## Câu 1 : So sánh UEFI so với LEGACY BIOS

UEFI và Legacy BIOS là hai giao diện firmware khác nhau được sử dụng trong các máy tính cá nhân và máy chủ.

* Kích thước: UEFI là một giao diện firmware mới hơn và có kích thước lớn hơn so với Legacy BIOS, điều này cho phép nó chứa thêm tính năng và công cụ quản lý.
* Hỗ trợ hệ điều hành: UEFI có thể hỗ trợ các hệ điều hành mới hơn và có thể chạy nhanh hơn so với Legacy BIOS, điều này giúp cho việc cài đặt hệ điều hành trở nên dễ dàng hơn.
* Bảo mật: UEFI có thể hỗ trợ các chức năng bảo mật cao hơn, chẳng hạn như xác thực chữ ký số và mã hóa dữ liệu, trong khi Legacy BIOS không hỗ trợ điều này.
* Tính năng mở rộng: UEFI cung cấp các tính năng mở rộng hơn, chẳng hạn như hỗ trợ định dạng lớn hơn cho đĩa cứng và khả năng chạy nhiều hệ điều hành cùng lúc, trong khi Legacy BIOS không hỗ trợ điều này.
* Quản lý thiết bị: UEFI cung cấp một giao diện quản lý thiết bị đồ họa, giúp người dùng dễ dàng quản lý và cấu hình các thiết bị cấu hình hệ thống, trong khi Legacy BIOS chỉ cung cấp một giao diện dòng lệnh.
* Tương thích: Một số thiết bị cũ không hỗ trợ UEFI, vì vậy nếu bạn muốn sử dụng chúng với máy tính mới, bạn cần phải sử dụng Legacy BIOS.

**Câu 3 :** BIOS là gì   
BIOS (Basic Input/Output System) là một phần của máy tính và không liên quan trực tiếp đến hệ điều hành Linux. BIOS là một lớp phần mềm nằm trên một chip nằm trên bo mạch chủ (mainboard) của máy tính và thường được cài đặt bởi nhà sản xuất máy tính hoặc mainboard. Nó có vai trò quản lý phần cứng của máy tính và chịu trách nhiệm khởi động hệ thống và truy cập các thiết bị như ổ cứng, bàn phím, màn hình, và nhiều thiết bị khác.

Hệ điều hành Linux hoạt động trên máy tính sau khi BIOS đã hoàn thành quá trình khởi động. Khi máy tính khởi động, BIOS thường kiểm tra các thiết bị phần cứng và sau đó nạp một trình boot loader như GRUB (GRand Unified Bootloader). Boot loader sau đó khởi động hệ điều hành Linux, và Linux tiếp tục quá trình khởi động và quản lý hệ thống từ đó.

Mặc dù BIOS không trực tiếp liên quan đến hệ điều hành Linux, nó vẫn đóng một vai trò quan trọng trong việc hỗ trợ và tương thích phần cứng cho hệ thống Linux, đặc biệt là trong quá trình khởi động ban đầu của máy tính.

**Câu 4 :** UEFI là gì

UEFI (Unified Extensible Firmware Interface) là một giao diện [firmware](https://bkhost.vn/blog/firmware/) có thể mở rộng, được thiết kế để thay thế cho BIOS (Basic Input/Output System) trong các máy tính cá nhân và máy chủ. Nó cung cấp các tính năng mở rộng hơn so với BIOS và có thể dễ dàng hỗ trợ các hệ điều hành mới hơn.  
**Câu 5 : Shell Script Order and Checklist**  
1. Shebang (#!/bin/bash)

2. Comment/ file header

3. Global variables

4. Functions

a. Use local variables

5. Main script contents

6. Exit with an exit status

a. exit <STATUS> at various exit points

**Câu 6**: **acpid là gì**

nó là tên của một dịch vụ (daemon) quản lý sự kiện năng lượng (power management events) trên hệ thống Linux. Dịch vụ này được sử dụng để theo dõi và xử lý các sự kiện liên quan đến năng lượng như khi máy tính được bật, tắt, đóng nắp laptop, hoặc xảy ra các sự kiện khác liên quan đến quản lý năng lượng.

Dịch vụ acpid thường chạy tự động khi hệ thống khởi động và là một phần quan trọng của quản lý năng lượng trên các hệ thống Linux. Nó cho phép hệ thống phản ứng đúng cách với các sự kiện năng lượng và thực hiện các hành động như đặt máy tính vào chế độ ngủ, đóng nắp laptop để tắt màn hình, và nhiều tác vụ khác.

Để kiểm tra trạng thái acpid sử dụng systemctl:

systemctl status acpid

Khởi động acpid:

systemctl start acpid

**Câu 7 : Thư mục trong linux**

1. /bin: Chứa các tệp tin thực thi cơ bản (binaries) cho các lệnh hệ thống.
2. /boot: Chứa các tệp tin và thư mục liên quan đến quá trình khởi động (boot process) của hệ thống, bao gồm kernel và tệp tin cấu hình của bootloader.
3. /etc: Là nơi chứa các tệp tin cấu hình hệ thống, bao gồm tệp tin cấu hình của hệ điều hành và ứng dụng.
4. /home: Thư mục chứa các thư mục cá nhân của người dùng. Mỗi người dùng thường có một thư mục con trong đây để lưu trữ tài liệu và cài đặt cá nhân.
5. /lib và /lib64: Chứa các thư viện cơ bản sử dụng bởi các tệp tin thực thi trong /bin và /sbin. /lib là cho các phiên bản 32-bit, trong khi /lib64 dành cho 64-bit.
6. /media và /mnt: Thư mục dùng để gắn (mount) các thiết bị lưu trữ ngoại vi, như ổ đĩa USB hoặc đĩa CD/DVD.
7. /opt: Thư mục dành cho cài đặt ứng dụng bên ngoài gói phân phối hệ thống.
8. /root: Thư mục home của người dùng root (superuser). Khác với /home, thư mục này chứa dữ liệu của người dùng root.
9. /srv: Thư mục này thường dùng để lưu trữ dữ liệu cho các dịch vụ hệ thống, ví dụ, các trang web (web server) có thể lưu dữ liệu tại đây.
10. /tmp: Chứa các tệp tạm thời (temporary files) được tạo và sử dụng bởi ứng dụng và quy trình hệ thống. Thường xóa đi sau khởi động lại hệ thống.
11. /usr: Chứa các ứng dụng và thư viện cài đặt cho toàn bộ hệ thống. Đây là một phần quan trọng của hệ thống Linux.
12. /var: Chứa các tệp tin biến thiên (volatile) như logs, tệp tin cache, và dữ liệu của các ứng dụng chạy trên hệ thống.

**Câu 8 : Các hệ thống tệp Linux**

EXT4 (Fourth Extended File System): EXT4 là một phiên bản nâng cấp của EXT3 và là một trong những hệ thống tệp phổ biến nhất trên Linux. Nó hỗ trợ dung lượng lớn, quản lý dữ liệu hiệu quả và có khả năng khắc phục lỗi.

EXT3 (Third Extended File System): EXT3 là phiên bản trước của EXT4 và nó cũng được sử dụng rộng rãi. EXT3 hỗ trợ journaling, giúp khôi phục dữ liệu sau mất điện hoặc sự cố hệ thống.

Btrfs (B-tree File System): Btrfs là một hệ thống tệp hiện đại với nhiều tính năng mạnh mẽ. Nó hỗ trợ snapshot, compression, deduplication và nhiều tính năng khác.

XFS (X File System): XFS được phát triển bởi Silicon Graphics (SGI) và được sử dụng phổ biến trong môi trường doanh nghiệp. Nó hỗ trợ quản lý dung lượng lớn và hiệu suất cao.

ZFS (Zettabyte File System): ZFS là một hệ thống tệp mạnh mẽ có nguồn gốc từ Sun Microsystems (nay là Oracle). Nó kết hợp hệ thống quản lý dung lượng và các tính năng bảo mật mạnh mẽ.

ReiserFS: ReiserFS là một hệ thống tệp được thiết kế để cải thiện hiệu suất đọc và ghi dữ liệu nhỏ. Tuy nhiên, nó không còn được phát triển hoặc hỗ trợ rộng rãi trong các bản phân phối Linux gần đây.

JFS (IBM Journaling File System): JFS là một hệ thống tệp phát triển bởi IBM và được thiết kế với mục tiêu tối ưu hóa hiệu suất và khôi phục lỗi.

NILFS (New Implementation of a Log-structured File System): NILFS là một hệ thống tệp log-structured, nghĩa là nó sử dụng journaling để quản lý dữ liệu.  
**Câu 9 :** Phân vùng hệ thống tệp EFI  
Phân vùng hệ thống tệp EFI (EFI System Partition, ESP) là một phân vùng trên ổ đĩa của máy tính được sử dụng trong các hệ thống dựa trên chuẩn UEFI (Unified Extensible Firmware Interface). Chuẩn UEFI thay thế chuẩn BIOS truyền thống và cung cấp một môi trường khởi động mới và linh hoạt hơn cho hệ thống máy tính.

