CHƯƠNG 4: HỆ THỐNG TẬP TIN LINUX (LINUX FILE SYSTEMS)

GV: LƯƠNG MINH HUẤN

NỘI DUNG

I. Khái niệm cơ bản của file systems

II. Các loại file systems của Linux

III. Các thao tác trên file systems

I. KHÁI NIỆM CƠ BẢN CỦA FILE SYSTEMS

- **File system là các phương pháp và các cấu trúc dữ liệu** mà một hệ điều hành sử dụng để theo dõi các tập tin trên ổ đĩa hoặc phân vùng. Có thể tạm dịch file system là hệ thống tập tin.
- Để một phân vùng hoặc một ổ đĩa có thể được sử dụng như một hệ thống tập tin, nó cần được khởi tạo và các cấu trúc dữ liệu của kiểu hệ thống tập tin đó cần phải được ghi vào ổ đĩa. Quá trình này được gọi là tạo hệ thống tập tin.

I. KHÁI NIỆM CƠ BẢN CỦA FILE SYSTEMS

- Phân biệt Partition và FileSystem:
  - Đĩa cứng được chia thành những partittion
  - Các partition được format với loại filesystem tương ứng giúp người dùng có thể lưu trữ dữ liệu
- Hầu hết các loại hệ thống tập tin UNIX đều có cấu trúc chung tương tự nhau, mặc dù các chi tiết cụ thể khác nhau khá nhiều.

I. KHÁI NIỆM CƠ BẢN CỦA FILE SYSTEMS

- Filesystem có ba thành phần chính
  - Superblock
  - Inode
  - Storageblock

I. KHÁI NIỆM CƠ BẢN CỦA FILE SYSTEMS

- Superblock là cấu trúc được tạo tại vị trí bắt đầu filesystem. Lưu trữ các thông tin:
  - Kích thước và cấu trúc filesystem.
  - Thời gian cập nhật filesystem cuối cùng.
  - Thông tin trạng thái.

I. KHÁI NIỆM CƠ BẢN CỦA FILE SYSTEMS

- Inode lưu những thông tin về tập tin và thư mục được tạo trong filesystem. Mỗi tập tin tạo ra sẽ được phân bổ một inode lưu thông tin sau:
  - Loại tập tin và quyền hạn truy cập.
  - Người sở hữu tập tin.
  - Kích thước và số hard link đến tập tin.
  - Ngày và giờ chỉnh sửa tập tin lần cuối cùng.
  - Vị trí lưu nội dung tập tin trong filesystem.

I. KHÁI NIỆM CƠ BẢN CỦA FILE SYSTEMS

- Storageblock là vùng lưu dữ liệu thực sự của tập tin và thư mục. Nó chia thành những datablock. Mỗi block chứa 1024 ký tự.
  - Data Block của tập tin thường lưu inode của tập tin và nội dung của tập tin.
  - Data Block của thư mục lưu danh sách những entry gồm inode number, tên tập tin và những thư mục con.

I. KHÁI NIỆM CƠ BẢN CỦA FILE SYSTEMS

- LOẠI FILESYSTEM:
- Trong Linux tập tin dùng lưu trữ dữ liệu, bao gồm thư mục và thiết bị lưu trữ. Các tập tin trong Linux được chia làm 3 loại chính
  - Tập tin dữ liệu: là dữ liệu lưu trữ trên các thiết bị như đĩa cứng.
  - Thư mục: chứa thông tin những tập tin và thư mục con trong nó.
  - Tập tin thiết bị: hệ thống Linux xem các thiết bị như là các tập tin. Ra vào dữ liệu trên các tập tin chính là ra vào cho thiết bị.

