Linux và Phần mềm Mã nguồn mở

Bài 4: Hệ thống file, tập tin và dẫn hướng vào/ra dữ liệu

Nhắc lại và chú ý

- Cấu trúc lệnh linux thường gồm 3 khối

  <lệnh> <lựa chọn> <tham số>

  - <lệnh>: cố định, phải học và nhớ
  - <lựa chọn>: tùy vào từng lệnh, thường bắt đầu với dấu -
  - <tham số>: tùy vào từng lệnh
- Với hầu hết các lệnh, có thể xem hướng dẫn sử dụng đơn giản (tiếng Anh) bằng cách thêm tham số --help ngay sau lệnh (ngoài ra có thể lệnh man để xem hướng dẫn chi tiết hơn)
- Các câu lệnh cung cấp thông tin về hệ thống

Nhắc lại và chú ý

- Các loại kí hiệu thay thế trong khi viết lệnh (*?~.)
- Các lệnh thao tác tập tin và thư mục
- Các lệnh làm việc với nội dung tập tin
- Các loại người dùng trong linux
- Khái niệm nhóm người dùng
- Thông tin về tập tin / thư mục
- Các loại quyền và phân quyền
- Các loại tập tin & quy cách đặt tên

Lệnh chmod (bổ sung)

- Nếu việc tính toán quyền phức tạp, có thể dùng hệ thống kí hiệu để thay thế
- Định danh quyền truy cập
  - u user, chủ sở hữu file
  - g group, nhóm có user là thành viên
  - o others, các user khác trên hệ thống
  - a all, tất cả user (u, g và o)
- Tác vụ trên quyền truy cập
  - + thêm quyền
  - - loại bỏ quyền
  - = gán quyền

Lệnh chmod (bổ sung)

- Cú pháp: chmod [options] mode file
- Option -R: thay đổi cả trong thư mục con
- Ví dụ sử dụng chmod
  - g+w thêm quyền ghi cho group
  - o-rwx loại bỏ tất cả các quyền của others
  - +x thêm quyền thực thi cho tất cả
  - a+rw thêm quyền ghi cho tất cả
  - ug+r thêm quyền đọc cho user và group
  - o=x chỉ cho phép thực thi với others

Lệnh chmod (bổ sung)

Bài tập: hãy giải thích các câu lệnh dưới đây

```
chmod -x *.php
chmod -R ug+rw lecture
chmod u=rwx,ug=r desktop.jpg
chmod 644 homelist.txt
chmod 755 myprogram
chmod 777 /tmp/tmp
chmod -R 777
```

Nội dung

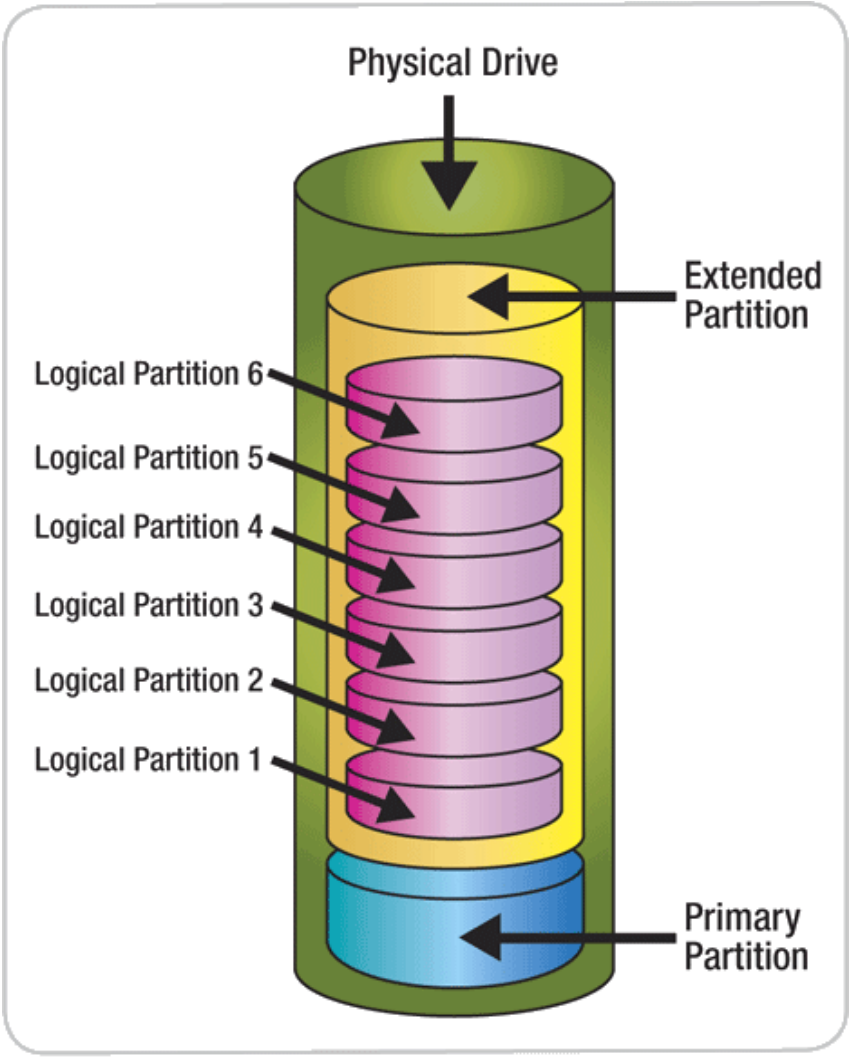
1. Một số hệ thống file và hệ điều hành linux
   - Hệ thống file
   - Hệ thống file của linux
   - Gắn kết (mount) hệ thống file
   - File cấu hình /etc/fstab
2. Một số thủ thuật làm việc với tập tin
   - Cắt & ghép tập tin
   - Xem và chỉnh sửa tập tin với “vi”
   - Một số tập tin hệ thống
3. Dẫn hướng vào/ra dữ liệu

