



# Tài liệu hướng dẫn học Linux **LPIC-1**

**Linux  
Professional  
Institute  
Certification  
LPIC-1 101**

# Tiêu đề

## Contents

101.1 – Xác định và cấu hình cài đặt phần cứng(Filesystems và Device Files) .....	1
101.1 Xác định và Cấu hình Cài đặt Phần cứng (Công cụ và Tiện ích để Tìm Hiểu Thiết bị Hệ thống Linux) 2	
101.2 Boot the System (Quá trình khởi động - sysvinit - Từ bật nguồn điện trên máy tính, đến vào chế độ dòng lệnh của hệ điều hành) .....	5
101.2 Khởi động hệ thống (Quá trình khởi động - systemd và upstart - Từ bật nguồn điện trên máy tính đến vào chế độ dòng lệnh của hệ điều hành).....	6
101.3 Thay đổi Runlevels/Boot Targets và Tắt (shutdown)hoặc khởi động(reboot) lại hệ thống Linux (systemd, sysvinit và upstart) .....	8
102.1 Thiết kế Layout Ổ cứng (Layout và Ví dụ Cài đặt) .....	9
102.1 Thiết kế Bố trí Ổ cứng (Cơ bản về LVM và Swap) .....	10
102.2 Cài đặt Boot Manager (Làm việc với GRUB Legacy).....	11
102.2 Cài đặt trình quản lý khởi động (Làm việc với trình khởi động GRUB2) .....	12
102.3 Quản lý Thư viện chia sẻ (Tổng quan về Thư viện chia sẻ (Shared Library)và Cấu hình các vị trí của Thư viện) .....	13
102.4 Sử dụng Quản lý Gói Debian (Kho chứa(Repositories) và Quản lý Gói(Package)) .....	14
102.4 Sử dụng Chương trình Quản lý Gói Debian.....	16
(Sử dụng Tiện ích dpkg).....	16
102.5 Sử dụng Quản lý Gói RPM và YUM .....	17
(Quản lý Gói YUM) .....	17
102.5 Sử dụng RPM và quản lý YUM Package .....	19
(sử dụng lệnh RPM) .....	19
Chủ đề 103 - Lệnh GNU và Unix 103.1 Làm việc trên Dòng lệnh (Shell, Sourcing và các Ký tự đặc biệt)... 22	
103.1 Làm việc trên Dòng lệnh (Các Biến môi trường - Hiển thị, Thiết lập và Sử dụng) .....	25
103.2 Xử lý luồng văn bản bằng Các lệnh Bộ lọc trong Textutils Package (sort, nl, wc, expand, cut, paste, join, uniq, head và tail).....	28
<b>103.2 Xử lý Luồng Văn bản Bằng Các Bộ Lọc (Các Lệnh Trong Gói Textutils: split, cat, od, pr, fmt, tr, sed, more và less).....</b>	<b>30</b>
103.3 Thực hiện việc Quản lý Cơ bản của file (Các Lệnh và Thuật ngữ cần biết - ls, cd, pwd, mkdir, file, globbing, touch, stat, cp và mv) .....	34
103.3 Thực hiện Quản lý File Cơ Bản (Các Lệnh và Thuật ngữ Cần Biết - dd, rmdir, rm và find) .....	39
103.3 Thực hiện Quản lý file Cơ Bản (Công cụ Lưu trữ và Nén) .....	41
103.4 Sử dụng Luồng(Streams), Ống(Pipes) và Chuyển hướng (Redirects) .....	44

103.5 Tạo, Giám sát và Hủy(kill) tiến trình process (Terms và Tiện ích- PID, ps, pstree, free, uptime, tín hiệu, kill, killall, pkill, pgrep).....	46
103.5 Tạo, Giám sát và Tiêu diệt Quá trình (Terms và Công cụ jobs, bg, fg, &, ưu tiên(priority), nohup và screen).....	48
103.6 Sửa đổi ưu tiên thực thi tiến trình (Sử dụng nice, renice và top).....	49
103.7 Tìm kiếm file văn bản bằng biểu thức chính quy Regular Expressions (Sử dụng grep, egrep, fgrep, sed và regex) .....	50
<b>103.8 Thực hiện các thao tác chỉnh sửa FILE cơ bản bằng VI .....</b>	<b>53</b>
104.1 Tạo Phân vùng (Partitions) và Filesystem (Sử dụng Công cụ Phân vùng - fdisk, gdisk và parted) ....	57
<b>104.1 Tạo phân vùng(partitions) và Filesystems (Các loại Filesystem và tạo chúng trên các phân vùng)</b> .....	<b>60</b>
<b>104.2 Bảo vệ tính toàn vẹn của filesystems .....</b>	<b>62</b>
<b>104.3 Kiểm soát việc gắn mount và hủy gắn umount filesystem (Gắn filesystem thủ công và tự động)</b>	<b>65</b>
104.4 Quản lý hạn ngạch(Quotas) ổ đĩa(disk).....	66
104.5 Quản lý Quyền Truy cập(Permissions) và Sở hữu (Ownership)File .....	69
104.6 Tạo và Thay đổi Liên kết Cứng(hard links) và Liên kết Tượng trưng (symbolic links hoặc gọi là liên kết mềm soft links).....	74
104.7 Tìm Kiếm Files Hệ Thống và Đặt file vào Vị Trí Chính Xác (Tiêu chuẩn phân cấp hệ thống file-Filesystem Hierarchy Standard) .....	75
104.7 Tìm Files Hệ Thống và Đặt Files vào Vị Trí Đúng (Sử dụng find, locate, updatedb, /etc/updatedb.conf, whereis, which, và type) .....	78
<b>Khắc phục lỗi yum update trong Centos6.....</b>	<b>79</b>

**Yêu cầu hệ điều hành:**

Linux - CentOS 6, CentOS 8 (hoặc bản phân phối SysVinit hoặc Systemd khác)

# Chương 101- kiến trúc hệ thống Linux

## 101.1 – Xác định và cấu hình cài đặt phần cứng(Filesystems và Device Files)

- Linux coi tất cả các thiết bị phần cứng là dưới dạng như là một file. Điều này có nghĩa là các thiết bị như ổ cứng, ổ đĩa CD/DVD, cổng USB, card mạng và các thiết bị khác được đại diện bởi các files trong file systems (hệ thống file) của Linux.
- **Procfs**
  - Thông tin về phần cứng và tình trạng hoạt động của hệ thống trong Linux được chứa trong 'pseudofilesystem' này.
  - **LƯU Ý:** Một 'pseudofilesystem' là một cái gì đó trông giống như một thiết bị trong hệ thống Linux, nhưng thực sự không liên quan đến phần cứng thật, nó chỉ hiển thị thông tin kernel dưới dạng đọc được cho chúng ta đọc thông tin.
- **/proc**
  - Thường là nơi mà procfs filesystem được gắn vào Linux
  - Trong thư mục này, bạn sẽ tìm thấy các thư mục tương ứng với nhãn (số) tương ứng với PID (Process ID) của các tiến trình đang chạy
  - Ngoài ra, các files và thư mục khác chứa thông tin về hệ thống hoặc phần cứng trong đó bao gồm:
    - **/proc/mounts** - một file(hoặc liên kết đến file khác) chứa thông tin về tất cả các filesystems được gắn vào hệ điều hành Linux
    - **/proc/interrupts** - hiển thị thông tin về các ngắt (interrupts) được sử dụng trong hệ thống và thiết bị phần cứng được liên kết đến dưới dạng files. Giúp quản trị hệ thống có thể hiểu được tài nguyên phần cứng đang được sử dụng và tương tác trên hệ thống.
    - **/proc/ioports** - một hoặc nhiều địa chỉ xác định một thiết bị và module kernel liên kết đến chúng
    - **/proc/dma** - (Direct Memory Access), các giá trị được sử dụng bởi phần cứng để truy cập trực tiếp vào bộ nhớ hệ thống (tức là không cần đến CPU)
    - **/proc/usb** - các ID thiết bị USB và module kernel liên kết đến chúng
    - **/proc/pci** - PCI các IDs của các thiết bị và kernel modules liên kết đến chúng
- **Sysfs**
  - Một 'pseudofilesystem' khác có thể chứa thông tin về phần cứng hệ thống (tương tự như procfs)

- Được thiết kế để giải quyết một số vấn đề với phương pháp procfs (dữ liệu không có cấu trúc và thông tin về phần cứng và tiến trình được tổng hợp trong cùng một cấu trúc)

- /sys

- Nơi mà 'sysfs' filesystem được gắn mounted vào hệ điều hành
- Trong thư mục này, bạn sẽ tìm thấy các thư mục tương ứng với phần cứng hệ thống và các module kernel liên quan đến hệ thống Linux (vật lý hoặc ảo)
- LƯU Ý: dữ liệu tiến trình (PID) KHÔNG tồn tại trong phương pháp này

- udev

- Chương trình quản lý thiết bị cho kernel

- /dev

- Thư mục chứa dữ liệu thiết bị (sử dụng thông tin từ /sys hoặc sysfs)
- Các thiết bị được thêm hoặc gỡ bỏ trong hệ điều hành hoạt động sẽ kích hoạt hệ thống udev, sẽ phát hiện và cấu hình nó, sau đó đưa thiết bị vào hệ điều hành để sử dụng (thường được gọi là 'hotplug' hoặc 'hotswap')

- D-Bus

- Liên quan đến hệ thống udev
- Cho phép các ứng dụng desktop gửi thông điệp tới các ứng dụng khác và/hoặc nhận thông điệp từ Linux kernel
- Ví dụ: khi gắn mount ổ CD-ROM vào Linux, hệ thống 'udev' sử dụng D-Bus để thông báo với trình quản lý cửa sổ đang chạy để hiển thị một biểu tượng lên trên màn hình desktop

## 101.1 Xác định và Cấu hình Cài đặt Phần cứng (Công cụ và Tiện ích để Tìm Hiểu Thiết bị Hệ thống Linux)

- lsmod

- Liệt kê các module kernel (drivers) đã được tải vào bộ nhớ (cùng với các thành phần phụ thuộc(dependencies) của chúng)
- Sẽ không tải một driver nếu nó bị thiếu

- lscpu

- Hiển thị một thông tin tổng kết về CPU và các tính năng/cấu hình của nó
- -a (hoặc --all) - sẽ hiển thị thông tin về CPU khi online và offline
- -b (hoặc --online) - chỉ hiển thị thông tin về CPU khi online
- -c (hoặc --offline) - chỉ hiển thị thông tin về CPU khi offline
- -e (--extended) =[list] - hiển thị thông tin dưới dạng định dạng dễ đọc hơn
- [list] - cột(s) để bao gồm (mặc định là tất cả - ví dụ bao gồm 'cpu' hoặc 'node')

- **-p** (hoặc **--parse**) - cung cấp thông tin hiển thị dưới dạng định dạng được phân tách bằng dấu phẩy (có thể được ghi lại và sử dụng bởi các ứng dụng khác để cho việc báo cáo thông tin)
- **lspci**
  - Hiển thị thông tin về bus PCI và các thiết bị liên quan đã được 'kết nối' vào đó
  - Trong trường hợp kernel không nhận ra (hoặc không hỗ trợ theo mặc định) một thiết bị, nó sẽ xuất hiện ở đây và chứa thông tin (khe cắm, ID thiết bị, phạm vi bộ nhớ, v.v.) có thể được sử dụng để tìm kiếm driver/module thiết bị
  - **-v** (hoặc **-vv** hoặc **-vvv**) - Kiểm soát mức độ chi tiết của thông tin hiển thị
  - **-m** (hoặc **-mm**) - Hiển thị thông tin PCI dưới dạng định dạng "dễ phân tích"
  - **-vmm** - Hiển thị danh sách được định dạng đẹp về tất cả các thiết bị PCI trên hệ thống
  - **-t** - Hiển thị một cấu trúc cây của các thiết bị PCI trên hệ thống (CHỈ khe cắm/cổng thiết bị)
  - **-tvmm** - liên kết các tên thiết bị với các ID trong cấu trúc cây
- **lsscsi**
  - Hiển thị thông tin về bất kỳ thiết bị SCSI nào được phát hiện
    - **LƯU Ý:** các bản phân phối Linux hiện đại sẽ hiển thị các thiết bị SATA được phát hiện thông qua lệnh này
  - **-c** (hoặc **--classic**) - đầu ra tương tự như việc chạy `cat /proc/scsi/scsi`
  - **-d** (hoặc **--device**) - cung cấp số 'major/minor' thiết bị bổ sung sau mỗi thiết bị được phát hiện
  - **-g** (hoặc **--generic**) - tên file thiết bị SCSI chung (sg)
  - **-l** (hoặc **--long**)
  - **-s** (hoặc **--size**) - hiển thị kích thước ổ đĩa trong định dạng dễ đọc hơn
  - **-v** (hoặc **-vv** hoặc **-vvv**) - điều khiển mức độ chi tiết (sự chi tiết)
- **lsdev**
  - Hiển thị tất cả các thiết bị được nhận dạng bởi kernel hệ thống Linux đang chạy
- **lsraid**
  - Hiển thị các hệ thống RAID (Redundant Array of Inexpensive Disks) có trong hệ thống Linux của bạn
  - **LƯU Ý:** không được cài đặt trên hầu hết các hệ điều hành Linux, không có sẵn ngoại trừ một số trình điều khiển drivers/ứng dụng cấu hình RAID
- **lsusb**
  - Hiển thị các ID thiết bị USB và thông tin chung về các thiết bị được phát hiện
  - **-s** [bus]:[device #] - hiển thị thông tin bus và số thiết bị chỉ định (tùy chọn)
  - **-d** [vendor]:[product #] - chỉ hiển thị các thiết bị có vendor ID và product ID chỉ định (dưới định dạng thập lục phân)
  - **-D** [device] - bỏ qua quá trình quét (scan) thư mục /dev/bus/usb, chỉ hiển thị thông tin về thiết bị chỉ định
    - **LƯU Ý:** chỉ có người dùng root mới có thể sử dụng tham số này, định dạng yêu cầu dựa trên đường dẫn, không phải ID thiết bị (**ví dụ:** /dev/bus/usb/001/001)
  - **-t** - hiển thị cấu trúc theo dạng cây
  - **-v** - đầu ra chi tiết về thiết bị

• **lsblk**

- Hiển thị các **block devices** (các ổ đĩa) được kết nối với hệ thống Linux đang chạy
- **-a** (hoặc **--all**) - liệt kê cả các thiết bị trống (mặc định là tắt)
- **-d** (hoặc **--nodeps**) - chỉ in thông tin thiết bị cấp cao nhất
  - **Ví dụ:** **lsblk --nodeps /dev/sdb** (chỉ hiển thị thông tin thiết bị trên ổ đĩa /dev/sdb)
- **-e** (**--exclude**) **[list]** - loại trừ các thiết bị chỉ định (một hoặc nhiều, phân cách bằng dấu phẩy)
- **-f** (hoặc **--fs**) - bao gồm thông tin về filesystem
- **-i** (hoặc **--ascii**) - sử dụng ký tự ASCII cho chế độ xem dạng cây
- **-l** (hoặc **--list**) - định dạng danh sách đầu ra
- **-t** - xem các thiết bị và phân vùng(partitions) ổ đĩa trong hệ điều hành dưới dạng cây.

## 101.2 Boot the System (Quá trình khởi động - sysvinit - Từ bật nguồn điện trên máy tính, đến vào chế độ dòng lệnh của hệ điều hành)

### • sysvinit

- Quản lý hệ thống và dịch vụ của hệ điều hành Linux cũ ví dụ từ Linux Centos 6 trở về trước (nhưng vẫn hợp lệ cho kỳ thi)
- Dựa trên các tập lệnh init
- Thứ tự khởi động hệ thống Linux (**ghi chú:** Thứ tự khởi động của các hệ điều hành Linux là khác nhau giữa **sysvinit**, **systemd** và **upstart** )

### • Thứ tự khởi động chung như sau:

1. Bật nguồn điện trên máy tính(Power on)
2. BIOS tải lên (LƯU Ý: trên các hệ thống Linux hiện đại, sử dụng EFI)
3. BIOS tìm và chuyển giao cho 'boot sector' của ổ đĩa chính (hoặc ổ đĩa được chọn)
4. Boot sector cung cấp MBR (Master Boot Record), nằm trong 512 byte đầu tiên của ổ đĩa hoạt động (hoặc ở đầu tiên, tùy thuộc vào quá trình cài đặt) (có thể là đĩa mềm, USB hoặc ổ đĩa cứng bất kỳ)
5. Boot loader được thực thi
6. Boot loader LILO/GRUB/GRUB2 bắt đầu
7. BƯỚC NHẬP DỮ LIỆU NGƯỜI DÙNG - tùy thuộc vào cấu hình boot loader, người dùng có thể chọn từ menu các loại khởi động/kernel hoặc cho phép sử dụng mặc định
8. Kernel Linux được đọc và thực thi
9. Khởi tạo thiết bị, tải các module và initial RAM disk (initrd)
10. File hệ thống gốc (Root filesystem) được gắn mount vào Linux
11. Chương trình init được tải lên (và trở thành PID đầu tiên - /sbin/init)
12. /etc/inittab được đọc và các tập lệnh runlevel tương ứng được chạy
  - Debian/Ubuntu - /etc/init.d/rc#
  - Red Hat/CentOS - /etc/rc.d/init.d/rc.sysinit
13. Các module được chỉ định trong các tập lệnh init được tải lên
14. Kiểm tra file hệ thống gốc
15. Các file hệ thống cục bộ còn lại được gắn mount vào hệ điều hành Linux
16. Thiết bị mạng được khởi động
17. Các filesystems từ xa được gắn mount vào hệ điều hành (nếu được cấu hình)



18. init process đọc lại **/etc/inittab** và chuyển đến runlevel mặc định được chỉ định và thực thi các tập lệnh tương ứng
19. Thực thi các tập lệnh **runlevel** theo thứ tự số (#dịch vụ(service) cho runlevel được chỉ định trong **/etc/inittab**)
20. Các phiên tty được tải theo thứ tự được liệt kê trong **/etc/inittab**
21. Màn hình nhắc đăng nhập hiển thị, hệ thống Linux đã sẵn sàng để đăng nhập
  - Các **runlevel** của **sysvinit**
    - 0 - Tắt hệ thống (**shutdown**)
    - 1 - Chế độ người dùng đơn lẻ **single user**
    - 2 - Chế độ nhiều người dùng **multi-user** (không có mạng hoặc file system từ xa)
    - 3 - Chế độ nhiều người dùng đầy đủ **multi-user** (có mạng)
    - 4 - Không sử dụng **unused**
    - 5 - **X11** (chế độ **desktop** đồ họa)
    - 6 - Khởi động lại **reboot**

## 101.2 Khởi động hệ thống (Quá trình khởi động - systemd và upstart - Từ bật nguồn điện trên máy tính đến vào chế độ dòng lệnh của hệ điều hành)

- **Systemd**
  - Mặc định trên hầu hết các bản phân phối Linux hiện đại, **VD**: Từ Linux Centos 7 trở lên.
  - Được thiết kế để thay thế phương pháp **sysvinit** quản lý khởi động và khởi tạo hệ thống
  - Hiệu suất cao hơn và cho phép hoạt động chạy song song
  - Ít các thành phần phụ thuộc (dependencies) hơn
  - Ưu tiên và thứ tự ưu tiên dịch vụ
  - Giảm thời gian khởi động
  - Khái niệm 'units' thay thế các tập lệnh khởi động (init scripts) của sysvinit
    - Dịch vụ có tên 'name.service' thay vì chỉ là tên tập lệnh khởi động (init scripts)
- Các thành phần chính cần biết
  - **systemd** - quản lý các dịch vụ (services) và hệ thống (systems)
  - **systemctl** - lệnh chính để điều khiển start (khởi động)/stop (dừng)/restart (khởi động lại)/status (trạng thái dịch vụ).
  - **systemd-analyze** - hiển thị thông tin về hiệu suất khởi động hệ thống, cho phép theo dõi và thông tin cho việc gỡ lỗi
  - **logind** - thay thế consolekit, hỗ trợ các trình quản lý X Windows
  - **console** - dịch vụ console daemon, thay thế các virtual terminals
  - **journald** - hệ thống ghi nhật ký (logging system), sử dụng ghi nhật ký nhị phân (mặc dù có thể được thay thế bằng các dịch vụ ghi nhật ký khác (other logging daemons))
  - **networkd** - dịch vụ hỗ trợ mạng trong Linux
- **CHÚ Ý**: các runlevel trong hệ thống systemd được chỉ định trong 'component groups (cgroup)'
  - Cho phép quản lý các thành phần phụ thuộc (dependencies) và gắn nhãn, tiết kiệm bộ nhớ và phân bổ tài nguyên

- **/usr/lib/systemd/system**
  - Các 'units' được cài đặt
- **/etc/systemd/system** - các units có ưu tiên hơn các units khác và được kiểm soát bởi quản trị hệ thống sysadmin
- **/run/systemd/system** - các units được tạo ra trong thời gian chạy và có thể có ưu tiên hơn các units không phải thời gian chạy (hoặc được cài đặt)
- Dịch vụ và phần mở rộng
  - **.service** - dịch vụ hệ thống
  - **.swap** - thiết bị device hoặc file swap
  - **.socket** - socket IPC
  - **.target** - unit (một hoặc một nhóm)
  - **.snapshot** - trạng thái đã lưu của quản lý systemd
  - **.slice** - các units được nhóm lại trong một cấu trúc phân cấp để quản lý các tiến trình
  - **.timer** - bộ hẹn giờ
  - **.mount** - điểm gắn kết (mount point) trên file system (cục bộ hoặc từ xa)
  - **.automount** - tự động gắn (automount) trên file system
  - **.scope** - tiến trình được tạo ra ở bên ngoài
  - **.path** - file hoặc thư mục trên file system
  - **.device** - thiết bị được sử dụng bởi system kernel
- **Các runlevels trong system HDH Linux**
  - 0 - **poweroff.target** - Tắt hệ thống- hệ điều hành.
  - 1 - **rescue.target** - Chế độ đơn người dùng(**single user**)/cứu hộ(**rescue shell**).
  - 2 - **multi-user.target** - Không có giao diện đồ họa, nhưng cho phép chạy mạng(full network) và đa người dùng(**multi-user**).
  - 3 - **multi-user.target** - Không có giao diện đồ họa, nhưng đầy đủ mạng và đa người dùng.
  - 4 - **multi-user.target** - Không có giao diện đồ họa, nhưng đầy đủ mạng (**full network**) và đa người dùng(**multi-user**)..
  - 5 - **graphical.target** - Môi trường đồ họa đầy đủ(**full graphical desktop**), đa người dùng(**multi-user**)..
  - 6 - **reboot.target** - Khởi động lại hệ thống- hệ điều hành(**reboot**).
- **Service target dependencies** - là cách thức xác định các phụ thuộc giữa các đơn vị dịch vụ (**service units**) hoặc nhóm đơn vị dịch vụ trên hệ thống Linux. Nó quy định thứ tự khởi động và dừng các dịch vụ, đảm bảo rằng các dịch vụ phụ thuộc sẽ được khởi động trước và các dịch vụ liên quan sẽ được khởi động sau.
  - Phương pháp chỉ định phụ thuộc(**dependencies**) giữa các đơn vị (**units**) hoặc các nhóm(**groups**) đơn vị trên hệ thống.
  - **Requires**
    - Chỉ khi yêu cầu được đáp ứng, các '**wanted**' units khác sẽ được khởi động.
  - **Wants**
    - Chỉ được thực thi sau khi các 'requires' units hoàn thành.
  - **LƯU Ý:** Xem ví dụ tại /usr/lib/systemd/system/graphical.target.
- Thay đổi quá trình khởi động