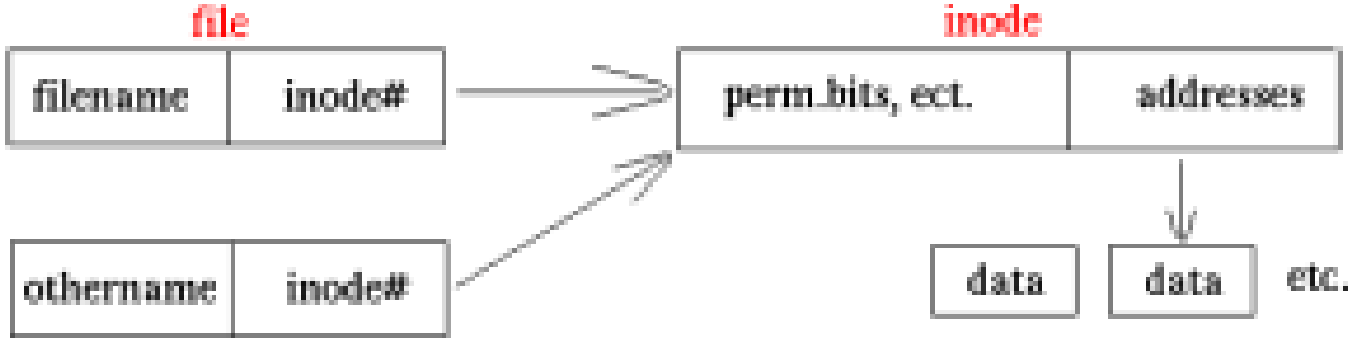
I. KHÁI NIỆM CƠ BẢN CỦA FILE SYSTEMS

- Tập tin liên kết: là tạo ra một tập tin thứ hai cho một tập tin.

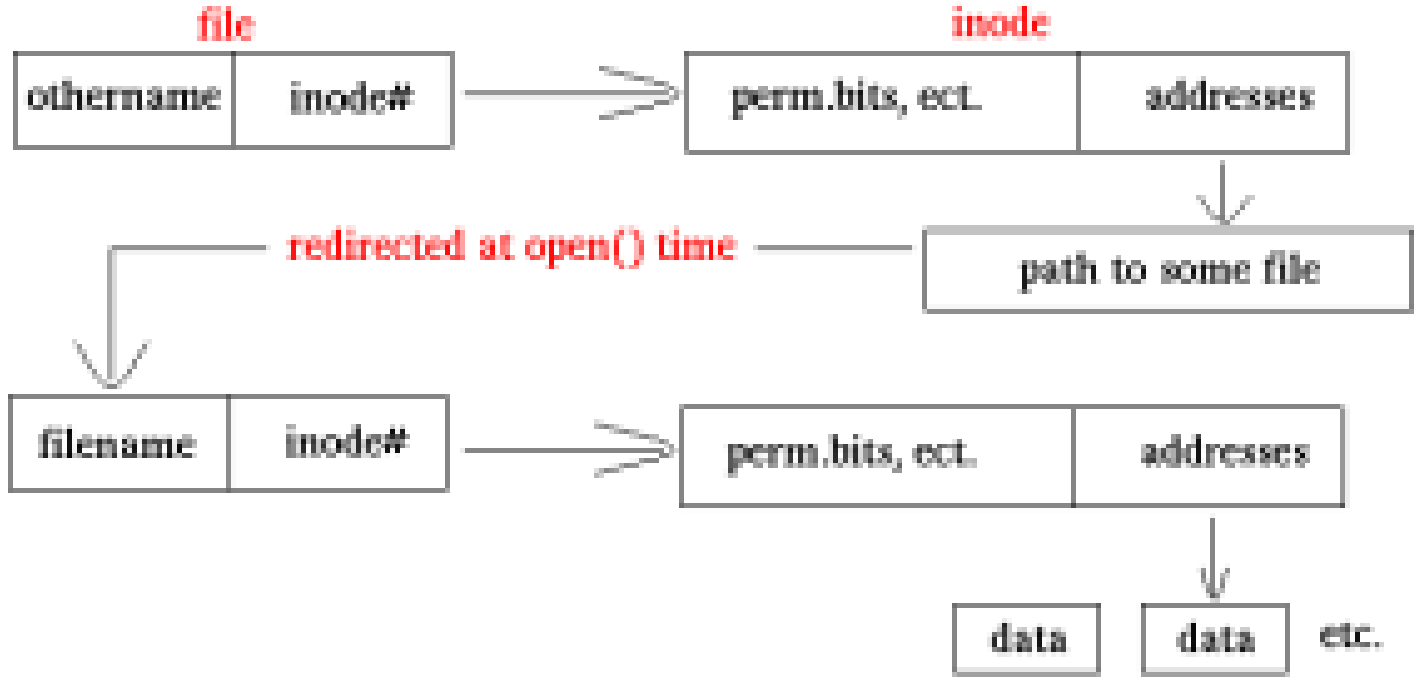
Cú pháp : **#ln [-s] <source> <destination>**