Phần 1

Một số hệ thống file và hệ điều hành linux

Hệ thống file

- Ổ cứng (vật lý)
- Phân vùng (partition)
- Hệ thống file bố trí trên từng phần vùng
- Với linux, các phân vùng ánh xạ vào hệ thống file thống nhất
- Có những hệ thống file khác không tuân theo nguyên tắc này



Một số hệ thống file thông dụng

Tên	Cỡ file tối đa	Cỡ phần vùng tối đa	Nhật ký thao tác	Ghi chú
Fat16	2 GiB	2 GiB	No	Đã cũ
Fat32	4 GiB	8 TiB	No	Đã cũ
NTFS	2 TiB	256 TiB	Yes	Dùng cho Windows, linux hỗ trợ đọc-ghi
ext2	2 TiB	32 TiB	No	Đã cũ
ext3	2 TiB	32 TiB	Yes	Dùng phổ biến trong đa số hệ thống linux
ext4	16 TiB	1 EiB	Yes	Mới nhất của linux
reiserFS	8 TiB	16 TiB	Yes	Đã dừng phát triển
JFS	4PiB	32PiB	Yes (metadata)	Phát triển bởi IBM
XFS	8 EiB	8 EiB	Yes (metadata)	Phát triển bởi SGI, chú trọng đến sự ổn định

Chú ý: GiB = Gibibyte (1024 MiB) / TiB = Tebibyte (1024 GiB) / PiB = Pebibyte (1024 TiB) / EiB = Exbibyte (1024 PiB)

Hệ thống file của linux

- Trên linux, có thể xem các phân vùng hiện tại bằng:

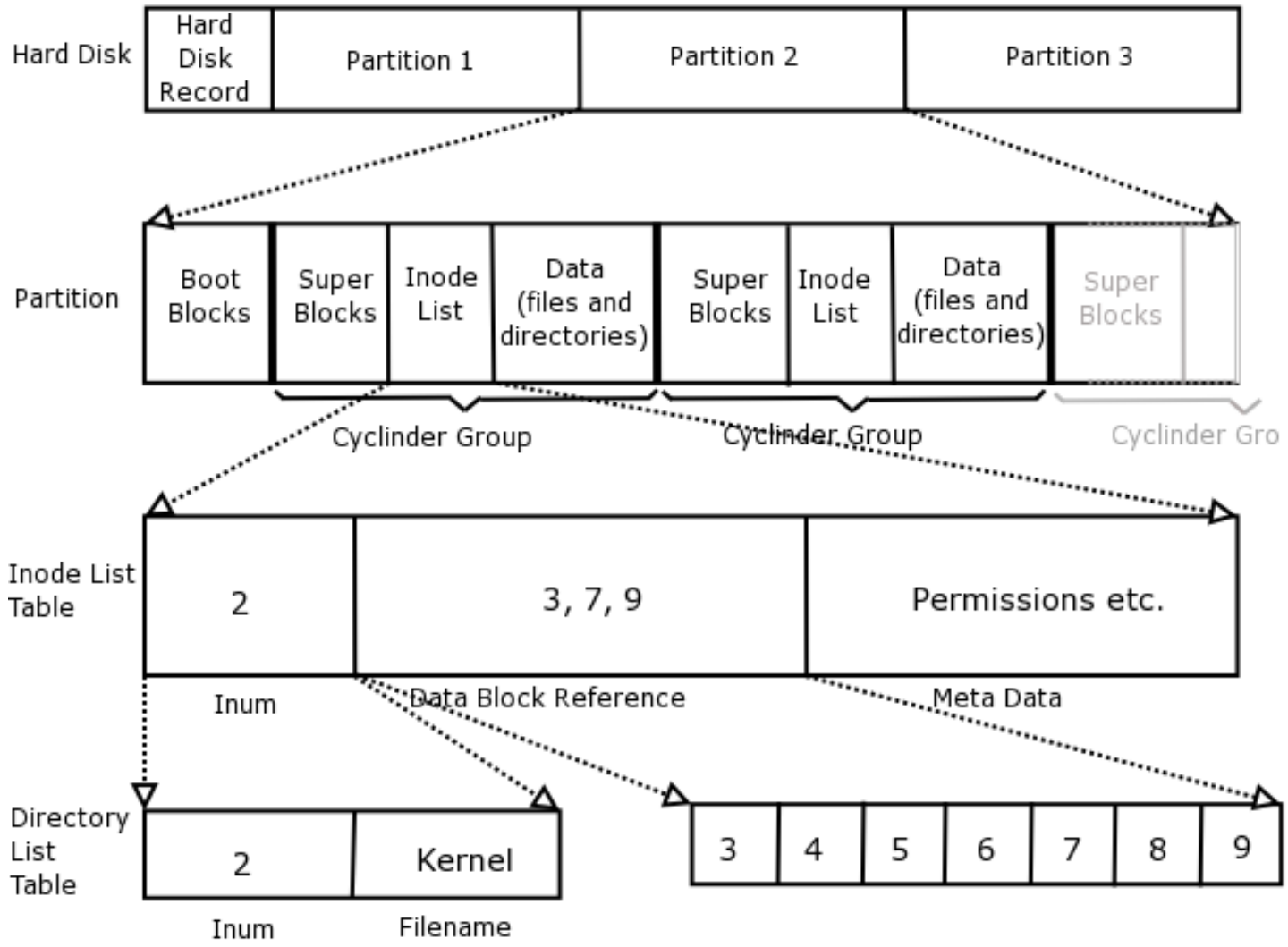
  ```
  cat /proc/partitions
  ```

- Hoặc xem thông tin chi tiết hơn (quyền root):

  ```
  fdisk -l
  ```

- Một số khái niệm cần biết về cơ chế cấp phát tài nguyên của hệ thống file trên linux
  - Super block
  - Storage block
  - i-node (ls –lia)

Hệ thống file của linux



Super block và storage block

- Super block:
  - Là cấu trúc được tạo tại vị trí bắt đầu hệ thống file
  - Lưu trữ các thông tin:
    - Thông tin về block size, free block
    - Thời gian gắn kết (mount) cuối cùng của tập tin
    - Thông tin trạng thái tập tin
- Storage block:
  - Là vùng lưu dữ liệu thực sự của tập tin và thư mục
  - Chia thành những data block (thường là 1024 byte)
    - Data block của tập tin lưu i-node và nội dung của tập tin
    - Data block của thư mục lưu danh sách những entry gồm i-node number, tên tập tin và những thư mục con