- `/etc/inittab` không được đọc (và không còn được sử dụng trong systemd nữa).
- Thay vào đó, một 'default.target' được khởi động (được liên kết với 'default' target hiện tại - đa người dùng, đồ họa, vv).
- Sau đó, các 'requires' và 'wants' được theo dõi trong suốt tiến trình.
- **runlevel**
  - Hiển thị runlevel hiện tại (và runlevel trước đó, nếu có).
- **systemctl get-default**
  - Hiển thị runlevel target mặc định hiện tại.
- **systemctl set-default [new.target]**
  - Sẽ đặt **runlevel target** mặc định thành giá trị chỉ định.
  - Tạo liên kết từ 'default.target' đến target được chỉ định.
    - **Ví dụ** phương pháp thủ công - `ln -sf /usr/lib/systemd/system/multi-user.target /etc/systemd/system/default.target`
- **systemctl list-units --type=target**
  - Liệt kê tất cả các system targets đang hoạt động
- **systemctl isolate [runlevel.target]**
  - Cho phép bạn thiết lập mức độ chạy của hệ thống mà không thay đổi mức độ mặc định
  - **Ví dụ:** `systemctl isolate graphical.target` sẽ chuyển hệ thống Linux sang chế độ đồ họa desktop đầy đủ
- **upstart**
  - Quá trình khởi động(boot) và tắt(shutdown) bất đồng bộ
  - Thiết kế để dễ triển khai vì nó duy trì tính tương thích sysinit (và có thể chạy các tập lệnh **sysvinit scripts** không thay đổi)
  - Được coi là dễ mở rộng

### 101.3 Thay đổi Runlevels/Boot Targets và Tắt (shutdown) hoặc khởi động(reboot) lại hệ thống Linux (systemd, sysvinit và upstart)

- **sysvinit**
  - **runlevel**
    - Hiển thị mức độ chạy hiện tại (và mức độ chạy runlevel trước đó, nếu có)
  - **init**
    - Có thể chạy để thay đổi mức độ chạy(runlevel) của hệ thống Linux
    - Ví dụ: `init 5` sẽ thay đổi mức độ chạy của hệ thống từ mức độ chạy hiện tại sang mức độ chạy 5 hoặc chế độ đồ họa, nếu được cấu hình
- **telinit**
  - Tương tự như lệnh `init` (lưu ý: Đây là một lệnh cũ không được sử dụng bình thường)
- Mục tiêu mặc định
  - Được thay đổi trong file `/etc/inittab`
    - Ví dụ: `id:3:initdefault` sẽ thiết lập giá trị mức độ chạy mặc định là 3 hoặc **multi-user target**
- Khởi động lại(reboot) hoặc tắt(shutdown) hệ thống Linux

- **reboot** - sẽ khởi động lại hệ thống
- **shutdown** - sẽ khởi động lại HOẶC tắt hệ thống
  - **-h** - tắt hệ thống (shutdown)
  - **-r** - khởi động lại hệ thống
  - **-P** - tắt nguồn (nếu ACPI có sẵn)
  - **-c** - hủy bỏ quá trình tắt máy tính
  - **-k** [message] - gửi thông điệp 'wall' đến người dùng đã đăng nhập
  - **[option]** - xác định thời điểm thực hiện quá trình tắt máy tính (theo giây, phút, thời gian cụ thể hoặc 'now')
- **halt**
  - **-f** (hoặc **--force**) - chỉ dừng hệ thống Linux, không tắt nguồn
  - **-p** (hoặc **--poweroff**) - tắt nguồn sau khi dừng hệ thống (nếu ACPI có sẵn)
  - **-w** (hoặc **--wtm-only**) - chỉ GHI LOG quá trình tắt máy mà không thực hiện tắt máy hoặc khởi động lại
  - **--verbose** - hiển thị thông tin chi tiết hơn để khắc phục sự cố
- **systemd**
  - Ngoài các phương pháp tắt/khởi động lại(**shutdown/reboot**) của sysvinit, bạn cũng có thể sử dụng **systemctl**
  - **systemctl halt** - dừng hệ thống Linux, không tắt nguồn
  - **systemctl poweroff** - dừng hệ thống và tắt nguồn (nếu ACPI có sẵn)
  - **systemctl reboot** - dừng hệ thống và khởi động lại
- **upstart**
  - Hỗ trợ tất cả các phương pháp của sysvinit nhưng không hỗ trợ các phương pháp của systemd
- **wall**
  - Cho phép bạn gửi thông điệp đến tất cả người dùng đã đăng nhập, giới hạn trong 20 dòng văn bản
  - **Ví dụ:** **wall** "Hệ thống này sẽ tắt máy sau năm phút - hãy kết thúc công việc và lưu lại file của bạn"
  - **LƯU Ý:** thông điệp này sẽ xuất hiện trên tất cả terminal, thường ghi đè hoặc làm gián đoạn văn bản trên terminal

## 102.1 Thiết kế Layout Ổ cứng (Layout và Ví dụ Cài đặt)

- **Cài đặt trên một ổ cứng duy nhất**
  - Linux được cài đặt trên một ổ cứng vật lý duy nhất, bao gồm một hoặc nhiều phân vùng(partitions)
- **Cài đặt trên nhiều ổ cứng**
  - Linux được cài đặt trên nhiều ổ cứng vật lý, mỗi ổ cứng gồm một hoặc nhiều phân vùng
- Quy ước đặt tên thiết bị (đối với ổ cứng SATA hoặc SCSI cục bộ)
  - **/dev/sda** - ổ cứng vật lý đầu tiên
  - **/dev/sda2** - ổ cứng vật lý đầu tiên, phân vùng thứ hai
- Các loại thiết bị khác

- hda (ổ cứng IDE)
    - /dev/hdc2 - ổ cứng IDE thứ ba, phân vùng thứ hai
  - scd (đĩa CDROM)
    - /dev/scd0 (đĩa CDROM đầu tiên)
      - ổ CDROM không có phân vùng
  - Bố cục hệ thống file(Linux filesystem layout) Linux
    - Mọi thứ trong Linux đều là một file
    - Mặc dù bạn có nhiều files/thư mục, bất kỳ thư mục nào (với một số ngoại lệ) có thể được gắn mount trên bất kỳ ổ đĩa/phân vùng nào
  - Lược đồ chung về gắn mount filesystem trong Linux
    - / - filesystem gốc(root) được gắn mount trên một thiết bị lưu trữ(Vd: ổ cứng sata, IDE, SCSI, vv..)/phân vùng và chứa tất cả các thư mục khác (một số trong số đó thường được gắn mount trên các phân vùng hoặc ổ đĩa khác).
    - /var - các files log, files/thư mục chia sẻ(share), thông tin thời gian chạy, file dữ liệu nhị phân
    - /home - chứa thư mục home của người dùng users trong Linux
    - /opt - thường là nơi cài đặt ứng dụng, phần mềm của bên thứ ba
    - /boot - chứa cấu hình của trình khởi động(boot loader) và các kernel files (trong các hệ thống Linux cũ, phân vùng(partition) đầu tiên được dành cho 512mb đầu tiên trên ổ đĩa đầu tiên)
    - swap - phân vùng dành riêng cho bộ nhớ ảo (swap)
  - Filesystem layout
    - Tạo bố cục phân vùng lưu dữ liệu ổ cứng cho mục đích cụ thể của bạn
  - Hệ thống hạn chế phát triển chia sẻ mở rộng không gian ổ lưu dữ liệu;
- Ví dụ:** bạn có thể tạo ra một phân vùng riêng cho thư mục /home để người dùng copy files hoặc cài đặt phần mềm/ứng dụng không gây chiếm dụng lượng làm đầy phân vùng gốc(root partition) (việc này giúp hệ thống Linux chạy ổn định).

## 102.1 Thiết kế Bố trí Ổ cứng (Cơ bản về LVM và Swap)

- **LVM**
  - Logical Volume Manager (Quản lý Logical Volume)
  - Cho phép tạo ra 'nhóm' ổ đĩa hoặc phân vùng(partitions) có thể gom lại thành một hoặc nhiều filesystems
- **PV**
  - Physical Volume (Ổ đĩa vật lý) - điểm khởi đầu hoặc đơn vị lưu trữ cơ bản
  - Tương ứng với một ổ đĩa hoặc phân vùng trên hệ thống Linux
    - Có thể là cục bộ trên hệ thống (như /dev/sdb1) hoặc một thiết bị lưu trữ SAN (storage area network)
- **VG**
  - Volume Group (Nhóm Ổ đĩa)
  - Kết hợp một hoặc nhiều PV để tạo thành một nhóm lưu trữ có sẵn
    - PE - Physical Extents (Các đơn vị vật lý)

- Được sử dụng để phân chia dễ dàng không gian lưu trữ của một PV
- **LV**
  - **Logical Volume** (Ổ đĩa Logic)
  - Mỗi nhóm ổ đĩa có thể được chia thành một hoặc nhiều Logical Volume, mỗi Logical Volume có thể được định dạng với một loại filesystem cụ thể và sau đó được gắn mount vào hệ điều hành
- Ưu điểm của **LVM**
  - **Linh hoạt** - bạn có thể tăng kích cỡ của một **Logical Volume**, chỉ bằng cách điều chỉnh kích thước, lấy thêm dung lượng(space) từ nhóm ổ đĩa(**volume group**), nếu nhóm ổ đĩa đã đầy dữ liệu, bạn có thể cấp thêm ổ đĩa vật lý cho nhóm ổ đĩa, và thiết đặt nó vào Logical Volume(s)
  - **Snapshots** - 'sao lưu điểm thời gian' trạng thái của bất kỳ Logical Volume nào có thể được sử dụng để sao lưu, khôi phục, kiểm tra, dịch chuyển, v.v. mà không ảnh hưởng đến Logical Volume 'trực tiếp'
- **Swap** (bộ nhớ ảo)
  - **Phân vùng swap** - một phân vùng dành riêng được định dạng đặc biệt làm dung lượng lưu trữ cho swap (thường được thực hiện trong quá trình cài đặt, nhưng cũng có thể thực hiện bất kỳ lúc nào bạn thêm ổ đĩa hoặc phân vùng)
  - **Files** - file được tạo ra đặc biệt và được sử dụng trên **filesystem** để làm dung lượng cho swap cho hệ thống Linux

## 102.2 Cài đặt Boot Manager (Làm việc với GRUB Legacy)

- **GRUB (phiên bản cũ)**
  - **Grand Unified Boot Loader** (Trình khởi động thống nhất toàn diện)
  - **/boot**
    - 'Thư mục khởi động' cho GRUB
    - **Kernel** (và các files liên quan) được lưu trữ ở đây
    - File cấu hình nhân (**kernel config**)
    - **Kernel initrd** (ramdisk) file
    - Thông báo khởi động mặc định
    - **Module 'symvers'** - File phiên bản chứa tất cả các ký hiệu của kernel các module
    - File bản đồ hệ thống (**system map**)
    - Kernel chính (**vmlinuz**)
  - **/boot/grub**
    - Thư mục cấu hình **GRUB** phiên bản cũ
      - Các 'stage' khác nhau cho quá trình khởi động
      - **menu.lst/grub.conf**
        - Các tùy chọn **GRUB** và tạo menu các lựa chọn hiển thị khi khởi động hệ thống Linux
        - Các thiết lập quan trọng:
          - **default=[#]** - xác định tùy chọn menu sẽ được khởi động lên, nếu không có lựa chọn khác
          - **timeout=[#]** - thời gian, tính bằng giây, bạn có để chọn một tùy chọn khác



- **Các phần (mục menu)**
  - **title [description]** - mô tả dựa trên tiêu đề của mục menu
  - **root [(hd#,#)]** - số thiết bị ổ đĩa cứng và phân vùng của ổ gốc (nơi **kernel** được lưu trữ)
  - **kernel [/boot/vmlinuz-# options root=LABEL=/ options]** - đường dẫn đến kernel, đọc các tùy chọn trên kernel, label và các tùy chọn khởi động khác được truyền cho kernel khi hệ thống Linux khởi động
  - **initrd [/boot/initrd#.img]** - đường dẫn đến initial RAM disk image cho mục menu đã chọn
- Các thay đổi trên danh sách menu sẽ được áp dụng trong lần khởi động hệ thống tiếp theo và có thể được nhìn thấy
- Chỉnh sửa **mục menu** khởi động bằng cách nhấn **'e'** trên danh sách menu trong quá trình hiển thị của trình khởi động
- Dòng **lệnh GRUB** có thể truy cập bằng cách nhấn **'c'** trong menu
- Cài đặt lại GRUB
  - **grub-install [device]**
    - Thiết bị(device) có thể được xác định bằng đường dẫn ổ đĩa (**/dev/hda** hoặc **/dev/hd0**)
- **grub**
  - Ứng dụng sẽ hiển thị dấu nhắc **GRUB**
  - Cho phép bạn tìm hiểu môi trường (tìm đường dẫn và số thiết bị(ổ cứng) chính)

## 102.2 Cài đặt trình quản lý khởi động (Làm việc với trình khởi động GRUB2)

- **GRUB2**
  - **Grand Unified Boot Loader** (Chương trình khởi động thống nhất cho các HĐH thiết bị - đã cải tiến)
  - **/boot**
    - Thư mục **'boot'** cho **GRUB2**
    - Kernel (và các files liên quan) được lưu trữ ở đây
      - File cấu hình kernel (**kernel config**)
      - File kernel initrd (**ramdisk**)
      - Thông báo khởi động mặc định
      - Module **'symvers'** - file phiên bản chứa tất cả các ký hiệu của các module nhân
      - File bản đồ hệ thống (**system map**)
      - Nhân (kernel) chính (**vmlinuz**)
  - **/boot/grub**
    - Thư mục tương thích với GRUB, đôi khi chứa **splash image** cho menu khởi động
  - **/boot/grub2**
    - **grub.cfg**
      - File menu được tạo ra, chứa cấu hình phức tạp hơn là sự kết hợp của nhiều files khi sử dụng tiện ích **grub2-mkconfig**
      - Những thay đổi chỉnh sửa ở đây sẽ bị mất trong lần chạy kế tiếp của tiện ích **grub2-mkconfig**

- **/boot/grub2/efi**
  - Các phần mở rộng **EFI** cho GRUB2 trên các bản phân phối Linux hiện đại (nằm ngoài phạm vi của kỳ thi LPIC-1)
- **/boot/grub2/fonts**
  - **Font(s)** chữ mặc định (và có thể là font tùy chỉnh) cho menu GRUB2
- **/boot/grub2/themes**
  - Giao diện (theme) cho menu đồ họa GRUB2
- **/etc/grub.d**
  - Các files được đánh số, khi tiện ích **grub2-mkconfig** chạy, tất cả các files này được nối lại (theo thứ tự số) để tạo ra file **/boot/grub2/grub.cfg (menu)**
- **/etc/default/grub**
  - File chứa các cấu hình cụ thể cho GRUB2 (thời gian chờ, tùy chọn mặc định, tùy chọn cho dòng lệnh)
    - **LƯU Ý:** Những tùy chọn này áp dụng cho hành vi chung của GRUB2 và không áp dụng cho bất kỳ mục **menu** nào
- Chỉnh sửa mục menu khởi động bằng cách nhấn **'e'** trên danh sách menu trong quá trình hiển thị của trình khởi động
- Dòng lệnh Grub có thể truy cập bằng cách nhấn **'c'** trong menu
- **grub2-mkconfig** (có thể là grub-mkconfig)
  - Dùng để tạo ra file cấu hình GRUB (menu)
  - **-o [/path/grub.cfg]** - đầu ra tới vị trí và file đã được chỉ định
- **grub2-install**
  - Cài đặt GRUB2 vào vị trí chỉ định (bản ghi khởi động chính (**drive master boot record**) hoặc đường dẫn đầy đủ được chỉ định trên thiết bị (ổ cứng))
  - **[[--boot-directory=/path/to/install] [/dev/drive]]**

## 102.3 Quản lý Thư viện chia sẻ (Tổng quan về Thư viện chia sẻ (Shared Library) và Cấu hình các vị trí của Thư viện)

- **Thư viện chia sẻ**
  - Các hàm có thể được sử dụng lại bởi các ứng dụng khác thông qua việc **'liên kết'** với chúng trong file thư viện
  - Bất kỳ tên file thư viện chia sẻ nào đều có đuôi mở rộng **'.so'** (có thể cũng chứa thông tin phiên bản sau phần mở rộng .so)
  - **'so'** trong tên đuôi mở rộng đại diện cho 'shared object' (đối tượng chia sẻ)
- **/lib**
- **/usr/lib/\*** (**/usr/lib64/\*** - cho thư viện 64 bit)
- **/usr/local/lib/\***
- **/usr/share/\***
  - Vị trí phổ biến cho các thư viện chia sẻ của hệ thống Linux
- **Liên kết mềm**
  - Rất nhiều lần, các phiên bản cụ thể của thư viện được liên kết với một tên chung
  - **Ví dụ - libstdc++.so > libstdc++.so.6.0.19**
- **Liên kết**
  - **Liên kết tĩnh** - ứng dụng chứa một bản sao đầy đủ của thư viện được sử dụng



- **Ưu điểm** - kiểm soát phiên bản (ứng dụng sẽ có phiên bản thư viện chính xác mà nó có thể cung cấp với các giao diện đã biết)
- **Nhược điểm** - kích thước (tăng kích cỡ của ứng dụng cần phân phối vì nó phải chứa các bản sao đầy đủ của nhiều thư viện được liên kết tĩnh), nâng cấp (ứng dụng cần được biên dịch lại và liên kết lại mỗi khi có yêu cầu cần phải cập nhật thư viện)
- **Liên kết động** - ứng dụng sử dụng thư viện từ bên ngoài bằng cách sử dụng **'stubs'** (điểm neo), thư viện được cài đặt trên hệ thống (HĐH) chính, nhưng không được cài đặt cùng với ứng dụng
  - **Ưu điểm** - kích thước (ứng dụng nhỏ hơn vì các thư viện tách biệt với nó), nâng cấp (không cần biên dịch lại hoặc liên kết lại khi/nếu thư viện được cập nhật)
- **ld.so**
  - Mỗi khi một ứng dụng cần sử dụng một thư viện chia sẻ, dịch vụ **ld.so** được gọi
  - Còn được gọi là **'dynamic linker'** (bộ liên kết động)
- **ldd [tên\_file]**
  - Hiển thị danh sách tất cả các thư viện mà ứng dụng thực thi động đã được chỉ định yêu cầu (và xem xét xem chúng có hiện diện không)
- **ldconfig**
  - Cấu hình các liên kết chạy thời gian của bộ liên kết động (tạo liên kết và lưu vào bộ nhớ cache các thư viện chia sẻ mới nhất được tìm thấy)
    - Tìm kiếm theo những gì được chỉ định trên dòng lệnh
    - Tìm kiếm theo những gì được chỉ định trong file **ld.so.conf**
  - Tạo ra file **/etc/ld.so.cache**, một tệp nhị phân liệt kê các thư viện trên hệ thống như phát hiện được (được sử dụng bởi **ld.so** khi cần thiết)
  - Thường chạy sau khi cập nhật hệ thống (tự động hoặc thủ công) hoặc sau khi cài đặt ứng dụng bên thứ ba (có thể đã thêm các thư viện)
- **/etc/ld.so.conf**
  - Danh sách tiêu chuẩn các vị trí file thư viện được liệt kê trong file này
- **LƯU Ý:** Nếu có một dòng tồn tại, **'include ld.so.conf.d/\*.conf'**, nó cũng sẽ đọc tất cả các files được chỉ định, tìm kiếm các thư viện trong những vị trí đó cũng (phương pháp mới hơn)
- Định dạng các mục đơn giản là đường dẫn đến thư mục mà bộ liên kết cần quét
  - **Ví dụ** - **/usr/lib/qt-3.3/lib**
- **LD\_LIBRARY\_PATH**
  - Một biến môi trường có thể chỉ định cho người dùng hoặc toàn bộ hệ thống, xác định một đường dẫn (theo định dạng thông thường, được phân tách bằng dấu hai chấm (:)), mà các ứng dụng **ld.so** và **ldconfig** có thể tìm kiếm thư viện
  - **Ví dụ** - **export LD\_LIBRARY\_PATH=/opt/myapp/lib:/opt/anotherapp/lib**
    - Sẽ tạo ra một biến môi trường phiên làm việc khi được thực thi trên dòng lệnh cho phép **ldconfig** tìm thấy các thư viện trong các đường dẫn đã chỉ định (ngoài các giá trị trong **/etc/ld.so.conf**)

## 102.4 Sử dụng Quản lý Gói Debian (Kho chứa(Repositories) và Quản lý Gói(Package))

- **Kho chứa**

- Là nơi lưu trữ các gói cài đặt & cập nhật của Linux tại các máy chủ từ xa mà bạn có thể tải xuống và cài đặt các gói sử dụng các chương trình quản lý gói.
- **/etc/apt/sources.list**
  - file chứa một hoặc nhiều địa chỉ URL kho chứa từ xa để tải xuống và cài đặt gói cho máy tính Linux của bạn
- **/etc/apt/sources.list.d**
  - Thư mục chứa các files riêng lẻ xác định một hoặc nhiều địa chỉ URL kho chứa từ xa để tải xuống và cài đặt gói
  - Mỗi files có thể chứa các dòng cho cả gói (**file nhị phân**) và mã nguồn gói
  - **Ví dụ - deb** <http://archive.canonical.com/ubuntu> **[nickname] partner**
    - Dòng trên sẽ thêm kho chứa từ xa '**partner**' cho bản phân phối Ubuntu được chỉ định (nickname), xem lệnh **apt-get update** để lấy danh sách gói vào bộ nhớ cache hệ thống Linux cục bộ

- **apt**

- Viết tắt của **Advanced Package Tool**
- Một bộ công cụ (**apt-\***) để làm việc với các gói Debian đầy đủ (và các thành phần phụ thuộc (**dependencies**)) của chúng

- **apt-get**

- Trình quản lý gói có thể cài đặt, cài đặt lại hoặc gỡ bỏ các gói và tất cả các thành phần phụ thuộc của chúng cùng một lúc
- **update** - đọc danh sách gói từ kho chứa từ xa (làm mới)
- **upgrade** - nâng cấp hệ thống Linux hiện có và phần mềm đã cài đặt bằng các phiên bản mới được báo cáo có sẵn trong kho chứa được cấu hình
- **install** [tên gói] - cài đặt gói được chỉ định (không phải là file **.deb**, mà là tên gói chung) cùng với tất cả các thành phần phụ thuộc của nó (nếu có sẵn và không gây xung đột)
- **dist-upgrade** - nâng cấp tất cả các gói có sẵn trong kho chứa trên hệ thống cùng một lúc trong phiên bản **Ubuntu/Debian** tiếp theo
- **remove** [tên gói] - gỡ bỏ các files của gói nhưng các files và thư mục cấu hình sẽ được giữ nguyên
- **purge** [tên gói] - gỡ bỏ các files của gói VÀ tất cả các files và thư mục cấu hình liên quan
- **-d** (hoặc **--download-only**) [tên gói] - chỉ tải xuống gói **.deb** và lưu nó trong **/var/cache/apt/archives**

- **apt-cache**

- Cho phép bạn tương tác với bộ nhớ cache các gói có sẵn từ tất cả các kho chứa được cấu hình (sau khi hệ điều hành được cập nhật bằng lệnh **apt-get update** ở trên)
- **search** [giá trị] - tìm kiếm các gói phù hợp với giá trị được chỉ định
- **show** [tên gói] - hiển thị thông tin có sẵn về gói được chỉ định
- **showpkg** [tên gói] - thông tin kỹ thuật bổ sung về gói (phụ thuộc, dịch vụ, v.v.)