Phân vùng EFI chứa các tệp và thư mục quan trọng cần thiết để khởi động hệ thống và quản lý quá trình khởi động. Các thành phần quan trọng bao gồm:

1. EFI Boot Manager: Một trình quản lý khởi động hoặc bootloader có trên phân vùng EFI. Nó quản lý danh sách các tùy chọn khởi động (boot entries) và cho phép người dùng lựa chọn hệ điều hành hoặc tải các ứng dụng khác.
2. EFI Bootloader Files: Các tệp thực thi bootloader như bootx64.efi cho hệ thống 64-bit. Những tệp này chịu trách nhiệm cho việc khởi động hệ thống và nạp kernel của hệ điều hành.
3. EFI Variables: Các biến EFI được sử dụng để lưu trạng thái và cấu hình của hệ thống, chẳng hạn như Secure Boot state, thứ tự khởi động ưu tiên, và nhiều tùy chọn quan trọng khác.
4. EFI Bootloader Configuration: Các tệp cấu hình cho bootloader như grub.cfg, refind.conf, hoặc loader.conf, tùy thuộc vào bootloader cụ thể được sử dụng.

Phân vùng EFI thường là một phân vùng FAT (File Allocation Table) với kích thước tối thiểu (thường khoảng 100MB) và đôi khi nằm ở đầu ổ đĩa hoặc ổ USB. Nó phải được đánh dấu là phân vùng EFI và được sử dụng bởi hệ thống để khởi động. Khi cài đặt một hệ điều hành mới hoặc quản lý bootloader, phân vùng EFI cần được chọn làm điểm gắn kết (mount point) cho phân vùng EFI.

------------------------------------------------/proc---------------------------------------------

1/proc và /sys**: Là các hệ thống tệp tin ảo để tương tác với kernel và cung cấp thông tin về quá trình và thiết bị hệ thống.**

**2. /proc/self/mounts**

Tệp /proc/self/mounts là một tệp ảo nằm trong hệ thống tệp /proc của Linux. Nó chứa thông tin về các hệ thống tệp đã được gắn kết (mounted) bởi quá trình hiện tại (tức là quá trình ghi tệp /proc/self/mounts đang chạy).

Cụ thể, tệp /proc/self/mounts cung cấp thông tin về các hệ thống tệp đã được gắn kết, bao gồm đường dẫn, loại hệ thống tệp, tùy chọn gắn kết (mount options), và thông tin khác về mỗi hệ thống tệp đã gắn kết.

* Mỗi dòng trong tệp /proc/self/mounts tương ứng với một hệ thống tệp đã được gắn kết.
* Các cột tương ứng với thông tin về hệ thống tệp đã gắn kết, bao gồm đường dẫn gắn kết, loại hệ thống tệp, tùy chọn gắn kết, và thông tin khác.

Lệnh mount và các công cụ quản lý hệ thống tệp thường sử dụng tệp /proc/self/mounts để hiển thị thông tin về các hệ thống tệp đã được gắn kết trong hệ thống và cũng để thực hiện việc gắn kết và unmount các hệ thống tệp.

**3./proc/cmdline**

Danh sách các tham số được truyền từ bootloader đến kernel được lưu trong tệp /proc/cmdline trong hệ thống tệp Linux. Đây là nơi mà kernel lấy các tham số được cung cấp từ bootloader để cấu hình khởi động hệ thống. Tham số này bao gồm các tùy chọn kernel, tùy chọn của hệ thống tệp gốc (root filesystem), và các thông số khác liên quan đến quá trình khởi động.

Bạn có thể đọc nội dung của tệp **/proc/cmdline** bằng lệnh cat hoặc bất kỳ trình soạn thảo văn bản nào

------------------------------------------------/sbin---------------------------------------------

**/sbin** là một thư mục quan trọng trong hệ thống tệp của các hệ điều hành Unix và Linux. Thư mục này chứa các tệp thực thi (executables) cần thiết để khởi động hệ thống và quản lý nó.

Cụ thể, Dưới đây là một số ví dụ về các tệp thực thi quan trọng có thể được tìm thấy trong /sbin:

* /sbin/init: Là tiến trình init, tiến trình đầu tiên được khởi chạy khi hệ thống Linux bắt đầu, và nó quản lý quá trình khởi động của hệ thống.
* /sbin/reboot: Là lệnh để khởi động lại hệ thống.
* /sbin/halt: Là lệnh để tắt máy tính một cách an toàn.
* /sbin/shutdown: Là lệnh để lập lịch hoặc thực hiện việc tắt máy tính hoặc khởi động lại dựa trên một lịch trình cụ thể.
* /sbin/fsck: Là lệnh kiểm tra và sửa chữa hệ thống tệp trong trường hợp hỏng hóc.
* /sbin/mount: Là lệnh để gắn kết (mount) các hệ thống tệp.
* /sbin/umount: Là lệnh để gỡ bỏ (unmount) các hệ thống tệp đã gắn kết.

Các lệnh và tệp thực thi trong /sbin thường được sử dụng trong quá trình khởi động và quản lý hệ thống, và chúng thường đòi hỏi quyền superuser (quyền root) để thực thi. Thư mục /sbin là một phần quan trọng của hệ thống tệp Linux và Unix.

-------------------------------------------------------**/usr**-----------------------------------------

**1. /usr/share/man** Thư mục **/usr/share/man** chứa các tệp manual pages (man pages) của hệ thống, không phải là một tệp duy nhất. Man pages là tài liệu dạng văn bản được sử dụng để cung cấp thông tin về các lệnh, hàm, file formats, conventions, và các khía cạnh khác của hệ thống và các ứng dụng.

Thư mục **/usr/share/man** thường được chia thành các thư mục con dựa trên các "sections" hoặc phần khác nhau của man pages. Các sections thông thường bao gồm:

**1. /usr/lib64/**

Trong nhiều hệ thống Linux 64-bit, thư mục /usr/lib64/ thường được sử dụng để chứa thư viện (libraries) 64-bit liên quan đến ứng dụng và các phần mềm của hệ thống. Thư mục này chứa các thư viện 64-bit được sử dụng bởi các ứng dụng và chương trình trên hệ thống 64-bit.

Thư mục /usr/lib64/ không phải lúc nào cũng tồn tại trên tất cả các phiên bản Linux và phụ thuộc vào cấu hình cụ thể của hệ thống. Nó có thể có tên khác trên các phiên bản Linux khác nhau hoặc có thể không tồn tại nếu các thư viện 64-bit được lưu trữ trong /usr/lib/ hoặc các thư mục khác.

Thường thì, /usr/lib64/ được sử dụng trên các hệ thống Linux 64-bit để phân biệt giữa các thư viện 32-bit và 64-bit, đồng thời hỗ trợ các ứng dụng và phần mềm 64-bit.

-------------------------------------------------------**/lib64**-----------------------------------------

Thư mục /lib64/ thường được sử dụng trong hệ thống Linux 64-bit để chứa thư viện (libraries) 64-bit. Thư viện này cần cho việc thực thi các ứng dụng 64-bit trên hệ thống. Thư mục /lib64/ cung cấp thư viện cho các ứng dụng 64-bit, trong khi thư mục /lib/ thường chứa thư viện 32-bit.

Trong hệ thống Linux, thư mục /lib64/ thường được tạo để hỗ trợ kiến trúc 64-bit và không phải lúc nào cũng tồn tại. Thư mục này có thể có tên khác trên các phiên bản Linux khác nhau hoặc không tồn tại tùy thuộc vào cấu hình cụ thể của hệ thống. Thường, nó được sử dụng trên các phiên bản Linux 64-bit để lưu trữ các thư viện cần cho các ứng dụng 64-bit.

Thư mục /boot/grub/ chứa các tệp và cấu hình liên quan đến GRUB (Grand Unified Boot Loader), một trình khởi động đa hệ thống phổ biến trong các hệ thống Linux và Unix. Dưới đây là một số chi tiết về thư mục /boot/grub/:

1. /boot/grub/grub.cfg: Đây là tệp cấu hình chính của GRUB 2. Nó chứa thông tin về cách GRUB sẽ hiển thị menu khởi động, các tùy chọn khởi động, và cách quản lý các hệ điều hành cài đặt trên hệ thống.
2. /boot/grub/i386-pc/ hoặc /boot/grub/x86\_64-pc/: Đây là các thư mục chứa tệp thực thi của GRUB cho kiến trúc máy tính của bạn (x86\_64-pc cho 64-bit và i386-pc cho 32-bit). Các tệp thực thi này bao gồm các phần của GRUB cần để khởi động hệ thống và tải kernel.
3. **\*\*/boot/grub/themes/** (tùy chọn): Nếu bạn sử dụng một giao diện (theme) cho GRUB, thư mục này chứa các tệp và hình ảnh liên quan đến giao diện đó.
4. **\*\*/boot/grub/fonts/** (tùy chọn): Nếu bạn đã cấu hình một font tùy chỉnh cho GRUB, thư mục này chứa các tệp font đó.
5. **\*\*/boot/grub/locale/ (tùy chọn):** Thư mục này chứa các tệp ngôn ngữ cho GRUB, giúp hiển thị menu và thông báo trong ngôn ngữ cụ thể.