i-node

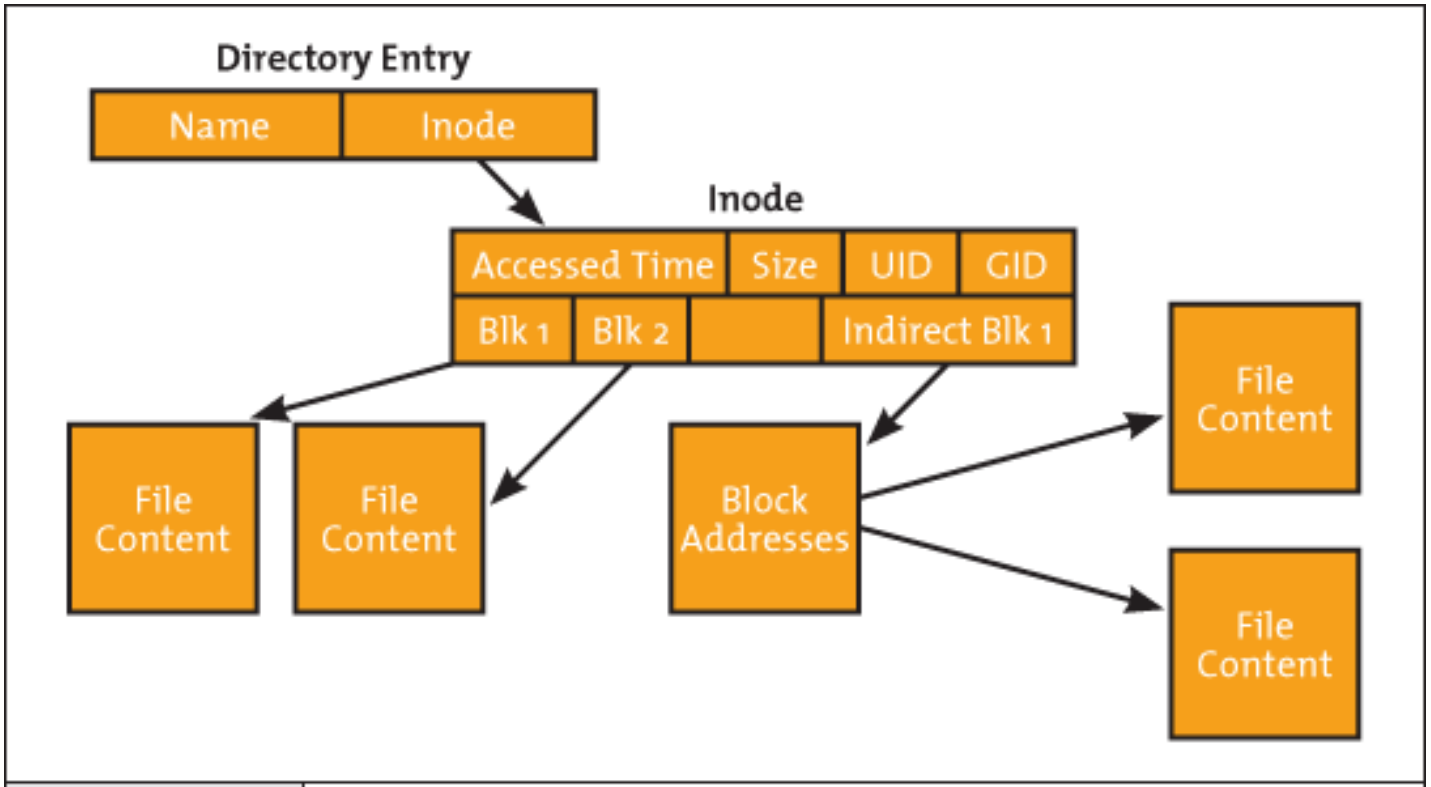


FIGURE 1 RELATIONSHIP BETWEEN THE DIRECTORY ENTRY, AN INODE, AND BLOCKS OF AN ALLOCATED FILE

i-node

- Lưu những thông tin về tập tin và thư mục được tạo trong hệ thống (nhưng không lưu tên)
- Mỗi tập tin có một i-node 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

Một lỗi phổ biến với linux: hệ thống file còn chỗ trống nhưng không tạo được file mới vì hết i-node

Gắn kết (mount) hệ thống file

- Khi lắp một thiết bị lưu trữ mới, linux có thể tự động nhận ra thiết bị đó (!) nhưng sẽ không tự động đưa thiết bị đó vào hệ thống file
- Người dùng phải yêu cầu hệ thống ánh xạ thiết bị đó vào một thư mục nào đó trong hệ thống file
- Công việc này gọi là "mount"
- Có thể xem những thiết bị nào đã được mount vào hệ thống và chúng nằm ở đâu bằng "mount -l"
- Có thể gỡ thiết bị đã mount bằng lệnh:

umount <tên thiết bị hoặc tên đường dẫn>

Gắn kết (mount) hệ thống file

- Cú pháp lệnh mount:

  mount [-t type] <device> <directory>

- Trong đó:
  - -t type kiểu hệ thống file trên thiết bị
  - device tên thiết bị vật lý muốn gắn kết
  - directory tên thư mục muốn ánh xạ tới
- Có thể bỏ qua tham số <directory>
- Trong hầu hết các tình huống, không cần chỉ ra kiểu hệ thống file trên thiết bị muốn gắn kết (hệ thống tự nhận ra)

Gắn kết (mount) hệ thống file

- Ví dụ: ta có ổ usb có dạng FAT32 đã được hệ thống phát hiện và đặt tên /dev/sdb1, muốn gắn kết ổ này thành thư mục /mnt/usb
- Các bước thực hiện như sau:

```
mkdir /mnt/usb (nếu chưa tồn tại)
mount -t vfat /dev/sdb1 /mnt/usb
```

- Sau khi thực hiện những lệnh này, mọi lệnh đọc/ghi vào thư mục /mnt/usb sẽ đọc/ghi vào ổ usb
- Gỡ ổ usb trên: `umount /dev/sdb1` hoặc `umout /mnt/usb`

Tự động gắn kết (auto mount)