- **aptitude**

- tương tự như **apt** (trong thực tế, nó sử dụng apt trong **nền background**), có giao diện đồ họa

- **synaptic**

- tương tự như apt (trong thực tế, nó sử dụng apt trong **nền background**), có giao diện đồ họa

## 102.4 Sử dụng Chương trình Quản lý Gói Debian

### (Sử dụng Tiện ích dpkg)

- **Gói Debian (\*.deb)**

- Chứa một ứng dụng, các file cấu hình mặc định, tài liệu và hướng dẫn hệ thống về cách/nơi mà các thành phần của ứng dụng sẽ được cài đặt trên hệ thống cũng như một danh sách đầy đủ các thành phần phụ thuộc

- **dpkg**

- Ứng dụng chịu trách nhiệm cài đặt, gỡ bỏ và truy vấn các gói Debian trên hệ thống
- **--info [tên gói]** - hiển thị nội dung của gói
- **--status [tên gói]** - hiển thị một phiên bản rút gọn của tùy chọn **--info**
- **-i (hoặc --install) [tên gói]** - sẽ cài đặt gói được chỉ định trên hệ thống
  - **CHÚ Ý:** Nếu thiếu các thành phần phụ thuộc (**dependencies**), quá trình cài đặt sẽ thất bại, các thành phần phụ thuộc nên được cài đặt trước hoặc cùng lúc trong cùng một lệnh (tùy chọn, sử dụng một trong các công cụ quản lý gói bên dưới)
- **-r (hoặc --remove) [tên gói]** - gỡ bỏ các files của gói nhưng các files và thư mục cấu hình sẽ được giữ nguyên
  - **-P (hoặc --purge) [tên gói]** - gỡ bỏ các files của gói VÀ bất kỳ files và thư mục cấu hình có liên quan
- **-L (hoặc --listfile) [tên gói]** - liệt kê tất cả các files đã được cài đặt với gói được chỉ định
- **-S (hoặc --search) [tên file]** - tìm kiếm file đã được cài đặt được chỉ định trong cơ sở dữ liệu gói cho tất cả các đề cập đến file hoặc chuỗi đã chỉ định
- **--force-reinstreq [tên gói]** - cho phép bạn buộc gỡ bỏ một gói được đánh dấu là yêu cầu cài đặt lại
- **--force-depends [tên gói]** - sử dụng với **-i** hoặc **--install** để buộc việc cài đặt một gói với các phụ thuộc bị thiếu
- **--force-conflicts [tên gói]** - sử dụng với **-i** hoặc **--install** để buộc việc cài đặt một gói với các thành phần phụ thuộc xung đột (conflict)
  - **LƯU Ý:** Việc bắt buộc cài đặt theo cách này có thể làm cho hệ thống Linux của bạn trở thành không thể sử dụng được, hãy chắc chắn rằng bạn đã hiểu những gì bạn đang làm
- **-l (hoặc --list) [tên gói hoặc tên một phần]** - liệt kê các gói đã được cài đặt khớp với giá trị đã chỉ định
- **dpkg-reconfigure [tên gói]**

- Một số ứng dụng sẽ bao gồm một công cụ để cấu hình ứng dụng trong quá trình cài đặt (**máy chủ mail**, **máy chủ cơ sở dữ liệu**, v.v.), nếu bạn cần chạy lại công cụ cấu hình đó cho một gói, bạn có thể chạy nó bằng lệnh này

## 102.5 Sử dụng Quản lý Gói RPM và YUM

### (Quản lý Gói YUM)

- **Repositories (kho lưu trữ)**

- Các hệ thống máy chủ từ xa được sử dụng để chứa các gói cài đặt Linux, mà bạn có thể tải xuống và cài đặt các gói từ sử dụng bất kỳ công cụ quản lý gói nào dưới đây

- **/etc/yum.conf**

- File cấu hình yum mặc định, có thể chứa một số định nghĩa kho lưu trữ

- **/etc/yum.repos.d**

- Chứa các files '**\*.repo**' độc lập chứa vị trí và tùy chọn cho kho lưu trữ từ xa
- Xem Phụ lục A để biết ví dụ về một file kho lưu trữ từ thư mục này (kho lưu trữ bên thứ ba)
  - **LƯU Ý:** trong nhật ký cấu hình kho lưu trữ, vị trí được xác định là '**baseurl**' (liên kết trực tiếp) đến một kho lưu trữ hoặc '**metalink**', sẽ trả về danh sách các trang web mirror để sử dụng
  - **LƯU Ý:** chỉ các kho lưu trữ được đặt thành '**ENABLED=1**' trong file repo mới được lệnh yum sử dụng

- **/var/log/yum.log**

- File mặc định nhật ký log giao dịch yum (được định nghĩa trong file **/etc/yum.conf**)

- **yum**

- **The Yellowdog Updater Modified**

- Đặt lên trên đầu của RPM, ngoài việc cài đặt gói, “**yum**” quản lý các thành phần phụ thuộc(**dependencies**)

- **update/upgrade** - sẽ đọc cấu hình các kho lưu trữ và bộ nhớ **cache**, và nâng cấp tất cả các ứng dụng hệ thống đã được cài đặt (**lưu ý** – việc này KHÔNG thực hiện nâng cấp phiên bản của ứng dụng/phần mềm)
- **--enablerepo [repo name]** - kích hoạt bật một kho lưu trữ bị vô hiệu hóa chỉ trong giao dịch hiện tại
- **install [packagename]** - cài đặt gói, cùng với tất cả các thành phần phụ thuộc cần phải có của gói
  - **--downloadonly** - khi thêm vào lệnh install, chỉ tải xuống gói được chỉ định
  - **/var/cache/yum/[architecture]/[version]/base/packages** - thư mục nơi gói sẽ được tải xuống
- **remove [packagename]** - gỡ bỏ gói, KHÔNG gỡ bỏ các thành phần phụ thuộc đã được cài đặt trước đó
  - Để gỡ bỏ các thành phần phụ thuộc không sử dụng (không nhất thiết liên quan đến việc gỡ bỏ một gói, nhưng bất cứ thứ gì không được sử dụng theo thời gian), bạn có thể thực hiện lệnh
    - **yum autoremove**
- **-y** - thêm vào bất kỳ giao dịch yum nào, sẽ bỏ qua bước xác nhận (xác nhận Y/N trước khi tiếp tục)
- Có thể được sử dụng trực tiếp để tải xuống một gói thay vì cài đặt nó bằng cách trả lời câu hỏi 'Y/N' với 'd'
  - **/var/cache/yum/[architecture]/[version]/base/packages** - thư mục nơi gói sẽ được tải xuống
- **yumdownloader**
  - Cho phép bạn chỉ tải xuống một gói, nhưng cung cấp các chức năng bổ sung nhiều hơn so với thực hiện lệnh **yum** (xem trên)
  - **--source** - chỉ tải xuống RPM nguồn
  - **--urls** - hiển thị URL của các files mà không tải xuống
  - **--destdir** - cho phép bạn chỉ định thư mục để lưu trữ việc tải xuống gói
  - **--resolve** - bao gồm các gói phụ thuộc

## 102.5 Sử dụng RPM và quản lý YUM Package

### (sử dụng lệnh RPM)

- Cơ sở dữ liệu RPM

- File cơ sở dữ liệu chứa các gói và các files liên quan đã được cài đặt trong Linux

- /var/lib/rpm

- Thư mục cơ sở dữ liệu RPM

- Gói rpm

- Có đuôi file mở rộng \*.rpm

- Chứa các files nguồn hoặc nhị phân, hướng dẫn xây dựng (nếu là nguồn), tên phần mềm, thông tin phiên bản, ngày xây dựng và hệ thống xây dựng, mô tả, các thành phần phụ thuộc và kiểm tra checksum

- Source RPM

- File nguồn cần thiết để xây dựng và cài đặt gói đã chỉ định

- rpm

- Nguyên bản, là Trình quản lý Gói RedHat

- --rebuilddb - xây dựng lại cơ sở dữ liệu RPM cục bộ trong thư mục /var/lib/rpm

- **Lưu ý:** chỉ người dùng root mới có thể thực hiện lệnh này (hoặc người dùng có đặc quyền sudo)

- Cài đặt, gỡ bỏ, truy vấn, v.v. có thể được thực hiện với tên gói dài hoặc tên gói ngắn

- -q [tên gói dài/ngắn] - hiển thị tất cả các gói phù hợp với các giá trị đã chỉ định đã được cài đặt

- -qi [tên gói dài/ngắn] - thông tin chi tiết về các giá trị đã chỉ định đã được cài đặt

- -ql [tên gói dài/ngắn] - danh sách tất cả các files đã được cài đặt cùng với gói

- -qip hoặc -qlp [gói.rpm] - hiển thị các chi tiết thích hợp về một file KHÔNG được cài đặt với tên đã đặt

- **--changelog [tên gói dài/ngắn]** - hiển thị lịch sử thay đổi cho gói đã chỉ định (phải được sử dụng với -q)
- **-qc [tên gói dài/ngắn]** - hiển thị tất cả các files cấu hình cho gói đã chỉ định
- **-qRp [gói.rpm]** - hiển thị các yêu cầu của file rpm đã chỉ định
- **qf [/đường/dẫn/của/file/kiểm tra]** - sẽ hiển thị gói chứa file và các files phụ thuộc
- **-K (hoặc --checksig) [gói.rpm]** - xác minh (sử dụng các checksum của gói) cho gói đã chỉ định
  - Kết quả **"NOT OK"** có thể có nghĩa là một vấn đề, nhưng cũng có thể chỉ đơn giản là chưa nhập khóa chữ ký (**signature key**)
- **--import [URL của file khóa cần nhập]** - nhập key file đã chỉ định từ người duy trì gói, chạy lại lệnh xác minh sau khi key được cài đặt
- **-i (hoặc --install) [package.rpm]** - cài đặt gói đã chỉ định (thao tác này sẽ thất bại nếu các gói phụ thuộc không được đáp ứng)
- **-v (hoặc --verbose) [package.rpm]** - hiển thị thông tin chi tiết (sử dụng cùng với các tùy chọn khác)
- **-h (hoặc --hash) [package.rpm]** - trong quá trình hoạt động cài đặt, sẽ hiển thị thanh trạng thái bằng ký tự '#' (sử dụng cùng với các tùy chọn khác)
  - **Ví dụ: rpm -ivh telnet-0.17-60.el6.x86\_64.rpm**
    - Sẽ cài đặt gói đã được chỉ định, với hiển thị dưới lệnh trên thanh trạng thái bằng ký tự '#' hiển thị lên
- **--force** - tùy chọn được thêm vào trong quá trình cài đặt hoặc gỡ bỏ để bắt buộc cài đặt/gỡ bỏ ngay cả khi có xung đột với các files hoặc gói hiện có
- **--replacefiles** - tùy chọn được thêm vào trong quá trình cài đặt (an toàn hơn một chút), chỉ thay thế các files trùng lặp
- **--nodeps** - tùy chọn được thêm vào trong quá trình cài đặt hoặc gỡ bỏ để bắt buộc cài đặt/gỡ bỏ gói mà không cần các phụ thuộc đã chỉ định



- **Lưu ý:** Việc Buộc cài đặt mà không giải quyết được các xung đột giữa các files hoặc thành phần phụ thuộc có thể làm hệ thống Linux của bạn trở nên không thể sử dụng được.

- **V [tên gói dài/ngắn]** - xác minh tính toàn vẹn của một gói, sẽ so sánh cài đặt và cấu hình hiện tại của gói với cài đặt và cấu hình mặc định của gói

- Không có đầu ra nghĩa là không tìm thấy sự khác biệt
- Nếu gói không được cài đặt, một thông báo lỗi sẽ xuất hiện để chỉ ra điều đó
- Những sự khác biệt tìm thấy sẽ được hiển thị dưới dạng đầu ra gồm chín ký tự

- Theo thứ tự - **(S)** thay đổi kích thước, **(M)** thay đổi chế độ, **(5)** MD5 sum khác nhau, **(D)** không khớp số lượng thiết bị chính/minor, vấn đề đường dẫn liên kết **(L)**, **(U)** thay đổi chủ sở hữu người dùng, **(G)** thay đổi chủ sở hữu nhóm, **(T)** thay đổi thời gian sửa đổi, **(P)** thay đổi thiết đặt quyền

- **LƯU Ý:** Chỉ hiển thị **NẾU** hạng mục đó đã thay đổi

- **Ví dụ - rpm -V vsftpd**

- Đầu ra - **S.5....T. c /etc/vsftpd/ftpusers**

- Trong trường hợp này, file cấu hình **(c) /etc/vsftpd/ftpusers** đã thay đổi về kích thước (S), MD5 sum(5) và thời gian sửa đổi (T) kể từ khi được cài đặt với gói, các hạng mục khác đã giữ nguyên

- **-Va** - xác minh **TẤT CẢ** các gói đã cài đặt trên hệ thống
- **-Vac** - xác minh **TẤT CẢ** các files cấu hình gói đã cài đặt trên hệ thống (thường được lưu lại để so sánh sau này)
- **-U (hoặc --upgrade) [tên gói]** - sẽ cài đặt hoặc nâng cấp gói được chỉ định, đồng thời gỡ bỏ các phiên bản trước nếu tồn tại
- **-F (hoặc --freshen) [tên gói]** - chỉ nâng cấp gói đã được cài đặt nếu đã tồn tại, nếu chưa được cài đặt, không có gì sẽ được thực hiện
- **-e (hoặc --erase) [tên gói dài/ngắn]** - xóa gói đã được chỉ định

- **LƯU Ý:** Nếu sử dụng tên ngắn và tìm thấy nhiều kết quả phù hợp cho tên ngắn đó, một thông báo lỗi sẽ xuất hiện và tên dài phải được sử dụng **HOẶC --allmatches** phải được thêm vào lệnh (**PHẢI CÂN THẬN KHI THỰC HIỆN LỆNH NÀY**)



- **rpm2cpio**

- Trích xuất nội dung của gói \*.rpm và chuyển đến đầu ra tiêu chuẩn (hoặc có thể chuyển hướng đến một file)
- Tiện ích cpio có thể được sử dụng để trích xuất các files, điều này cho phép cài đặt trên các hệ thống Linux không phụ thuộc vào RPM
- Ví dụ - **rpm2cpio telnet-0.17-60.el6.x86\_64.rpm > telnet.cpio**
  - Sẽ giải nén gói rpm được chỉ định và tạo ra bản lưu trữ **cpio 'telnet.cpio'**
  - Sau đó, file có thể được trích xuất bằng **cpio -idmv < telnet.cpio**, hiệu quả 'cài đặt' nó trong thư mục cục bộ (xem đầu ra)

## Chủ đề 103 - Lệnh GNU và Unix 103.1 Làm việc trên Dòng lệnh (Shell, Sourcing và các Ký tự đặc biệt)

- **Bash**

- **Shell Linux** mặc định được sử dụng để diễn giải các lệnh và **kịch bản(script)** trong hệ thống Linux của chúng ta
- **Các shell khác**
  - **ash, c-shell, ksh, zsh**
- **Các shell đặc biệt**
  - **/bin/false, /etc/nologin** (cảnh báo – Điều quan trọng cần nhớ là **/bin/false** và **/etc/nologin** không tồn tại theo mặc định trong hệ thống Linux. Tuy nhiên, bạn có thể tạo file này và cấu hình hệ thống để sử dụng chúng để ngăn người dùng thường đăng nhập vào Linux. Cần cẩn thận trước khi thực hiện)
- Thực thi kịch bản(**script**)
  - **Bash** đọc nội dung của file kịch bản được chỉ định và, nếu các quyền thích hợp được thiết lập, các lệnh bên trong nó được thực thi một cách tuần tự (giống như bạn gõ trên dòng lệnh **bash shell**), nhưng trong một '**shell**' 'mới'
  - Bất kỳ biến nào được thiết lập trong không gian thực thi của shell trong quá trình thực thi kịch bản sẽ bị mất (gọi là '**ra khỏi phạm vi**') sau khi kịch bản hoàn tất và thoát ra, và trở lại dấu nhắc lệnh

- Ví dụ - `./myscript.sh`

- Sẽ thực thi kịch bản `myscript.sh` từ thư mục làm việc hiện tại (dấu `./` trong lệnh của chúng ta)

- Sourcing kịch bản(script)

- Bash đọc nội dung của file kịch bản được chỉ định và, nếu các quyền thích hợp được thiết lập, các lệnh bên trong nó được thực thi một cách tuần tự (giống như bạn gõ các câu lệnh trên dòng lệnh), nhưng trong môi trường **shell HIỆN TẠI**
- Cho phép giữ nguyên tất cả các thiết lập trong shell đó (**ví dụ:** biến môi trường) sau khi kịch bản hoàn thành

- Ví dụ - `source myscript.sh` hoặc `. myscript.sh`

- `/etc/profile`

- Tập lệnh cấu hình cho toàn bộ hệ thống Linux, có tác động ảnh hưởng đến môi trường của mỗi người dùng (miễn là họ sử dụng **shell bash**)
- Đó là một tập lệnh 'được **source**' và được thực thi mỗi khi người dùng đăng nhập(**login**) thông qua một shell đăng nhập

- `/etc/bashrc`

- Đây là một tập lệnh cấu hình cho toàn bộ hệ thống Linux, ảnh hưởng đến môi trường của mỗi người dùng (miễn là họ sử dụng **shell bash**).
- Đây là một tập lệnh 'được **source**' và được thực thi mỗi khi người dùng đăng nhập thông qua một shell đăng nhập HOẶC khi họ thực thi một phiên không đăng nhập (xem bên dưới).

- `./bash_profile`

- Đây là một tập lệnh 'được source' khác được thực thi khi người dùng đăng nhập thông qua một shell đăng nhập, nhưng chỉ ảnh hưởng đến môi trường của người dùng đăng nhập (nó nằm trong thư mục chủ(**home**) của họ, được chỉ định bằng ký tự viết tắt `~` trong đường dẫn).
- Đôi khi nó có thể được gọi là `~/profile` hoặc `~/bash_login`; tuy nhiên, nếu có bất kỳ file nào NỔI BẬT HƠN ĐƯỢC TÌM THẤY bổ sung vào `~/bash_profile`, chỉ `~/bash_profile` mới được thực thi.

- Thông thường, nó được sử dụng để thiết lập các biến môi trường cụ thể cho người dùng (ví dụ: thêm thư mục home cục bộ `~/bin` vào biến `PATH`).
- `./bashrc`
  - Đây là một tập lệnh 'được source' khác, thường được gọi bởi tập lệnh `./bash_profile`.
  - Nếu có, nó sẽ được sử dụng để 'source' tập lệnh `/etc/bashrc`.
  - Thông thường được sử dụng để tùy chỉnh giao diện dòng lệnh, phím tắt bàn phím, bí danh (`aliases`) lệnh, v.v.
- `~/.bash_logout`
  - Đây là một tập lệnh 'được source' khác, được thực thi khi lệnh `logout` hoặc `exit` được thực hiện.
- Ví dụ về thứ tự phiên đăng nhập vào Linux của user:
  1. Người dùng đăng nhập.
  2. `/etc/profile` (được source).
  3. `~/.bash_profile` của người dùng (được source).
  4. `~/.bashrc` của người dùng (được source từ `~/.bash_profile`).
  5. Người dùng thực hiện công việc của họ theo nhu cầu sử dụng.
  6. Người dùng đăng xuất bằng lệnh `logout` hoặc `exit`.
  7. `~/.bash_logout` của người dùng (được source).
- Phiên 'không đăng nhập' (Non-login session):
  - Khi người dùng root hoặc người dùng khác root sử dụng lệnh `su` để trở thành người dùng khác (hoặc người dùng root), mặc định không tải toàn bộ môi trường của người dùng đó.
- Thực hiện các thiết đặt môi trường
  - Hoàn thành lệnh
    - Gõ một lệnh nằm trong đường dẫn hệ thống Linux hoặc bất kỳ đường dẫn hợp lệ nào, một phần của câu lệnh, và nhấn phím `<TAB>` sẽ tự động hoàn thành phần còn lại của lệnh hoặc đường dẫn.
  - Các ký tự đặc biệt

- `~` - thư mục gốc của người dùng (ví dụ - `/home/user`)
- `\` - ký tự thoát
- `$` - xác định một biến
- `?` - ký tự đại diện cho một ký tự duy nhất
- `*` - ký tự đại diện cho 0 đến n ký tự
- `&` - Chuyển một tiến trình vào chế độ chạy nền background
- `&&` - thực thi lệnh thứ hai chỉ khi lệnh đầu tiên thành công (ví dụ - `/bin/true && echo 'It Works'`)
- `|` - đường ống(**pipe line**), dùng để chuyển đầu ra của lệnh sang lệnh khác
- `||` - thực thi lệnh thứ hai chỉ khi lệnh đầu tiên không thành công (ví dụ - `/bin/false || echo 'Got Here'`)
- `;` - thực thi nhiều lệnh trên cùng một dòng
- `[]` - xác định một phạm vi ký tự (số hoặc chữ)
- `>` - chuyển hướng đầu ra chuẩn (**redirect standard**) vào một tập file
- `<` - chuyển hướng đầu vào chuẩn (**standard input**) từ một chương trình
- **Mã thoát lệnh**
  - Mỗi lệnh thành công hoặc thất bại và cung cấp một mã thoát lệnh bạn có thể xem sau đó
  - `echo $?` (0 cho thành công, 1 hoặc cao hơn cho thất bại)

## 103.1 Làm việc trên Dòng lệnh (Các Biến môi trường - Hiện thị, Thiết lập và Sử dụng)

### • Biến môi trường

- '**Phím tắt**' cho phép bạn chỉ định đường dẫn, lệnh, bí danh (**aliases**) hoặc thông tin khác với một ký hiệu dễ sử dụng và dễ nhớ.
- **env**

- Lệnh để liệt kê tất cả các biến môi trường được thiết lập cho phiên làm việc hiện tại TRỪ các thiết lập của **shell**

- **set**

- Dùng để xem các thiết lập của shell hoặc biến shell nào cho phiên làm việc
- Có thể dùng để bật hoặc tắt các tùy chọn shell

- **-o [option]** – ON (bật) tùy chọn
- **+o [option]** – OFF (tắt) tùy chọn

- **Các tùy chọn có sẵn**

- **vi hoặc emacs** - kiểu bàn phím cho dòng lệnh
- **hashall** - mặc định BẬT, cho phép bảng băm lệnh và vị trí được sử dụng lặp lại
- **history** - mặc định BẬT, cho phép đọc **HISTFILE** để tìm file lịch sử (xem bên dưới)
- **monitor** - chạy các tiến trình nền trong một nhóm khác và thông báo cho console khi chúng hoàn thành
- **noclobber** - mặc định TẮT, không cho phép chuyển hướng '>' ghi đè lên một file đã tồn tại
- **noexec** - mặc định TẮT, các files kịch bản(**script**) sẽ chạy kiểm tra cú pháp khi bật trên tất cả các lệnh khi chạy
- **notify** - công việc kết thúc thông báo ngay lập tức trên **console** thay vì trong lần thực thi tiếp theo của lệnh '**jobs**'
- **verbose** - hiển thị lệnh ra terminal/màn hình trước khi chúng được thực thi

- **shopt**

- Dùng để xem bất kỳ thiết lập của shell hoặc biến shell nào cho phiên làm việc

- **PATH**

- Một biến môi trường chứa danh sách các thư mục được phân tách bằng dấu hai chấm (:) được tìm kiếm (theo thứ tự) để tìm các file thực thi lệnh (nghĩa là các file có quyền thực thi)