Ví dụ: #ln /usr/bill/testfile /usr/tim/testfile

- Hard link file là hình thức tạo một hay nhiều file tạm có cùng nội dung với file nguồn, các file này đều trỏ về cùng một địa chỉ lưu trữ nội dung hay nói cách khác chúng có cùng idnode number. .



- Symbolic link file là hình thức tạo một liên kết tạm dùng để trỏ về file nguồn, symbolic link giúp cho người quản trị có thể đơn giản hóa các thao tác truy cập file hệ thống, bằng cách tạo ra liên kết file trỏ về file hệ thống .



II. CÁC LOẠI FILE SYSTEMS CỦA LINUX

- Một số hệ thống tập tin Linux hổ trợ:
  - Ext
  - Ext2
  - Ext3
  - Ext4
  - BtrFS
  - ReiserFS

II. CÁC LOẠI FILE SYSTEMS CỦA LINUX

EXT

- **Ext** – Extended file system: là định dạng file hệ thống đầu tiên được thiết kế dành riêng cho Linux.
- Có tổng cộng 4 phiên bản và mỗi phiên bản lại có 1 tính năng nổi bật.
- Phiên bản đầu tiên của Ext là phần nâng cấp từ file hệ thống **Minix** được sử dụng tại thời điểm đó, nhưng lại không đáp ứng được nhiều tính năng phổ biến ngày nay.