- Đôi khi việc gắn kết cần thực hiện ngay khi hệ thống khởi động (chẳng hạn như cần dữ liệu trên ổ đĩa)
- Linux lưu danh sách những thiết bị được gắn kết khi khởi động trong file "/etc/fstab"
- Có thể xem file này bằng lệnh: "`cat /etc/fstab`"
- Nếu chỉnh sửa file này sẽ thay đổi cấu hình tự động gắn kết của hệ thống (đây là cách quản trị viên hay sử dụng)
- Nếu sửa file xong muốn tự động gắn kết luôn (mà không khởi động lại máy) dùng: "`mount -a`"

File cấu hình /etc/fstab

- Hình chụp một file /ect/fstab thông thường

```
[root@localhost root]# more /etc/fstab
LABEL=/                 /                       ext3    defaults        1 1
none                    /dev/pts                devpts  gid=5,mode=620  0 0
/dev/hda2               /home                   ext3    defaults        1 2
none                    /proc                   proc    defaults        0 0
none                    /dev/shm                tmpfs   defaults        0 0
/dev/hda3               swap                    swap    defaults        0 0
/dev/fd0                /mnt/floppy             auto    noauto,owner,kudzu 0 0
/dev/cdrom              /mnt/cdrom              udf,iso9660 noauto,owner,kudzu,r
o 0 0
```

- Chứa các dòng khai báo thiết bị / phân vùng được mount tự động
- Mỗi thiết bị trên một dòng
- Mỗi dòng có 6 cột tham số

File cấu hình /etc/fstab

- Cột 1 – /dev/hda2 – tên phân vùng được kết gán
- Cột 2 – /home – ánh xạ đến thư mục /home
- Cột 3 – ext3 – hệ thống file là ext3 (nên để auto)
- Cột 4 – defaults – các lựa chọn mặc định để một phân vùng hoạt động bình thường (rw, suid, dev, exec, auto, nouser, async,…)
- Cột 5 – 1/0 – có sao lưu phân vùng này khi chạy lênh dump hay không?
- Cột 6 – 1/0 – có cần kiểm tra phân vùng này (bằng lệnh fsck) khi khởi động hay ko? (lớn hơn 1 là có)

File cấu hình /etc/fstab

- Các tham số ứng với cột 4 (default):
  - rw / ro: cho đọc ghi (read write) / chỉ đọc (read only)
  - exec / noexec: cho phép / không cho thực thi các file nhị phân (chẳng hạn như dùng với ổ đĩa của Windows)
  - auto / noauto: mount tự động / không tự động mount phân vùng tương ứng (khi boot hoặc gõ mount -a)
  - nouser / user: không cho phép / cho phép các user khác được mount thiết bị ngoài root
  - async / sync: không / có ghi dữ liệu lên đĩa vật lý ngay khi thực hiện các thao tác ghi dữ liệu (sync nên được dùng với các thiết bị removeable)

Phần 2

Một số thủ thuật làm việc với tập tin

Cắt & ghép tập tin

- Chia tập tin lớn ra thành nhiều tập tin con:

```
split -b20m largefile smallfile
```

- Câu lệnh trên sẽ chia "largefile" thành các file con có kích thước 20mb, với các tên tăng dần:
  - smallfilea
  - smallfileb
  - …
- Ghép các tập tin con để phục hồi lại tập tin gốc:

```
cat smallfile* > largefile
```

“vi”: xem và chỉnh sửa file

- Sử dụng công cụ vi: vi <tên file>
- Là công cụ chuẩn của linux, có 3 chế độ làm việc:
  - Lệnh (command mode): phím nhập vào là lệnh
  - Soạn thảo (edit mode): phím nhập vào thành nội dụng
  - Dòng lệnh (“:” mode): thực hiện các lệnh sau dấu “:”
- Thông dụng nhất:
  - Dùng các phím mũi tên để di chuyển quanh văn bản
  - Nhấn INS chỉnh sửa văn bản
  - Nhấn ESC thoát khỏi chế độ chỉnh sửa
  - :wq lưu và thoát
  - :q! thoát (không lưu)

“vi”: chế độ soạn thảo

- a chèn ngay sau vị trí con trỏ
- A chèn vào cuối dòng
- i chèn ngay trước vị trí con trỏ
- I chèn vào đầu dòng
- o chèn một hàng mới dưới vị trí con trỏ
- O chèn một hàng mới trên vị trí con trỏ
- r thay thế ký tự tại vị trí con trỏ
- R thay thế bắt đầu từ vị trí con trỏ
- S thay thế dòng hiện tại
- C thay thế từ vị trí con trỏ đến cuối dòng

“vi”: di chuyển – theo ký tự

- Sử dụng phím mũi tên để di chuyển con trỏ từng ký tự (tuỳ hỗ trợ của terminal)
- h,j,k,l thay thế cho các phím mũi tên
- [n]h dịch trái [n] ký tự
- [n]j dịch xuống [n] ký tự
- [n]k dịch lên [n] ký tự
- [n]l dịch phải [n] ký tự

