
-rw-rw-r-- 1 huepth huepth 441261 Oct 9 10:39 todolist.zip
-rw-rw-r-- 1 huepth huepth 3437908 Oct 30 10:09 vision.zip
huepth@lab-server:~$ ls -a
. .bashrc .profile todolist.zip
.. .cache shoeshop-eCommerce.zip .viminfo
.bash_history data .ssh vision.zip
.bash_logout data.tar .sudo_as_admin_successful .vscode-server
huepth@lab-server:~$ ls -t
vision.zip data.tar data shoeshop-eCommerce.zip todolist.zip
huepth@lab-server:~$ ls -lta
total 29876
-rw----- 1 huepth huepth 668 Oct 30 10:57 .bash_history
-rw-rw-r-- 1 huepth huepth 3437908 Oct 30 10:09 vision.zip
drwxr-xr-x 6 huepth huepth 4096 Oct 30 10:09 .
-rw-rw-r-- 1 huepth huepth 10240 Oct 30 10:03 data.tar
drwxrwxr-x 2 huepth huepth 4096 Oct 30 09:59 data
-rw----- 1 huepth huepth 1312 Oct 30 08:55 .viminfo
drwxr-x--- 2 huepth huepth 4096 Oct 29 16:24 .vscode-server
drwxr-xr-x 7 root root 4096 Oct 9 13:46 ..
-rw-rw-r-- 1 huepth huepth 26650385 Oct 9 13:44 shoeshop-eCommerce.zip
-rw-rw-r-- 1 huepth huepth 441261 Oct 9 10:39 todolist.zip
-rw-r--r-- 1 huepth huepth 0 Oct 9 09:38 .sudo_as_admin_successful
drwx----- 2 huepth huepth 4096 Oct 9 09:33 .cache
drwx----- 2 huepth huepth 4096 Oct 9 09:32 .ssh
-rw-r--r-- 1 huepth huepth 220 Feb 25 2020 .bash_logout
-rw-r--r-- 1 huepth huepth 3771 Feb 25 2020 .bashrc
-rw-r--r-- 1 huepth huepth 807 Feb 25 2020 .profile
huepth@lab-server:~$ |
```

PHẦN 3: LÀM VIỆC VỚI FILE VÀ THƯ MỤC

7. TẠO, XEM, XÓA VÀ DI CHUYỂN FILE

- **touch** : tạo 1 file mới
- **cat**: hiển thị nội dung file ra ngoài màn hình
- **less**: xem nội dung dài
- **head**: hiện thị những dòng đầu của file
- **tail**: hiển thị những dòng cuối của file

```
huepth@lab-server:~$ touch text.txt
huepth@lab-server:~$ cat text.txt
huepth@lab-server:~$ ls -a
. .bash_history .bashrc .profile .. .ssh text.txt .viminfo .vscode-server
.. .bash_logout .cache shoeshop-e-commerce.zip .sudo_as_admin_successful todolist.zip vision.zip
huepth@lab-server:~$ vi text.txt
huepth@lab-server:~$ cat text.txt
Xin chào
huepth@lab-server:~$ less /etc/passwd
huepth@lab-server:~$ vi text.txt
huepth@lab-server:~$ head text.txt
Xin chào
6. **Hiểu cấu trúc đường dẫn**
- Đường dẫn tuyệt đối vs tương đối.
- Đường dẫn tuyệt đối: là đường dẫn đầy đủ bắt đầu từ thư mục gốc, không phụ thuộc vào vị trí hiện tại.
- Đường dẫn tương đối: là đường dẫn tính từ thư mục hiện tại, không bắt đầu bằng '/'.
- Dấu '.', '..', '..' ý nghĩa và cách dùng.
- '.': thư mục **home** của người dùng hiện tại
- '..': thư mục hiện tại
- '../': thư mục cha

huepth@lab-server:~$ -3 head text.txt
-3: command not found
huepth@lab-server:~$ -n 3 head text.txt
-n: command not found
huepth@lab-server:~$ head -n 20 text.txt
Xin chào
6. **Hiểu cấu trúc đường dẫn**
- Đường dẫn tuyệt đối vs tương đối.
- Đường dẫn tuyệt đối: là đường dẫn đầy đủ bắt đầu từ thư mục gốc, không phụ thuộc vào vị trí hiện tại.
- Đường dẫn tương đối: là đường dẫn tính từ thư mục hiện tại, không bắt đầu bằng '/'.
- Dấu '.', '..', '..' ý nghĩa và cách dùng.
- '.': thư mục **home** của người dùng hiện tại
- '..': thư mục hiện tại
- '../': thư mục cha

---
## Phần 3: Làm việc với file và thư mục
7. **Tạo, xem, xóa và di chuyển file**
huepth@lab-server:~$ head -n 3 text.txt
Xin chào
6. **Hiểu cấu trúc đường dẫn**
- Đường dẫn tuyệt đối vs tương đối.
huepth@lab-server:~$ tail text.txt
- Dấu '.', '..', '..' ý nghĩa và cách dùng.
- '.': thư mục **home** của người dùng hiện tại
- '..': thư mục hiện tại
```

PHẦN 3: LÀM VIỆC VỚI FILE VÀ THƯ MỤC

7. TẠO, XEM, XÓA VÀ DI CHUYỂN FILE

- **cp:** sao chép file/ thư mục
- **mv:** di chuyển file/thư mục
- **rm:** xóa file hoặc thư mục
- **mkdir:** tạo thư mục mới
- **rmdir:** xóa thư mục trống