- Đường dẫn tuyệt đối - `/home/user/command`
- Đường dẫn tương đối - `./command`
- Thông thường biến môi trường **PATH** được thiết lập ban đầu (cho hệ thống Linux) trong file `/etc/profile`
- **PATH** có thể được thay đổi trên dòng lệnh hoặc các files cấu hình khác chứa các tập lệnh thực thi (ví dụ: `/home/user/.bashrc`)
  - Ví dụ (trong file `/etc/profile`) - `export PATH=/sbin:/bin:/usr/bin:/usr/local/bin`
    - Thiết lập các vị trí thư mục 'bin' chung cho các files thực thi trên hệ thống Linux
  - Ví dụ về **PATH** được thay đổi trong file `/home/user/.bashrc` - `export PATH=$PATH:/home/user/bin`
    - Lấy giá trị hiện tại (không bao gồm **\$PATH** sẽ ghi đè lên giá trị hiện tại, làm mất nó) và thêm vào cuối `/home/user/bin` để người dùng có thể đặt các files tập lệnh riêng trong thư mục đó và thực thi mà không cần cung cấp đường dẫn tuyệt đối đến chúng
- HISTFILE
  - Lệnh **history** sử dụng giá trị này để hiển thị danh sách các lệnh đã chạy trước đó
  - Theo mặc định, giá trị của nó cho bất kỳ người dùng nào sẽ là `/home/user/.bash_history`
  - Các biến điều khiển việc sử dụng lịch sử của bash shell là:
    - **HISTCMD** - chỉ mục của lệnh hiện tại
    - **HISTCONTROL** - khi được thiết lập thành **ignorespace**, bất kỳ lệnh nào đi trước bởi một khoảng trắng sẽ KHÔNG được ghi lại trong file lịch sử, khi được thiết lập thành **ignoredups**, hai dòng liên tiếp giống nhau sẽ có một dòng bị bỏ qua
    - **HISTFILESIZE** - số dòng có thể chứa các lệnh trước đó (mặc định là 500)
  - Thay đổi biến môi trường
    - Tất cả các biến có thể được thay đổi theo hai cách:
      1. Ghi đè (`export VARIABLE=newValue`)
      2. Thêm vào (`export VARIABLE=$VARIABLE:newValue`)
    - Sử dụng biến môi trường trên dòng lệnh

- **echo** [**\$** + **tên biến**] - sẽ hiển thị giá trị hiện tại
- **Ví dụ - echo \$HOME**
  - Sẽ hiển thị thư mục gốc của người dùng hiện tại đã được thiết lập trong biến đó
- **Ví dụ - cd \$HOME** (hoặc **cd ~**)
  - Sẽ đổi thư mục hiện tại của shell thành thư mục gốc của người dùng hiện tại (xem danh sách các **ký tự đặc biệt** trong shell ở phần trên)

## 103.2 Xử lý luồng văn bản bằng Các lệnh Bộ lọc trong Textutils Package (sort, nl, wc, expand, cut, paste, join, uniq, head và tail)

- **sort**
  - Sẽ sắp xếp (theo thứ tự số hoặc chữ cái, mặc định bắt đầu từ cột 0) mỗi dòng trong file được chỉ định
  - **sort** [**tên file**] - hiển thị một sắp xếp mặc định của mỗi dòng (từ trên xuống dưới) lên trên màn hình
  - **-k[#]** [**tên file**] - thực hiện sắp xếp ở đầu của trường phân tách thứ hai (ký tự phân tách mặc định là dấu cách hoặc tab)
  - **-n** [**tên file**] - sắp xếp file theo thứ tự số (chú ý tới toàn bộ số, không chỉ cột 0)
- **nl**
  - Đánh số dòng trong một file (hoặc luồng đầu vào), có thể là tất cả các dòng hoặc chỉ các dòng có dữ liệu
  - **nl** [**tên file**] - sẽ đánh số tất cả các dòng có dữ liệu (mặc định)
  - **-ba** [**tên file**] - sẽ đánh số tất cả các dòng, kể cả có dòng trống
- **wc**
  - Từ viết tắt của '**word count**', nhưng có thể làm nhiều hơn thế
  - **-l** - số dòng
  - **-w** - số từ
  - **-c** - số ký tự (byte)

- Được sử dụng như một bổ sung cho đầu ra của một lệnh khác

- **Ví dụ - `cat /etc/passwd | wc -l`**

- Lệnh này sẽ hiển thị nội dung của file `/etc/passwd`, lọc nó qua lệnh `wc` và hiển thị số dòng trong file

- **expand**

- Chuyển đổi các tab trong một file thành một số ký tự dấu cách cụ thể (mặc định là 8)
  - `-t (hoặc --tabs=[#]) [#]` - chuyển đổi các tab thành # số dấu cách đã chỉ định

- **cut**

- Lệnh cho phép trích xuất dữ liệu trường hoặc cột từ một vị trí cụ thể trong file được chỉ định
  - `-c [#][-#] [filename]` - sẽ chỉ hiển thị cột (hoặc phạm vi cột) từ tệp được chỉ định
  - `-d[delimiter]` - thiết lập ký tự phân cách để sử dụng khi xử lý các trường (mặc định - **TAB**)
  - `-f [field1[,field2[,field3]]] [filename]` - xác định số trường (được xác định bằng ký tự phân cách) để hiển thị từ file được chỉ định

- **paste**

- Kết hợp hai files lại với nhau mà không loại bỏ bất kỳ dữ liệu nào (phong cách nối tiếp)

- **join**

- Kết hợp hai files lại với nhau nhưng loại bỏ các trường lặp lại (giống với cơ sở dữ liệu) - dựa trên trường đầu tiên (trường khóa)
  - `-t[character]` - sử dụng ký tự được chỉ định làm bộ phân cách trường (ký tự phân cách)

- **uniq**

- Cho phép bạn trích xuất chỉ các dòng dữ liệu '**duy nhất**' từ file
  - Áp dụng sắp xếp vào file trước khi chạy lệnh này (đặc biệt là trên các file lớn)



- **-u [filename]** - chỉ in ra các dòng duy nhất trong file
- **-d [filename]** - in ra ví dụ của mỗi dòng bị lặp lại trong file
- **-D [filename]** - in ra TẤT CẢ các trường hợp của các dòng bị lặp lại trong file
- **head**
  - Tương tự như lệnh **cat** vì nó sẽ hiển thị nội dung của một file, nhưng chỉ hiển thị một số dòng nhất định từ đầu (mặc định là **10**)
  - **-n [#] [filename]** - hiển thị 'n' số dòng, bắt đầu từ đầu, của tệp được chỉ định
- **tail**
  - Mặc dù tương tự như lệnh **cat** vì nó sẽ hiển thị nội dung của một file, nhưng nó là đối lập với lệnh **head**, chỉ hiển thị một số dòng nhất định từ cuối file (mặc định là **10**)
  - **-n [#] [filename]** - hiển thị 'n' số dòng, bắt đầu từ cuối, của file được chỉ định
  - **-f [tên file]** - hiển thị file được chỉ định, (mặc định là 10) các mục cuối cùng để bắt đầu theo dõi file, và sau đó sẽ 'theo dõi' đầu ra của file, giúp có thể theo dõi xem các giá trị bổ sung vào hiển thị khi chúng được ghi vào file.

## 103.2 Xử lý Luồng Văn bản Bằng Các Bộ Lọc (Các Lệnh Trong Gói Textutils: **split**, **cat**, **od**, **pr**, **fmt**, **tr**, **sed**, **more** và **less**)

- **split**
  - Cho phép bạn chia một file có một số bản ghi thành nhiều file chứa 'phần' của dữ liệu gốc
    - **-a [#]** - khi tạo các file chia, đặt tên chúng là 'x#' (ví dụ: **-a 5** sẽ tạo file 'aaaaa', các files tiếp theo theo thứ tự chữ cái)
    - **-b [#][b/k/m]** - các files mới chứa số lượng **byte/kilobyte/megabyte** được chỉ định
    - **-l [#]** - các files mới chứa số lượng byte hoặc dòng được chỉ định
    - **-l [#]** - các files mới chứa số lượng dòng được chỉ định

- **cat**

- Hiển thị nội dung của một file từ đầu đến cuối trên đầu ra tiêu chuẩn

- **tac**

- Cùng chức năng nhưng ngược lại với lệnh **cat**, hiển thị nội dung của một file từ cuối lên đầu trên đầu ra tiêu chuẩn (**standard output**)

- **od**

- Cho phép hiển thị an toàn nội dung của một file nhị phân trên **đầu ra/terminal** tiêu chuẩn (mà không cần phục hồi với lệnh '**clear**' hoặc '**reset**' của terminal)

- **-a** - hiển thị file nhị phân 'đặt tên'

- **-d** - định dạng thập phân

- **-f** - định dạng số chấm động

- **-o** - định dạng bát phân

- **-x** - định dạng thập lục phân

- **pr**

- Thường được sử dụng để '**định dạng**' một file nguồn hoặc các files văn bản khác để in ra màn hình

- Thêm tiêu đề với ngày của '**job**', tên file và phân trang (số trang) ở đầu trang

- **--columns=[#]** - định dạng dữ liệu văn bản trong file thành số cột được chỉ định để in

- **-a** (hoặc **--across**) - in các cột qua ngang thay vì dọc (sử dụng kết hợp với **--columns**)

- **-d** (hoặc **--double-space**) - tăng gấp đôi khoảng cách giữa các dòng

- **-h** (hoặc **--header=[text]**) - tùy chỉnh nội dung tiêu đề, thay thế tên file bằng **[text]** được chỉ định

- **-t** (hoặc **--omit-header**) - loại bỏ tiêu đề khỏi đầu ra

- **fmt**

- Cũng được sử dụng để định dạng các files cho việc in, tuy nhiên, nó chỉ giới hạn trong việc xếp dòng các dòng đầu ra dài hơn
- Thường được sử dụng kết hợp với lệnh **pr** (xem ở trên)
- **-[#]** - chiều rộng mong muốn để chia mỗi dòng
- **-s** - chia các dòng dài (hơn 50 ký tự) mà không cần điền lấp đầy
- **Ví dụ - `fmt -40 mytext.txt | pr --columns=2`**
  - Sẽ chia mỗi dòng thành 40 ký tự (hoặc gần nhất có thể là một số nguyên) và chuẩn bị in với tiêu đề trong hai cột
- **tr**
  - Biên dịch, cho phép bạn thay đổi một hoặc nhiều ký tự (khớp với một mẫu) trong một file hoặc luồng
  - Không dùng để thay thế toàn bộ từ hoặc cụm từ
  - **"[giá trị-từ-dài]" "[giá trị-đến-dài]" [tên-file]** - cung cấp một hoặc nhiều ký tự thông qua một dải (như **'a-f'**) để thay thế bằng các ký tự khác trong một dải (như **'1-6'**)
  - Không thể được sử dụng để chạy trực tiếp trên một file, file cần xử lý phải được coi như một luồng
  - **Ví dụ - `tr 'A' 'a' < myfile.txt`**
    - Sẽ chuyển đổi tất cả các chữ cái in hoa **'A'** thành chữ thường **'a'** trong luồng được cung cấp bằng cách chuyển hướng nội dung của **'myfile.txt'** làm đầu vào cho lệnh
  - **`[:upper:][:lower:] < [tên-file]`** - chuyển đổi tất cả các chữ in hoa thành chữ thường trong luồng đầu vào (có thể đảo ngược)
- **sed**
  - Một chương trình(lệnh) sử dụng để chỉnh sửa luồng rất mạnh mẽ có thể xử lý và thực hiện các hành động phức tạp trên luồng văn bản (có thể là một khóa học riêng về nó. Ở đây chúng ta tập trung vào các trường hợp sử dụng căn bản cho kỳ thi LPIC-1)
  - Sử dụng 'biểu thức chính quy'(regular expressions), thường được gọi là **'regex'**, để xác định những gì cần thực hiện các hoạt động trên luồng (và **regex** có thể cực kỳ

phức tạp, nhưng linh hoạt, **regex** là một chủ đề khác có thể làm một khóa học riêng)

- **s/** - tùy chọn cơ bản phổ biến nhất, nó thay thế các chuỗi ký tự tiếp theo
- **[giá trị(s)]/** - các giá trị cần tìm kiếm trong luồng
- **[giá trị(s)]/** - các giá trị để thay thế giá trị(s) đầu tiên
- **g** - khi kết thúc lệnh bằng 'g', thay thế TẤT CẢ các trường hợp của giá trị đầu tiên bằng giá trị thứ hai, nếu bỏ qua, chỉ thay thế sự xuất hiện đầu tiên của giá trị đầu tiên bằng giá trị thứ hai
- Ứng dụng **sed** có thể sử dụng chuỗi các tùy chọn để thực hiện nhiều thay thế
  - **Ví dụ - sed 's/a/A/g' filename.txt**
    - Sẽ thay thế tất cả các trường hợp của chữ cái 'a' viết thường bằng chữ cái 'A' (bỏ 'g' ở cuối để chỉ thay thế lần đầu tiên)
  - **Ví dụ - sed 's/me/ME/g' ; 's/ME/YOU/g' filename.txt**
    - Sẽ thay thế tất cả các trường hợp của chuỗi 'me' viết thường bằng chuỗi 'ME' và ngay sau đó, thay thế 'ME' bằng 'YOU' từ 'filename.txt'
- **-e** - Có thể sử dụng để kết nối nhiều bộ lọc với nhau (thay vì ký tự ';')
- **CHÚ Ý:** nếu giá trị đầu tiên hoặc thứ hai không chứa khoảng trắng, không cần sử dụng dấu nháy đơn; tuy nhiên, sử dụng dấu nháy đơn luôn hoạt động, đây là một thói quen tốt
- **-f [scriptlistfile] [filename]** - có thể sử dụng một file chứa nhiều dòng tùy chọn để thực hiện trên tên file đã chỉ định
- **CHÚ Ý:** **sed** xem file như một luồng và đầu ra vào đầu ra chuẩn, nó không thay đổi file chính nó
- **-pg** - chỉ hiển thị các dòng đã thay đổi (thay vì 'g' hiển thị tất cả)
- **-n** (hoặc **--quiet**) – chặn việc in 'pattern space', có thể ngăn các dòng xuất hiện nhiều lần trong đầu ra
- **[#]{-/,}[#]/** - cung cấp một số dòng duy nhất, danh sách hoặc khoảng dòng để thực hiện thay thế

- **more**

- Cho phép phân trang qua các file văn bản (một màn hình mỗi lần)
- **-d** - hiển thị lời nhắc 'nhấn space để tiếp tục' hoặc 'q để thoát' trên mỗi màn hình (thay vì phải sử dụng **CTL-C** hoặc biết phải nhấn 'q')
- **-num [#]** - chỉ định kích thước màn hình (theo số dòng), giá trị mặc định là **50** hoặc kích thước của terminal (nếu nó là terminal GUI cung cấp thông tin đó)
- **-p** - xóa màn hình trước khi bắt đầu hiển thị màn hình đầu tiên
- **CHÚ Ý:** Chỉ cho phép di chuyển về phía trước qua một file (không thể quay lại màn hình trước đó)

- **less**

- Tương tự như '**more**'; tuy nhiên, cho phép di chuyển qua file văn bản theo chiều ngược lại (các giá trị trước đó), nó cho phép di chuyển nhanh hơn vì không đọc toàn bộ file trước khi bắt đầu hiển thị.
- **CHÚ Ý:** các tùy chọn được liệt kê dưới đây áp dụng trong quá trình hiển thị file (phím tắt bàn phím)
  - **space** - di chuyển đến màn hình văn bản tiếp theo
  - **q** - thoát khỏi lệnh **less**
  - **d[#]** - cuộn qua số dòng được chỉ định (nếu không chỉ định số, mặc định là một màn hình đầy đủ)
  - **b[#]** - cuộn ngược lại số dòng được chỉ định (nếu không chỉ định số, mặc định là một màn hình đầy đủ)

### 103.3 Thực hiện việc Quản lý Cơ bản của file (Các Lệnh và Thuật ngữ cần biết - ls, cd, pwd, mkdir, file, globbing, touch, stat, cp và mv)

- **ls**

- '**liệt kê**' thư mục, files hoặc cả hai
- **-l** - liệt kê chi tiết, bao gồm quyền truy cập, chủ sở hữu, kích cỡ và ngày/giờ tạo và sửa file

- **-a** - liệt kê file, bao gồm cả file 'ẩn' (file bắt đầu bằng dấu .)
- **-d** - chỉ liệt kê thư mục mà không liệt kê các files trong đó
- **-i** - hiển thị số inode cho file hoặc thư mục
- **-h** - thêm định dạng dễ đọc cho kích thước và chi tiết của file
- **CHÚ Ý:** có thể kết hợp tất cả các tùy chọn

- **Ví dụ - ls -ailh /home/user**

- Sẽ hiển thị các file và thư mục trong /home/user, bao gồm cả các file ẩn, quyền truy cập, chủ sở hữu, nhóm chủ sở hữu, kích thước (dưới định dạng dễ đọc) và số inode

- **cd**

- Thay đổi thư mục làm việc
- Có thể sử dụng đường dẫn tuyệt đối, đường dẫn tương đối hoặc giá trị trong biến làm phần của đường dẫn
- **~** - một ký tự đặc biệt đại diện cho 'thư mục gốc' (home directory)

- **CHÚ Ý:** hãy cẩn thận, để đi đến thư mục gốc của bạn, sử dụng '**cd ~**', nhưng '**cd ~user**' sẽ là thư mục gốc của người dùng đó

- '**cd ~/tmp**' - thư mục con "**tmp**" trong thư mục gốc của bạn
- '**cd ~tmp**' - thư mục gốc của người dùng 'tmp' (lưu ý thiếu dấu /)

- **.** - một tham chiếu đến thư mục hiện tại (**cd .**)
- **..** - di chuyển lên một thư mục (**cd ..**)
- Có thể kết hợp trong đường dẫn tương đối

- **Ví dụ - cd ../../mydirectory**

- Lệnh trên sẽ di chuyển lên HAI thư mục trong cây thư mục và vào thư mục '**mydirectory**' (nếu tồn tại)

- **pwd**

- Hiện thị thư mục hiện tại mà bạn đang ở trong đó

- **mkdir**

- Tạo một thư mục (nếu chưa tồn tại)
- **-p** - tạo tất cả các thư mục trong đường dẫn chỉ định, nếu chúng chưa tồn tại
  - Ví dụ - **mkdir -p /home/user/dir1/dir2/dir3/dir4**
  - Sẽ tạo TẤT CẢ các thư mục trong đường dẫn chỉ định nếu chúng chưa tồn tại

- **file**

- Có thể xác định loại file của file được chỉ định
  - Ví dụ - **file myfile.txt**
    - Giả sử đó là một file văn bản, sẽ hiển thị **myfile.txt: ASCII text**
  - Có thể chạy đối với một tập hợp các file trong một thư mục, nó sẽ liệt kê loại cho mỗi file

- Ví dụ - **file /home/user/\***
  - Sẽ hiển thị tất cả các files trong /home/user và loại file của thư mục này

- **Globbering**

- Một ký tự hoặc toán tử đa dạng/mảng có khớp với một hoặc nhiều file dựa trên một mẫu đã xác định

- Ví dụ - **ls -al /usr/bin/.sh**

- Sẽ hiển thị các thuộc tính phù hợp của tất cả các file trong /usr/bin có đuôi '.sh' (bất kỳ số lượng ký tự trước đó)

- Ví dụ - **ls -al /usr/bin/.?h**

- Sẽ hiển thị các thuộc tính phù hợp của tất cả các file trong /usr/bin có đuôi '[một ký tự]h' (bất kỳ số lượng ký tự trước đó)

- Ví dụ - **ls -al /usr/bin/[a-f]\*.sh**

- Sẽ hiển thị các thuộc tính phù hợp của tất cả các file trong `/usr/bin/` bắt đầu với các ký tự từ 'a' đến 'f' với bất kỳ số lượng ký tự nào sau đó, kết thúc với đuôi file mở rộng `.sh`

- `touch`

- Tạo một file (nếu chưa tồn tại) hoặc cập nhật ngày/giờ cho file nếu nó đã tồn tại
- Thường được sử dụng để tạo một file mới mà một ứng dụng hoặc tiện ích ghi `logging` nhật ký cần có
- Có thể sử dụng với tên đường dẫn tuyệt đối hoặc tương đối
  - **CHÚ Ý:** bạn phải có quyền tạo/cập nhật file trong vị trí thư mục chỉ định
- `-t [ngày] [tên file]` - đặt ngày/giờ như đã chỉ định cho file
  - **Định dạng ngày/giờ** - `yyyymmddhhmm` (y=năm, m=tháng, d=ngày, h=giờ, m=phút)
- `-r [filethamchieu] [tên file]` - áp dụng ngày/giờ đã chỉ định trong 'filethamchieu' cho 'tên file' (hoặc nhiều file)
- Được sử dụng để xác định một 'điểm bắt đầu' cho các file mà không cần nhớ giá trị đó

- `stat`

- Hiển thị trạng thái file (hoặc file system)
- `-f (hoặc --file-system)` - hiển thị trạng thái file system thay vì trạng thái file
- `-t (hoặc --terse)` - hiển thị thông tin dưới dạng ngắn gọn

- `cp`

- Sao chép file và/hoặc thư mục từ một vị trí này đến vị trí khác
- `-d` - không copy các liên kết tượng trưng(symbolic links), chỉ sao chép liên kết(link)



- **-f** - ghi đè mà không hỏi (nếu file đã tồn tại ở điểm đến)
  - **-i** - hỏi trước khi ghi đè lên một file đã tồn tại
  - **-l** - tạo một liên kết cứng đến file gốc
  - **-s** - tạo một liên kết tượng trưng đến file gốc
  - **-r** (hoặc **-R**) - áp dụng đệ quy cho các tùy chọn cho các thư mục và các thư mục con trong đường dẫn được chỉ định
  - **-u** - chỉ cập nhật copy nếu file gốc mới hơn file đích hiện có (hoặc tạo file đích nếu chưa tồn tại)
  - **-x** - không bao gồm bất kỳ file/thư mục nào từ các filesystems khác (chia sẻ mạng)
  - **NOTE:** việc sao chép có thể được thực hiện với đường dẫn tuyệt đối hoặc tương đối
- **mv**
    - Di chuyển một file hoặc thư mục từ vị trí này sang vị trí khác (khác với **copy**(sao chép), **cp** tạo bản sao tại vị trí đích)
    - Có thể sử dụng với đường dẫn tuyệt đối hoặc tương đối
    - **-i** - kiểm tra xem đích đã tồn tại chưa, sẽ hỏi có ghi đè không
    - **-u** - không ghi đè file hoặc thư mục đích nếu nó mới hơn file hoặc thư mục gốc
    - **-f** - không hỏi về thay đổi mục là nhập tên thư mục (**LƯU Ý:** thường là mặc định trong các bản phân phối Linux)
    - **LƯU Ý:** cần cẩn thận khi di chuyển nội dung của một thư mục vào một thư mục khác
      - **Ví dụ - mv mydir1 mydir2**
        - **Việc này có thể có hai hành vi khác nhau:**
          - Nếu **'mydir2'** chưa tồn tại, nó sẽ đổi tên **'mydir1'** thành **'mydir2'**

- Nếu 'mydir2' đã tồn tại, nó di chuyển 'mydir1' VÀO TRONG 'mydir2'
- Ví dụ - mv mydir1/\* mydir2
- Lệnh này sẽ di chuyển file(s) trong 'mydir1' vào thư mục 'mydir2', trừ khi nó không tồn tại, sau đó nó sẽ báo lỗi