II. CÁC LOẠI FILE SYSTEMS CỦA LINUX

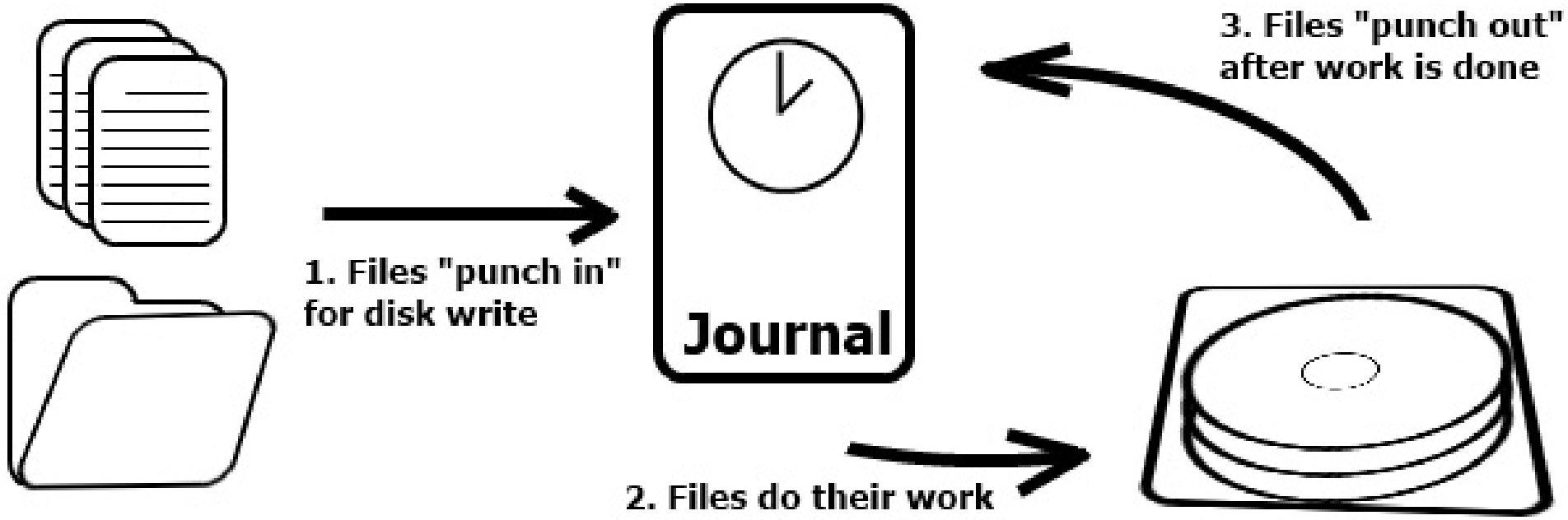
EXT2

- Ext2 thực chất không phải là file hệ thống **journaling**, được phát triển để kế thừa các thuộc tính của file hệ thống cũ, đồng thời hỗ trợ dung lượng ổ cứng lên tới 2 TB.
- Ext2 không sử dụng journal cho nên sẽ có ít dữ liệu được ghi vào ổ đĩa hơn.
- Do lượng yêu cầu viết và xóa dữ liệu khá thấp, cho nên rất phù hợp với những thiết bị lưu trữ bên ngoài như thẻ nhớ, ổ USB..

II. CÁC LOẠI FILE SYSTEMS CỦA LINUX

- **Journaling** chỉ được sử dụng khi ghi dữ liệu lên ổ cứng và đóng vai trò như những chiếc đục lỗ để ghi thông tin vào phân vùng.
- Đồng thời, nó cũng khắc phục vấn đề xảy ra khi ổ cứng gặp lỗi trong quá trình này, nếu không có journal thì hệ điều hành sẽ không thể biết được file dữ liệu có được ghi đầy đủ tới ổ cứng hay chưa.
- Một filesystem sử dụng journaling cũng được gọi là hệ thống tập tin journaling.
- Một hệ thống tập tin journaling duy trì bản ghi, biên bản, về những gì đã xảy ra trên hệ thống tập tin.

II. CÁC LOẠI FILE SYSTEMS CỦA LINUX



II. CÁC LOẠI FILE SYSTEMS CỦA LINUX

EXT3

- Ext3 về căn bản chỉ là **Ext2** đi kèm với journaling.
- Mục đích chính của **Ext3** là tương thích ngược với **Ext2**, và do vậy những ổ đĩa, phân vùng có thể dễ dàng được chuyển đổi giữa 2 chế độ mà không cần phải format như trước kia. Tuy nhiên, vẫn còn tồn tại những giới hạn của **Ext2** trong **Ext3**, và ưu điểm của **Ext3** là hoạt động nhanh, ổn định hơn rất nhiều.
- Không thực sự phù hợp để làm file hệ thống dành cho máy chủ bởi vì không hỗ trợ tính năng tạo **disk snapshot** và file được khôi phục sẽ rất khó để xóa bỏ sau này.

II. CÁC LOẠI FILE SYSTEMS CỦA LINUX

EXT4

- Ext4: cũng giống như **Ext3**, lưu giữ được những ưu điểm và tính tương thích ngược với phiên bản trước đó.
- Trên thực tế, **Ext4** có thể giảm bớt hiện tượng phân mảnh dữ liệu trong ổ cứng, hỗ trợ các file và phân vùng có dung lượng lớn...
- Thích hợp với ổ SSD so với **Ext3**, tốc độ hoạt động nhanh hơn so với 2 phiên bản **Ext** trước đó, cũng khá phù hợp để hoạt động trên server, nhưng lại không bằng **Ext3**.