Lưu ý: lệnh có thể thêm chữ số đứng trước để chỉ số lần lặp lại lệnh đó

“vi”: di chuyển – theo màn hình

- Sử dụng các phím PageUP, PageDown để cuộn một khung màn hình (tuỳ hỗ trợ của terminal)
- ctrl + F cuộn xuống 1 khung màn hình
- ctrl + B cuộn lên 1 khung màn hình
- ctrl + D cuộn xuống 1/2 khung màn hình
- ctrl + U cuộn lên 1/2 khung màn hình

“vi”: di chuyển – theo từ, dòng

- G đến dòng cuối file
- [n]G đến cuối file hoặc dòng thứ [n]
- :n đến dòng thứ n
- gg đến dòng đầu file
- $ về cuối dòng (End)
- ^ về đầu dòng (Home)
- [n]w tới [n] từ (word)
- [n]b lùi [n] từ
- e về cuối từ

“vi”: nhóm lệnh xóa

- [n]x xoá [n] ký tự tại vị trí con trỏ (Del)
- X xoá ký tự trước vị trí con trỏ (Backspace)
- [n]dw xoá [n] từ
- D xoá từ vị trí con trỏ đến cuối dòng
- [n]dd xoá [n] dòng từ vị trí con trỏ
- d$ xoá đến cuối dòng
- dG xoá đến cuối file

Văn bản bị xoá luôn được lưu tạm trong một bộ đệm (ý nghĩa giống như “cut”)

“vi”: copy, cut, paste

- [n]yw copy [n] từ vào bộ đệm (yank)
- [n]yy copy (yank) [n] dòng vào bộ đệm
- [n]dw cắt [n] từ vào bộ đệm
- [n]dd cắt [n] dòng vào bộ đệm
- p dán từ bộ đệm vào sau con trỏ
- P dán từ bộ đệm vào trước con trỏ

“vi”: một số lệnh đặc biệt

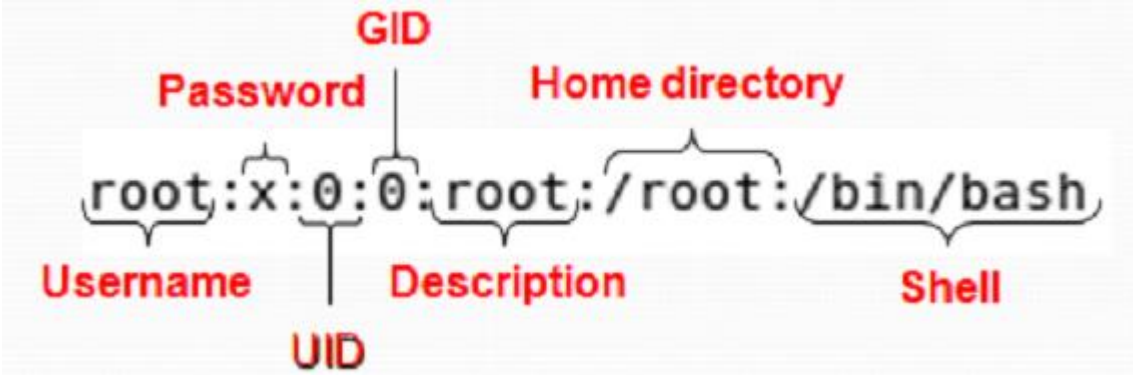
- J nối dòng hiện tại và dòng kế
- u undo thay đổi cuối cùng
- U khôi phục dòng như trước khi bị sửa đổi
- ^R redo thay đổi sau đó
- . lặp lại thay đổi cuối cùng
- /[pattern] tìm kiếm theo hướng tới
- ?[pattern] tìm kiếm theo hướng lùi
- n lặp lại tìm kiếm theo cùng chiều
- N lặp lại tìm kiếm theo ngược chiều

“vi”: lưu và thoát tập tin

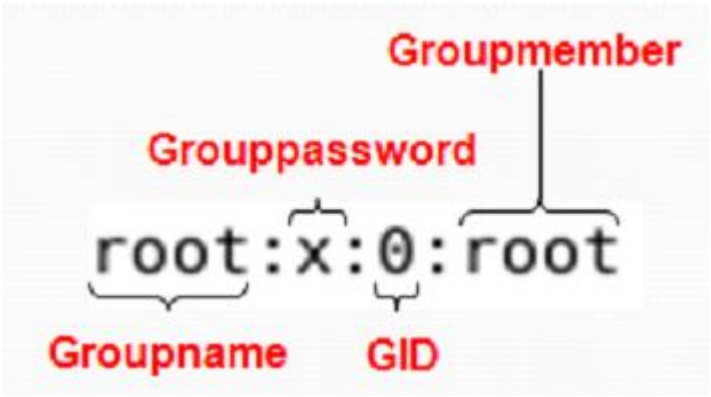
- ZZ ghi nội dung bộ đệm ra file và thoát
- x ghi nội dung bộ đệm ra file và thoát
- :w ghi nội dung bộ đệm ra file
- :q! huỷ phiên làm việc hiện tại và thoát
- :wq ghi nội dung bộ đệm ra file và thoát
- ! buộc thi hành lệnh (force operation)