### 103.3 Thực hiện Quản lý File Cơ Bản (Các Lệnh và Thuật ngữ Cần Biết - dd, rmdir, rm và find)

- dd
  - Được sử dụng để tạo bản sao lưu hình ảnh, các files định dạng đặc biệt cho dung lượng swap, file ISO CD/DVD và các file khác
  - if=[file hình ảnh hoặc thiết bị] - sử dụng làm đầu vào (như một tệp hình ảnh như .img hoặc .iso) hoặc như một thiết bị (như /dev/sr0)
  - of=[file hình ảnh hoặc thiết bị] - sử dụng làm đầu ra (như một file hình ảnh như .img hoặc .iso) hoặc như một thiết bị (như /dev/sdc)
  - Ví dụ - dd if=/dev/sda of=/dev/sdb
    - Sẽ sao lưu toàn bộ ổ cứng /dev/sda vào ổ cứng backup /dev/sdb (giả sử /dev/sdb có cùng hoặc nhiều dung lượng hơn)
  - Ví dụ - dd if=/dev/sr0 of=/tmp/cdimage.iso
    - Sẽ lấy một bản sao hình ảnh(image) từ thiết bị CDROM tại /dev/sr0 và tạo một file hình ảnh ISO từ nó tại /tmp/cdimage.iso
  - Ví dụ - dd if=/dev/sda of=/tmp/mbrbkup.img count=1 bs=512
    - Sẽ sao lưu master boot record(bản ghi khởi động chính) từ /dev/sda bằng cách đọc một khối block (count) gồm 512 byte (bs)
    - Có thể phục hồi bằng cách đảo ngược (if và of)
- rmdir
  - Xóa một thư mục, nhưng chỉ khi nó RỎNG

- **-p** - xóa tất cả các thư mục trong một đường dẫn, miễn là TẤT CẢ chúng đều trống (ngoại trừ các thư mục trống khác)
- **--ignore-fail-on-non-empty** - cho phép xóa các thư mục có file hoặc thư mục không trống (đệ quy)

- **Lưu ý:** ít khi được sử dụng, hãy xem lệnh **rm** và các tùy chọn bên dưới

- **rm**

- Xóa file (hoặc thư mục)
- **-r** - xóa đệ quy file (và các thư mục con nếu mục tiêu là một thư mục)
- **-f** - không yêu cầu xác nhận khi xóa
- **-i** - yêu cầu xác nhận ĐỐI VỚI MỖI file/thư mục sẽ bị xóa

- **find**

- Phương pháp tốt nhất để 'tìm' các files dựa trên tên, loại hoặc các đặc điểm khác trên hệ thống Linux (mặc dù nó có thể tốn kém về hiệu suất CPU và I/O)
- **[đường dẫn bắt đầu]** - nơi bắt đầu tìm kiếm
- **-[tùy chọn]** - loại 'thứ' muốn tìm
- **[giá trị(s)]** - điều bạn đang cố gắng tìm (tên file, thư mục, tên được so sánh với **globbing**, v.v.)
- **Ví dụ - find / -name "messages"**
  - Sẽ tìm kiếm trên **filesystem**, bắt đầu từ thư mục **/**, đệ quy cố gắng tìm bất kỳ file hoặc thư mục nào có tên **'messages'**
- **Các tùy chọn bạn có thể tìm thấy:**
  - **group** - các tệp tin/thư mục thuộc nhóm được chỉ định
  - **user** - các files/thư mục thuộc về người dùng được chỉ định
  - **newer** - các files/thư mục mới hơn file được chỉ định
  - **name** - các files/thư mục có tên khớp hoàn toàn (bao gồm cả chữ hoa chữ thường) với tên được chỉ định

- **iname** - các files/thư mục có tên khớp (không phân biệt chữ hoa chữ thường) với tên được chỉ định
- **mtime** - các files/thư mục phù hợp với thời gian sửa đổi được chỉ định (cũ hơn +n ngày)
- **atime** - các files/thư mục phù hợp với số ngày kể từ lần truy cập cuối cùng được chỉ định
- **ctime** - các files/thư mục phù hợp với số ngày kể từ lần thay đổi cuối cùng được chỉ định
- **'-exec [cmd] {} [output] ;'**
  - Cho phép thực thi [cmd] trên tất cả các giá trị tìm thấy, được chuyển vào lệnh đó ({} là đại diện cho mỗi giá trị), với bất kỳ đầu ra liên quan nào
  - **Ví dụ:** `find /home/user/bin -name ".sh" -exec cp -f {} /home/user/backup/bin ;`
    - Đây sẽ tìm kiếm đệ quy thư mục `/home/user/bin` để tìm các tệp tin kết thúc bằng `.sh` và sao chép chúng, ghi đè mà không cần xác nhận nếu chúng đã tồn tại, vào thư mục `/home/user/backup/bin`
- **CHÚ Ý**
  - Có thể sử dụng nhiều giá trị
    - **Ví dụ -** `find /var -name ".log" -group "apache"`
      - Sẽ tìm kiếm đệ quy thư mục `/var` để tìm các file kết thúc bằng `.log` thuộc nhóm `'apache'`

### 103.3 Thực hiện Quản lý file Cơ Bản (Công cụ Lưu trữ và Nén)

- **tar**
  - Dùng để tạo các lưu trữ dữ liệu (nén file/thư mục) (ban đầu được phát triển để sao lưu file systems vào thiết bị băng gốc tape)

- Là một trong những tiện ích phổ biến nhất để sao lưu và (qua các đuôi file mở rộng), nén các files và thư mục
- Có thể sử dụng các loại nén khác nhau (**gzip** và **bzip**)
- **-c** - tạo lưu trữ
- **-t** - hiển thị nội dung của lưu trữ
- **-x** - giải nén nội dung của lưu trữ
- **NOTE:** Luôn cần một trong ba tùy chọn trên
- **-f** - tên của file cần tạo
- **-j** - nén/giải nén với bzip2 (mặc định không luôn có sẵn trong Linux, nhưng là phương pháp nén tốt nhất)
- **-z** - nén/giải nén với gzip (sẵn có mặc định và là phương pháp nén phổ biến nhất)
- **-v** - thông báo chi tiết (đầu ra này gần giống với lệnh `ls -al` khi tạo/giải nén/xem lưu trữ)
- **NOTE:** nhiều tùy chọn có thể được kết hợp trong một danh sách tùy chọn duy nhất
- **Ví dụ** - `tar -cvjf mybackup.tar.bz2 /home/user`
- Sẽ tạo một file có tên **mybackup.tar.bz2** trong thư mục hiện tại, chứa tất cả các file trong /thư mục người dùng, được nén bằng công cụ nén **bzip2**
- Một hoặc nhiều files hoặc thư mục có thể được nén trong một lưu trữ bằng cách truyền danh sách chúng với khoảng trắng giữa mỗi tên
- **cpio**
- Thường được sử dụng để nhận đầu vào từ một file hoặc lệnh khác và sau đó gửi các files đến đầu ra chuẩn (mặc định) hoặc một file nếu được chuyển hướng
- **-o (hoặc --create)** - chạy ở chế độ sao chép ra (mặc định, đầu ra chuẩn hoặc màn hình)
- **-O [file lưu trữ]** - tạo file được chỉ định thay vì sử dụng đầu ra chuẩn
- **Ví dụ** - `ls -al /home/user/bin/.sh | cpio -o > scriptbkup.cpio`

- Liệt kê tất cả các files trong /home/usr/bin kết thúc bằng **.sh** và chuyển đầu ra đó đến lệnh **cpio** chạy ở chế độ sao chép ra đến terminal và chuyển hướng vào file có tên **scriptbkup.cpio**
- **Ví dụ - `ls -al /home/user/bin/.sh | cpio -O scriptbkup.cpio`**
  - Tương tự như lệnh trên, nhưng tạo trực tiếp file lưu trữ được chỉ định mà không chuyển hướng đầu ra chuẩn
- **-d (hoặc `--make-directories`)** - tạo các thư mục chính nếu cần thiết
- **-i (hoặc `--extract`)** - trích xuất nội dung trong chế độ sao chép vào (mặc định, đầu ra chuẩn hoặc màn hình)
- **-I [file lưu trữ]** - sử dụng file lưu trữ chỉ định thay vì đầu vào chuẩn từ một lệnh hoặc file được chuyển hướng
- **-v** - thông báo chi tiết quá trình chạy của lệnh
- **gzip (và/hoặc `gzip2`)**
  - Tiện ích nén dữ liệu cho file và thư mục
  - **-r** - đệ quy, bao gồm tất cả các files và thư mục
  - **CHÚ Ý:** nén file gốc và thay thế bằng phiên bản nén mới
    - **Ví dụ - `gzip /home/user/myfile.txt`**
      - Lệnh này sẽ nén file **/home/user/myfile.txt**, để lại **/home/user/ myfile.txt.gz** (phiên bản chưa nén sẽ bị xóa)
- **gunzip**
  - Tiện ích trích xuất **gzip** cho các files nén
- **bzip2**
  - Tiện ích nén tốt hơn các tiện ích nén khác (về kích cỡ) cho file và thư mục
  - **CHÚ Ý:** nén file gốc và thay thế bằng phiên bản nén mới
    - **Ví dụ - `bzip2 /home/user/myfile.txt`**
      - Lệnh này sẽ nén file **/home/user/myfile.txt**, để lại **/home/user/ myfile.txt.bz** (hoặc **.bz2**) (phiên bản chưa nén sẽ bị xóa)
- **xz**
  - Tiện ích nén mới hơn cho file và thư mục
  - Hiệu suất tốt hơn so với **bzip2** (nhưng không nén tốt hơn)
  - **-z (hoặc `--compress`)** - nén file được chỉ định
  - **-d (hoặc `--uncompress` hoặc `--decompress`)** - giải nén file được chỉ định
  - **CHÚ Ý:** nén file gốc và thay thế bằng phiên bản nén mới
    - **Ví dụ - `xz -z /home/user/myfile.txt`**
      - Lệnh này sẽ nén file **/home/user/myfile.txt**, để lại **/home/user/ myfile.txt.xz** (phiên bản chưa nén sẽ bị xóa)

## 103.4 Sử dụng Luồng(Streams), Ống(Pipes) và Chuyển hướng (Redirects)

- **Luồng đầu vào tiêu chuẩn (Standard input stream)**
  - Một luồng cung cấp đầu vào cho các thiết bị đầu cuối, files, các ứng dụng và/hoặc tiện ích (phương thức mà tất cả các chương trình được cho là có thể chấp nhận trong một số hình thức)
- **Luồng đầu ra tiêu chuẩn(Standard output stream)**
  - Luồng đầu ra tiêu chuẩn
  - Một luồng cung cấp đầu ra cho các thiết bị đầu cuối, file, ứng dụng và/hoặc tiện ích
- **Luồng lỗi tiêu chuẩn(Standard error stream)**
  - Một luồng chứa các thông báo lỗi được đưa ra đến các thiết bị đầu cuối, file, ứng dụng và/hoặc tiện ích (được coi là tập hợp con của luồng đầu ra tiêu chuẩn)
- **Chuyển hướng(Redirection)**
  - Quá trình lấy một luồng và gửi nó "đến nơi khác", không phải là mặc định
- **Các thiết bị**
  - Đầu vào tiêu chuẩn(Standard input) - `/dev/stdin`
  - Đầu ra tiêu chuẩn(Standard output) - `/dev/stdout`
  - Luồng lỗi tiêu chuẩn(Standard error) - `/dev/stderr`
- **Bộ chỉ số tập tin và tập tin liên quan**
  - Đầu vào tiêu chuẩn - chỉ số file 0 - file `/proc/self/fd/0`
  - Đầu ra tiêu chuẩn - chỉ số file 1 - file `/proc/self/fd/1`
  - Luồng lỗi tiêu chuẩn - chỉ số file 2 - file `/proc/self/fd/2`
- **Trình chuyển hướng(Redirectors)**
  - `|` - đường ống, để gửi đầu ra lệnh đến lệnh khác
    - **Ví dụ** - `cat /var/log/messages | more`
  - `>` - chuyển hướng đầu ra tiêu chuẩn đến một file hoặc thiết bị (tạo hoặc ghi đè đích nếu là một file)
    - **Ví dụ** - `find /user -name "*.sh" > output.txt`
  - `>>` - chuyển hướng đầu ra tiêu chuẩn đến một file hoặc thiết bị (nối thêm vào đích nếu là một file)
    - **Ví dụ** - `find /user -name "*.txt" >> output.txt`
  - `<` - chuyển hướng đầu vào tiêu chuẩn đến một chương trình
    - **Ví dụ** - `sort < /home/user/listfile.txt`
    - **CHÚ Ý:** hành vi tương tự như khi sử dụng `cat /home/user/listfile.txt | sort`
  - **CHÚ Ý** - chuyển hướng có thể xảy ra mặc định (dựa trên trình chuyển hướng) hoặc bằng cách sử dụng chỉ số file với trình chuyển hướng
- **Chuyển hướng luồng lỗi(Redirecting standard error)**
  - Thông thường, `stderr` được chuyển hướng đến một file (ghi nhật ký `log`) hoặc một thiết bị đặc biệt gọi là `/dev/null`

- Việc này cho phép bạn xóa các lỗi từ đầu ra tiêu chuẩn bình thường
- **Ví dụ** - `find / -iname "*.sh" 2> /dev/null`
  - Sẽ hiển thị kết quả của các kết quả tìm thấy mà không hiển thị các thông báo lỗi liên quan đến quyền truy cập
- Kết hợp chuyển hướng
  - **Ví dụ** - `find / -iname "*.sh" 2> /dev/null > output.txt`
    - Tương tự như ví dụ trên, nhưng các kết quả sẽ được chuyển hướng đến file `output.txt` thay vì hiển thị lên trên màn hình
  - **Ví dụ** - `sort < listfile | nl`
    - Chuyển hướng 'listfile' như một luồng đầu vào đến lệnh `sort`, đẩy đầu ra đó vào lệnh `nl` để thêm số dòng, hiển thị lên màn hình
- Kết hợp đặc biệt - `2>&1`
  - Thường được sử dụng để loại bỏ toàn bộ đầu ra cho một công việc(job) hoặc tiến trình process (như một công việc cron), để đảm bảo không gây ra các lỗi trong hiển thị dữ liệu
  - **Ví dụ** - `find / -iname "*.sh" > /dev/null 2>&1`
    - Chuyển hướng lỗi tiêu chuẩn thành đầu ra tiêu chuẩn (`2>&1`) và toàn bộ luồng đầu ra được chuyển hướng vào `/dev/null` (bị loại bỏ)
- **tee**
  - Nhận luồng đầu vào tiêu chuẩn và gửi một luồng đầu ra (giống nhau) đến một file được chỉ định
  - Thường được sử dụng khi bạn muốn thu thập đầu ra của một ứng dụng nhưng cũng cần nhìn thấy kết quả trên màn hình
  - **Ví dụ** - `find / -name "*.sh" | tee visibleresults.txt`
    - Sẽ tìm tất cả các file có đuôi mở rộng `*.sh` từ phân vùng gốc(root partition), đẩy kết quả đó như một luồng đầu vào cho `tee`, sau đó mở hai luồng đầu ra tiêu chuẩn, gửi một luồng đến console và một luồng đến file được chỉ định `visibleresults.txt`
- **xargs**
  - Nhận một luồng đầu vào (kết quả của lệnh khác - thường là lệnh `find`), sau đó đẩy chúng vào một lệnh khác theo yêu cầu
  - **Ví dụ** - `find / -name "*.sh" | xargs ls -al > myresults.txt`
    - Sẽ tìm tất cả các files có đuôi mở rộng `.sh`, sau đó `xargs` sẽ lấy kết quả đó và cung cấp cho lệnh `ls -al` để hiển thị chi tiết của mỗi file và chuyển hướng đầu ra vào file `myresults.txt`



## 103.5 Tạo, Giám sát và Hủy(kill) tiến trình process (Terms và Tiện ích- PID, ps, pstree, free, uptime, tín hiệu, kill, killall, pkill, pgrep)

### • PID

- Mã xác định tiến trình - một số duy nhất (trên hệ thống Linux) có thể được sử dụng để tham chiếu đến một tiến trình đang chạy

### • ps

- Hiển thị các tiến trình đã được khởi động và đang chạy với quyền người dùng hiện tại hoặc được chỉ định

- **-a** - hiển thị tất cả các quá trình đang chạy trên hệ thống, cho bất kỳ người dùng nào
- **-u** - hiển thị thông tin người dùng cho các tiến trình được hiển thị
- **-x** - hiển thị quá trình không có tty (terminal) kết nối

- **LƯU Ý:** các tùy chọn này thường được sử dụng cùng nhau và có thể được sử dụng có hoặc không có ký tự **-** ở trước

### • Ví dụ - ps aux

- Sẽ hiển thị tất cả các tiến trình đang chạy, bởi bất kỳ người dùng nào, hiển thị chủ sở hữu tiến trình và bao gồm các tiến trình không liên kết với một terminal
- Đây là cú pháp Linux để xem các mục này

### • Ví dụ - ps -ef

- Sẽ hiển thị tất cả các tiến trình đang chạy, bởi bất kỳ người dùng nào, hiển thị chủ sở hữu tiến trình và bao gồm các tiến trình không liên kết với một terminal
- Đây là cú pháp Linux để xem các mục này

- Theo tính chất chuyển hóa, mặc dù có sự khác biệt về định dạng, **ps aux = ps -ef**

### • pstree

- Hiển thị cây tiến trình đang chạy theo dạng cây (hiển thị ASCII phân cấp hoặc ncurses)
- **-A** - hiển thị cây sử dụng ký tự ASCII
- **-a** - hiển thị các tiến trình bao gồm các tham số sử dụng
- **-p** - hiển thị PID

### • free

- Cung cấp thông tin về bộ nhớ RAM hệ thống Linux (bao gồm tổng cộng, đang sử dụng, được chia sẻ, bộ đệm và bộ nhớ cache, VV).
- **-b (hoặc --bytes)** - hiển thị bộ nhớ theo byte
- **-k (hoặc --kilo)** - hiển thị bộ nhớ theo kilobyte (mặc định)
- **-m (hoặc --mega)** - hiển thị bộ nhớ theo megabyte
- **-g (hoặc --giga)** - hiển thị bộ nhớ theo gigabyte
- **-h (hoặc --human)** - hiển thị bộ nhớ trong định dạng dễ đọc cho con người
- **-c (hoặc --count)** - số lần hiển thị kết quả (phải được sử dụng với tùy chọn **-s**)
- **-s (hoặc --seconds)** - thời gian chờ giữa mỗi lần hiển thị kết quả (sử dụng với **-c**)
- **-t (hoặc --total)** - hiển thị dòng chỉ số tổng của từng cột
- **-l (hoặc --lohi)** - hiển thị thống kê bộ nhớ thấp và cao

### • uptime

- Hiển thị thời gian hệ thống Linux đã chạy kể từ lần khởi động(boot)/khởi động lại(reboot) cuối cùng
- Cung cấp thông tin về số lượng người dùng hiện đang có trên hệ thống Linux và tải trung bình trên hệ thống trong 1, 5 và 15 phút (ba giá trị)
- Tín hiệu (Signals)
  - Là những gì được gửi đến tiến trình và tiến trình sau đó phản ứng tương ứng, tùy thuộc vào loại tín hiệu
    - **SIGHUP** - tín hiệu(signal) 1 - tắt và khởi động lại tiến trình (hang up)
    - **SIGINT** - tín hiệu 2 - gián đoạn một tiến trình (CTL-C)
    - **SIGKILL** - tín hiệu 9 - kill tiến trình (không thể bị bỏ qua hoặc bắt được)
    - **SIGTERM** - tín hiệu 15 - kết thúc tiến trình (tiến trình có thể bỏ qua hoặc bắt được)
- tín hiệu)
  - **SIGSTOP** - tín hiệu 19 - dừng tiến trình (không thể bị bỏ qua hoặc bắt được)
  - **SIGTSTP** - tín hiệu 20 - dừng terminal (CTL-Z)
- kill
  - Gửi một lệnh **kill 'nice'** đến PID chỉ định (mặc định lệnh kill là tín hiệu **SIGTERM** signal 15)
  - Phương pháp này cho phép tiến trình dừng lại một cách 'sạch sẽ' (kết thúc, giải phóng bộ nhớ, đóng file đang mở, v.v.)
  - **-[số tín hiệu]** - cố gắng tiêu diệt quá trình bằng tín hiệu chỉ định
  - **-1 (-HUP)** - tham chiếu đặc biệt để yêu cầu tiến trình khởi động lại (để đọc lại file cấu hình hoặc thực hiện thay đổi)
    - **-9 (-KILL)** - **kill/stop/end/dump** ngay lập tức (thường dùng để **kill** ngay cả một tiến trình **zombie** hoặc đang treo)
- killall
  - Tiêu diệt kill tất cả các phiên bản của tiến trình được đặt tên (hữu ích khi có nhiều tiến trình sử dụng cùng tên tiến trình/tên ứng dụng - ví dụ **'httpd'**)
- pkill
  - Tiêu diệt kill các tiến trình dựa trên tên, ID, người dùng, phiên hoặc thiết bị đầu cuối (**tty**)
  - **LƯU Ý:** nếu có nhiều tiêu chí được sử dụng, **TẤT CẢ** các tùy chọn phải khớp để áp dụng cho một tiến trình
  - **-signal (hoặc --signal) [#]** - gửi số tín hiệu đến tiến trình khớp
  - **-t [terminal]** - khớp với thiết bị đầu cuối/tty được chỉ định
  - **-U (hoặc --uid) [user]** - khớp với ID người dùng
- pgrep
  - Cho phép kiểm tra (debug) lệnh pkill (hiển thị những gì sẽ xảy ra)
  - Hiển thị ID quá trình sẽ bị ảnh hưởng
  - Chấp nhận **TẤT CẢ** các tùy chọn pkill để xác định **PID** sẽ bị ảnh hưởng
  - **LƯU Ý:** nếu cung cấp nhiều tùy chọn, cách liệt kê sẽ xác định cách chúng được hiểu
    - **Ví dụ - pgrep -u root,apache httpd**
      - Sẽ hiển thị bất kỳ tiến trình **httpd** nào thuộc sở hữu của **root** HOẶC **apache**
    - **Ví dụ - pgrep -u root apache**
      - Chỉ hiển thị các tiến trình thuộc sở hữu của **root** VÀ **apache**