Thư mục /boot/grub/ là một phần quan trọng của quá trình khởi động của hệ thống, và nó được sử dụng để quản lý cách GRUB tải và khởi động các hệ điều hành trên máy tính của bạn. Các tệp và cấu hình trong thư mục này có thể được chỉnh sửa để tùy chỉnh quá trình khởi động của bạn hoặc để thêm tính năng như chọn nền, giao diện người dùng, và ngôn ngữ khởi động.

-----------------------------Yum (Yellowdog Updater, Modified) trong CentOS:---------------------------

Yum là một tiện ích quản lý gói phần mềm được sử dụng trong CentOS, một hệ điều hành Linux nguồn mở, được cộng đồng phát triển dựa trên mã nguồn của Red Hat Enterprise Linux (RHEL). Yum giúp dễ dàng cài đặt, cập nhật và quản lý các gói phần mềm trên CentOS. Dưới đây là một số thông tin cơ bản về Yum trong CentOS:

Cài đặt và Sử dụng Cơ bản: Yum thường được cài sẵn trên các hệ thống CentOS. Để cài đặt hoặc cập nhật các gói phần mềm, bạn có thể sử dụng các lệnh sau:

Để cài đặt một gói:

yum install tên-gói

Để cập nhật các gói đã cài đặt:

yum update

Cấu hình Kho Lưu trữ (Repository): Yum phụ thuộc vào các kho lưu trữ phần mềm để lấy các gói. CentOS đi kèm với các kho lưu trữ chính thức của riêng nó, nhưng bạn có thể thêm các kho lưu trữ của bên thứ ba để có thêm phần mềm. Các tệp cấu hình kho lưu trữ được lưu trong /etc/yum.repos.d/.

Phân Giải Sự Phụ Thuộc Gói: Yum tự động giải quyết sự phụ thuộc gói, đảm bảo rằng các thư viện và thành phần cần được cài đặt khi bạn cài đặt một gói.

Tìm kiếm và Liệt kê Gói Phần Mềm: Bạn có thể tìm kiếm các gói và liệt kê các gói đã cài đặt bằng các lệnh như yum search tên-gói và yum list installed.

Gỡ Bỏ Gói Phần Mềm: Để gỡ bỏ một gói phần mềm, sử dụng lệnh yum remove tên-gói.

Xóa Bộ Nhớ Cache: Yum lưu trữ các dữ liệu về gói phần mềm để tăng tốc các thao tác trong tương lai. Bạn có thể xóa bộ nhớ cache bằng lệnh yum clean all.

Cài Đặt theo Nhóm: Yum hỗ trợ cài đặt các nhóm gói, điều này hữu ích khi bạn muốn cài đặt một bộ gói liên quan cùng nhau. Ví dụ, để cài đặt môi trường desktop, bạn có thể sử dụng yum groupinstall "Desktop".

Cập nhật Bảo Mật: Bạn có thể cấu hình Yum để tự động cài đặt cập nhật bảo mật để bảo vệ hệ thống của bạn.

Các Plugin của Yum: Yum có thể được mở rộng với các plugin để thêm các tính năng bổ sung. Các plugin phổ biến bao gồm yum-utils, cung cấp các công cụ bổ sung như yumdownloader để tải các gói mà không cài đặt chúng.

DNF (Dandified Yum): Trong CentOS 8 và các phiên bản sau đó, Yum đã được thay thế bằng Dnf (Dandified Yum) như trình quản lý gói mặc định. Dnf cung cấp trải nghiệm quản lý gói hiện đại và hiệu quả hơn, trong khi vẫn duy trì tính tương thích với các lệnh Yum. Tuy nhiên, bạn vẫn có thể sử dụng các lệnh Yum, mà Dnf cung cấp dưới dạng alias.

Hãy nhớ rằng trong khi Yum vẫn có sẵn và có thể sử dụng trong CentOS, nếu bạn đang sử dụng CentOS 8 hoặc các phiên bản sau, nên làm quen với Dnf. Dnf giữ lại các lệnh Yum mà bạn quen thuộc và cung cấp hiệu suất và tính năng nâng cao.

-----------------------------rpm (Red Hat Package Manager):---------------------------

Lệnh rpm (Red Hat Package Manager) là một công cụ dòng lệnh được sử dụng trong các hệ điều hành dựa trên RPM (Red Hat Package Manager) như CentOS, Fedora và Red Hat Enterprise Linux để quản lý các gói phần mềm. Dưới đây là một số lệnh phổ biến và tùy chọn của rpm:

Cài đặt gói phần mềm:

rpm -i package.rpm

Tùy chọn:

* -v: Hiển thị thông tin chi tiết về quá trình cài đặt.
* -h: Hiển thị tiêu đề và hiệu chỉnh tiến trình.

Cài đặt gói phần mềm (không cần thông báo):

rpm -ivh package.rpm

Cập nhật gói phần mềm:

rpm -U package.rpm

Tùy chọn:

* -v: Hiển thị thông tin chi tiết về quá trình cập nhật.
* -h: Hiển thị tiêu đề và hiệu chỉnh tiến trình.

Gỡ bỏ gói phần mềm:

rpm -e package-name

Tùy chọn:

* --nodeps: Gỡ bỏ gói mà không kiểm tra phụ thuộc.

Hiển thị thông tin về gói đã cài đặt:

rpm -q package-name

Tùy chọn:

* -i: Hiển thị thông tin chi tiết về gói.
* -l: Liệt kê các tệp tin thuộc gói.
* -p: Hiển thị thông tin từ gói RPM chưa được cài đặt.

Kiểm tra xem gói có được cài đặt hay không:

rpm -q package-name

Liệt kê tất cả các gói đã cài đặt:

rpm -qa

Hiển thị thông tin về gói RPM:

rpm -qi package-name

Liệt kê các tệp thuộc gói RPM:

rpm -ql package-name

Kiểm tra sự phụ thuộc của gói RPM:

rpm -qR package-name

Tải gói RPM từ kho lưu trữ RPM:

rpm -ivh <ftp://example.com/package.rpm>

Lưu ý rằng rpm là công cụ quản lý gói cơ bản và không giải quyết tự động các phụ thuộc, trong khi các trình quản lý gói cao cấp như Yum hoặc Dnf thực hiện điều này một cách tự động.

-----------------------------stream redirection---------------------------

Trong hệ thống Unix và Linux, stream redirection là một cách để điều hướng dữ liệu đầu ra hoặc đầu vào từ và đến các tệp hoặc thiết bị khác. Có một số cách để thực hiện stream redirection. Dưới đây là tất cả các loại stream redirection chính:

1. Standard Output (stdout):

**> :** Dùng để ghi đè dữ liệu đầu ra tiêu chuẩn vào một tệp tin mới hoặc hiện tại hoặc ghi đè dữ liệu một tệp tin hiện có. Ví dụ:

command > output.txt

**>> :** Dùng để nối dữ liệu đầu ra tiêu chuẩn vào một tệp tin mới hoặc hiện tại hoặc nối dữ liệu vào một tệp tin hiện có. Ví dụ:

command >> output.txt

Standard Input (stdin):

**< :** Dùng để đọc dữ liệu đầu vào từ một tệp tin hoặc thiết bị khác. Ví dụ:

command < input.txt

Standard Error (stderr):

**2>**: Dùng để điều hướng dữ liệu đầu ra lỗi (stderr) vào một tệp tin hoặc thiết bị khác. Ví dụ:

command 2> error.txt

**2>>** : Dùng để nối dữ liệu đầu ra lỗi vào một tệp tin hoặc thiết bị khác. Ví dụ:

command 2>> error.txt

Combine Standard Output and Standard Error:

**2>&1**: Dùng để kết hợp stdout và stderr, điều hướng cả hai sang cùng một nơi, thường là màn hình hoặc tệp tin. Ví dụ:

command > output.txt 2>&1

Here Documents:

**<<** : Được sử dụng để cung cấp đầu vào cho một lệnh từ văn bản trong mã shell. Ví dụ:

command << EOF

This is the input text.

More input text.

EOF

1. Null Device:
   * **/dev/null** : Dùng để loại bỏ dữ liệu đầu ra hoặc ngăn việc ghi dữ liệu ra màn hình. Ví dụ:

command > /dev/null

Những kỹ thuật redirection này rất hữu ích trong việc xử lý dữ liệu và quản lý luồng ra vào trong hệ thống Unix và Linux. Chúng cho phép bạn kiểm soát nơi dữ liệu đầu vào và đầu ra được điều hướng, ghi lại, hoặc loại bỏ.

**---------------------------------------------------/var/log------------------------------------------------------------------**

Thư mục /var/log trong hệ thống Linux và Unix là nơi chứa các tập tin nhật ký (log files). Các tập tin này chứa thông tin về hoạt động của hệ thống, ứng dụng và dịch vụ chạy trên hệ thống. Việc theo dõi các tệp nhật ký này có thể rất hữu ích để gỡ lỗi sự cố hệ thống, xem thông tin về tài khoản người dùng và theo dõi hoạt động của ứng dụng.