```
huepth@lab-server:~$ cp text.txt text2.txt
huepth@lab-server:~$ mkdir data
huepth@lab-server:~$ mv text.txt /data
mv: cannot move 'text.txt' to '/data/text.txt': Permission denied
huepth@lab-server:~$ ls -a
.                 .bash_logout  data                  .ssh
..                .bashrc       .profile               .sudo_as_admin_successful
.bash_history   .cache        shoeshop-eCommerce.zip  text2.txt
huepth@lab-server:~$ mv text.txt data/
huepth@lab-server:~$ cd data/
huepth@lab-server:~/data$ ls
text.txt
huepth@lab-server:~/data$ rm text.txt
huepth@lab-server:~/data$ cd .
huepth@lab-server:~/data$ cd ~
huepth@lab-server:~$ rmdir data

Command 'rmdir' not found, did you mean:

  command 'mkdir' from deb coreutils (8.30-3ubuntu2)
  command 'hmdir' from deb hfsutils (3.2.6-14)
  command 'rmdir' from deb coreutils (8.30-3ubuntu2)
  command 'mmkdir' from deb mblaze (0.6-1)

Try: sudo apt install <deb name>

huepth@lab-server:~$ rmdir data
huepth@lab-server:~$ |
```

PHẦN 3: LÀM VIỆC VỚI FILE VÀ THƯ MỤC

8. SAO CHÉP VÀ NÉN FILE

- **tar**: Đóng gói và giải nén nhiều file/thư mục:
 - c: create - tạo file nén .tar
 - x: extract - giải nén
 - v: verbose: hiển thị chi tiết quá trình
 - f: file - chỉ định tên file nén.
 - -z: nén bằng gzip
- **gzip**: Nén và giải nén file đơn lẻ
- **zip**: nén theo định dạng zip
- **unzip**: giải nén theo định dạng zip
- **scp**: Sao chép file giữa các máy qua SSH

```
h@lab-server:~$ tar -cvf data.tar data/
text2.txt
h@lab-server:~$ ls -a
.bash_history  .bashrc  data      .profile          .ssh           todolist.zip  vision.zip
.bash_logout   .cache    data.tar  shoeshop-eCommerce.zip  .sudo_as_admin_successful  .viminfo       .vscode-ser
h@lab-server:~$ tar -xvf data.tar
text2.txt
```

```
PS D:\> scp vision.zip huepth@192.168.126.99:/home/huepth
huepth@192.168.126.99's password:
vision.zip
```

PHẦN 3: LÀM VIỆC VỚI FILE VÀ THƯ MỤC

8. SAO CHÉP VÀ NÉN FILE

- Giải thích khái niệm stream (`stdin`, `stdout`, `stderr`).
 - **stdin**: Standard Input - Dữ liệu nhập (thường là bàn phím)
 - **stdout**: Standard Output - Dữ liệu xuất ra (thường là màn hình)
 - **stderr**: Standard Error - Dòng lỗi của chương trình

9. TÌM KIẾM FILE

- **find**: Tìm file/ thư mục trong hệ thống.
 - Tìm file theo tên:
 - Tìm theo phần mở rộng
 - Tìm theo kích thước

```
find /home -name "file.txt"
```

```
find / -type f -name "*.log"
```

```
find /var -size +100M
```

PHẦN 3: LÀM VIỆC VỚI FILE VÀ THƯ MỤC

9. TÌM KIẾM FILE

- **locate**: Tìm file nhanh bằng cơ sở dữ liệu (nhanh hơn find nhưng cần cập nhật CSDL định kỳ)
- **grep**: tìm chuỗi trong file
- **Kết hợp grep với pipe (|)**: Lọc kết quả từ lệnh khác: **ps aux | grep python** => hiển thị các tiến trình có chứa từ "python"

```
updatedb
```

```
locate file.txt
```

```
grep "xin chao file.txt"
```

PHẦN 4: QUYỀN TRUY CẬP VÀ NGƯỜI DÙNG

10. NGƯỜI DÙNG VÀ NHÓM (USER & GROUP)

- **whoami**: kiểm tra xem đang đăng nhập với user nào
- **id**: hiển thị UID (User ID), GID (Group ID) và các nhóm (groups) mà người dùng thuộc về.
- **adduser**: tạo user mới và tạo ra các thông tin thư mục liên quan tới user đó
- **deluser**: xóa user
- **su**: (**switch user**) cho phép bạn chuyển sang một người dùng khác trong cùng hệ thống.
- **sudo**: (**superuser do**) cho phép user thực thi lệnh với quyền root tạm thời và chỉ những user thuộc nhóm sudoers mới có quyền này. => giúp an toàn hơn thay vì đăng nhập trực tiếp root.
- **/etc/passwd**: file chứa danh sách tất cả user của hệ thống.

```
huepth@lab-server:~$ whoami
huepth
huepth@lab-server:~$ id huepth
uid=1000(huepth) gid=1000(huepth) groups=1000(huepth),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),117(lxd)
huepth@lab-server:~$ |
```

su [username]: chỉ chuyển user, giữ nguyên môi trường hiện tại.

su - [username]: chuyển user và tải luôn môi trường, biến môi trường của user đó.

PHẦN 4: QUYỀN TRUY CẬP VÀ NGƯỜI DÙNG

11. PHÂN QUYỀN FILE

- Phân quyền trong Linux:
 - User (u): chủ sở hữu của file/ thư mục
 - Group (g): nhóm người dùng có quyền chung
 - Other (o): những người dùng khác còn lại trong hệ thống
- ls -l, quyền đọc/ghi/thực thi (rwx).
 - ls -l: xem danh sách các file/ thư mục kèm quyền truy cập, chủ sở hữu, nhóm và các thông tin khác.
 - rwx : đọc/ ghi/ thực thi: r = 4, w = 2, x = 1 => Người dùng full quyền: rwx = 7
- chmod, chown, chgrp.
 - chmod: thay đổi quyền truy cập file/ thư mục
 - chown: thay đổi chủ sở hữu của file/ thư mục
 - chgrp: thay đổi nhóm sở hữu của file/ thư mục

- Numeric: Mỗi quyền được biểu diễn bằng tổng số r = 4, w = 2, x = 1