II. CÁC LOẠI FILE SYSTEMS CỦA LINUX

BtrFS

- BtrFS – thường phát âm là **Butter** hoặc **Better FS**, hiện tại vẫn đang trong giai đoạn phát triển bởi Oracle và có nhiều tính năng giống với ReiserFS.
- Đại diện cho **B-Tree File System**, hỗ trợ tính năng pool trên ổ cứng, tạo và lưu trữ snapshot, nén dữ liệu ở mức độ cao, chống phân mảnh dữ liệu nhanh chóng... được thiết kế riêng biệt dành cho các doanh nghiệp có quy mô lớn

II. CÁC LOẠI FILE SYSTEMS CỦA LINUX

➢ So sánh giữa BtrFS và EXT

- Sự khác biệt cơ bản nhất giữa ext và btrfs là với ext khi thay đổi dữ liệu của một tập tin thì dữ liệu cũ sẽ bị ghi đè, do đó để an toàn chúng ta cần copy dữ liệu cũ ra một vị trí mới để lưu dự phòng.
- Tuy nhiên đối với btrfs thì khi thay đổi dữ liệu của một tập tin thì hệ thống tự động tạo ra một bản sao của tập tin và ghi các thay đổi của bạn vào bản sao đó, rồi cập nhật con trỏ nội bộ đến vị trí bản sao và tạo ghi chú nhắc nhở xóa tập tin cũ sau một khoảng thời gian nào đó.

II. CÁC LOẠI FILE SYSTEMS CỦA LINUX

- Ngoài ra btrfs còn hơn ext ở giới hạn dung lượng phân vùng và dung lượng tập tin, ở ext chỉ hỗ trợ đến 1 exbibyte (khoảng 1,152,921.5 terabytes) dung lượng phân vùng và 16 tebibytes dung lượng tập tin, còn btrfs hỗ trợ dung lượng phân vùng là 16 exbibytes và dung lượng tập tin cũng là 16 exbibytes.

II. CÁC LOẠI FILE SYSTEMS CỦA LINUX

ReiserFS

- ReiserFS: có thể coi là 1 trong những bước tiến lớn nhất của file hệ thống Linux, lần đầu được công bố vào năm 2001 với nhiều tính năng mới mà file hệ thống Ext khó có thể đạt được.
- Đến năm 2004, **ReiserFS** đã được thay thế bởi Reiser4 với nhiều cải tiến hơn nữa. Tuy nhiên, quá trình nghiên cứu, phát triển của **Reiser4** khá "chậm chạp" và vẫn không hỗ trợ đầy đủ hệ thống kernel của **Linux**.
- Đạt hiệu suất hoạt động rất cao đối với những file nhỏ như file log, phù hợp với database và server email.

II. CÁC LOẠI FILE SYSTEMS CỦA LINUX

- Ngoài ra, Linux còn hỗ trợ khá nhiều hệ thống tập tin khác như:
  - XFS được phát triển bởi **Silicon Graphics** từ năm 1994 để hoạt động với hệ điều hành riêng biệt của họ, và sau đó chuyển sang Linux trong năm 2001.
  - JFS được **IBM** phát triển lần đầu tiên năm 1990, sau đó chuyển sang **Linux**.
  - **ZFS** hiện tại vẫn đang trong giai đoạn phát triển bởi Oracle với nhiều tính năng tương tự như **Btrfs** và **ReiserFS**

III. CÁC THAO TÁC TRÊN FILE SYSTEMS

- Kiểm tra dung lượng ổ đĩa
- Tạo phân vùng ổ cứng
- Tạo file systems
- Gắn kết file systems
- Lệnh Chattr và Isattr

III.1 KIỂM TRA DUNG LƯỢNG Ổ ĐĨA

➢QUẢN LÝ DUNG LƯỢNG ĐĨA

- Để quản lý và theo dõi dung lượng đĩa ta có thể sử dụng nhiều cách khác nhau, thông thường ta dùng hai lệnh df và fdisk.