Dưới đây là một số ví dụ về các tệp nhật ký phổ biến mà bạn có thể tìm thấy trong thư mục /**var/log:**

1. **/var/log/syslog hoặc /var/log/messages:** Chứa thông tin về hoạt động của hệ thống và các thông báo từ hệ thống.
2. **/var/log/auth.log:** Ghi lại các sự kiện liên quan đến xác thực và quyền truy cập người dùng, chẳng hạn như đăng nhập và thay đổi mật khẩu.
3. **/var/log/dmesg:** Là một nhật ký của thông tin từ kernel, chứa thông tin về phần cứng và sự kiện liên quan đến boot.
4. **/var/log/apache2/access.log và /var/log/apache2/error.log:** Các tệp nhật ký của máy chủ web Apache, chứa thông tin về truy cập và lỗi của trang web.
5. **/var/log/mysql/error.log:** Ghi lại các lỗi và thông báo từ máy chủ cơ sở dữ liệu MySQL.
6. **/var/log/mail.log hoặc /var/log/maillog:** Chứa thông tin về hoạt động của máy chủ thư điện tử (email server), chẳng hạn như gửi và nhận thư.
7. **/var/log/auth.log hoặc /var/log/secure:** Tệp nhật ký liên quan đến xác thực và an ninh hệ thống.

-----------------------------VI---------------------------

Vi là một trình soạn thảo dòng lệnh mạnh mẽ nhưng có cú pháp và tùy chọn phức tạp. Dưới đây là một số lệnh và tùy chọn cơ bản trong Vi:

Chế độ Command (Normal Mode):

1. i: Chuyển sang chế độ Insert trước dấu nháy vị trí con trỏ.
2. I: Chuyển sang chế độ Insert ở đầu dòng hiện tại.
3. a: Chuyển sang chế độ Insert sau dấu nháy vị trí con trỏ.
4. A: Chuyển sang chế độ Insert ở cuối dòng hiện tại.
5. o: Thêm dòng mới dưới con trỏ và chuyển sang chế độ Insert.
6. O: Thêm dòng mới trên con trỏ và chuyển sang chế độ Insert.
7. x: Xóa ký tự tại vị trí con trỏ.
8. dd: Xóa dòng hiện tại.
9. dw: Xóa một từ (khoảng trắng và dấu cách làm kết thúc từ).
10. D: Xóa từ vị trí con trỏ tới cuối dòng.
11. yy: Sao chép (copy) dòng hiện tại.
12. yw: Sao chép (copy) một từ (khoảng trắng và dấu cách làm kết thúc từ).
13. p: Dán (paste) văn bản ở sau dấu nháy vị trí con trỏ.
14. P: Dán (paste) văn bản ở trước dấu nháy vị trí con trỏ.
15. u: Hoàn tác thay đổi cuối cùng.
16. Ctrl-R: Làm lại thay đổi sau khi đã hoàn tác.

Chế độ Insert:

1. Esc: Trở lại chế độ Command (Normal Mode).
2. Ctrl-C: Trở lại chế độ Command (Normal Mode).

Lưu và thoát:

1. :w: Lưu tệp.
2. :q: Thoát (nếu không có thay đổi).
3. :wq hoặc :x: Lưu và thoát.
4. :q!: Thoát mà không lưu thay đổi.

Tìm kiếm và thay thế:

1. /pattern: Tìm kiếm mẫu "pattern" xuống (sử dụng "n" để tìm kiếm tiếp theo và "N" để tìm kiếm ngược lại).
2. ?pattern: Tìm kiếm mẫu "pattern" lên (sử dụng "n" để tìm kiếm tiếp theo và "N" để tìm kiếm ngược lại).
3. :s/old/new: Thay thế "old" bằng "new" trên dòng hiện tại.
4. :%s/old/new/g: Thay thế "old" bằng "new" trên toàn bộ tệp.

Đây là một số lệnh và tùy chọn cơ bản trong Vi. Tuy nhiên, Vi có rất nhiều tính năng và tùy chọn khác, và để tận dụng được Vi, bạn cần nắm rõ cú pháp và cách hoạt động của nó. Để tìm hiểu thêm, bạn có thể sử dụng lệnh man vi hoặc vimtutor để mở trình hướng dẫn sử dụng Vi.

------------------------------------------------------------------------------------------------------------------

**lệnh umask :**

Lệnh umask là một lệnh dùng để thiết lập hoặc hiển thị mặt nạ (mask) quyền truy cập mặc định cho các tệp và thư mục mới được tạo trong môi trường Unix và Linux. Mặt nạ quyền (umask) xác định quyền mặc định mà tệp và thư mục mới sẽ có sau khi chúng được tạo.

Mặt nạ quyền (umask) hoạt động theo nguyên tắc "bỏ đi" các quyền mà bạn muốn từ tập quyền đầy đủ (rwx) của tệp và thư mục. Umassk được định nghĩa dưới dạng một số bát phân (octal), và nó đại diện cho các quyền mà bạn muốn bỏ đi. Dưới đây là cách sử dụng lệnh umask:

--------------------------------------------------------------------------------------------------------------------

**lệnh Tune2fs :**

Tune2fs là một lệnh trong các hệ điều hành dựa trên hệ thống tệp EXT2, EXT3 và EXT4 (các hệ thống tệp phổ biến trên Linux) để điều chỉnh và cấu hình các tham số của hệ thống tệp. Dưới đây là một số tùy chọn (options) quan trọng của lệnh tune2fs:

-c count: Đặt số lần kiểm tra tự động để kiểm tra hệ thống tệp. Đối số count là số ngày mà bạn muốn kiểm tra. Ví dụ: tune2fs -c 30 /dev/sdX sẽ kiểm tra hệ thống tệp mỗi 30 ngày.

-i interval: Đặt thời gian kiểm tra tự động dựa trên thời gian. Đối số interval là số ngày. Ví dụ: tune2fs -i 2m /dev/sdX sẽ kiểm tra hệ thống tệp mỗi 2 tháng.

-e errors\_behavior: Đặt cách xử lý lỗi cho hệ thống tệp khi phát hiện lỗi. errors\_behavior có thể là "continue" (tiếp tục hoạt động dù có lỗi), "remount-ro" (remount chỉ đọc khi có lỗi), hoặc "panic" (gây ra sự cố và tạo ra core dump khi có lỗi).

-m reserved\_blocks\_percent: Đặt phần trăm các khối (blocks) dành riêng cho root (người dùng có quyền cao nhất) để sử dụng khi đĩa đầy. Điều này đảm bảo rằng root vẫn có khả năng ghi dữ liệu khi đĩa đã đầy. Ví dụ: tune2fs -m 1 /dev/sdX để đặt 1% khối dành riêng.

-L label: Đặt nhãn (label) cho hệ thống tệp. Điều này có thể giúp bạn xác định hệ thống tệp một cách dễ dàng.

-U user\_options: Đặt các tùy chọn người dùng (user options) cho hệ thống tệp. Các tùy chọn này là tùy chỉnh và phụ thuộc vào người dùng.

-r reserved\_blocks\_count: Đặt số khối (blocks) dành riêng cho root để sử dụng khi đĩa đầy. Tùy chọn này sử dụng số lượng thay vì phần trăm. Ví dụ: tune2fs -r 1000 /dev/sdX để đặt 1000 khối dành riêng.

-S features: Điều chỉnh các tính năng hệ thống tệp. Đây là một tùy chọn nâng cao và thường được sử dụng bởi người quản trị hệ thống có kiến thức sâu về hệ thống tệp.

--------------------------------------------------------------------------------------------------------------------

**lệnh mount --bind :**

Lệnh mount --bind được sử dụng để gắn kết (bind) một thư mục (hoặc thiết bị) vào một vị trí (mount point) khác trong hệ thống tệp Linux. Điều này cho phép bạn tạo một bản sao của một thư mục hoặc thiết bị trong một vị trí khác mà không cần sao chép dữ liệu.

mount --bind nguồn đích

Gắn kết thư mục vào vị trí khác:

mount --bind /path/to/source /path/to/destination

Gắn kết một phân vùng vào một thư mục trong hệ thống tệp:

mount --bind /dev/sdX /mnt/myfolder

Sử dụng trong tệp /etc/fstab:

Bạn cũng có thể sử dụng lệnh mount --bind trong tệp /etc/fstab để tự động gắn kết thư mục hoặc thiết bị vào một vị trí cụ thể khi hệ thống khởi động.

Lệnh mount --bind thường được sử dụng trong các tình huống như tạo các sao chép tạm thời của các thư mục mà bạn muốn sửa đổi mà không ảnh hưởng đến thư mục gốc hoặc khi bạn cần chia sẻ dữ liệu giữa các vị trí khác nhau trong hệ thống tệp.