## 103.5 Tạo, Giám sát và Tiêu diệt Quá trình (Terms và Công cụ jobs, bg, fg, &, ưu tiên(priority), nohup và screen)

- **jobs**
  - Sẽ hiển thị trạng thái của các công việc(jobs) đã bị tạm dừng
  - Hiển thị:
    - **[#]- Stopped [cmd]** - cho biết công việc trước đó hoặc công việc trước đó sẽ được thao tác
    - **[#]+Stopped [cmd]** - cho biết công việc hiện tại với bất kỳ lệnh (**bg** hoặc **fg**) nào sẽ tác động vào nó
- **bg**
  - Gửi tiến trình đã chỉ định vào chế độ nền background (thường được sử dụng sau lệnh jobs để đưa một tiến trình tạm dừng vào nền để nó tiếp tục chạy)
- **fg**
  - Gửi tiến trình đã chỉ định ra chế độ phía trước **foreground** (dùng để đưa một tiến trình đang chạy nền ra phía trước để sử dụng)
- **&**
  - Đặt một chương trình vào chế độ nền background khi chạy sau một lệnh
    - **Ví dụ: vim &**
      - Sẽ chạy và đưa trình soạn thảo **vim** vào chế độ nền
- **priority**
  - Lập lịch và ưu tiên là cách Linux sử dụng để chạy nhiều ứng dụng/dịch vụ/tiến trình trên một máy tính duy nhất một cách đa nhiệm
  - Ưu tiên tiến trình mặc định là '0'
  - Dải ưu tiên là từ '0' đến '19' và từ '0' đến '-20' (điểm ưu tiên càng thấp, tiến trình càng nhận được nhiều tài nguyên hơn, -20 là điểm ưu tiên cao nhất)
    - Mọi người dùng có thể bắt đầu tiến trình với điểm ưu tiên từ 0 đến 19
    - Chỉ root mới có thể bắt đầu tiến trình với điểm ưu tiên từ 0 đến -20
- **nohup**
  - Cho phép bạn khởi chạy một ứng dụng hoặc tiến trình từ dòng lệnh/terminal và sau đó **stop/exit/logout** khỏi terminal hoặc phiên làm việc, trong khi vẫn giữ ứng dụng hoặc tiến trình đang chạy
    - **Ví dụ: nohup find / -name "tên\_file" > output.txt**
      - Lệnh tìm file này có thể chạy trong thời gian dài (giả sử trên một hệ thống file lớn) và sẽ dừng nếu chúng ta đăng xuất khỏi máy tính, đóng terminal - **nohup** cho phép lệnh nó tiếp tục chạy nếu chúng ta thoát/đóng phiên
- **screen**
  - Là một trình quản lý phiên (session) văn bản cho phép chạy nhiều phiên shell độc lập với nhau trong cùng một cửa sổ
  - Có thể sử dụng phiên để kết nối cục bộ hoặc từ xa (tương tự như sử dụng nhiều terminal)
  - Hỗ trợ chức năng sao chép/dán giữa các phiên làm việc
  - Mỗi phiên làm việc có bộ đệm cuộn riêng
  - Hỗ trợ ghi log đầu ra nếu được cấu hình

- Có khả năng tách rời (thường được gọi là "bỏ lại") một cửa sổ, cho phép các chương trình được khởi chạy bên trong tiếp tục chạy
- Các tùy chọn khi **screen** được khởi động
  - **ctl+a** sau đó **c** - mở một phiên làm việc mới
  - **ctl+a** sau đó **p** - di chuyển quay lại phiên làm việc trước đó
  - **ctl+a** sau đó **n** - di chuyển đến phiên làm việc tiếp theo
  - **ctl+a** sau đó **"** - hiển thị danh sách các phiên làm việc có sẵn (được định danh bằng số)
    - Sau khi các số được hiển thị, nhấn số tương ứng và enter để chuyển sang phiên làm việc đó
  - **ctl+a** sau đó **d** - tách rời khỏi screen (các tiến trình vẫn tiếp tục chạy)
  - **ctl+a** sau đó **x** - khóa màn hình
    - Sẽ yêu cầu mật khẩu (hai lần) để mở khóa
  - **screen -ls** - hiển thị danh sách các màn hình
  - **screen -r** - kết nối lại với màn hình

## 103.6 Sửa đổi ưu tiên thực thi tiến trình (Sử dụng nice, renice và top)

- **nice**
  - Cho phép người dùng khởi chạy một tiến trình với ưu tiên thấp hơn mặc định (mặc định là '0', nice sẽ khởi chạy với '10')
    - Ví dụ - **nice myscript.sh**
      - Sẽ khởi chạy **myscript.sh** với ưu tiên thấp hơn là '10'
  - **-n [#] [chương trình]** - cho phép bạn thay đổi ưu tiên mặc định của chương trình đã chỉ định
- **renice**
  - Thay đổi ưu tiên của một tiến trình
  - Thường được sử dụng để giảm ưu tiên của một tiến trình mà có thể tiêu thụ quá nhiều tài nguyên hoặc tăng ưu tiên của một tiến trình để nó hoàn thành nhanh hơn
  - **+[#] [PID]** - thay đổi ưu tiên thành một số cao hơn (tương ứng với ưu tiên thấp hơn)
    - Tất cả người dùng có thể giảm ưu tiên của một tiến trình đang chạy (tăng số của tiến trình).
  - **-[#] [PID]** - thay đổi ưu tiên thành một số thấp hơn (tương ứng với ưu tiên cao hơn)
  - Chỉ có người dùng root mới có thể tăng ưu tiên của một tiến trình đang chạy (giảm số)
- **top**
  - Cấu hình được đọc từ **/etc/toprc** (hoặc **/home/user/.toprc** cục bộ cho cấu hình người dùng)
  - Có thể được sử dụng để hiển thị các tiến trình **'top'** đang chạy (hoạt động)
  - Cũng có thể được sử dụng để thay đổi ưu tiên của các tiến trình đang chạy

- **d [#]** - chạy và cập nhật hiển thị tiến trình mỗi '#' giây
- **i** - chỉ hiển thị các tiến trình hoạt động
- **-b** - chạy ở chế độ batch
- **-c [#]** - chạy '#' lần cập nhật và thoát
  - **Ví dụ - top -b -n 5 > output.txt**
    - Sẽ chạy top ở chế độ batch và cập nhật 5 lần, ghi kết quả vào file output.txt và thoát
- Trong quá trình chạy, các phím tắt bàn phím sau được áp dụng:
  - **spacebar** - cập nhật ngay lập tức
  - **h** - hiển thị màn hình trợ giúp
  - **k** - yêu cầu PID để kết thúc
  - **i** - hiển thị hoặc bỏ qua các tiến trình không hoạt động/ru idle/zombie (chuyển đổi)
  - **n** - yêu cầu số tiến trình để hiển thị trên màn hình
  - **r** - yêu cầu PID để áp dụng 'renice' và ưu tiên mới (nhớ - số càng cao, ưu tiên càng thấp)
  - **R** - sắp xếp các PID của tiến trình từ cao đến thấp (ngược lại so với mặc định)

## 103.7 Tìm kiếm file văn bản bằng biểu thức chính quy Regular Expressions (Sử dụng grep, egrep, fgrep, sed và regex)

- **grep**
  - Một tiện ích để tìm chuỗi và cụm từ trong tập tin, luồng hoặc thư mục
  - Đại diện cho - in ra các biểu thức chính quy toàn cầu
  - **-c** - chỉ hiển thị số lượng tìm thấy
  - **-C [#]** - bao quanh kết quả tìm kiếm với '#' dòng thông tin liên quan (trước và sau dữ liệu)
  - **-E [ext. regex]** - sử dụng biểu thức chính quy mở rộng chỉ định để tìm kiếm
  - **-F [fixed regex]** - sử dụng biểu thức chính quy cố định chỉ định để tìm kiếm
  - **-H** - hiển thị tên file của mỗi chuỗi hoặc cụm từ khớp
  - **-h** - ngăn không hiển thị tên file
  - **-i** - không phân biệt chữ hoa chữ thường
  - **-l** - chỉ hiển thị tên file, không hiển thị chuỗi/cụm từ khớp
  - **-L** - chỉ hiển thị tên file không chứa chuỗi/cụm từ khớp
  - **-w** - chỉ khớp các dòng chứa toàn bộ chuỗi hoặc cụm từ

- **-x** - chỉ hiển thị các dòng khớp chính xác toàn bộ chuỗi hoặc cụm từ
- **-v** - chỉ hiển thị các dòng trong file không khớp với chuỗi hoặc cụm từ (ngược lại với mặc định)
- **Ví dụ** - `find / -name "*.sh" -exec grep -iH "modprobe" {} ;`
  - Lệnh này sẽ tìm file, đệ quy, bắt đầu từ thư mục root và kết thúc bằng `.sh`, truyền file đó cho lệnh `grep`, lệnh này sẽ không phân biệt chữ hoa chữ thường và hiển thị tên file và nội dung khớp của bất kỳ file nào chứa từ `'modprobe'`
- **Ví dụ** - `grep -cv "/bin/bash" /etc/passwd`
  - Sẽ chỉ hiển thị các dòng không chứa giá trị `'/bin/bash'` và hiển thị số lượng (ví dụ: bạn sẽ biết số lượng tài khoản người dùng không có bash được liệt kê là shell mặc định)
- **egrep**
  - Lệnh `grep` mà không cần chỉ định **-E**
  - Cho phép sử dụng biểu thức chính quy mở rộng để khớp các mẫu văn bản
  - **Ví dụ** - `'egrep '^ope(n|r)' /etc/passwd'`
    - Lệnh này sẽ hiển thị các mục nhập trong file `/etc/passwd` mà bắt đầu bằng `'ope'` và tiếp theo là `'n'` hoặc `'r'` (sau đó là bất kỳ ký tự nào khác) - **ví dụ** như `openvpn` hoặc `operator`.
  - Bạn cũng có thể sử dụng toán tử OR (`|`) để tạo mẫu tìm kiếm phức tạp hơn.
    - **Ví dụ** - `egrep '(bin|bash)' /etc/passwd`
      - Lệnh này sẽ hiển thị các dòng chứa từ `'bin'` hoặc `'bash'` trong file `/etc/passwd`.
- **fgrep**
  - Là lệnh **Grep** mà không cần chỉ định **-F**
  - Cho phép sử dụng một file chứa một hoặc nhiều mục để tìm kiếm (tìm kiếm file)
  - **-f [itemfile]** - sử dụng file chỉ định là danh sách các mục để tìm kiếm trong file được chỉ định
    - **Ví dụ** - `fgrep -f itemfile.txt searchfile.txt`
      - Sẽ sử dụng các mục trong `itemfile.txt` làm tham số tìm kiếm cho tất cả các dòng trong `searchfile.txt` và hiển thị các kết quả tìm kiếm.
- **LƯU Ý:** Tất cả các lệnh `grep` đều hỗ trợ các phạm vi (ví dụ: `[a-z]`) hoặc ký tự đại diện (\*) làm tham số tìm kiếm.
- Biểu thức chính quy (**Regular expressions**)
  - Là phương pháp tìm kiếm chuỗi hoặc cụm từ không biết chính xác bằng cách sử dụng chuỗi bán phần và ký tự đặc biệt.
  - Các ký tự biểu thức chính quy:
    - `.` - phù hợp với bất kỳ ký tự đơn nào

- **?** - phù hợp với một mục tùy chọn, nhưng chỉ một lần
- **\*** - phù hợp với từ '0' đến 'n' ký tự trong một chuỗi
- **+** - mục PHẢI được phù hợp ít nhất một lần nhưng có thể được phù hợp nhiều hơn
- **{#}** - phù hợp '#' lần
- **{#,}** - phù hợp '#' lần hoặc nhiều hơn
- **{#,#}** - phù hợp giữa số đầu tiên và số thứ hai (ví dụ: {3,10} có nghĩa là phù hợp từ 3 đến 10 lần)
- **<** - từ bắt đầu bằng chuỗi sau
- **^** - từ bắt đầu bằng chuỗi sau
- **>** - từ kết thúc bằng chuỗi sau
- **\$** - từ kết thúc bằng chuỗi sau
- **[aA]** - từ chứa 'a' hoặc 'A'
  - **Ví dụ - `grep "d[iou]g" textfile.txt`**
    - Sẽ trả về các từ bắt đầu bằng 'd' và kết thúc bằng 'g' và có 'i' hoặc 'o' hoặc 'u' trong chúng (ví dụ: **dig, dug, dog**)
  - **[^o]** - các từ không chứa chữ cái 'o'
    - **Ví dụ - `grep "d[^o]g" textfile.txt`**
      - sẽ trả về các từ bắt đầu bằng 'd' và kết thúc bằng 'g' và có bất kỳ chữ cái nào ngoại trừ 'o' ở giữa (**ví dụ: dig, dug**)
  - **^...\$** - bất kỳ dòng nào chứa chính xác ba ký tự
    - **Ví dụ - `grep "<ope[^r]" /etc/passwd`**
      - sẽ hiển thị các từ bắt đầu bằng 'ope' theo sau bởi bất kỳ chữ cái nào TRỪ 'r' trong file **/etc/passwd** (**ví dụ:** tìm tài khoản '**openvpn**' nhưng không tìm '**operator**')
    - **Ví dụ - `grep "nologin>" /etc/passwd`**
      - sẽ hiển thị tất cả các từ kết thúc bằng 'nologin' trong tệp **/etc/passwd**



## 103.8 Thực hiện các thao tác chỉnh sửa FILE cơ bản bằng VI

- **vi/vim**

- Một trình soạn thảo phổ biến có sẵn trong tất cả các bản phân phối Linux
- Hoạt động trong ba chế độ riêng biệt
  - Chế độ lệnh - được sử dụng để di chuyển con trỏ và thực hiện các hoạt động khác nhau trên văn bản
  - Chế độ chèn - thêm/chèn văn bản vào tài liệu của bạn
  - Chế độ Ex (còn gọi là LastLine) - chế độ nâng cao để tìm/thay thế, thực thi các lệnh bên ngoài và sử dụng các khung chia

- **/etc/vimrc**

- File cấu hình vim toàn cầu

- **/home/user/.vimrc**

- Cấu hình cụ thể cho người dùng của vim
- Phím tắt số dòng trong chế độ lệnh để nắm được cho thi LPIC-1

- **:number/nonumber** - bật/tắt hiển thị số dòng trên màn hình

- **:nu/no nu** - phím tắt cho lệnh trên

- Màn hình

- dưới cùng - chứa 'dòng thông điệp'

- Chứa đường dẫn đầy đủ đến file, số dòng, kích thước file, dòng và cột hiện tại và vị trí con trỏ hiện tại

- **Ví dụ "/etc/passwd" 33L, 3359C 10,1 Top**

- File /etc/passwd, dài 33 dòng, 3359 ký tự, dòng 10, cột 1 và con trỏ đang ở phía trên của tài liệu

- **Vim** sẽ bắt đầu ở chế độ **'command'** (lệnh)



- **i** - chuyển sang chế độ 'insert mode' (chèn) tại vị trí con trỏ hiện tại để bạn có thể bắt đầu nhập văn bản vào tài liệu
- **I** - di chuyển đến đầu dòng hiện tại và chuyển sang 'insert mode'
- **a** - chuyển sang chế độ 'insert mode', đặt con trỏ một ký tự sang phải so với vị trí hiện tại (**append**)
- **A** - di chuyển đến cuối dòng hiện tại, đặt con trỏ một ký tự sang phải so với vị trí cuối (line append)
- **o** - chèn một dòng mới dưới dòng hiện tại, đặt con trỏ ở vị trí đầu tiên trên dòng mới trong chế độ 'insert mode'
- **O** - chèn một dòng mới phía trên dòng hiện tại, đặt con trỏ ở vị trí đầu tiên trên dòng mới trong chế độ 'insert mode'
- **c**[tùy chọn] - thay đổi văn bản tại vị trí hiện tại
- **cw** - thay đổi từ hiện tại tại vị trí con trỏ
- **cc** - thay đổi dòng hiện tại tại vị trí con trỏ
- **c\$** - thay đổi từ vị trí hiện tại đến cuối dòng
- **r** - thay thế ký tự tại vị trí hiện tại
- **R** - thay thế văn bản trên cùng một dòng cho đến khi thoát khỏi chế độ 'insert mode' hoặc đến cuối dòng
- **x** - xóa ký tự sau con trỏ
- **X** - xóa ký tự trước con trỏ
- **dw** - xóa từ sau con trỏ
- **dd** - xóa toàn bộ dòng mà con trỏ đang đứng trên đó
- **D** - xóa văn bản từ vị trí con trỏ đến cuối dòng
- **dL** - xóa văn bản từ vị trí con trỏ đến cuối màn hình hiện tại
- **dG** - xóa văn bản từ vị trí con trỏ đến cuối file
- **d^** - xóa toàn bộ văn bản từ đầu dòng đến vị trí con trỏ hiện tại

- **u** - hoàn tác thao tác/thay đổi cuối cùng
- **yy** - sao chép dòng hiện tại vào bộ đệm (thường được gọi là yank)
- **yw** - sao chép từ vị trí con trỏ hiện tại đến cuối từ hiện tại
- **p** - dán nội dung của bộ nhớ đệm sau con trỏ
- **P** - dán nội dung của bộ nhớ đệm trước con trỏ
- **:e!** - hoàn tác TẤT CẢ các thay đổi kể từ lần cuối cùng file được lưu vào đĩa
- **:w** - lưu tệp xuống đĩa (lưu lại)
- **:q** - thoát khỏi trình chỉnh sửa (sẽ cảnh báo nếu tệp đã được thay đổi nhưng chưa được lưu)
- **:q!** - thoát ngay lập tức, không quan tâm đến việc file đã được thay đổi và lưu
- **:x** - phím tắt để lưu và thoát
- **ZZ** - phím tắt để lưu và thoát
- Phím tắt di chuyển
- **h** - di chuyển một ký tự sang trái
- **j** - di chuyển một dòng xuống
- **k** - di chuyển một dòng lên
- **l** - di chuyển một ký tự sang phải
- bất kỳ thao tác di chuyển nào ở trên có thể lặp lại bằng cách đặt số lần mong muốn phía trước
  - **Ví dụ** - **3k** (sẽ di chuyển lên 3 dòng) **2h** (sẽ di chuyển sang trái 2 ký tự)
- **ctl-u** - di chuyển lại một nửa trang
- **ctl-b** - di chuyển lại một trang
- **ctl-d** - di chuyển tiếp một nửa trang
- **ctl-f** - di chuyển tiếp một trang

- **ctl-G** - hiển thị tên file, số dòng và vị trí theo phần trăm so với tổng độ dài file
- **[#]G** - di chuyển trực tiếp đến dòng được chỉ định
- **[#]W** - di chuyển 12 từ sang phải
- Tìm kiếm
  - Phải ở chế độ lệnh (nhấn phím **ESC** để chắc chắn bạn đang ở chế độ lệnh)
  - **/[chuỗi]** - tìm kiếm từ vị trí con trỏ trở đi cho 'chuỗi'
  - **?[chuỗi]** - tìm kiếm từ vị trí con trỏ ngược lại cho 'chuỗi'
  - **N** - khi được sử dụng sau một tìm kiếm, sẽ tìm kiếm 'tiếp theo' sự xuất hiện của 'chuỗi' trong hướng được chỉ định
- Thay thế
  - phải ở chế độ lệnh (nhấn phím **ESC** để chắc chắn bạn đang ở chế độ lệnh)
  - cú pháp tương tự như sed
  - **s** - thay thế (trong dòng hiện tại)
  - **%s** - thay thế (trong toàn bộ file)
  - **/[giá trị cần tìm]** - giá trị cần tìm
  - **/[giá trị cần thay thế]/** - giá trị để thay thế
  - **g** - tùy chọn, sẽ thay thế TẤT CẢ các sự xuất hiện thay vì chỉ sự xuất hiện đầu tiên
    - **Ví dụ** - **:s/Mar/Apr/g**
      - Sẽ tìm kiếm trong dòng tất cả các sự xuất hiện của **'Mar'** và thay thế chúng bằng **'Apr'**, thêm **'%'** trước **'s'** và sau đó thay thế tất cả các sự xuất hiện trong toàn bộ file
  - **LƯU Ý**: có thể sử dụng cú pháp biểu thức chính quy phức tạp
- Thực hiện lệnh
  - **:! [cmd]** - thực thi lệnh được chỉ định trên dòng lệnh

## Chủ đề 104 - Thiết bị(Devices), Linux Filesystems và Tiêu chuẩn Phân cấp Hệ thống File(Filesystem Hierarchy Standard)

### 104.1 Tạo Phân vùng (Partitions) và Filesystem (Sử dụng Công cụ Phân vùng - fdisk, gdisk và parted)

- Thứ tự chung để chuẩn bị sử dụng một ổ đĩa(hard disk) trong Linux

1. Lắp đặt vật lý (hoặc, trong trường hợp máy ảo, cấp phát ổ đĩa(disk) cho máy ảo)
2. Phân vùng thiết bị (sử dụng một trong vài công cụ phân vùng, tùy thuộc vào kích thước/ tính năng của thiết bị)
3. Định dạng các phân vùng đã tạo với loại Filesystem đã chọn
4. Tạo điểm gắn(mount point) hệ thống (thư mục) và xác định quyền truy cập
5. Gắn thiết bị/phân vùng vào thư mục đã chọn
6. Nếu muốn sử dụng vĩnh viễn ổ đĩa, thêm một mục vào file /etc/fstab

- /dev

- Thư mục thiết bị tiêu chuẩn trong Linux nơi các ổ đĩa sẽ có các mục tương ứng
- Thiết bị IDE

- Ví dụ - /dev/hda (và sau đó là số phân vùng như /dev/hda1)

- Thiết bị SATA/SCSI

- Ví dụ - /dev/sda (và sau đó là số phân vùng như /dev/sda1)

- Phân vùng chính(Primary partition)

- Các phân vùng độc lập với bất kỳ phân vùng mở rộng và phân vùng logic nào (xem bên dưới), thông thường bạn có thể có tối đa BỐN phân vùng chính (vốn là giới hạn của MBR) trên một thiết bị ổ đĩa (với một số ngoại lệ - xem GUID bên dưới)

- Phân vùng mở rộng(Extended partition)

- Chỉ có một phân vùng mở rộng trên mỗi ổ đĩa
  - Hãy nghĩ về nó như một 'container' cho các phân vùng logic

- Phân vùng logic(Logical partition)

- Các phân vùng trong một phân vùng mở rộng, thường cần thiết khi cần nhiều hơn bốn phân vùng (xem phân vùng chính ở trên)

- Phân vùng swap(Swap partitions)

- Các phân vùng được định dạng đặc biệt và được dành riêng cho bộ nhớ ảo để hỗ trợ việc thiếu bộ nhớ RAM trong hệ thống Linux

- LƯU Ý: nên có ít nhất kích cỡ bằng bộ nhớ hệ thống (khuyến nghị lịch sử là gấp đôi bộ nhớ hệ thống Linux- đề thi!)