```
chmod 750 file.txt => rwxr-x---
```

7->user: đọc, ghi, thực thi
5->group: đọc, thực thi
0->other: không có quyền

- Symbolic: biểu diễn bằng ký tự và toán tử

- + : thêm quyền
- - : bỏ quyền
- = : gán lại quyền cụ thể

```
chmod u+x file.txt - thêm quyền thực thi cho user  
chmod g-w file.txt - bỏ quyền ghi của group  
chmod o=r file.txt - gán cho other chỉ có quyền đọc  
chmod ug=rw,o=r file.txt - user & group có rw, other không có quyền  
chmod u=rwx,g=rx,o=r file.txt - thiết lập quyền chi tiết cho từng nhóm
```

PHẦN 4: QUYỀN TRUY CẬP VÀ NGƯỜI DÙNG

12. QUYỀN ROOT VÀ AN TOÀN

- **Tại sao không nên chạy mọi thứ bằng root.**
 - Tài khoản root (còn gọi là superuser) là tài khoản người dùng có đặc quyền cao nhất trên hệ thống. Nó được thiết kế để thực hiện các tác vụ quản trị hệ thống quan trọng
 - Không nên chạy mọi thứ bằng root:
 - Rủi ro bảo mật cao:
 - Vi phạm nguyên tắc “nguyên tắc đặc quyền tối thiểu” (Principle of Least Privilege)
 - Lỗi người dùng có thể gây hại lớn:
 - Giảm khả năng kiểm soát và phân quyền:
- **Phân biệt sudo và su.**
 - Sudo là một cơ chế quản lý quyền, phụ thuộc vào /etc/sudoers, để cho phép người dùng thực hiện một số lệnh quản trị với quyền root hoặc quyền của người dùng khác mà không cần đăng nhập trực tiếp vào tài khoản đó. Khi dùng sudo, người dùng nhập mật khẩu của chính mình.
 - su: (viết tắt của substitute user hoặc switch user) được dùng để chuyển đổi sang tài khoản khác (thường là root) bằng cách nhập mật khẩu của tài khoản đích. Nó mở một phiên shell mới với quyền của người dùng đó cho đến khi thoát ra (exit).

PHẦN 5: QUẢN LÝ TIẾN TRÌNH VÀ HỆ THỐNG

13. QUẢN LÝ TIẾN TRÌNH (PROCESS)

- **ps**: liệt kê các dòng lệnh đang chạy.
- **top**: theo dõi tiến trình đang chạy theo thời gian thực, sắp xếp theo mức sử dụng CPU. Cung cấp cái nhìn động về tải hệ thống.
- **htop**: phiên bản cải tiến hơn của top hỗ trợ thao tác bằng chuột và hiển thị đồ họa dễ nhìn.
- **kill**: Gửi tín hiệu tới một tiến trình dựa trên PID (Process ID)
- **killall**: Gửi tín hiệu tới các tiến trình dựa trên tên tiến trình (process name).

```
huepth@lab-server:~$ ps
  PID TTY      TIME CMD
 2455 pts/0    00:00:00 bash
 9735 pts/0    00:00:00 ps

huepth@lab-server:~$ top
top - 19:03:24 up 4:43, 2 users, load average: 0.05, 0.10, 0.09
Tasks: 226 total, 1 running, 225 sleeping, 0 stopped, 0 zombie
%Cpu(s): 1.7 us, 6.7 sy, 0.0 ni, 88.3 id, 0.0 wa, 0.0 hi, 3.3 si, 0.0 st
MiB Mem : 1939.7 total, 121.6 free, 499.3 used, 1318.9 buff/cache
MiB Swap: 2048.0 total, 2048.0 free, 0.0 used. 1247.6 avail Mem

          PID USER      PR  NI    VIRT    RES   SHR S %CPU %MEM     TIME+ COMMAND
 2454 huepth    20   0  13972  6024  4548 S  3.3  0.3  0:01.28 sshd
 9740 huepth    20   0  9396  4048  3232 R  3.3  0.2  0:00.13 top
    10 root     20   0      0      0      0 I  1.6  0.0  0:12.82 rcu_sched
 2867 root     20   0      0      0      0 I  1.6  0.0  3:33.78 kworker/0:+
 9638 root     20   0      0      0      0 I  1.6  0.0  0:00.35 kworker/u2+
    1 root     20   0 104124 13140  8544 S  0.0  0.7  0:23.56 systemd
    2 root     20   0      0      0      0 S  0.0  0.0  0:00.05 kthreadd
    3 root     0 -20      0      0      0 I  0.0  0.0  0:00.00 rcu_gp
    4 root     0 -20      0      0      0 I  0.0  0.0  0:00.00 rcu_par_gp
    6 root     0 -20      0      0      0 I  0.0  0.0  0:00.00 kworker/0:+
    8 root     0 -20      0      0      0 I  0.0  0.0  0:00.00 mm_percpu_+
    9 root     20   0      0      0      0 S  0.0  0.0  0:02.25 ksoftirqd/0
   11 root     rt  0      0      0      0 S  0.0  0.0  0:01.06 migration/0
   12 root    -51  0      0      0      0 S  0.0  0.0  0:00.00 idle_inject+
   14 root     20   0      0      0      0 S  0.0  0.0  0:00.00 cpuhp/0
   15 root     20   0      0      0      0 S  0.0  0.0  0:00.00 kdevtmpfs
   16 root     0 -20      0      0      0 I  0.0  0.0  0:00.00 netns
```

PHẦN 5: QUẢN LÝ TIẾN TRÌNH VÀ HỆ THỐNG

13. QUẢN LÝ TIẾN TRÌNH (PROCESS)

- **Foreground và background process (&, fg, bg).**

- **Foreground:** Tiến trình chạy trực tiếp, chiếm quyền điều khiển terminal, phải chờ lệnh hoàn thành hoặc nhấn Ctrl+C để dừng.
- **Background :** Tiến trình chạy ngầm, không chiếm quyền điều khiển terminal, cho phép tiếp tục thực hiện các lệnh khác.
 - **Dấu & –** Chạy tiến trình ở chế độ nền (background)
 - **fg –** Chuyển một tiến trình đang chạy ngầm về chế độ foreground để tương tác hoặc dừng nó bằng Ctrl + C.
 - **bg –** Tiếp tục chạy tiến trình ở chế độ nền