Để unmount (gỡ bỏ) một thư mục hoặc thiết bị đã được gắn kết trong hệ thống tệp Linux, bạn có thể sử dụng lệnh umount. Dưới đây là cách sử dụng lệnh umount:

umount [đường dẫn-thư-mục-đã-gắn-kết]

Unmount một thư mục đã gắn kết:

umount /path/to/mount-point

Unmount tất cả các thư mục và thiết bị đã gắn kết:

umount -a

----------------------------------------------------------------------------------------------------------------------------------

**lệnh dpkg**

Lệnh **dpkg** là một công cụ quản lý gói cho hệ thống Debian và các phiên bản dựa trên Debian, như Ubuntu. Nó cho phép bạn cài đặt, cập nhật, xóa và quản lý các gói phần mềm trên hệ thống của bạn. Dưới đây là một số tùy chọn (options) quan trọng của lệnh dpkg:

Cài đặt gói: Để cài đặt một gói .deb, bạn có thể sử dụng tùy chọn -i (hoặc –install):

dpkg -i package.deb

Gỡ bỏ gói: Để gỡ bỏ một gói cài đặt, bạn có thể sử dụng tùy chọn -r (hoặc –remove):

dpkg -r package-name

Gỡ bỏ gói và cấu hình cùng lúc: Sử dụng tùy chọn --purge để gỡ bỏ một gói và cả cấu hình liên quan của nó:

dpkg --purge package-name

Kiểm tra tình trạng gói: Để kiểm tra tình trạng cài đặt của một gói, bạn có thể sử dụng tùy chọn -l (hoặc –list):

dpkg -l package-name

Liệt kê tất cả gói đã cài đặt: Sử dụng tùy chọn --get-selections để liệt kê tất cả các gói đã cài đặt:

dpkg –get-selections

Cập nhật danh sách các gói đã cài đặt: Để cập nhật danh sách các gói đã cài đặt, sử dụng tùy chọn –update-avail:

dpkg –update-avail

Xem thông tin về gói: Để xem thông tin chi tiết về gói đã cài đặt, sử dụng tùy chọn -s (hoặc –status):

dpkg -s package-name

Kiểm tra sự phụ thuộc gói: Để kiểm tra sự phụ thuộc (dependencies) của một gói, sử dụng tùy chọn –info:

dpkg --info package.deb

---------------------------------------------------------------------------------------------------------------------

**lệnh ln:**

Lệnh ln trong Unix và Linux được sử dụng để tạo liên kết (links) giữa các tệp hoặc thư mục. Dưới đây là một số tùy chọn (options) quan trọng của lệnh ln:

-s (Symbolic Link): Tạo liên kết tượng trưng (symbolic link) thay vì liên kết cứng (hard link). Liên kết tượng trưng là một tệp mới có đường dẫn riêng, trỏ đến tệp hoặc thư mục gốc.

-b (Backup Existing Files): Tạo một bản sao lưu của tệp tiềm năng sẽ bị ghi đè khi tạo liên kết mới.

-i (Interactive): Yêu cầu xác nhận từ người dùng nếu liên kết đã tồn tại hoặc sẽ bị ghi đè.

-f (Force): Ghi đè tệp liên kết nếu nó đã tồn tại mà không cần xác nhận từ người dùng

-n (Do Not Dereference): Sử dụng chế độ không giải quyết đường dẫn (do not dereference) để tạo liên kết đến tệp hoặc thư mục gốc, thay vì tạo liên kết đến mục tiêu của đường dẫn.

Nếu bạn chỉ muốn sử dụng lệnh ln mà không có tùy chọn (options), thì việc tạo liên kết cơ bản sẽ tự động tạo liên kết cứng (hard link). Dưới đây là cách cơ bản để sử dụng lệnh ln để tạo liên kết cứng

---------------------------------------------------------------------------------------------------------------------

**lệnh dmesg**

Lệnh **dmesg** trong hệ thống Linux được sử dụng để xem và kiểm tra các thông báo hạt nhân (kernel messages). Nó thường hiển thị các thông tin liên quan đến phần cứng, kernel, và các sự kiện hệ thống. Dưới đây là một số sử dụng phổ biến của lệnh dmesg:

**-c hoặc --clear:** Xóa bộ nhớ đệm của dmesg. Điều này sẽ làm cho lệnh dmesg chỉ hiển thị các thông báo mới nhất sau khi lệnh này được thực hiện.

**dmesg -c**

-n level hoặc --console-level level: Hiển thị thông báo hạt nhân với mức độ (log level) chỉ định. Ví dụ, để chỉ hiển thị các thông báo lỗi (level 3) hoặc cao hơn:

**dmesg -n 3**

--ctime: Hiển thị thời gian bắt đầu của mỗi thông báo dmesg.

**dmesg –ctime**

**---------------------------------------------------------------------------------------------------------------------**

--zypper--

zypper là một công cụ quản lý gói dựa trên dòng lệnh được sử dụng trong các hệ điều hành Linux dựa trên RPM như openSUSE. zypper cho phép bạn quản lý các gói phần mềm, cài đặt, gỡ cài đặt, cập nhật và tìm kiếm gói trên hệ thống của bạn. Dưới đây là một số lệnh zypper phổ biến:

Cài đặt gói phần mềm:

sudo zypper install <package-name>

Cập nhật tất cả các gói:

update

**---------------------------------------------------------------------------------------------------------------------**

**lệnh grub-install**

Lệnh **grub-install** trong Linux được sử dụng để cài đặt GRUB (Grand Unified Boot Loader) vào một ổ đĩa hoặc phân vùng cụ thể. GRUB là một trình khởi động đa hệ thống phổ biến sử dụng trong các hệ thống Linux và Unix để quản lý quá trình khởi động và cho phép bạn chọn các hệ điều hành hoặc môi trường khởi động khác.

Cú pháp cơ bản của lệnh grub-install là:

**grub-install [OPTIONS] DEVICE**

Trong đó, DEVICE là đối tượng bạn muốn cài đặt GRUB lên, có thể là ổ đĩa hoặc phân vùng.

Một số tùy chọn (options) thông dụng cho grub-install bao gồm:

* --target=TARGET: Xác định kiến trúc mục tiêu của GRUB. Thông thường, bạn sẽ sử dụng --target=i386-pc cho các hệ thống 32-bit hoặc --target=x86\_64-efi cho các hệ thống 64-bit với giao diện EFI.
* --boot-directory=DIR: Chỉ định thư mục chứa tệp cấu hình và hình ảnh GRUB.
* --efi-directory=DIR: Chỉ định thư mục chứa các tệp cho kiến trúc EFI.
* --recheck: Kiểm tra lại GRUB đã được cài đặt trước đó.
* --force: Cài đặt GRUB mà không kiểm tra kiến trúc mục tiêu.

Ví dụ cơ bản:

1. Cài đặt GRUB lên MBR của ổ đĩa /dev/sda:

**sudo grub-install /dev/sda**

**---------------------------------------------------------------------------------------------------------------------**

**lệnh find**

Lệnh find trong hệ thống Unix và Linux được sử dụng để tìm kiếm và xác định các tệp và thư mục dựa trên các tiêu chí cụ thể. Dưới đây là một số lệnh và tùy chọn phổ biến của find:

1. Cú pháp cơ bản của lệnh find:

find [đường dẫn] [tùy chọn] [kết quả]

* [đường dẫn]: Đường dẫn xuất phát cho việc tìm kiếm (mặc định là thư mục hiện tại).
* [tùy chọn]: Các tùy chọn cho lệnh find.
* [kết quả]: Các hành động cần thực hiện với kết quả tìm kiếm.

1. Tìm kiếm tệp hoặc thư mục dựa trên tên:

find /đường/dẫn -name "pattern"

Ví dụ: find /var/www -name "\*.html" tìm kiếm tất cả các tệp HTML trong thư mục /var/www.

1. Tìm kiếm tệp hoặc thư mục dựa trên kiểu:

find /đường/dẫn -type [d/f/l/c/b/p/s]

* + d: Thư mục
  + f: Tệp thường
  + l: Liên kết tượng trưng (symbolic link)
  + c: Thư mục kịch bản thiết bị ký tự
  + b: Thư mục kịch bản thiết bị khối
  + p: Ổ đĩa FIFO (named pipe)
  + s: Socket

1. Tìm kiếm dựa trên thời gian sửa đổi:

find /đường/dẫn -mtime [+/-]n

* +n: Lớn hơn n ngày trước.
* -n: Ít hơn n ngày trước.
* n: Chính xác n ngày trước.

1. Tìm kiếm dựa trên kích thước tệp:

arduinoCopy code

find /đường/dẫn -size [+/-]n[c]

* +n[c]: Lớn hơn n (n có thể là ký tự "c" để biểu thị byte).
* -n[c]: Ít hơn n (n có thể là ký tự "c" để biểu thị byte).
* n[c]: Chính xác n (n có thể là ký tự "c" để biểu thị byte).

1. Tìm kiếm tệp dựa trên quyền truy cập:

find /đường/dẫn -perm mode

Ví dụ: find /var -perm 644 tìm kiếm các tệp với quyền truy cập 644.

1. Tìm kiếm dựa trên quyền sở hữu và nhóm:

sqlCopy code

find /đường/dẫn -user owner -group group

Ví dụ: find /var/www -user user1 -group group1 tìm kiếm các tệp và thư mục trong /var/www mà có chủ sở hữu là user1 và thuộc nhóm group1.