Các file cấu hình

- /etc/passwd:



- /etc/group:



Cấu trúc 1 dòng của /etc/passwd

- username:password:uid:gid:gecos:homedir:shell
- Trong đó:
  - username tên dùng để login
  - password mật khẩu đã được mã hóa
  - uid user ID
  - gid group ID
  - gecos thông tin thêm về user (ghi chú)
  - homedir thư mục home của user
  - shell shell đăng nhập của người dùng
- Ví dụ: root:x:0:0:root,home:/root:/bin/bash

Cấu trúc 1 dòng của /etc/shadow

- usr:pwd:d1:d2:d3:d4:d5:d6:reserved
- Trong đó:
  - usr tên trong /etc/passwd
  - pwd mật khẩu đã được mã hoá
  - d1 ngày thay đổi mật khẩu gần nhất
  - d2 số ngày cần chờ để có thể thay đổi mật khẩu
  - d3 số ngày mật khẩu có giá trị
  - d4 số ngày cảnh báo thay đổi mật khẩu
  - d5 số ngày tài khoản hết hạn đăng nhập
  - d6 ngày mà tài khoản bị khoá
- Ngày trong linux tính theo mốc từ 1/1/1970

Cấu trúc 1 dòng của /etc/shadow

- Các trường có thể để trống
- Tài khoản bị khóa nếu có ký tự ! đứng trước pwd
- Tài khoản không có mật khẩu và không để đăng nhập hệ thống nếu có giá trị !! ở trường pwd
- Tài khoản không được phép đăng nhập hệ thống nếu có giá trị * ở trường pwd

```
root:$1$dxtC0nf$SCITrkSH5tjw0s/:12148:0:99999:7:::
daemon:*:12148:0:99999:7:::adm:*:12148:0:99999:7:::
nobody:*:12148:0:99999:7:::
xfs:!!:12148:0:99999:7:::
```

Cấu trúc 1 dòng của /etc/group

- groupname:password:gid:members
- Trong đó:
  - groupname chuỗi ký tự bất kỳ, xác định tên group
  - password mật khẩu (tùy chọn)
  - gid group id
  - members danh sách thành viên, cách nhau bằng “,”
- Ví dụ:

```
root:x:0:
bin:x:1:bin,daemon
student:x:500:
```

Phần 3

Dẫn hướng vào/ra dữ liệu

Các luồng vào ra dữ liệu chuẩn

- Khái niệm "luồng": dãy dữ liệu được xử lý tuần tự
  - Tương tự như khái niệm stream trong lập trình C++
- "luồng vào": dãy dữ liệu được gửi vào chương trình
- "luồng ra": dữ liệu kết quả, được chương trình gửi trả lại từng thành phần cho chương trình gọi
- Khi thực thi một chương trình trên linux, hệ thống mặc định tạo 3 luồng cho chương trình đó
  - Luồng 0 (luồng vào chuẩn): thường là bàn phím
  - Luồng 1 (luồng ra chuẩn): thường là màn hình console
  - Luồng 2 (luồng lỗi chuẩn): thường là màn hình console

Các luồng vào ra dữ liệu chuẩn

- Chương trình luôn hoạt động theo nguyên tắc:
  - Đọc dữ liệu đầu vào từ luồng 0
  - Nếu có kết quả thì ghi ra luồng 1
  - Nếu có báo lỗi thì ghi ra luồng 2
- Chính vì hoạt động mặc định trên, thông thường ta luôn nhập liệu từ bàn phím vào chương trình, và khi hoạt động chương trình in ra màn hình kết quả hoạt động hoặc báo lỗi
- Người dùng có thể thay đổi các luồng vào/ra chuẩn để phục vụ những ý đồ riêng của mình

Đổi hướng nhập (input redirection)

- Sử dụng "<" hoặc "0<" để đổi hướng việc nhập liệu
- Cú pháp:

```
lệnh < tập-tin
lệnh 0< tập-tin
```

- Cách làm việc: thay vì nhận dữ liệu từ bàn phím, câu lệnh sẽ nhận dữ liệu từ tập-tin chỉ định
- Ví dụ :

```
cat < /etc/passwd
more 0< /etc/passwd
```