```
huepth@lab-server:~$ ping google.com
PING google.com (142.250.66.46) 56(84) bytes of data.
64 bytes from nchkbg-ag-in-f14.1e100.net (142.250.66.46): icmp_seq=1 ttl=128 time=32.3 ms
64 bytes from nchkbg-ag-in-f14.1e100.net (142.250.66.46): icmp_seq=2 ttl=128 time=25.6 ms
64 bytes from nchkbg-ag-in-f14.1e100.net (142.250.66.46): icmp_seq=3 ttl=128 time=26.8 ms
^C
--- google.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2012ms
rtt min/avg/max/mdev = 25.611/28.249/32.299/2.907 ms
huepth@lab-server:~$ ping google.com &
[1] 10051
huepth@lab-server:~$ PING google.com (142.250.66.46) 56(84) bytes of data.
64 bytes from hkg12s26-in-f14.1e100.net (142.250.66.46): icmp_seq=1 ttl=128 time=25.8 ms
64 bytes from hkg12s26-in-f14.1e100.net (142.250.66.46): icmp_seq=2 ttl=128 time=26.1 ms
64 bytes from hkg12s26-in-f14.1e100.net (142.250.66.46): icmp_seq=3 ttl=128 time=25.5 ms
64 bytes from hkg12s26-in-f14.1e100.net (142.250.66.46): icmp_seq=4 ttl=128 time=27.2 ms
64 bytes from hkg12s26-in-f14.1e100.net (142.250.66.46): icmp_seq=5 ttl=128 time=27.0 ms
64 bytes from hkg12s26-in-f14.1e100.net (142.250.66.46): icmp_seq=6 ttl=128 time=26.3 ms
|
```

PHẦN 5: QUẢN LÝ TIẾN TRÌNH VÀ HỆ THỐNG

14. KIỂM TRA TÀI NGUYÊN HỆ THỐNG

- **df -h:** hiển thị dung lượng đĩa còn trống (-h: hiển thị dung lượng ở dạng dễ đọc (GB, MB).)
- **free -m:** hiển thị thông tin RAM
- **du:** kiểm tra dung lượng thư mục hoặc file đang chiếm dụng du -sh /home/huepth
- **uname:** xem thông tin hệ điều hành và nhân
- **lscpu:** xem thông tin CPU chi tiết.
- **lsblk:** hiển thị danh sách ổ đĩa và phân vùng

```
huepth@lab-server:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            922M   0    922M  0% /dev
tmpfs           194M  1.5M  193M  1% /run
/dev/mapper/ubuntu--vg-ubuntu--lv 18G  14G  3.6G  79% /
tmpfs           970M   0    970M  0% /dev/shm
tmpfs           5.0M   0    5.0M  0% /run/lock
tmpfs           970M   0    970M  0% /sys/fs/cgroup
/dev/loop2       331M  331M  0    100% /snap/code/211
/dev/loop0       92M   92M  0    100% /snap/1xd/32662
/dev/loop1       92M   92M  0    100% /snap/1xd/24061
/dev/loop4       159M  159M  0    100% /snap/code-server/71
/dev/loop3       67M   67M  0    100% /snap/core24/1225
/dev/loop5       64M   64M  0    100% /snap/core20/2669
/dev/loop6       51M   51M  0    100% /snap/snapd/25202
/dev/loop7       64M   64M  0    100% /snap/core20/1828
/dev/loop8       51M   51M  0    100% /snap/snapd/25577
/dev/sda2        1.8G  114M  1.5G  7% /boot
tmpfs           194M   0    194M  0% /run/user/1000

huepth@lab-server:~$ free -m
              total        used       free     shared  buff/cache   available
Mem:       1939         484        447          1      1007       1273
Swap:      2047          0          0          0          0          0

huepth@lab-server:~$ uname -a
Linux lab-server 5.4.0-216-generic #236-Ubuntu SMP Fri Apr 11 19:53:21 UTC 2025 x86_64 x86_64
GNU/Linux
huepth@lab-server:~$
```

```
Vulnerability Spectre v1:
Vulnerability Spectre v2:
Vulnerability Srbds:
Vulnerability Tsx async abort:
Flags:
omp
Mitigation: usercopy/swapgs barriers and __user pointer sanitization
Mitigation: IBRS; IBPB conditional; STIBP disabled; RSB filling; PBRSB-eIBRS Not affected; BHI SW loop, KVM SW loop
Not affected
Not affected
fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pa t pse36 clflush mmx fxsr sse sse2 ss syscall nx pdpe1gb rdtscl m constant_tsc arch_perform rep_good nogl xtopology tsc_reliable nonstop_tsc cpuid pri pcimulqdq ssse3 fma cx16 pcd sse4_1 sse4_2 x2apic movbe popcnt tsc_deadline_timer aes xsave avx f16c rdr and hypervisor lshf_lm abm 3dnowprefetch invocd_single ssbd lbr s ibpb stibp fsgsbase tsc_adjust bmii smep bmi2 erms invpcid rdseed adx snap clflushopt club sha_ni xsaveopt xsavec xgetbv1 xsaves arat umip gfnr vaes vpcinulqdq rdpid movdir movdir64b m d_clear flush_lid arch_capabilities