1. Tìm kiếm và thực hiện hành động trên kết quả:

find /đường/dẫn -name "pattern" -exec [lệnh] {} \;

Ví dụ: find /var/www -name "\*.txt" -exec mv {} /tmp/ \; tìm kiếm tất cả các tệp với phần mở rộng .txt trong thư mục /var/www và di chuyển chúng đến thư mục /tmp/.

1. Hiển thị kết quả dưới dạng cây thư mục:

find /đường/dẫn -print

Ví dụ: find /var/www -type f -print tìm kiếm và hiển thị tất cả các tệp trong thư mục /var/www.

Lưu ý rằng find là một công cụ rất mạnh mẽ và có nhiều tùy chọn. Bạn có thể kết hợp nhiều tùy chọn để xác định cụ thể các tệp và thư mục bạn đang tìm kiếm.

**---------------------------------------------------------------------------------------------------------------------**

**lệnh watch**

Lệnh watch trong hệ thống Unix và Linux được sử dụng để thực hiện theo dõi và thực thi một lệnh cụ thể theo khoảng thời gian định sẵn. Nó giúp bạn xem dữ liệu cập nhật liên tục từ một lệnh mà bạn quan tâm. Dưới đây là các tùy chọn cơ bản của lệnh watch:

1. Cú pháp cơ bản:

watch [options] command

1. Cách sử dụng thời gian cập nhật mặc định (2 giây):

watch command

1. Tùy chọn thời gian cập nhật (giây):

-n, --interval=seconds

Ví dụ:

watch -n 5 command

1. Hiển thị kết quả dưới dạng tiêu đề (mặc định):

-t, --title

Ví dụ:

watch -t command

1. Không hiển thị tiêu đề:

-t, --no-title

Ví dụ:

watch -t command

1. Xóa màn hình trước mỗi cập nhật:

-c, --clear

Ví dụ:

watch -c command

1. Xem chi tiết về sự thay đổi giữa các cập nhật:

-d, --differences

Ví dụ:

watch -d command

1. Số lần lặp của lệnh (mặc định là vô hạn):

-n, --count

Ví dụ:

watch -n 10 -c command

1. Tắt sau một số lần lặp (kết hợp với -c để xóa màn hình):

-c -n

Ví dụ:

watch -n 10 -c command

1. Thoát sau khi một lần lặp hoàn thành (số lần lặp là 1):

-n 1

Ví dụ:

watch -n 1 command

Lệnh watch rất hữu ích khi bạn muốn theo dõi sự thay đổi trong dữ liệu hoặc kết quả của một lệnh mà bạn thực thi và muốn xem cập nhật liên tục sau mỗi khoảng thời gian cụ thể.

**---------------------------------------------------------------------------------------------------------------------**

**---uptime---**

Lệnh "uptime" trong hệ thống Linux được sử dụng để hiển thị thông tin về thời gian hoạt động của hệ thống và tải trung bình trong một khoảng thời gian cụ thể. Dưới đây là cú pháp cơ bản của lệnh "uptime" và một số tùy chọn phổ biến:

Cú pháp cơ bản:

**- uptime**

Kết quả mẫu có thể trông như sau:

13:45:08 up 1 day, 2:30, 2 users, load average: 0.23, 0.19, 0.22

Trong đó:

- "13:45:08" là thời gian hiện tại của hệ thống.

- "up 1 day, 2:30" cho biết thời gian mà hệ thống đã hoạt động (1 ngày và 2 giờ 30 phút).

- "2 users" cho biết có 2 người dùng đang đăng nhập vào hệ thống.

- "load average: 0.23, 0.19, 0.22" là tải trung bình trong 1 phút, 5 phút và 15 phút gần nhất.

Một số tùy chọn phổ biến của lệnh "uptime" bao gồm:

- -p: Hiển thị thời gian hoạt động và tải trung bình dưới dạng số nguyên, dễ dàng cho việc sử dụng trong các tập lệnh hoặc scripts.

- -s: Hiển thị thời gian hệ thống bắt đầu hoạt động.

- -h: Hiển thị trợ giúp về cách sử dụng lệnh "uptime".

Lệnh "uptime" là một công cụ hữu ích để kiểm tra thời gian hoạt động của hệ thống và tải trung bình, đặc biệt khi bạn quan tâm đến hiệu suất hoặc theo dõi tình trạng của máy chủ Linux.

**---------------------------------------------------------------------------------------------------------------------**

**---top---**

Lệnh "top" trong hệ thống Linux là một tiện ích dòng lệnh mạnh mẽ được sử dụng để hiển thị thông tin thời gian thực về hoạt động của các tiến trình và tài nguyên hệ thống, như CPU, bộ nhớ, và các tiến trình đang chạy. "top" thường được sử dụng để theo dõi hiệu suất hệ thống và xác định các tiến trình tiêu thụ nhiều tài nguyên. Dưới đây là cú pháp cơ bản và một số tùy chọn phổ biến của lệnh "top":

Cú pháp cơ bản:

**top**

Khi bạn chạy lệnh "top" theo cách này, nó sẽ hiển thị một bảng với danh sách các tiến trình, thông tin về CPU, bộ nhớ và các thống kê khác. Bạn có thể sử dụng các phím tắt để tương tác với "top". Ví dụ, để thoát khỏi "top", bạn có thể sử dụng phím tắt "q".

Một số tùy chọn phổ biến của lệnh "top" bao gồm:

**- -b:** Chạy "top" trong chế độ batch, làm cho nó xuất kết quả ra một tệp thay vì hiển thị trực tiếp trên màn hình.

**- -n** NUM: Xác định số lần cập nhật trước khi thoát.

**- -d** SECONDS: Xác định khoảng thời gian giữa các cập nhật.

**- -c:** Hiển thị cả các tiến trình con (child processes).

**- -u** USERNAME: Hiển thị chỉ thông tin về tiến trình của một người dùng cụ thể.

- **-p** PID1,PID2,PID3,...: Hiển thị chỉ thông tin về các tiến trình có số hiệu tiến trình (PID) cụ thể.

Ví dụ sử dụng một số tùy chọn:

1. Chạy "top" với tùy chọn "b" để xuất kết quả ra một tệp văn bản:

**top -b -n 1 > top\_output.txt**

2. Hiển thị thông tin về tiến trình của người dùng "username":

**top -u username**

3. Hiển thị chỉ thông tin về các tiến trình có PID cụ thể:

**top -p 1234,5678**

"Lệnh top" là một công cụ hữu ích để theo dõi tình trạng hoạt động của hệ thống và các tiến trình, và nó cung cấp rất nhiều thông tin quan trọng.

**---------------------------------------------------------------------------------------------------------------------**

**---uname---**

Lệnh "uname" trong Linux được sử dụng để hiển thị thông tin về hệ điều hành của hệ thống. Nó trả về các thông tin như tên hệ điều hành, máy chủ, phiên bản hệ thống, kiến trúc máy tính, và nhiều thông tin liên quan khác. Dưới đây là cú pháp cơ bản của lệnh "uname" và một số tùy chọn phổ biến:

Cú pháp cơ bản:

**uname [OPTIONS]**

Một số tùy chọn phổ biến của lệnh "uname" bao gồm:

**- -a, --all:** Hiển thị tất cả thông tin có sẵn.

**- -s, --kernel-name:** Hiển thị tên hạt nhân (kernel name).

- **-n, --nodename:** Hiển thị tên máy chủ (node name).

**- -r, --kernel-release:** Hiển thị phiên bản của hạt nhân (kernel release).

**- -v, --kernel-version:** Hiển thị phiên bản của hạt nhân (kernel version).

**- -m, --machine**: Hiển thị tên kiến trúc máy tính (machine hardware name).

**- -o, --operating-system**: Hiển thị tên hệ điều hành (operating system).

Ví dụ sử dụng "uname" với các tùy chọn:

1. Hiển thị tất cả thông tin:

uname -a

2. Hiển thị tên hạt nhân:

uname -s

3. Hiển thị tên máy chủ:

uname -n

4. Hiển thị phiên bản của hạt nhân:

uname -r

5. Hiển thị phiên bản của hạt nhân (kernel version):

uname -v

6. Hiển thị tên kiến trúc máy tính:

uname -m

7. Hiển thị tên hệ điều hành:

uname -o

Lệnh "uname" là một công cụ hữu ích để xác định các thông tin cơ bản về hệ điều hành và máy tính mà bạn đang sử dụng.

**---------------------------------------------------------------------------------------------------------------------**

**---** **mkfs** **---**

Lệnh mkfs là một lệnh trong hệ điều hành Linux được sử dụng để tạo hệ thống tệp (filesystem) trên một thiết bị lưu trữ như ổ cứng, ổ đĩa USB, hoặc phân vùng. Cụ thể, mkfs được sử dụng để định dạng (format) thiết bị để tạo ra các cấu trúc dữ liệu cần thiết để lưu trữ các tệp và thư mục trên đó.

mkfs -t <filesystem\_type> <device>

Trong đó:

* <filesystem\_type> là loại hệ thống tệp bạn muốn tạo, ví dụ: ext4, ext3, xfs, vfat, ntfs, và nhiều loại khác.
* <device> là thiết bị lưu trữ bạn muốn tạo hệ thống tệp trên, thường là đường dẫn đến thiết bị như /dev/sda1 hoặc /dev/sdb.