- Mặc dù có thể sử dụng như bộ nhớ hệ thống trong trường hợp thiếu bộ nhớ hệ thống, hiệu suất hệ thống có thể giảm đáng kể khi sử dụng

- Số thứ tự

- Phân vùng chính - từ 1 đến 4
  - Phân vùng logic - từ 5 trở đi

- fdisk (mô tả)

- Công cụ phân vùng đĩa có sẵn trong các bản phân phối Linux (quy tắc phân vùng tiêu chuẩn, không có khả năng chỉnh sửa các ổ đĩa GUID)
- Xử lý với bảng phân vùng MBR truyền thống
- Giới hạn - giới hạn kích cỡ phân vùng đến 2TB, giới hạn bốn phân vùng (phân vùng chính và mở rộng - các ổ đĩa logic được tạo ra để giải quyết vấn đề này), không hỗ trợ kiểm tra định danh ổ đĩa
- **gdisk** (mô tả)
  - Công cụ phân vùng đĩa được cài đặt trên hầu hết các bản phân phối Linux hiện đại, có khả năng tạo và sửa đổi bảng GUID
  - Giải quyết hạn chế của bảng phân vùng MBR
- **parted** (mô tả)
  - Công cụ phân vùng đĩa với một bộ tính năng mở rộng (bao gồm tổ chức lại/phóng to/thu nhỏ ổ đĩa/phân vùng)
  - Thường được sử dụng với một giao diện đồ họa mở rộng (ví dụ như **gparted**)
- **fdisk** (sử dụng)
  - Dòng lệnh, kèm theo thiết bị
    - **Ví dụ** - **fdisk /dev/sda**
  - **-l** [thiết bị] - hiển thị tóm tắt trạng thái MBR và thông tin phân vùng đĩa cho các thiết bị được chỉ định
  - Tùy chọn tương tác (sau khi công cụ được khởi động)
    - **LƯU Ý**: không có gì được ghi vào ổ đĩa cho đến khi bạn chỉ định rõ (xem danh sách tùy chọn dưới đây)
    - **p** - hiển thị tóm tắt đĩa/phân vùng
    - **n** - tạo một phân vùng mới
      - **p** - tạo phân vùng chính
        - **[#]** - số phân vùng chính (từ 1 đến 4)
      - **e** - tạo phân vùng mở rộng
        - **[#]** - số phân vùng mở rộng (từ 2 đến 4 - **LƯU Ý**: tính là một trong bốn phân vùng chính được phép)
      - **l** - tạo phân vùng logic
        - **[#]** - số phân vùng logic (phạm vi sẽ phụ thuộc vào các lựa chọn trước, nhưng sẽ là số của phân vùng mở rộng + 1)
    - **LƯU Ý**: mỗi phân vùng sẽ tự động được cấp phát dưới dạng 'Linux' (loại **83**) trừ khi thay đổi
    - **t** - thay đổi loại phân vùng
    - **[#]** - mã hexa chỉ định loại ('**L**' sẽ liệt kê chúng)
      - Các loại phân vùng cần biết cho kỳ thi:
        - **82** - Linux swap
        - **83** - Linux
        - **85** - Linux extended
        - **8e** - Linux LVM
        - **fd** - Linux RAID
    - **w** - lưu các thay đổi xuống ổ đĩa
  - **gdisk** (sử dụng)
    - Tương tự về chức năng nhưng hỗ trợ các loại filesystem bổ sung và bảng phân vùng GUID

- **-l** [thiết bị] - liệt kê trạng thái GUID (gpt) và thông tin phân vùng đĩa cho các thiết bị được chỉ định
- Tùy chọn tương tác (sau khi công cụ được khởi động)
  - **LƯU Ý:** Giống như với ổ đĩa, không có gì được ghi vào ổ đĩa cho đến khi bạn chỉ định rõ (xem danh sách tùy chọn dưới đây)
  - **p** - hiển thị tóm tắt đĩa/phân vùng
  - **n** - tạo một phân vùng mới
    - **p** - tạo phân vùng chính
      - **[#]** - số phân vùng chính (từ 1 đến 128)
  - **LƯU Ý:** Mỗi phân vùng sẽ tự động được cấp phát dưới dạng 'Linux' (loại **8300**) trừ khi thay đổi
  - **t** - thay đổi loại phân vùng
    - **[#]** - mã hexa chỉ định loại ('L' sẽ liệt kê các mã)
      - Các loại phân vùng cần biết cho kỳ thi:
        - **8200** - Linux swap
        - **8300** - Linux
        - **8301** - Linux reserved
        - **8e00** - Linux LVM
        - **fd00** - Linux RAID
- **w** - ghi các thay đổi xuống ổ đĩa
- **parted** (sử dụng)
  - Có khả năng làm việc với cả ổ đĩa MBR truyền thống và ổ đĩa GPT mới hơn
  - **-l** [thiết bị] - liệt kê RẤT NHIỀU thông tin về thiết bị và các phân vùng được chỉ định
  - Tùy chọn tương tác (sau khi công cụ được khởi động)
    - **help** - liệt kê các lệnh có sẵn
    - **mkpart** [tên phân vùng] [đầu] [cuối] - tạo một phân vùng với tên phân vùng chỉ định, bắt đầu từ giá trị đầu và kết thúc tại giá trị cuối
    - **LƯU Ý:** mặc dù hỗ trợ thay đổi kích cỡ, nhưng việc thay đổi kích cỡ ổ đĩa, nhưng không phải là mục tiêu trong kỳ thi và nằm ngoài phạm vi khóa học của chúng ta
    - **LƯU Ý:** **parted** không thiết lập loại filesystem (và thực tế, Linux thường không quan tâm tới điều này, tuy nhiên, việc có loại sai hoặc không xác định có thể gây nhầm lẫn đôi khi đối với các ứng dụng cũ hơn)

## 104.1 Tạo phân vùng(partitions) và Filesystems (Các loại Filesystem và tạo chúng trên các phân vùng)

- **Superblock**

- Phần của ổ đĩa có thể đọc và chứa thông tin về Filesystem (kích thước, thống kê inode và thời gian kiểm tra cuối cùng)
- Thông thường được lưu trữ trên ổ đĩa trong phân đoạn đầu tiên (và tồn tại ở nhiều vị trí khác nhau để hỗ trợ khôi phục)
- Khối sao lưu đầu tiên cho Filesystem '**ext**' - **8193** (đề thi!)

- **Inode**

- Liên kết với mỗi file và thư mục trong hệ thống
- Chứa thông tin về file (ngoại trừ tên file - đề thi!)
- Bao gồm một danh sách các khối tạo nên file

- **Ví dụ** - **ls -li /var/log/messages**

- Sẽ hiển thị inode của file /var/log/messages

- **Ví dụ** - **stat /var/log/messages**

- Sẽ hiển thị thông tin chi tiết về inode liên quan đến file /var/log/messages

- Số lượng inode được thiết lập vào thời điểm tạo filesystem và không thể thay đổi sau đó.

- **Ví dụ**: **df -i**

- Sẽ hiển thị số lượng inode có sẵn, đã sử dụng và còn trống trên các filesystems đã được gắn vào hệ điều hành.

- Các loại hệ filesystems (phạm vi trong kỳ thi - còn có những loại khác)

- **ext2** - filesystem mở rộng Linux (cũ)

- **ext3** - filesystem mở rộng Linux có ghi chép nhật ký

(có khả năng khôi phục dễ dàng và nhanh chóng khi gặp vấn đề lỗi ổ đĩa)

- **ext4** - filesystem mở rộng Linux có ghi chép nhật ký (có cải tiến hiệu suất so với **ext3**)
- **xfs** - filesystem mở rộng, cải tiến hiệu suất đặc biệt trên các filesystems có nhiều files nhỏ hơn
- **ReiserFS** - một trong những filesystems đầu tiên giới thiệu việc ghi chép nhật ký và cung cấp khả năng điều chỉnh kích thước động
- **btrfs** - xây dựng trên các tính năng của ReiserFS và thêm các tính năng quản trị bổ sung trong khi tăng hiệu suất trên các filesystem lớn hơn
- **iso9660** - filesystem cho CD-ROM
- **udf** - filesystem cho DVD
- **vfat** - loại phân vùng DOS cũ (được sử dụng để tương thích với các hệ điều hành khác)
- Định dạng một ổ đĩa (hard disk)
  - Quá trình chuẩn bị kích cỡ và cấu trúc trên một phân vùng và áp dụng loại filesystem chỉ định, để nó có thể được gắn và sử dụng
  - **mkfs**
    - Tạo hệ filesystem
    - **-t [loại-fs] [thiết-bị/phân-vùng]** - tạo hệ thống tệp loại chỉ định trên phân vùng đĩa đã chọn
    - **-b [#]** - tạo filesystem với kích thước khối block chỉ định (mặc định - 4096)
    - **-m [#]** - phần trăm không gian dành riêng cho người dùng 'root'
    - **-L [nhãn]** - đặt volume label phân vùng
    - **-O [tùy-chọn]** - tùy chọn bổ sung
      - **Ví dụ:** `mkfs -t ext4 -b 8192 -m 10 -L LargeData -O sparse_super /dev/sde4`
      - Sẽ tạo ra một phân vùng '**ext4**', với kích thước khối là '**8192**' (nghĩa là không gian tối thiểu được phân bổ cho BẤT KỲ file nào là 8192 - điều này có thể được áp dụng cho một ổ đĩa dữ liệu với nhiều file lớn), dành 10% cho người dùng 'root', và sử dụng tùy chọn 'sparse\_super' (giảm số lượng bản sao superblock có sẵn cho khôi phục).



- **mkfs.[fstype]**
  - Lệnh tương đương cho mỗi loại filesystem để định dạng phân vùng được chỉ định
    - Ví dụ: **mkfs.ext3 /dev/sda3**
      - Sẽ tạo ra filesystem **ext3** trên phân vùng **/dev/sda3**
- **mke2fs**
  - Lệnh tương đương để tạo filesystem **ext2/3/4**
    - **/etc/mke2fs.conf**
      - File cấu hình cho cách thức hoạt động của **mke2fs**
    - **-O [option]** - thay đổi hành vi mặc định được cấu hình trong file cấu hình
- **mkswap**
  - Tạo swap space từ phân vùng (hoặc file) được chỉ định
- **mkraid**
  - Tạo một tập hợp RAID array từ các ổ đĩa được chỉ định
- **mknod**
  - Tạo các files 'đặc biệt' (ví dụ: thiết bị)
- **mkisofs** - tạo filesystem ISO để ghi vào đĩa CD-ROM

## 104.2 Bảo vệ tính toàn vẹn của filesystems

- **du**
  - Được sử dụng để 'ước tính' việc sử dụng dung lượng ổ đĩa(hard disk) cho các files và/hoặc thư mục
  - **-a (hoặc --all)** - hiển thị số lượng cho tất cả các files, không chỉ thư mục
  - **-c (hoặc --total)** - tính tổng cộng
  - **-h (hoặc --human-readable)** - hiển thị kích thước dưới dạng đọc được (K/M/G/TB)
  - **-s (hoặc --summarize)** - chỉ hiển thị tổng kết cho mọi đối số
  - **LƯU Ý:** có thể sử dụng với globbing file
    - Ví dụ: **du -sh /home/user/.bash\***
      - Sẽ cung cấp một tóm tắt dễ đọc cho mỗi file hoặc thư mục phù hợp với **'.bash\*'** trong thư mục **/home/user**
- **df**
  - Báo cáo việc sử dụng dung lượng ổ đĩa(hard disk) của filesystem
  - **-a (hoặc --all)** - bao gồm tất cả các filesystem (bao gồm cả filesystems 'dummy')
  - **--direct** - hiển thị thống kê cho một file thay vì mount
  - **--total** - hiển thị tổng cộng

- **-h** (hoặc **--human-readable**) - hiển thị kích cỡ dung lượng ổ dưới dạng đọc được (K/M/G/TB)
- **-l** (hoặc **--local**) - chỉ bao gồm các filesystem cục bộ
- **-t** (hoặc **--type**) - giới hạn danh sách cho loại được chỉ định
  - **Ví dụ: df -lh --total**
    - Sẽ tạo ra một danh sách dễ đọc được của tất cả các filesystems cục bộ bao gồm một dòng tổng kết ở cuối
- **debugfs**
  - Trình gỡ lỗi filesystem, có thể hiển thị một lượng thông tin lớn về phân vùng được chỉ định
  - **Ví dụ: debugfs /dev/sdb1**
    - Sẽ hiển thị nhiều thông tin về bất kỳ file hoặc thư mục nào đã được chỉ định chọn trên ổ đĩa
  - Tương tác (sau khi bắt đầu)
    - **?** - hiển thị các lệnh có sẵn
    - **cd** [đường dẫn] - chuyển đến đường dẫn đã chỉ định (**LƯU Ý**: phải tồn tại trên ổ đĩa mà **debugfs** đang chạy)
    - **features** - hiển thị các tính năng của filesystem
    - **logdump** - hiển thị nội dung nhật ký (nếu có sẵn)
    - **ls** - hiển thị nội dung của thư mục hiện tại
    - **pwd** - hiển thị thư mục làm việc
    - **open** - mở một filesystem để gỡ lỗi
    - **stats** - hiển thị thống kê cho filesystem
    - **undelete** - khôi phục tệp đã bị xóa (**LƯU Ý**: **PHẢI** được sử dụng ngay sau khi xóa file trước khi tất cả các thay đổi được đồng bộ hóa với ổ đĩa - thường không tìm thấy file đã bị xóa trên các phiên bản hiện đại vì 'sync' là cài đặt mặc định cho đồng bộ hóa đĩa của filesystem)
    - **quit** - exit
- **fsck**
  - Kiểm tra filesystem
    - **e2fsck** - kiểm tra các loại filesystem 'ext'
    - **reiserfsck** - kiểm tra các loại filesystem 'ReiserFS'

- **dosfsck** - kiểm tra các loại filesystem DOS (ví dụ **VFAT**)
- Sử dụng **/etc/fstab** để kiểm tra các filesystems tự động khi khởi động (những filesystems có '1' ở cột thứ sáu trong file), khi một filesystem được đánh dấu là '**unclean**' (bị crash, hỏng, không được **unmount** một cách sạch sẽ, v.v.), một lệnh scan quét sâu hơn sẽ được thực hiện
- Nếu filesystem không thể được sửa chữa tự động khi khởi động, bạn sẽ được nhắc nhở phải nhấn '**CTRL-D**' để tiếp tục hoặc cung cấp mật khẩu root để thực hiện quét sâu hơn (hiển thị lời nhắc cứu hộ)
- **-A** - **fsck** sẽ lặp lại qua file **/etc/fstab** và kiểm tra tất cả các filesystem
- **-C** - hiển thị thanh tiến trình băm/dấu
- **-N** - chạy thử, không thực hiện thay đổi nhưng hiển thị những gì sẽ được thực hiện
- **-V** - đầu ra chi tiết (cẩn thận, rất nhiều thông tin và có thể cuộn để xem thông tin)
- **-a** - chỉ thực hiện mà không yêu cầu xác nhận hoặc phản hồi khác, chạy kiểm tra và sửa chữa một cách phi tương tác
- **-f** - bắt buộc kiểm tra ngay cả khi filesystem báo cáo là sạch sẽ
- Thứ tự các sự kiện kiểm tra của **fsck**:
  1. Kiểm tra inode, khối, kích thước
  2. Kiểm tra cấu trúc thư mục
  3. Kiểm tra kết nối thư mục
  4. Kiểm tra số lượng tham chiếu của tệp/thư mục
  5. Kiểm tra thông tin tóm tắt của nhóm
- **LƯU Ý**: chỉ chạy trên các filesystem chưa được gắn **mount** hoặc hành động thực hiện có thể gây không ổn định hoặc hỏng hóc ổ cứng.
- **tune2fs**
  - Sử dụng để đặt các tham số filesystem sau khi tạo
  - **-c [#]** - đặt số lần tối đa mà một filesystem có thể được gắn kết để **fsck** tự động xảy ra
  - **-e [option]** - thay đổi hành vi của filesystem với tùy chọn được chỉ định (tiếp tục, **remount-ro**, **panic**)
  - **-g [tên nhóm]** - thêm nhóm được chỉ định như là người dùng tiềm năng của dung lượng dự trữ trên filesystem (thường dành cho root)
- **xfsprogs**
  - Công cụ và tiện ích cho filesystem XFS
  - **HÃY LƯU Ý** các tiện ích sau và chức năng chung của chúng cho kỳ thi!
- **xfs\_check**
  - Tương đương **fsck** cho filesystem XFS để kiểm tra filesystem

- **xfs\_repair**
  - Tương đương fsck cho filesystem XFS để sửa chữa filesystem
- **xfs\_metadump**
  - Tạo thông tin gỡ lỗi có thể được sử dụng bởi một bên thứ ba để hỗ trợ khôi phục một filesystem XFS khi quá trình sửa chữa thất bại
- **xfs\_growfs**
  - Được sử dụng để mở rộng một filesystem XFS (tuy nhiên, không thể thu nhỏ)

### 104.3 Kiểm soát việc gắn mount và hủy gắn umount filesystem (Gắn filesystem thủ công và tự động)

- **/media**
  - Thư mục 'cha' đa năng thường được sử dụng cho filesystem 'removeable' (ổ đĩa CD, DVD, v.v.)
- **/mnt**
  - Thư mục 'cha' đa năng thường được sử dụng để gắn kết ổ đĩa/phân vùng không thuộc filesystem cài đặt hoặc nằm ngoài cấu trúc 'root' của hệ điều hành (ví dụ, ổ sao lưu hoặc ổ đĩa dữ liệu thường)
- **blkid**
  - Lệnh sử dụng để lấy UUID (định danh duy nhất chung) cho các phân vùng ổ đĩa cục bộ trên hệ thống Linux, cũng sẽ hiển thị labels (nếu có)
  - Bảng giá trị với UUID lấy được trong **/dev/disk/by-uuid**
- **mount**
  - Khi lệnh này thực thi mà không có tham số, sẽ hiển thị các filesystems được gắn vào Linux (đọc từ **/etc/mtab**, cũng có sẵn trong **/proc/mounts**)
  - Lệnh được sử dụng để gắn **mount** thủ công một thiết bị với một filesystem hiện có trên một điểm gắn kết (thư mục)
  - **-a** - gắn mount tất cả các filesystems trong **/etc/fstab** (nếu chưa được gắn mount, sẽ KHÔNG gỡ bỏ/mount lại những filesystems đã được mount)
  - **-f** - giả lập gắn **mount** tất cả các filesystems trong **/etc/fstab**
  - **-r** - gắn mount filesystem chỉ định ở chế độ chỉ đọc
  - **-t [fstype]** - loại filesystem để gắn **mount** thiết bị được chỉ định (có thể đọc thông qua các giá trị trong superblock tự động nếu kernel hỗ trợ filesystem đó)
  - **-o [option(s)]** - chỉ định một hoặc nhiều tùy chọn nằm ngoài các giá trị mặc định (xem **/etc/fstab** cột 4 dưới đây)
  - **-w** - gắn mount ở chế độ có thể ghi (mặc định)
- **umount**
  - Giống như bạn nghĩ, hủy gắn mount filesystems được chỉ định
  - **Ví dụ - umount /mnt/data**
    - Sẽ hủy gắn mount thư mục **/mnt/data** từ bất kỳ thiết bị nào đã được gắn mount tại đó
  - **LƯU Ý:** một filesystem chỉ có thể được hủy gắn mount nếu nó không được sử dụng

- **-f** - cố gắng hủy gắn **mount filesystems** một cách bắt buộc (ngay cả khi nó đang được sử dụng hoặc có các files đang mở)
- **fuser**
  - Nếu **umount** cho biết một filesystem đang được sử dụng, lệnh này sẽ cho bạn biết người dùng nào đang sử dụng nó
  - **-m [mount]** - xác định người đang sử dụng điểm gắn kết (mount point) được chỉ định
- **Ví dụ - fuser -m /mnt/data**
  - Sẽ hiển thị điểm gắn kết (mount points) và bất kỳ PID nào đang sử dụng nó cùng với chữ cái 'c' nếu nó được sử dụng làm thư mục hiện tại (nói cách khác, một người dùng đang 'trong' thư mục)

- **/etc/fstab**

- Cấu hình mount trên một dòng duy nhất cho các filesystems local (và từ xa) sẽ được gắn **mount** khi khởi động Linux

- **Nội dung của các cột là:**

1. Thiết bị (ví dụ: **/dev/sda1** HOẶC LABEL=data HOẶC UUID=44d27f92-d3df-4207-80ea-22830afccf03)
  2. Điểm gắn mount (ví dụ: **/** HOẶC **/mnt/data**, v.v.)
  3. Filesystem - loại filesystem được hỗ trợ (ví dụ: ext3 HOẶC xfs, v.v.)
  4. Tùy chọn - danh sách phân tách bằng dấu phẩy
    - **noauto** - không tự động gắn mount khi khởi động (ngăn không cho các thiết bị ổ đĩa (External HDD) cắm ngoài được mount khi boot, do có thể gây ra vấn đề truy cập khi khởi động)
    - **defaults** - tùy chọn ổ đĩa thông thường gồm rw, setuid, dev, exec, auto, nouser, async
    - **user** - chỉ người dùng gắn mount ổ đĩa (CD-ROM/DVD/Ổ đĩa External HDD) có thể gỡ bỏ nó
    - **users** - bất kỳ người dùng nào cũng có thể gỡ bỏ ổ đĩa (CD-ROM/DVD/Ổ đĩa External HDD)
    - **ro** - gắn **mount** chỉ cho phép đọc
  5. **dump** - giá trị '0' sẽ ngăn lệnh 'dump' ảnh hưởng đến ổ đĩa
  6. **fsck** - giá trị '1' sẽ cho biết filesystem cần được kiểm tra đầu tiên
- **LƯU Ý:** **/etc/fstab** được sử dụng cho TẤT CẢ các loại **mounts**, cả mount local và từ xa, các filesystems từ xa không thuộc phạm vi của Bài kiểm tra LPIC-1, nhưng sẽ được đề cập trong Bài kiểm tra LPIC-2!

## 104.4 Quản lý hạn ngạch(Quotas) ổ đĩa(disk)

- **quota**

- Tên gói cung cấp quản lý hạn ngạch ổ đĩa

- Giới hạn người dùng sử dụng chỉ một lượng dung lượng ổ được cấu hình
- Lệnh để hiển thị bất kỳ hạn ngạch nào được áp dụng
- Các lệnh và/hoặc tùy chọn liên quan cho người dùng/filesystem:
  - **quotaon** - bật hạn ngạch
  - **quotaoff** - tắt hạn ngạch
  - **quotacheck** - xác minh việc sử dụng của người dùng và cập nhật các giá trị đã lưu
  - **edquota** - chỉnh sửa số lượng hạn ngạch người dùng
  - **aquota.user** - file nhị phân chứa thông tin hạn ngạch người dùng
  - **aquota.group** - file nhị phân chứa thông tin hạn ngạch nhóm
  - **usrquota** - tùy chọn trong file /etc/fstab để bật tính năng hạn ngạch người dùng
  - **grpquota** - tùy chọn trong file /etc/fstab để bật tính năng hạn ngạch nhóm
  - **repquota** - được sử dụng để tạo báo cáo về việc sử dụng hạn ngạch
- Các loại hạn ngạch
  - Giới hạn mềm(**Soft limit**) - hiển thị cảnh báo khi vượt quá giới hạn cho đến khi kết thúc khoảng thời gian ân hạn
  - Giới hạn cứng(**Hard limit**) - không thể vượt quá
  - Khoảng thời gian ân hạn(**Grace period**) - thời gian mà giới hạn mềm có thể vượt quá
- **LƯU Ý:** không khuyến nghị đặt hạn ngạch cho người dùng root
  - Thông thường, hạn ngạch được đặt trên filesystem **/home** (xem bài học trước về cách thiết lập cấu trúc đĩa trong quá trình cài đặt)
- Kích hoạt(bật enable) hạn ngạch cho người dùng và nhóm
  - **Ví dụ** mục dòng cấu hình trong file **/etc/fstab**
    - **/dev/sdc1 /mnt/data ext4 defaults,usrquota,grpquota 0 0**
    - Khi khởi động hoặc remount(gắn lại) filesystem, hạn ngạch có thể được kích hoạt

- Cập nhật các files 'aquota.\*'
- Ví dụ - quotacheck -avugc
  - Sẽ kích hoạt hạn ngạch người dùng và nhóm cho tất cả các filesystem được chỉ định trong file /etc/fstab và được mount với các tùy chọn đã được kích hoạt (kiểm tra bằng lệnh 'mount')
- Chỉnh sửa cài đặt hạn ngạch cho người dùng được chỉ định cấp sử dụng hạn ngạch
- Ví dụ - edquota -u người\_dùng
  - Sẽ mở trình soạn thảo mặc định (vi hoặc nano trong hầu hết các trường hợp) với các giá trị hạn ngạch có thể chỉnh sửa
  - Blocks (đang sử dụng), soft, hard, inodes (đang sử dụng), soft, hard (soft và hard áp dụng cho block hoặc inodes trực tiếp trước đó)
    - Giới hạn khối (soft, hard) sẽ đặt giới hạn DUNG LƯỢNG
    - Giới hạn inodes (soft, hard) sẽ đặt giới hạn SỐ LƯỢNG FILE
- Xác minh giới hạn mới của người dùng
  - Ví dụ - quota user
- Kích hoạt hạn ngạch cho filesystem
  - Ví dụ (ví dụ thư mục lưu dữ liệu của chúng ta /mnt/data) - quotaon /mnt/data
    - Sẽ kích hoạt các hạn ngạch đã được cấu hình từ các files aquota.\* và các filesystems được cấu hình cho các người dùng đã được chỉ định.
  - Đặt khoảng thời gian bù cho 'giới hạn mềm'
    - Ví dụ - edquota -t
      - Sẽ mở trình soạn thảo để đặt thời gian bù trong 'ngày, giờ, phút hoặc giây' cho cả giới hạn block và giới hạn inode
- Báo cáo về việc sử dụng hạn ngạch
  - Ví dụ - repquota -a