huepth@lab-server:~$ lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
fd0      2:0    1  1.4M  0 disk
loop0     7:0    0 91.9M  1 loop /snap/1xd/32662
loop1     7:1    0 91.9M  1 loop /snap/1xd/24061
loop2     7:2    0 330.2M  1 loop /snap/code/211
loop3     7:3    0  66.9M  1 loop /snap/core24/1225
loop4     7:4    0 158.4M  1 loop /snap/code-server/71
loop5     7:5    0  63.8M  1 loop /snap/core20/2669
loop6     7:6    0  50.8M  1 loop /snap/snapd/25202
loop7     7:7    0  63.3M  1 loop /snap/core20/1828
loop8     7:8    0  50.9M  1 loop /snap/snapd/25577
sda      8:0    0   20G  0 disk
└─sda1    8:1    0   1M  0 part
└─sda2    8:2    0  1.8G  0 part /boot
└─sda3    8:3    0 18.2G  0 part
└─ubuntu--vg-ubuntu--lv 253:0  0 18.2G  0 lvm /
sr0      11:0   1 99.5M  0 rom
sr1      11:1   1  1.4G  0 rom

huepth@lab-server:~$ _
```

PHẦN 5: QUẢN LÝ TIẾN TRÌNH VÀ HỆ THỐNG

15. DỊCH VỤ VÀ TIẾN TRÌNH NỀN (DAEMON)

- **systemctl**: Quản lý dịch vụ trong hệ thống dùng systemd
- **journalctl**: Dùng để xem log của systemd và các dịch vụ (daemon).
- **service**: Lệnh cũ, tương thích hệ thống dùng SysVinit
- **Start/stop/restart dịch vụ.**
 - Start – Khởi động dịch vụ
 - Stop – Dừng dịch vụ đang chạy.
 - Restart – Khởi động lại dịch vụ dùng khi thay đổi cấu hình hoặc muốn làm mới dịch vụ.

```
service <service> start  
service <service> stop  
service <service> restart  
service <service> status
```

```
systemctl start <service> - Khởi động dịch vụ  
systemctl stop <service> - Dừng dịch vụ  
systemctl restart <service> - Khởi động lại dịch vụ  
systemctl status <service> - Kiểm tra trạng thái dịch vụ  
systemctl enable <service> - Tự động khởi động khi boot  
systemctl disable <service> - Ngừng tự động khởi động
```

```
journalctl - Xem toàn bộ log hệ thống  
journalctl -u nginx - Xem log của dịch vụ nginx  
journalctl -u ssh --since today - Xem log ssh từ hôm nay  
journalctl -xe - Xem log lỗi chi tiết nhất
```

PHẦN 6: QUẢN LÝ GÓI PHẦN MỀM

16. TRÌNH QUẢN LÝ GÓI (PACKAGE MANAGER)

- **apt** là công cụ quản lý gói được dùng trong hệ điều hành Debian và các bản phân phối dựa trên nó như Ubuntu.
 - => Cho phép người dùng cài đặt, cập nhật, gỡ bỏ và quản lý các phần mềm từ kho lưu trữ (repository) của hệ thống.
- **apt-get**: công cụ cũ nhưng mạnh mẽ hơn apt, thường được dùng trong script tự động.
- **yum** (Yellowdog Updater Modified) là công cụ quản lý gói trên hệ điều hành họ Red Hat.
- **dnf** là phiên bản kế nhiệm của yum, hỗ trợ xử lý phụ thuộc nhanh hơn và có hiệu suất tốt hơn.

```
sudo apt install nginx
sudo apt update
sudo apt upgrade
sudo apt-get install apache2
sudo apt-get remove apache2
```

17. CÀI ĐẶT VÀ GỠ BỎ PHẦN MỀM

- **sudo apt install:** tải và cài đặt phần mềm từ kho chính thức. sudo: cho phép chạy lệnh với quyền quản trị (root).
- **sudo apt remove:** Xóa gói phần mềm khỏi hệ thống nhưng vẫn giữ lại file cấu hình (để cài lại nhanh sau này).
- **apt update:** Lấy về danh sách mới nhất của các phần mềm từ repository. Không cài đặt hay nâng cấp phần mềm – chỉ cập nhật thông tin.
- **apt upgrade:** Cập nhật tất cả gói phần mềm hiện có lên phiên bản mới nhất (theo danh sách từ apt update). Dùng sau khi chạy apt update.
- Kiểm tra package đang cài đặt: dpkg -l.
 - **dpkg -l:** liệt kê tất cả các gói đã được cài đặt.
 - Tìm 1 gói cụ thể dùng grep để lọc theo tên phần mềm: dpkg -l | grep nginx

PHẦN 6: QUẢN LÝ GÓI PHẦN MỀM

18. TẠO VÀ SỬ DỤNG ALIAS

- **Tạo lệnh tắt trong file ~/.bashrc.**

- alias ll='ls -alF'.
 - alias giúp tạo lệnh rút gọn thay cho các lệnh dài, giúp tiết kiệm thời gian và tránh gõ sai.
 - Alias chỉ tồn tại trong phiên làm việc hiện tại. Nếu muốn lưu vĩnh viễn, thêm vào file ~/.bashrc hoặc ~/.bash_aliases.

Thay vì gõ: ls -alF --color=auto

Chuyển thành: alias ll='ls -alF --color=auto'

=> Bây giờ chỉ cần gõ ll là hệ thống sẽ tự gọi và thực thi lệnh gốc

PHẦN 7: LÀM VIỆC VỚI MẠNG (NETWORKING)

19. LỆNH KIỂM TRA MẠNG

- **ping**: kiểm tra xem máy có thể kết nối tới internet hay không ping 192.168.126.100
- **ifconfig**: dùng để hiển thị các thông tin của máy tính đang muốn kiểm tra như Host, địa chỉ IP và DNS.
- **ip addr**: thay thế ifconfig xem cấu hình địa chỉ IP
- **netstat**: liệt kê các kết nối mạng, cổng đang mở, và các dịch vụ đang lắng nghe
- **netstat -tlpun**:
 - t: Hiển thị các kết nối TCP.
 - u: Hiển thị các kết nối UDP.
 - l: Hiển thị các cổng đang lắng nghe (listening).
 - p: Hiển thị ID chương trình (PID) và tên chương trình đang sử dụng cổng đó.
 - n: Hiển thị địa chỉ IP và số cổng theo định dạng số, thay vì cố gắng phân giải chúng thành tên máy chủ và tên dịch vụ, giúp tăng tốc độ hiển thị.
- **ss** công cụ mới nhanh hơn netstat dùng để kiểm tra các kết nối mạng và cổng
- **curl**: gửi yêu cầu đến một địa chỉ url kiểm tra phản hồi của mảng.
- **wget**: tải file hoặc nội dung phản hồi của website

PHẦN 7: LÀM VIỆC VỚI MẠNG (NETWORKING)

20. KẾT NỐI SSH VÀ TRUYỀN FILE

- **ssh user@ip:** kết nối từ xa tới máy chủ Linux thông qua giao thức SSH.
 - VD: ssh huepth@192.168.126.99
- **scp:** sao chép file qua SSH
 - VD: scp text.txt huepth@192.168.126.99:/home/huepth
- **rsync:** Dùng để đồng bộ dữ liệu giữa hai máy, chỉ sao chép phần thay đổi (nhanh hơn scp)