Ví dụ, để tạo một hệ thống tệp ext4 trên thiết bị /dev/sda1, bạn có thể sử dụng lệnh sau:

mkfs -t ext4 /dev/sda1

--tune2fs--

Lệnh tune2fs là một công cụ trên hệ điều hành Linux được sử dụng để điều chỉnh và thay đổi các thuộc tính của hệ thống tệp ext2, ext3 và ext4 (Extended Filesystem). Đây là một số công việc thường được thực hiện bằng tune2fs:

1. Thay đổi tần suất kiểm tra hệ thống tệp: Bạn có thể sử dụng tune2fs để đặt tần suất kiểm tra hệ thống tệp bằng cách sử dụng tùy chọn -c (thay đổi sau mỗi n lần khởi động) hoặc -i (thay đổi sau mỗi n ngày).
2. Thay đổi tên hệ thống tệp: Bạn có thể đặt tên cho hệ thống tệp sử dụng tùy chọn -L.
3. Thay đổi thông báo trước khi hệ thống tệp bị kiểm tra: Bạn có thể sử dụng tune2fs để thay đổi thông báo trước khi hệ thống tệp bị kiểm tra bằng cách sử dụng tùy chọn -T.

**---------------------------------------------------------------------------------------------------------------------**

**---** **mkfs** **---**

**---------------------------------------------------------------------------------------------------------------------**

**---tar---**

Lệnh tar (tên viết tắt từ "tape archive") là một công cụ mạnh mẽ trong các hệ thống Linux và Unix để nén và giải nén tệp và thư mục. Dưới đây là một số tùy chọn phổ biến của lệnh tar:

**1. Tạo một tệp nén (archive):**

- **tar -cvf archive.tar file1 file2**: Tạo một tệp archive.tar chứa file1 và file2.

**2. Thêm tệp vào một tệp nén hiện có:**

- **tar -rvf archive.tar newfile:** Thêm newfile vào archive.tar.

**3. Tạo tệp nén và nén một thư mục:**

- t**ar -cvf archive.tar directory/:** Tạo tệp archive.tar chứa toàn bộ thư mục directory.

**4. Xem nội dung của một tệp nén:**

**- tar -tf archive.tar:** Hiển thị danh sách tất cả các tệp trong archive.tar.

**5. Giải nén một tệp nén:**

**- tar -xvf archive.tar:** Giải nén tất cả các tệp từ archive.tar.

**6. Giải nén một tệp nén vào một thư mục cụ thể:**

**- tar -xvf archive.tar -C /path/to/directory/:** Giải nén archive.tar vào thư mục /path/to/directory/.

**7. Nén bằng gzip:**

**- tar -czvf archive.tar.gz file1 file2:** Nén file1 và file2 vào một tệp archive.tar.gz sử dụng gzip.

**8. Giải nén tệp tar.gz:**

**- tar -xzvf archive.tar.gz:** Giải nén một tệp archive.tar.gz.

**9. Nén bằng bzip2:**

**- tar -cjvf archive.tar.bz2 file1 file2:** Nén file1 và file2 vào một tệp archive.tar.bz2 sử dụng bzip2.

**10. Giải nén tệp tar.bz2:**

**- tar -xjvf archive.tar.bz2:** Giải nén một tệp archive.tar.bz2.

**11. Xác định thư mục mục tiêu:**

**- tar -C /path/to/directory/ -cvf archive.tar .:** Nén toàn bộ thư mục hiện tại và lưu vào archive.tar.

**12. Xóa các tệp sau khi nén (cùng với tùy chọn khác):**

**- tar --remove-files -cvf archive.tar file1 file2:** Nén file1 và file2 và sau đó xóa chúng.

Đây chỉ là một số tùy chọn phổ biến của lệnh tar. Bạn có thể tìm hiểu thêm bằng cách tham khảo tài liệu hệ thống hoặc sử dụng man tar để xem hướng dẫn chi tiết.

**---------------------------------------------------------------------------------------------------------------------**

**---file---**

Lệnh "file" trong Linux được sử dụng để xác định kiểu file của một tập tin hoặc thư mục cụ thể. Điều này giúp bạn biết được loại dữ liệu bạn đang làm việc, chẳng hạn như văn bản, hình ảnh, thư mục, thư mục nén, thư mục thư mục, v.v. Dưới đây là cách cơ bản để sử dụng lệnh "file" trong Linux:

Một số tùy chọn thông dụng:

1. `-b` hoặc `--brief`: In kết quả ngắn gọn, chỉ hiển thị loại file.

2. `-i` hoặc `--mime`: In thông tin MIME type của tập tin.

3. `-z` hoặc `--uncompress`: Cố gắng giải nén tập tin và sau đó kiểm tra loại file.

4. `-L` hoặc `--dereference`: Kiểm tra kiểu file của tệp liên kết mềm thay vì tệp nguồn.

Ví dụ sử dụng lệnh "file" với các tùy chọn:

**1. Kiểm tra loại file của một tập tin cụ thể:**

file example.txt

**2. In ra thông tin MIME type của một tập tin:**

file -i example.jpg

**3. Kiểm tra loại file của một tập tin sau khi giải nén (nếu tập tin là một tệp nén):**

file -z compressed.tar.gz

**4. Kiểm tra kiểu file của tệp liên kết mềm thay vì tệp nguồn:**

file -L symbolic\_link

**---------------------------------------------------------------------------------------------------------------------**

**---gunzip---**

Lệnh `gunzip` trong Linux và Unix được sử dụng để giải nén (decompress) các tệp nén bằng gZIP compression. Dưới đây là cú pháp cơ bản và một số tùy chọn phổ biến của lệnh `gunzip`:

**Cú pháp cơ bản:**

gunzip [tùy chọn] [tên\_tệp\_nén.gz]

Một số tùy chọn thông dụng:

**1. `-c` hoặc `--stdout`: In nội dung tệp giải nén ra màn hình thay vì ghi vào tệp mục tiêu.**

gunzip -c file.gz

**2. `-d` hoặc `--decompress`: Sử dụng tùy chọn này để chỉ ra rằng bạn muốn giải nén tệp (đây là tùy chọn mặc định khi bạn sử dụng `gunzip`, vì vậy nó thường không cần thiết).**

gunzip -d file.gz

**3. `-f` hoặc `--force`: Làm cho `gunzip` bỏ qua các cảnh báo và ghi đè lên tệp mục tiêu nếu tệp đã tồn tại.**

gunzip -f file.gz

**4. `-k` hoặc `--keep`: Giữ tệp nén gốc sau khi giải nén (mặc định, tệp gốc bị xóa).**

gunzip -k file.gz

**5. `-t` hoặc `--test`: Kiểm tra tính toàn vẹn của tệp nén mà không giải nén nó.**

gunzip -t file.gz

Lệnh `gunzip` là một công cụ phổ biến trong môi trường Linux và Unix để làm việc với các tệp nén bằng gZIP compression.

**---------------------------------------------------------------------------------------------------------------------**

**---at---**

Lệnh **"at"** trong hệ điều hành Linux và Unix được sử dụng để thực hiện các nhiệm vụ một lần hoặc đặt lịch thời gian để chạy các tác vụ vào một thời điểm cụ thể trong tương lai. Dưới đây là cú pháp cơ bản và một số tùy chọn phổ biến của lệnh "at":

Cú pháp cơ bản:

→ at [tùy\_chọn] thời\_gian

Một số tùy chọn thông dụng:

**1. `-l` hoặc `--list`: Liệt kê các công việc đã được đặt lịch bằng "at".**

at -l

**2. `-d` hoặc `--delete`: Xóa một công việc đã được đặt lịch theo số công việc đã liệt kê trong danh sách công việc.**

at -d job\_number

**3. `-m` hoặc `--mail`: Gửi email cho người dùng khi công việc hoàn thành.**

at -m thời\_gian

**4. `-f` hoặc `--file`: Đọc tác vụ từ một tệp văn bản thay vì nhập tác vụ từ dòng lệnh.**

at -f file\_name thời\_gian

**5. `-c` hoặc `--command`: Đặt tác vụ trực tiếp từ dòng lệnh sau tùy chọn thời gian.**

at -c "command" thời\_gian

Ví dụ sử dụng lệnh "at" với các tùy chọn:

**1. Đặt lịch để thực hiện một công việc vào 2 giờ chiều ngày hôm nay:**

at 2:00 PM

**2. Đặt lịch để chạy một tác vụ từ tệp văn bản vào 3 giờ sáng ngày mai:**

at -f myscript.sh 3:00 AM

**3. Liệt kê các công việc đã được đặt lịch:**

at -l

**4. Xóa một công việc đã được đặt lịch, ví dụ công việc số 1:**

at -d 1

**5. Đặt lịch gửi email cho người dùng khi công việc hoàn thành vào 5 giờ chiều:**

at -m 5:00 PM

Lệnh "at" là một cách tiện lợi để tự động hóa các tác vụ trong hệ thống Linux hoặc Unix tại một thời điểm cụ thể.