Cú pháp: **#df <option>**

#fdisk <option> <parameters>

Ví dụ:

```
[root@server /]# df -l
Filesystem           1K-blocks      Used Available Use% Mounted on
/dev/sda1              2830408   2376896    309732  89% /
none                     30608         0     30608   0% /dev/shm
[root@server /]#
```

III.1 KIỂM TRA DUNG LƯỢNG Ổ ĐĨA

- Để báo cáo lượng không gian trên đĩa được dùng bởi các tập tin và thư mục ta dùng lệnh du
- *Cú pháp:* **#du <option> <file>**
- *Ví dụ:*

```
-bash-3.2# du -sh *
0       aquota.group
0       aquota.user
5.8M    bin
4.0K    boot
4.0K    dev
27M     etc
```

III.1 KIỂM TRA DUNG LƯỢNG Ổ ĐĨA

➢KIỂM TRA FILESYSTEM VỚI fsck

Cú pháp : **#fsck <option> <partition>**

Ví dụ : #fsck –V –a /

Bảng mô tả các tùy chọn:

Tùy chọn	Mô tả
-A	Duyệt khắp tập tin /etc/fstab và cố gắng kiểm tra tất cả các hệ thống tập tin chỉ trong một lần duyệt.
-V	Chế độ chi tiết. Cho biết lệnh fsck đang làm gì.
-t loại-fs	Xác định loại hệ thống tập tin cần kiểm tra.
-a	Tự động sửa chữa hệ thống tập tin mà không cần hỏi.
-l	Liệt kê tất cả các tên tập tin trong hệ thống tập tin.
-r	Hỏi trước khi sửa chữa hệ thống tập tin.
-s	Liệt kê các superblock trước khi kiểm tra hệ thống tập tin.

III.2 TẠO PHÂN VÙNG Ổ CỨNG

- Để kiểm tra xem một ổ cứng đã gắn vào máy tính của mình hay chưa, ta dùng lệnh fdisk như sau:

#fdisk –l

- Khi đó, hệ thống sẽ báo ra có bao nhiêu ổ cứng, partition được gắn trên máy tính.
- Ví dụ:

```
# fdisk -l
Disk /dev/hda: 80.0 GB, 80060424192 bytes
255 heads, 63 sectors/track, 9733 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

Device Boot Start End Blocks Id System
/dev/hda1 1 262 2104483+ 82 Linux swap / Solaris
/dev/hda2 * 263 2873 20972857+ 83 Linux
/dev/hda3 2874 9733 55102950 83 Linux

Disk /dev/sda: 40.0 GB, 40007761920 bytes
64 eads, 32 sectors/track, 38154 cylinders
65 Units = cylinders of 2048 * 512 = 1048576 bytes
```

III.2 TẠO PHÂN VÙNG Ổ CỨNG

- Để format partition, ta dùng lệnh:

  Fdisk <đường dẫn thư mục chứa ổ đĩa>

- Ví dụ:

```
[root@vnitnews ~]# fdisk /dev/sdb
```

III.2 TẠO PHÂN VÙNG Ổ CỨNG

- Khi đó, ta sẽ có một số tùy chọn.
- Tại đây, dựa trên các tùy chọn, ta sẽ lựa chọn để tạo ra một partition phù hợp.

```
Building a new DOS disklabel with disk identifier 0xb395e9b3.
Changes will remain in memory only, until you decide to write them.
After that, of course, the previous content won't be recoverable.

Warning: invalid flag 0x0000 of partition table 4 will be corrected by w(rite)

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to
         switch off the mode (command 'c') and change display units to
         sectors (command 'u').

Command (m for help): m
Command action
   a   toggle a bootable flag
   b   edit bsd disklabel
   c   toggle the dos compatibility flag
   d   delete a partition
   l   list known partition types
   m   print this menu
   n   add a new partition
   o   create a new empty DOS partition table
   p   print the partition table
   q   quit without saving changes
   s   create a new empty Sun disklabel
   t   change a partition's system id
   u   change display/entry units
   v   verify the partition table
   w   write table to disk and exit
   x   extra functionality (experts only)

Command (m for help):
```