```
rsync -avz [thư_mục_nguồn]/ [user]@[IP]:[thư_mục_đích]/
```

VD: rsync -avz /home/huepth/Documents/ huepth@192.168.126.99:/home/huepth/backup/
-a: giữ nguyên quyền và cấu trúc thư mục
-v: hiển thị chi tiết tiến trình
-z: nén dữ liệu khi truyền

20. KẾT NỐI SSH VÀ TRUYỀN FILE

- **Thiết lập SSH key (`ssh-keygen`, `ssh-copy-id`).**
 - Mục đích: đăng nhập không cần mật khẩu bằng cặp khóa công khai - riêng tư.
 - Tạo SSH key trên máy cục bộ: `ssh-keygen` => tạo khóa trong thư mục `~/.ssh/id_rsa` và `id_rsa.pub`
 - Sao chép khóa công khai lên máy chủ `ssh-copy-id huepth@192.168.126.99`
 - Đăng nhập không cần mật khẩu

PHẦN 7: LÀM VIỆC VỚI MẠNG (NETWORKING)

21. KIỂM TRA CỔNG VÀ FIREWALL

- - **ufw, iptables, ss -tuln.**
 - ufw (Uncomplicated Firewall): Công cụ quản lý tường lửa đơn giản trên Ubuntu/Debian.
 - sudo ufw status: kiểm tra trạng thái tường lửa
 - sudo ufw allow 22: mở cổng dịch vụ
 - sudo ufw deny 21: chặn cổng cụ thể
 - sudo ufw enable: kích hoạt tường lửa
 - sudo ufw disable: tắt tường lửa
 - iptables: Công cụ mạnh mẽ hơn để quản lý firewall ở mức kernel (thường dùng trên các bản CentOS/RedHat).
 - ss -tuln:
 - t: Hiển thị các kết nối TCP.
 - u: Hiển thị các kết nối UDP.
 - l: Hiển thị các cổng đang lắng nghe (listening).
 - -n: Hiển thị địa chỉ IP và số cổng theo định dạng số, thay vì cố gắng phân giải chúng thành tên máy chủ và tên dịch vụ, giúp tăng tốc độ hiển thị.

22. SHELL SCRIPT LÀ GÌ

- **Shell Script** là tập hợp các lệnh Linux được viết trong một file văn bản (thường có phần mở rộng .sh), giúp tự động hóa các công việc lặp đi lặp lại như sao lưu dữ liệu, kiểm tra hệ thống, cài đặt phần mềm,...
 - File .sh:
 - Dòng khai báo trình thông dịch (Shebang). Dòng đầu tiên luôn bắt đầu bằng `#!/bin/bash` → Cho hệ điều hành biết rằng script này sẽ được chạy bằng Bash shell.
 - Phần nội dung (Body) bao gồm các lệnh, biến, vòng lặp, điều kiện,... mà bạn muốn thực thi.
 - Phần kết thúc (Exit code – tùy chọn)
 - Dòng exit 0 thường được dùng để kết thúc script với trạng thái thành công.
 - 0 nghĩa là không lỗi, còn các số khác (1, 2, 127, ...) báo lỗi
 - Cách chạy: `bash script.sh` hoặc `chmod +x script.sh`.
 - **bash script.sh**: Chạy script bằng trình thông dịch Bash, dù file chưa có quyền thực thi.
 - **chmod +x script.sh**: Cấp quyền thực thi, chạy trực tiếp file như một chương trình;

23. BIẾN, VÒNG LẶP VÀ ĐIỀU KIỆN

- Biến là cách để lưu trữ dữ liệu tạm thời và tái sử dụng trong chương trình
- **Biến hệ thống:** được shell tạo sẵn.

\$HOME - Thư mục chính của user

\$USER - Tên người dùng hiện tại

\$PATH - Danh sách thư mục chứa lệnh hệ thống

\$BASH - Đường dẫn đến shell hiện tại (ví dụ: /bin/bash)

- **Biến người dùng:** tự định nghĩa