In summary,

The logic for at usage is: If /etc/at.allow exists, only non-root users listed within it can schedule jobs.

If /etc/at.deny exists, only non-root users listed within it cannot schedule jobs. If none of them exists: permission for non-root scheduling is distro dependent.

**---------------------------------------------------------------------------------------------------------------------**

**---jobs---**

Trong hệ điều hành Linux và Unix, lệnh "jobs" được sử dụng để quản lý và xem danh sách các công việc (jobs) đang chạy hoặc đã dừng trong phiên làm việc hiện tại của bạn trong môi trường dòng lệnh. Dưới đây là cú pháp cơ bản và một số tùy chọn thông dụng của lệnh "jobs":

Cú pháp cơ bản:

- jobs [tùy\_chọn]

Một số tùy chọn thông dụng:

**1. `-l`: Hiển thị danh sách các công việc kèm theo các thông tin chi tiết như số công việc (job number), trạng thái của công việc (Running hoặc Stopped), và lệnh được chạy.**

jobs -l

**2. `-n`: Hiển thị chỉ các công việc gần đây được tạo ra hoặc thay đổi trạng thái trong phiên làm việc hiện tại.**

jobs -n

**3. `-p`: Hiển thị danh sách các công việc trong tiến trình (process) chạy hiện tại.**

jobs -p

**4. `-%N` hoặc `+%N`: Chọn công việc dựa trên số công việc (job number). `-%N` sẽ chọn công việc với số âm (được đánh số ngược), trong khi `+%N` sẽ chọn công việc với số dương.**

jobs -%2

**5. `-r`: Hiển thị chỉ các công việc đang chạy (Running).**

jobs -r

**6. `-s`: Hiển thị chỉ các công việc đã dừng (Stopped).**

jobs -s

Ví dụ sử dụng lệnh "jobs" với các tùy chọn:

**1. Hiển thị danh sách các công việc đang chạy cùng với thông tin chi tiết:**

jobs -l

**2. Hiển thị danh sách các công việc đã dừng trong phiên làm việc hiện tại:**

jobs -s

**3. Hiển thị danh sách các công việc gần đây đã tạo ra hoặc thay đổi trạng thái:**

jobs -n

Lệnh "jobs" rất hữu ích khi bạn làm việc với nhiều tiến trình và muốn quản lý chúng trong môi trường dòng lệnh.

**---------------------------------------------------------------------------------------------------------------------**

**---renice---**

Lệnh **"renice"** trong Linux được sử dụng để thay đổi ưu tiên của một tiến trình đang chạy. Bằng cách điều chỉnh ưu tiên, bạn có thể kiểm soát mức ưu tiên của tiến trình trong hệ thống, làm cho nó hoạt động chậm hơn hoặc nhanh hơn. Dưới đây là cú pháp cơ bản và một số tùy chọn của lệnh "renice":

Cú pháp cơ bản:

**renice [tùy\_chọn] [độ ưu tiên] -p [ID\_tiến\_trình]**

Một số tùy chọn thông dụng:

**1. `-n` hoặc `--priority`: Điều chỉnh độ ưu tiên của tiến trình dựa trên một giá trị ưu tiên. Giá trị ưu tiên thường là một số từ -20 (ưu tiên cao nhất) đến 19 (ưu tiên thấp nhất). Bạn cũng có thể sử dụng số dương để tăng độ ưu tiên hoặc số âm để giảm độ ưu tiên.**

renice -n 10 -p 1234

**2. `-g` hoặc `--group`: Điều chỉnh độ ưu tiên của một nhóm tiến trình. Bạn cung cấp ID của nhóm tiến trình thay vì ID của tiến trình cụ thể.**

renice -n 5 -g 5678

**3. `-u` hoặc `--user`: Điều chỉnh độ ưu tiên của tất cả tiến trình của một người dùng. Bạn cung cấp tên người dùng thay vì ID tiến trình hoặc nhóm.**

renice -n -5 -u user123

Ví dụ sử dụng lệnh "renice" với các tùy chọn:

**1. Điều chỉnh độ ưu tiên của tiến trình có ID 1234 để đặt độ ưu tiên là 15:**

renice -n 15 -p 1234

**2. Giảm độ ưu tiên của tất cả tiến trình của người dùng "user123" xuống -5:**

renice -n -5 -u user123

**3. Điều chỉnh độ ưu tiên của tất cả tiến trình thuộc nhóm có ID 5678 lên 5:**

renice -n 5 -g 5678

Lệnh "renice" là một công cụ quan trọng để quản lý độ ưu tiên của tiến trình trong hệ thống Linux và Unix, và nó có thể được sử dụng để cải thiện hoạt động của hệ thống trong môi trường đa nhiệm.

**---------------------------------------------------------------------------------------------------------------------**

**---modprobe---**

`modprobe` là một công cụ dòng lệnh trong hệ thống Linux được sử dụng để tải (load) mô-đul kernel và tất cả các mô-đul phụ thuộc cần thiết (dependencies). Nó tự động quản lý việc giải quyết các phụ thuộc mô-đul và nạp chúng vào hạt nhân (kernel).

Cú pháp cơ bản của `modprobe` là:

**sudo modprobe [tùy\_chọn] [tên\_mô\_đul]**

Một số tùy chọn thông dụng của `modprobe` bao gồm:

**- `-r`:** Sử dụng để gỡ bỏ một mô-đul khỏi hạt nhân. Ví dụ: `sudo modprobe -r module\_name`.

**- `-v` hoặc `--verbose`:** Hiển thị thông tin chi tiết về các hoạt động `modprobe`.

**- `-a` hoặc `--all`:** Tải tất cả các mô-đul có sẵn trong các thư mục mô-đul (thông thường `/lib/modules/`).

**- `-c` hoặc `--showconfig`:** Hiển thị cấu hình cơ bản của `modprobe`.

- **`-n`:** Chỉ hiển thị các mô-đul sẽ được tải mà không thực hiện thực sự thao tác.

Ví dụ sử dụng `modprobe` để tải một mô-đul là:

sudo modprobe module\_name

Nó sẽ tự động tải mô-đul được chỉ định và tất cả các mô-đul phụ thuộc cần thiết, giúp bạn quản lý mô-đul kernel một cách thuận tiện trong Linux.

**---------------------------------------------------------------------------------------------------------------------**

**---lspci---**

Lệnh `lspci` có thể hiển thị thông tin về phần cứng của hệ thống, bao gồm các thông tin sau:

B. Cài đặt IRQ của thiết bị: **`lspci`** có thể hiển thị các cài đặt IRQ cho các thiết bị PCI, thông tin quan trọng để hiểu cách phần cứng hệ thống được cấu hình.

D. Địa chỉ MAC Ethernet: Lệnh **`lspci`** có thể hiển thị thông tin về bộ điều khiển Ethernet, bao gồm địa chỉ MAC (Media Access Control).

E. Nhận dạng nhà sản xuất và thiết bị của thiết bị: **`lspci`** cung cấp thông tin về nhà sản xuất và mã thiết bị, thường bao gồm tên nhà sản xuất và tên thiết bị cùng với mã nhà sản xuất và mã thiết bị.

**---------------------------------------------------------------------------------------------------------------------**

**---ldd---**

`ldd` là một tiện ích dòng lệnh trong hệ thống Linux và Unix được sử dụng để kiểm tra các phụ thuộc thư viện (shared libraries) của một tệp thực thi. Nó giúp xác định các thư viện nào cần được tải vào bộ nhớ để chạy một chương trình. Dưới đây là cú pháp cơ bản và một số tùy chọn thông dụng của lệnh `ldd`:

Cú pháp cơ bản:

**ldd [tùy\_chọn] tệp\_thực\_thi**

Một số tùy chọn thông dụng:

**1. `-v` hoặc `--verbose`: Hiển thị thông tin chi tiết về các thư viện, bao gồm địa chỉ và phiên bản.**

ldd -v executable\_file

**2. `-u` hoặc `--unused`: Hiển thị danh sách các thư viện không được sử dụng bởi tệp thực thi.**

ldd -u executable\_file

**3. `-r` hoặc `--function-relocs`: Hiển thị các relocatable functions (hàm có thể được di chuyển) trong thư viện, nếu có.**

ldd -r executable\_file

**4. `-d` hoặc `--data-relocs`: Hiển thị các relocatable data objects (đối tượng dữ liệu có thể được di chuyển) trong thư viện, nếu có.**

ldd -d executable\_file

**5. `-f` hoặc `--file`: Hiển thị đường dẫn tuyệt đối của thư viện được sử dụng.**

ldd -f executable\_file

Lệnh `ldd` giúp bạn kiểm tra xem một tệp thực thi cụ thể cần các thư viện nào để chạy. Điều này có thể hữu ích trong việc xác định sự phụ thuộc của các ứng dụng và giải quyết các vấn đề liên quan đến thư viện.

**---------------------------------------------------------------------------------------------------------------------**