III.3 TẠO FILE SYSTEMS

- Sau khi thực hiện xong phân vùng ổ đĩa (format partition), ta vẫn chưa sử dụng được ổ đĩa này.
- Để sử dụng được, ta cần phải format phân vùng này với một hệ thống tập tin đã lựa chọn.
- Dùng lênh mkfs để tạo file hệ thống

#mkfs [option] Device name (partition)

- Một số option thường dùng
  - -t Chỉ định type cho file hệ thống. Nếu không chỉ định type thì mặc định sẽ dùng là ext2

Ví dụ:

#mkfs.ext2 : định dạng partition theo loại ext2.

#mkfs.ext3 : định dạng partition theo loại ext3.

#mkfs –t ext2 /dev/hda1

III.3 TẠO FILE SYSTEMS

➢ Ví dụ:

```
[root@vnitnews ~]# mkfs.ext3 /dev/sdb1
mke2fs 1.41.12 (17-May-2010)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
655360 inodes, 2620595 blocks
131029 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=2684354560
80 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632

Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 39 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
[root@vnitnews ~]#
```

III.4 GẮN KẾT FILE SYSTEMS

- **Mount** là một quá trình mà trong đó hệ điều hành làm cho các tập tin và thư mục trên một thiết bị lưu trữ (như ổ cứng, CD-ROM hoặc tài nguyên chia sẻ) có thể truy cập được bởi người dùng thông qua hệ thống tập tin của máy tính.
- Quá trình mount bao gồm việc hệ điều hành được truy cập vào phương tiện lưu trữ, công nhận, đọc và xử lý cấu trúc hệ thống tệp cùng với siêu dữ liệu trên nó, sau đó, đăng ký chúng vào thành phần hệ thống tệp ảo (VFS).
- Vị trí đăng ký trong VFS của phương tiện mới được mount gọi là **điểm mount**. Đây là điểm mà người dùng có thể truy cập tập tin, thư mục của phương tiện sau khi quá trình mount hoàn thành.

III.4 GẮN KẾT FILE SYSTEMS

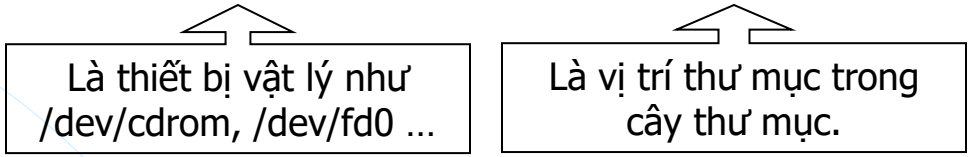
- Ngược với mount là **unmount**, trong đó, hệ điều hành huỷ tất cả quyền truy cập tập tin, thư mục của người dùng tại điểm mount, ghi tiếp những dữ liệu người dùng đang trong hàng đợi vào thiết bị, làm mới siêu dữ liệu hệ thống tệp, sau đó, tự huỷ quyền truy cập thiết bị và làm cho thiết bị có thể tháo ra an toàn.
- Bình thường, khi tắt máy tính, mỗi thiết bị lưu trữ sẽ trải qua quá trình unmount để đảm bảo rằng tất cả các dữ liệu trong hàng đợi được ghi và để duy trì tính toàn vẹn của cấu trúc hệ thống tệp trên các phương tiện.