```
name="Hue"  
echo "Xin chào, $name!"
```

PHẦN 8: SCRIPT & AUTOMATION CƠ BẢN

23. BIẾN, VÒNG LẶP VÀ ĐIỀU KIỆN

- Câu lệnh điều kiện
 - if-then-else-fi để kiểm tra và thực thi tùy thuộc điều kiện

```
if [ điều_kiện ]; then
    lệnh_nếu_đúng
else
    lệnh_nếu_sai
fi
```

- Vòng lặp giúp thực thi một nhóm lệnh nhiều lần.

- for:

```
for i in {1..5}
do
    echo "Số thứ tự: $i"
done
```

- while: chạy cho đến khi sai

```
count=1
while [ $count -le 5 ]
do
    echo "While loop: $count"
    count=$((count+1))
done
```

24. TỰ ĐỘNG HÓA TÁC VỤ

- **cron** là một dịch vụ nền (daemon) trong Linux dùng để tự động chạy các tác vụ (scripts, lệnh, chương trình) theo thời gian định kỳ – ví dụ như sao lưu dữ liệu hàng ngày, gửi email tự động, xóa log cũ, cập nhật hệ thống,... Dịch vụ này hoạt động thông qua bảng lịch crontab (cron table).
- **crontab (cron table)**: là file cấu hình lịch chạy các lệnh tự động. Mỗi người dùng (kể cả root) đều có một file crontab riêng.

*	*	*	*	*	command to be executed
-	-	-	-	-	
				-----	day of week (0 - 6) (Sunday=0)
			-----	-----	month (1 - 12)
		-----	-----	-----	day of month (1 - 31)
	-----	-----	-----	-----	hour (0 - 23)
-----	-----	-----	-----	-----	min (0 - 59)

24. TỰ ĐỘNG HÓA TÁC VỤ

- **crontab -e**: Mở file crontab để chỉnh sửa lịch chạy
- Lên lịch chạy script.
 - File script /usr/local/bin/backup_logs.sh với nội dung:

```
#!/bin/bash
tar -czf /backup/logs/system_logs_$(date +%F).tar.gz /var/log
```

- Cấp quyền chạy cho script: **chmod +x /usr/local/bin/backup_logs.sh**
- Mở crontab: **crontab -e**
- Thêm dòng **0 3 * * * /usr/local/bin/backup_logs.sh** => chạy script sao lưu log tự động vào 3h sáng mỗi ngày.

PHẦN 9: THỰC HÀNH TỔNG HỢP

25. BÀI TẬP THỰC TẾ

- Tạo user mới, cấp quyền hạn chế.

```
root@lab-server:~# adduser huepth2
adduser: The user 'huepth2' already exists.
root@lab-server:~# deluser huepth2
Removing user 'huepth2' ...
Warning: group 'huepth2' has no more members.
Done.
root@lab-server:~# adduser huepth2
Adding user 'huepth2' ...
Adding new group 'huepth2' (1002) ...
Adding new user 'huepth2' (1002) with group 'huepth2' ...
The home directory '/home/huepth2' already exists. Not copying from '/etc/skel'.
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for huepth2
Enter the new value, or press ENTER for the default
      Full Name []:
      Room Number []:
      Work Phone []:
      Home Phone []:
      Other []:
Is the information correct? [y/n] y
```

PHẦN 9: THỰC HÀNH TỔNG HỢP

25. BÀI TẬP THỰC TẾ

- Tạo user mới, cấp quyền hạn chế.

```
root@lab-server:~# chown -R root:huepth2 datas1/
root@lab-server:~# ls -l
total 20
drwxr-xr-x 2 root root 4096 Oct  9 10:25 datas
drwxr-xr-x 2 root huepth2 4096 Oct 31 14:19 datas1
drwxr-xr-x 4 root root 4096 Oct 13 08:02 nginx-proxy-manager
drwxr-xr-x 6 root root 4096 Oct 14 16:39 npm
drwx----- 5 root root 4096 Oct 29 16:33 snap
root@lab-server:~# ls -l datas1/
total 0
-rw-r--r-- 1 root huepth2 0 Oct 31 14:19 datas1.txt
root@lab-server:~# chmod u=rwx, g =rx,o=- datas1/
chmod: invalid mode: 'u=rwx,'
Try 'chmod --help' for more information.
root@lab-server:~# chmod u=rwx, g=rx,o=- datas1/
chmod: invalid mode: 'u=rwx,'
Try 'chmod --help' for more information.
root@lab-server:~# chmod u=rwx,g=rx,o= datas1/
root@lab-server:~# ls -l
total 20
drwxr-xr-x 2 root root 4096 Oct  9 10:25 datas
drwxr-x--- 2 root huepth2 4096 Oct 31 14:19 datas1
drwxr-xr-x 4 root root 4096 Oct 13 08:02 nginx-proxy-manager
drwxr-xr-x 6 root root 4096 Oct 14 16:39 npm
drwx----- 5 root root 4096 Oct 29 16:33 snap
root@lab-server:~# chmod 750 datas1
root@lab-server:~# ls -l
total 20
drwxr-xr-x 2 root root 4096 Oct  9 10:25 datas
drwxr-x--- 2 root huepth2 4096 Oct 31 14:19 datas1
drwxr-xr-x 4 root root 4096 Oct 13 08:02 nginx-proxy-manager
drwxr-xr-x 6 root root 4096 Oct 14 16:39 npm
drwx----- 5 root root 4096 Oct 29 16:33 snap
root@lab-server:~# |
```

PHẦN 9: THỰC HÀNH TỔNG HỢP

25. BÀI TẬP THỰC TẾ

- Tạo và nén backup thư mục.