- Sẽ tạo ra một báo cáo văn bản về việc sử dụng hạn ngạch cho tất cả người dùng đã định nghĩa trên các filesystems được kích hoạt

## 104.5 Quản lý Quyền Truy cập(Permissions) và Sở hữu (Ownership)File

- **UID**
  - **User ID** (duy nhất cho mỗi người dùng trên mỗi hệ thống Linux)
- **GID**
  - **Group ID** (duy nhất cho mỗi nhóm trên mỗi hệ thống)
- Quyền truy cập được thiết lập trong các nhóm 3x3 cho mỗi file và thư mục (khi được cung cấp dưới dạng giá trị biểu tượng - ngoại trừ bit đặc biệt/sticky, xem bên dưới)
- Quyền truy cập được thiết lập bằng 3 số cho mỗi file và thư mục (khi được cung cấp dưới dạng giá trị số - ngoại trừ bit đặc biệt/sticky, xem bên dưới)
  - Quyền truy cập biểu tượng(**Symbolic permissions**)
    - **[owner rwx][group rwx][everyone rwx]** - quyền read(đọc)/write(ghi)/execute(thực thi) cho user(người dùng), group( nhóm) và everyone(mọi người)
  - Quyền truy cập số (**numeric permissions**)
    - **[owner #][group #][everyone #]** - read/write/execute-read(đọc)/write(ghi)/execute(thực thi) cho user(người dùng), group( nhóm) và everyone(mọi người)
  - Ánh xạ quyền truy cập (ví dụ về danh sách tệp)
    - Biểu tượng(**Symbolic**)
      - **r** - đọc (xem nội dung thư mục nhưng không xem thuộc tính file)
      - **w** - ghi (thêm và xóa file, bao gồm cả các file bạn không sở hữu nếu bạn có quyền ghi vào thư mục đó)
      - **x** - thực thi (có thể điều hướng đến thư mục [**cd**])
    - Số(**Numeric**)



- 4 – **read**(đọc) (xem nội dung thư mục nhưng không xem thuộc tính file)
- 2 - ghi (**write**)(thêm và xóa file, bao gồm cả các file bạn không sở hữu nếu bạn có quyền ghi vào thư mục chứa các files này)
- 1 - thực thi(**execute**) (có thể điều hướng đến thư mục [cd hoặc thay đổi thư mục])
- **LƯU Ý:** Quyền truy cập có thể được thiết lập trên file hoặc thư mục bằng cách sử dụng NOTATION số hoặc chữ
  - **Ví dụ** (symbolic) - **-rwxr-xr-- root admins 2456 Mar 25 01:05 myfile.txt**
    - Sẽ cho biết rằng file **myfile.txt**, thuộc sở hữu của root, có quyền của chủ sở hữu (**root**) là 'đọc(**read**)/ghi(**write**)/(**execute**)thực thi', quyền của nhóm (admins) là 'đọc(**read**)/(**execute**)thực thi' và quyền của mọi người khác là 'đọc(read)'
  - **Ví dụ** (numeric) - **-754 root admins 2456 Mar 25 01:05 myfile.txt**
    - Sẽ cho biết rằng file myfile.txt, thuộc sở hữu của root, có quyền của chủ sở hữu (**root**) là 'đọc(**read**)/ghi(**write**)/thực thi(**execute**) (4+2+1)', quyền của nhóm (admins) là 'đọc(read)/thực thi(execute) (4+1)' và quyền của mọi người khác là 'read (4)'
- Trong danh sách trên (symbolic), ký tự ban đầu trước quyền truy cập, chỉ ra loại đối tượng như sau:
  - **-** - file thường
  - **l** - liên kết tượng trưng(symlink)
  - **b** - thiết bị khối (như một ổ cứng)
  - **c** - thiết bị ký tự
  - **d** - thư mục
- **Hierarchy** của quyền truy cập
  - Nếu người dùng là chủ sở hữu của file - áp dụng quyền của CHỦ SỞ HỮU
  - Nếu người dùng là thành viên của nhóm sở hữu của file - áp dụng quyền của NHÓM
  - Ngược lại, áp dụng quyền của MỌI NGƯỜI

- **LƯU Ý:** chỉ ÁP DỤNG MỘT bộ quyền, bộ quyền đầu tiên mà người dùng đủ điều kiện

- **chmod**

- Công cụ(lệnh) được sử dụng để thay đổi quyền truy cập của file và thư mục

- **Ví dụ - chmod 755 myscript.sh**

- Sẽ thiết lập quyền truy cập là 'đọc(read)/ghi(write)/thực thi(execute) của người dùng(user)', 'đọc(read)/thực thi(execute) của nhóm(group)' và 'đọc(read)/thực thi(execute) của người khác(other)'

- Có thể sử dụng các giá trị biểu tượng mỗi lần một loại (u - người dùng(user), g - nhóm(group), a - tất cả(all)) cho mỗi quyền truy cập

- **Ví dụ - chmod g=rx myscript.sh**

- Sẽ thiết lập quyền truy cập là 'đọc/thực thi của nhóm'

- **LƯU Ý:** cũng có thể sử dụng +/- để thêm hoặc xóa giá trị quyền truy cập (ví dụ: g+r-w+x cho 'đọc/thực thi')

- **LƯU Ý:** có thể thiết lập nhiều loại quyền truy cập (người dùng(user)/nhóm(group)/tất cả(all)) bằng cách chèn dấu phẩy giữa vào mỗi loại (ví dụ: u=rwx,g=rw,a=r)

- **-R** - đệ quy, cho phép bạn áp dụng quyền truy cập cho toàn bộ cấu trúc thư mục, bao gồm tất cả các files và thư mục khác trong đó

- **LƯU Ý:** hãy cẩn thận khi áp dụng, việc đặt dấu gạch chéo cuối (hoặc không) có ý nghĩa khác nhau

- **Ví dụ - chmod 744 -R /home/user/data**

- Đặt quyền truy cập cho thư mục '/home/user/data' và tất cả các files và thư mục trong đó

- **Ví dụ - chmod 744 -R /home/user/data/\***

- Đặt quyền truy cập cho tất cả các files và thư mục trong /home/user/data nhưng KHÔNG thay đổi quyền truy cập của thư mục /home/user/data chính nó

- **-c** - báo cáo các files đã được thay đổi

- **-v** - hiển thị tất cả các files

- **-h** - chỉ thay đổi các liên kết tượng trưng(symbolic links), không phải là các files gốc
- **-f** - bỏ qua/không hiển thị thông báo lỗi

### • Quyền đặc biệt

- Giải quyết vấn đề của mỗi file/thư mục chỉ thuộc về một nhóm duy nhất trong một thời điểm
- Hữu ích trong các cấu hình chung nơi bạn có người dùng thuộc các nhóm khác nhau, nhưng một số người dùng là thành viên của nhiều hơn một nhóm
  - Nhóm chính (nhóm đầu tiên mà bạn là thành viên)
  - Nhóm phụ (tài khoản người dùng của bạn xuất hiện trong /etc/group)
- Đại diện 'quyền đặc biệt' của '**s**' thay thế cho quyền thực thi trong danh sách (ví dụ: **rwX** trở thành **rws**)
- **SUID** - Đặt User ID
  - Cho phép người dùng truy cập/chạy một chương trình như người sở hữu chương trình đó
  - **4** - giá trị của quyền SUID
  - **u+s** - ký hiệu tượng trưng
- **SGID** - Set Group ID
  - Cung cấp quyền sở hữu nhóm cho bất kỳ file mới nào được tạo ra trong một thư mục cho nhóm sở hữu của thư mục (hoặc khi được áp dụng cho một file, quyền truy cập/chạy một chương trình như người sở hữu nhóm)
  - **g+s** - ký hiệu tượng trưng
- **Sticky bit**
  - Được sử dụng để ngăn người dùng không sở hữu file xóa file trong một thư mục mà họ vẫn có quyền ghi của nhóm/quyền ghi toàn bộ thư mục (thay đổi quyền 'ghi' cho một thư mục - chỉ chủ sở hữu file mới có thể xóa file, người sở hữu thư mục hoặc người dùng root, ngay cả khi người dùng bình thường có quyền ghi đầy đủ vào thư mục chứa các file cũng không thể xóa file)

- **a+t** (hoặc +t) - ký hiệu tương trưng
- **umask**
  - **/etc/bashrc**
    - Cài đặt toàn bộ hệ thống Linux áp dụng cho tất cả người dùng trừ khi được chỉ định trong file **.bashrc** cục bộ của người dùng
  - **~/.bashrc**
    - Có thể đặt ở đây cho người dùng được chỉ định, ghi đè lên cài đặt hệ thống
  - Một phương pháp sửa đổi quyền mặc định được áp dụng cho các file và thư mục mà người dùng tạo ra
  - Quyền mặc định (không có umask)
    - File - **rw-rw-rw-** (666)
    - Thư mục - **rw-rw-rw-** (777)
  - Sử dụng lệnh umask sẽ cung cấp giá trị mặc định hiện tại cho người dùng
    - **Ví dụ - umask** (có thể hiển thị - **0022**)
      - **Đối với người dùng khi nhận được giá trị này, có nghĩa là**
        - **0** - các quyền đặc biệt bị che giấu(đã loại bỏ) - LƯU Ý: luôn là 0 (đã loại bỏ)
        - **0** - các quyền người dùng(**user**) bị che giấu (đã loại bỏ)
        - **2** - các quyền của nhóm(**group**) bị che giấu (đã loại bỏ)
        - **2** - các quyền người dùng khác(**other**) bị che giấu (đã loại bỏ)
      - Điều này có nghĩa là bất kỳ file nào được tạo ra sẽ có quyền mặc định là '**644**' và các thư mục được tạo bởi người dùng đó sẽ có quyền mặc định là '**755**'
  - Quyền mặc định cho người dùng đang hoạt động có thể được thay đổi tạm thời bất kỳ lúc nào
    - **Ví dụ - umask 0027**
      - Sau đó sẽ đặt quyền tạo ra cho file là **640** và thư mục là **750** cho người dùng đó

- **LƯU Ý:** Quyền sẽ được đặt lại thành mặc định khi kết thúc phiên và bắt đầu phiên tiếp theo

## 104.6 Tạo và Thay đổi Liên kết Cứng(hard links) và Liên kết Tượng trưng (symbolic links hoặc gọi là liên kết mềm soft links)

- Liên kết tượng trưng(Symbolic link)
  - Rất hữu ích, nó là một lối tắt(shortcut) trỏ từ một file này đến một file khác
  - File nhỏ (chứa inode riêng của nó và một đường dẫn đến file gốc)
  - Có thể 'bắc qua' các filesystems và ổ đĩa (cục bộ hoặc từ xa) vì nó có một inode độc lập
  - Quyền không thể sửa đổi, các thay đổi sẽ (cố gắng được) áp dụng cho file gốc (miễn là người dùng chạy lệnh có quyền thích hợp)
  - Nếu Xóa một liên kết tượng trưng (đôi khi được gọi là liên kết 'mềm'), chỉ xóa file liên kết và inode liên kết đi kèm và không ảnh hưởng đến file gốc
  - Nếu Xóa file gốc KHÔNG xóa liên kết nhưng để lại file liên kết 'mò côi'
- Liên kết cứng (hard link)
  - Thêm một tên bổ sung cho một file hoặc thư mục thực sự trỏ đến inode GỐC (thực sự là cùng dữ liệu trên ổ đĩa)
  - Nếu Xóa bất kỳ MỘT trong các liên kết cứng hiện có là chỉ xóa tham chiếu đến inode, miễn là còn tồn tại bất kỳ liên kết cứng nào khác (hoặc file gốc), dữ liệu trên inode vẫn tồn tại
  - Kết quả là do mỗi liên kết cứng chia sẻ cùng một inode, liên kết cứng KHÔNG thể 'bắc qua' các filesystems hoặc ổ đĩa (cục bộ hoặc từ xa)
  - Để xóa hoàn toàn một file với các liên kết cứng, cần xóa TẤT CẢ các liên kết cứng
  - Liên kết cứng hữu ích để sử dụng cho các files rất quan trọng 'bất biến' (ví dụ: đặt một liên kết cứng trong một thư mục chỉ có quyền truy cập bởi quản trị hệ thống system admin đến một thư mục hoặc file không bao giờ được xóa hoàn toàn)
- **LƯU Ý:** hãy nhớ là một tên và vị trí của file không được lưu trữ trong một inode
- **ln**

- Lệnh được sử dụng để tạo liên kết cứng hoặc liên kết tượng trưng
- **-s** - tạo một liên kết tượng trưng
- **Ví dụ: `ln -s myscript.sh yourscript.sh`**
  - Tạo một liên kết tượng trưng có tên là '**yourscript.sh**' trỏ đến '**myscript.sh**'
- **Ví dụ: `ln myscript.sh yourscript.sh`**
  - Tạo một liên kết cứng (mặc định) có tên là '**yourscript.sh**' trỏ đến dữ liệu trong inode được trỏ đến bởi '**myscript.sh**'
- **ls**
- **-i [tên file]** - có thể được sử dụng để tìm tất cả các liên kết cho một file

## 104.7 Tìm Kiếm Files Hệ Thống và Đặt file vào Vị Trí Chính Xác (Tiêu chuẩn phân cấp hệ thống file-Filesystem Hierarchy Standard)

- Tiêu chuẩn phân cấp hệ thống **file-Filesystem Hierarchy Standard (FHS)**
  - Tóm tắt, nó xác định cấu trúc thư mục và nội dung trong các hệ điều hành kiểu Unix (duyên nhiên, áp dụng cho Linux).
  - Chuẩn này được duy trì bởi Linux Foundation và (kể từ tháng 3 năm 2017) hiện đang ở phiên bản 3.0.
  - Hầu hết các bản phân phối Linux tự nguyện tuân theo FHS và tuân thủ khi có các bản cập nhật được phát hành, tuy nhiên, một số bản phân phối có thể có một số chênh lệch nhỏ về một số cách.
- **Cấu trúc thư mục**
  - Quy định chính là tất cả các files và thư mục xuất hiện dưới thư mục gốc (/), ngay cả khi lưu trữ trên các thiết bị vật lý hoặc ảo khác nhau (mặc dù một số thư mục không cần tồn tại nếu các hệ thống phụ thuộc không tồn tại, chẳng hạn như **X Windows**).
  - **/** - Thư mục gốc(**root directory**) và là thư mục gốc của **FHS**
  - **/bin** – Chứa Các files nhị phân của các lệnh thực thi, dành cho tất cả người dùng, phải có sẵn trong chế độ người dùng độc lập(single user mode)

- **/boot** - Chứa Các files khởi động (kernel, initrd, v.v.)
- **/dev** - Chứa Các files thiết bị phần cứng
- **/etc** - Chứa Các files cấu hình cụ thể cho Linux, không có files phân
- **/etc/opt** – Chứa Các files cấu hình cho các gói ứng dụng bên thứ 3 được lưu trữ trong /opt
- **/etc/sgml** – Chứa Các tệp cấu hình cho phần mềm xử lý SGML
- **/etc/X11** – Chứa Config files cho X Windows
- **/etc/XML** - Config files cho phần mềm xử lý XML
- **/home** – Chứa thư mục của người dùng - User's home directories
- **/lib** - Thư viện cần thiết cho các files thực thi trong /bin và /sbin
- **/lib[qual]** - Thư viện định dạng thay thế (ví dụ: 64 bit)
- **/media** - Điểm gắn kết(mount points) cho thiết bị cắm ngoài (CD, DVD, v.v.)
- **/mnt** - filesystems tạm thời được gắn mount vào Linux
- **/opt** - Phần mềm ứng dụng tùy chọn (thường là bên thứ ba)
- **/proc** - filesystem ảo cung cấp thông tin về tiến trình và kernel dưới dạng files (tương ứng với procfs)
- **/root** - Thư mục gốc của người dùng root
- **/run** - Dữ liệu biến thời gian chạy, thông tin về hệ thống Linux hoạt động kể từ lần khởi động trước
- **/sbin** - Các chương trình nhị phân hệ thống cần thiết
- **/srv** - Dữ liệu cụ thể của trang web được cung cấp bởi hệ thống cục bộ
- **/sys** - Chứa thông tin về các thiết bị được kết nối vào Linux
- **/tmp** - Files tạm, không được giữ lại qua các lần khởi động lại
- **/usr** - Phân cấp phụ cho dữ liệu người dùng chỉ đọc

- **/usr/bin** - Các chương trình nhị phân không cần thiết cho tất cả người dùng
- **/usr/include** - Các files bao gồm chuẩn (ví dụ: file.h cho C/C++)
- **/usr/lib** - Thư viện cho các chương trình nhị phân trong /usr/bin và /usr/sbin
- **/usr/lib[qual]** - Thư viện định dạng thay thế (ví dụ: 64 bit)
- **/usr/local** - Phân cấp phụ cho dữ liệu cục bộ, định rõ cho máy chủ
- **/usr/sbin** - Các chương trình nhị phân hệ thống không cần thiết
- **/usr/share** - Dữ liệu chia sẻ không phụ thuộc vào kiến trúc
- **/usr/src** - Mã nguồn (ví dụ: kernel)
- **/usr/X11R6** - Cấu hình X Windows cũ (tùy chọn)
- **/var** - Các files biến đổi có nội dung dự kiến sẽ thay đổi trong quá trình sử dụng và hoạt động bình thường của hệ thống Linux
- **/var/cache** - Dữ liệu cache ứng dụng được sử dụng do quá trình I/O hoặc tính toán tốn thời gian. Có thể xóa mà không gây mất dữ liệu.
- **/var/lib** - Thông tin trạng thái.
- **/var/lock** - File khóa giữ vị trí tài nguyên đang sử dụng.
- **/var/log** - Các files nhật ký (log)
- **/var/mail** - Các files hòm thư
- **/var/opt** - biến Dữ liệu từ các gói bổ sung được lưu trữ trong /opt
- **/var/run** - biến Dữ liệu Run-time (được thay thế bằng /run trong phiên bản 3.0)
- **/var/spool** - Spool cho các nhiệm vụ đang chờ xử lý (ví dụ: hàng đợi của máy in, /var/spool/mail cho mục khác)
- **/var/tmp** - Các files tạm được duy trì qua các khởi động lại



## 104.7 Tìm Files Hệ Thống và Đặt Files vào Vị Trí Đúng (Sử dụng find, locate, updatedb, /etc/updatedb.conf, whereis, which, và type)

- **locate**

- Tìm các files theo tên, trả về kết quả cho chuỗi được chỉ định tồn tại trong bất kỳ phần nào của đường dẫn hoặc tên file (quan trọng cho kỳ thi!)
- Có thể sử dụng biểu thức chính quy(**regular expressions**), nhưng thường được sử dụng để tìm một tên file đã biết hoặc một phần tên bao gồm **globbing**
- Sử dụng cơ sở dữ liệu locate được cập nhật bởi tiện ích **updatedb**
- **-i** [chuỗi] - không phân biệt chữ hoa/chữ thường trong file được tìm kiếm

- **/etc/updatedb.conf**

- File cấu hình cho lệnh **updatedb** quản lý cơ sở dữ liệu locate
- Chứa tên file và đuôi mở rộng đầy đủ hoặc một phần để bỏ qua hoặc 'bỏ qua' khỏi các cập nhật cơ sở dữ liệu

- **which**

- Chỉ ra lệnh nào (bao gồm đường dẫn) sẽ chạy nếu lệnh chỉ định được chạy mà không có đường dẫn tuyệt đối được cung cấp
- Có thể quan trọng nếu có nhiều phiên bản tiềm năng của một ứng dụng vì việc này sẽ hiển thị đường dẫn chứa chương trình ĐẦU TIÊN trong biến **PATH**
- **-a** [lệnh] - hiển thị tất cả các kết quả phù hợp trong PATH, theo thứ tự chúng sẽ được chạy/hiển thị

- **whereis**

- Cung cấp 'thông tin thêm' về một lệnh so với lệnh which chỉ cung cấp đường dẫn phù hợp đầu tiên cho nó
- Cung cấp tên của lệnh, tất cả các vị trí trong **PATH** và bất kỳ trang man nào liên quan đến nó

- **-b** - tìm kiếm các files nhị phân (binaries)

- **-m** - tìm kiếm các trang hướng dẫn (man pages)
- **-s** - tìm kiếm mã nguồn (sources)
- **-u** - tìm kiếm các mục không phổ biến hoặc không được tài liệu hóa
- **type**
  - Cho biết liệu một file đã được 'mở rộng' hay không
  - **Ví dụ - type ll**
    - Nếu bạn đã tạo một bí danh cho nó, bạn có thể thấy một cái gì đó như 'll được đặt là 'ls -al --color=auto''
  - **-a** - liệt kê tất cả các biến thể của lệnh được chỉ định (ví dụ: type -a ls)
- **find**
  - Như đã nói trước đó, là phương pháp tốt nhất để tìm kiếm các file trong hệ thống Linux (vì nó không phụ thuộc vào cơ sở dữ liệu), nhưng có thể tốn kém về hiệu suất
  - **-perm -[#####]** - tìm kiếm các files phù hợp với quyền được chỉ định
  - **Ví dụ - 'find /usr/bin -perm -4000 -exec cp { } /mnt/data/binbkup ;'**
    - Sẽ tìm kiếm tất cả các files được bật SUID trong thư mục **/usr/bin** và sao chép chúng vào thư mục **/mnt/data/binbkup**

## Khắc phục lỗi yum update trong Centos6

**Centos 6 đã kết thúc vòng đời và dừng hỗ trợ update, vì vậy khi chạy yum update sẽ bị lỗi**

Loaded plugins: fastestmirror, refresh-packagekit, security

Setting up Update Process

Determining fastest mirrors

YumRepo Error: All mirror URLs are not using ftp, http[s] or file.

Eg. Invalid release/repo/arch combination/

removing mirrorlist with no valid mirrors: /var/cache/yum/x86\_64/6/base/mirrorlist.txt

Error: Cannot find a valid baseurl for repo: base

**Để khắc phục lỗi này chúng ta cần edit lại repo sang vault theo hướng dẫn sau:**

### 1. Backup lại file repo cũ

```
cp /etc/yum.repos.d/CentOS-Base.repo /etc/yum.repos.d/CentOS-Base.repo_bk
```

### 2. Edit lại file repo CentOS-Base.repo

- Sử dụng lệnh:

```
vim /etc/yum.repos.d/CentOS-Base.repo
```

- Và thay đổi nội dung của file CentOS-Base.repo như sau:

```
[base]
name=CentOS-$releasever - Base
# mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=os&infra=$infra
# baseurl=http://mirror.centos.org/centos/$releasever/os/$basearch/
baseurl=https://vault.centos.org/6.10/os/$basearch/
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-6

# released updates
[updates]
name=CentOS-$releasever - Updates
#
mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=updates&infra=$infra
# baseurl=http://mirror.centos.org/centos/$releasever/updates/$basearch/
baseurl=https://vault.centos.org/6.10/updates/$basearch/
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-6

# additional packages that may be useful
[extras]
name=CentOS-$releasever - Extras
# mirrorlist=http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&repo=extras&infra=$infra
# baseurl=http://mirror.centos.org/centos/$releasever/extras/$basearch/
baseurl=https://vault.centos.org/6.10/extras/$basearch/
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-6
```

### 3. Xóa cache

```
yum clean all
```

### 4. Chạy lại lệnh update

```
yum update
```