III.4 GẮN KẾT FILE SYSTEMS

➢ MOUNT VÀ UMOUNT FILESYSTEM

- **Mount thủ công**

Cú pháp : **#mount –t <device_name> <mount_point>**



Một số tùy chọn:

-v : chế độ chi tiết

-w: mount hệ thống tập tin với quyền đọc và ghi.

-r : mount hệ thống tập tin với quyền đọc.

-t loai-fs : xác định hệ thống tập tin đang mount : ext2, ext3, ...

-a : mount tất cả hệ thống tập tin khai báo trong /etc/fstab.

-o remount <fs> : chỉ định việc mount lại 1 filesystem nào đó.

III.4 GẮN KẾT FILE SYSTEMS

- **Mount tự động**

Tập tin /etc/fstab liệt kê các hệ thống cần được mount tự động.

LABEL=/	/	ext3	defaults	1	1
LABEL=/boot	/boot	ext3	defaults	1	1
None	/dev/pts	devpts	gid=5,mode=620	0	0

cột 1: chỉ ra thiết bị hoặc hệ thống tập tin cần mount

cột 2: xác định mount point

cột 3: chỉ ra loại filesystem như : vfat, ext2 …

cột 4: các tùy chọn phân cách nhau bởi dấu phẩy.

cột 5: xác định thời gian để lệnh dump sao chép hệ thống tập tin.

cột 6: khai báo lệnh fsck biết thứ tự kiểm tra các hệ thống tập tin.

III.4 GẮN KẾT FILE SYSTEMS

- **Umount hệ thống tập tin**

  Cú pháp : **#umount <device_name> <mount_point>**

  Ví dụ: Loại bỏ tất cả các filesystem đang mount

  ```
  #umount -a
  ```

  Lưu ý: umount không loại bỏ những hệ thống tập tin đang sử dụng

III.5 LỆNH CHATTR VÀ ISATTR

- Các tập tin trên các hệ thống tập tin mở rộng, phổ biến của Linux (như *ext2*, *ext3*, *ext4* …) có thể được làm cho không thể chỉnh sửa bằng việc sử dụng 1 loại thuộc tính cụ thể.
- Khi 1 tập tin ở trạng thái bất biến, không thể chỉnh sửa (***immutable***), bất cứ tài khoản người dùng nào cũng không thể xóa các tập tin này cho đến chi trạng thái bất biến này được loại khỏi tập tin (kể cả tài khoản root)

III.5 LỆNH CHATTR VÀ ISATTR

- Lệnh ***chattr*** có thể được dùng để làm cho tập tin không thể chỉnh sửa.
- Một tập tin có thể được làm cho bất biến bằng việc sử dụng lệnh sau:

  #chattr +i file
- Để làm cho tập tin có thể chỉnh sửa trở lại, loại bỏ thuộc tính bất biến ra khỏi tập tin như sau:

  #chattr -i file

III.5 LỆNH CHATTR VÀ ISATTR

- Sử dụng lệnh lsattr để hiển thị thuộc tính của các tập tin nhị phân trong hệ thống của bạn tại các vị trí như */bin*, */sbin* và */usr/bin*, như ví dụ:

#lsattr /usr/bin

```
------------e- /usr/bin/watch
------------e- /usr/bin/gvfs-mount
------------e- /usr/bin/dh_installudev
lsattr: Operation not supported While reading flags on /usr/bin/nawk
------------e- /usr/bin/dbus-cleanup-sockets
------------e- /usr/bin/dh_python
------------e- /usr/bin/pnmdepth
------------e- /usr/bin/hcitool
lsattr: Operation not supported While reading flags on /usr/bin/createdb
------------e- /usr/bin/dh_undocumented
------------e- /usr/bin/sputoppm
------------e- /usr/bin/iceauth
------------e- /usr/bin/gs
------------e- /usr/bin/gettext

(josh)-(jobs:0)-(~)
(! 136)->
```