```
root@lab-server:~# mkdir data2
root@lab-server:~# touch /data2/text1.txt
touch: cannot touch '/data2/text1.txt': No such file or directory
root@lab-server:~# touch data2/text1.txt
root@lab-server:~# ls -l
total 20
drwxr-xr-x 2 root root 4096 Oct 31 14:39 data2
drwxr-xr-x 2 root root 4096 Oct  9 10:25 datas
drwxr-xr-x 4 root root 4096 Oct 13 08:02 nginx-proxy-manager
drwxr-xr-x 6 root root 4096 Oct 14 16:39 npm
drwx----- 5 root root 4096 Oct 29 16:33 snap
root@lab-server:~# tar -czvf backup_data2_$(date +%F).tar.gz data2
data2/
data2/text1.txt
root@lab-server:~# tar -xzvf backup_data2_$(date +%F).tar.gz
data2/
data2/text1.txt
root@lab-server:~# |
```

PHẦN 9: THỰC HÀNH TỔNG HỢP

25. BÀI TẬP THỰC TẾ

- Viết script tự động sao lưu log hệ thống hàng ngày.

```
#!/bin/bash
# Script: backup_logs.sh
# Mục đích: Tự động sao lưu log hệ thống hàng ngày

BACKUP_DIR="/backup/logs"
LOG_DIR="/var/log"
DATE=$(date +%F)
FILENAME="logs_backup_$DATE.tar.gz"

# Tạo backup
tar -czf $BACKUP_DIR/$FILENAME $LOG_DIR

# In thông báo
echo "Backup hoàn tất: $BACKUP_DIR/$FILENAME"
```

```
root@lab-server:~# mkdir -p /backup/logs
root@lab-server:~# vi /usr/local/bin/backup_logs.sh
root@lab-server:~# chmod +x /usr/local/bin/backup_logs.sh
root@lab-server:~# crontab -
no crontab for root - using an empty one

Select an editor. To change later, run 'select-editor'.
1. /bin/nano      <---- easiest
2. /usr/bin/vim.basic
3. /usr/bin/vim.tiny
4. /bin/ed

Choose 1-4 [1]: 1
crontab: installing new crontab
root@lab-server:~# crontab -l

0 8 * * * /usr/local/bin/backup_logs.sh
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
root@lab-server:~#
```

PHẦN 9: THỰC HÀNH TỔNG HỢP

25. BÀI TẬP THỰC TẾ

- **Dò tìm file lớn nhất trong thư mục home.**

- du -ah /home | sort -rh | head -n 2 : Hiển thị 2 file hoặc thư mục lớn nhất trong /home
- find /home -type f -exec du -h {} + | sort -rh | head -n 5 : chỉ liệt kê 5 file lớn nhất không tính thư mục

```
root@lab-server:~# du -ah /home | sort -rh | head -n 2
754M    /home
467M    /home/vision/.npm
root@lab-server:~# find /home -type f -exec du -h {} + | sort -rh | head -n 5
94M    /home/todolist/.nvm/versions/node/v20.19.5/bin/node
26M    /home/huepth/shoeshop-eCommerce.zip
25M    /home/todolist/.nvm/.cache/bin/node-v20.19.5-linux-x64/node-v20.19.5-linux-
x64.tar.xz
21M    /home/vision/.npm/_cacache/content-v2/sha512/20/16/83b8789285d66a99f73f4f01
eeebe77a93696d3427f702ccf13668c97352947d774479124e2b236f2e600769aae1cc6993495ca035d
caef3fa295efda06b
17M    /home/vision/.npm/_cacache/content-v2/sha512/93/a2/62eca52668ba0adb8d982879
49d752bc993d4e99fe153bd2eeac657433621b64a2e7758f9952f98cfafdf345e9bddc67acfd4949bf1
2af7cb74b16a57634
root@lab-server:~# |
```

PHẦN 9: THỰC HÀNH TỔNG HỢP

25. BÀI TẬP THỰC TẾ

- Cấu hình SSH và kiểm tra kết nối.

```
root@lab-server:~# apt install openssh-server -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
openssh-server is already the newest version (1:8.2p1-4ubuntu0.13).
0 upgraded, 0 newly installed, 0 to remove and 58 not upgraded.
root@lab-server:~# enable ssh
-bash: enable: ssh: not a shell builtin
root@lab-server:~# systemctl enable ssh
Synchronizing state of ssh.service with SysV service script with /lib/systemd/system
md-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable ssh
root@lab-server:~# systemctl start ssh
root@lab-server:~# systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
    Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enab
    Active: active (running) since Fri 2025-10-31 14:08:40 UTC; 1h 2min ago
      Docs: man:sshd(8)
            man:sshd_config(5)
      Main PID: 957 (sshd)
         Tasks: 1 (limit: 2212)
        Memory: 5.5M
       CGroup: /system.slice/ssh.service
               └─957 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups

Oct 31 14:08:33 lab-server systemd[1]: Starting OpenBSD Secure Shell server...
Oct 31 14:08:40 lab-server sshd[957]: Server listening on 0.0.0.0 port 22.
Oct 31 14:08:40 lab-server sshd[957]: Server listening on :: port 22.
Oct 31 14:08:40 lab-server systemd[1]: Started OpenBSD Secure Shell server.
Oct 31 14:10:14 lab-server sshd[2381]: Accepted password for huepth from 192.168.1>
Oct 31 14:10:14 lab-server sshd[2381]: pam_unix(sshd:session): session opened for >
lines 1-17/17 (END)
```

```
C:\Users\Hue>ssh huepth@192.168.126.99
huepth@192.168.126.99's password:
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.4.0-216-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

System information disabled due to load higher than 1.0
```

Q&A: XIN MỜI CÁC REVIEWER



Tìm hiểu về Git



CLOUD

THANK YOU