Đổi hướng xuất (output redirection)

- Sử dụng ký tự “>” để đổi hướng việc xuất dữ liệu
- Cú pháp: lệnh > tập-tin
- Cách làm việc:
  - Thay vì ghi dữ liệu ra màn hình, kết quả thực hiện câu lệnh sẽ được ghi vào tập-tin chỉ định (ghi đè nội dung đã có)
  - Nếu muốn ghi dữ liệu vào tập-tin chỉ định, nhưng giữ nguyên dữ liệu cũ, ghi thêm vào cuối tập-tin, dùng “>>”
- Ví dụ :

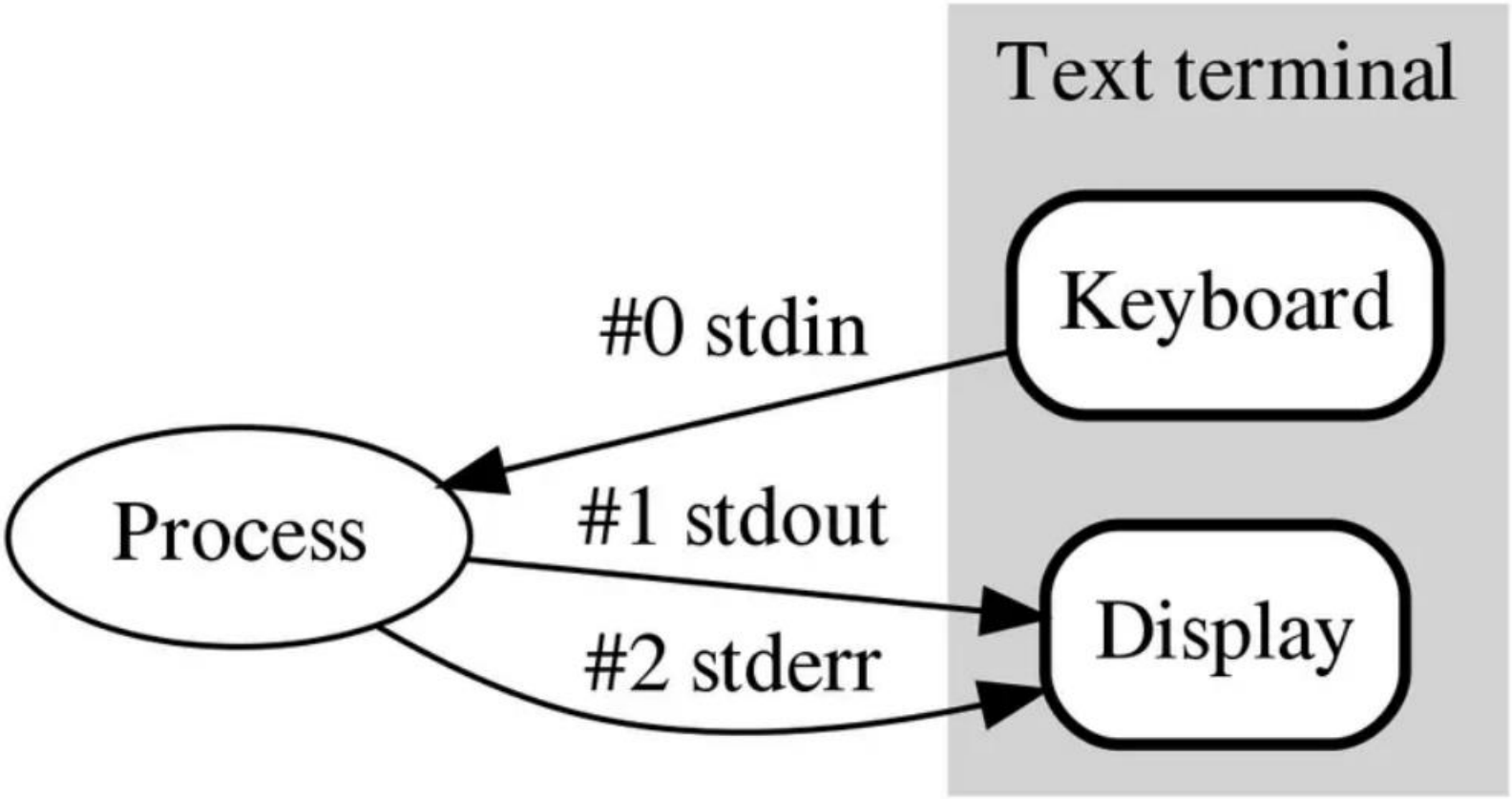
```
ls -l /tmp/ > /home/txnam/t1.out
ls -l /etc/ >> t1.out
```

Đổi hướng lỗi (error redirection)

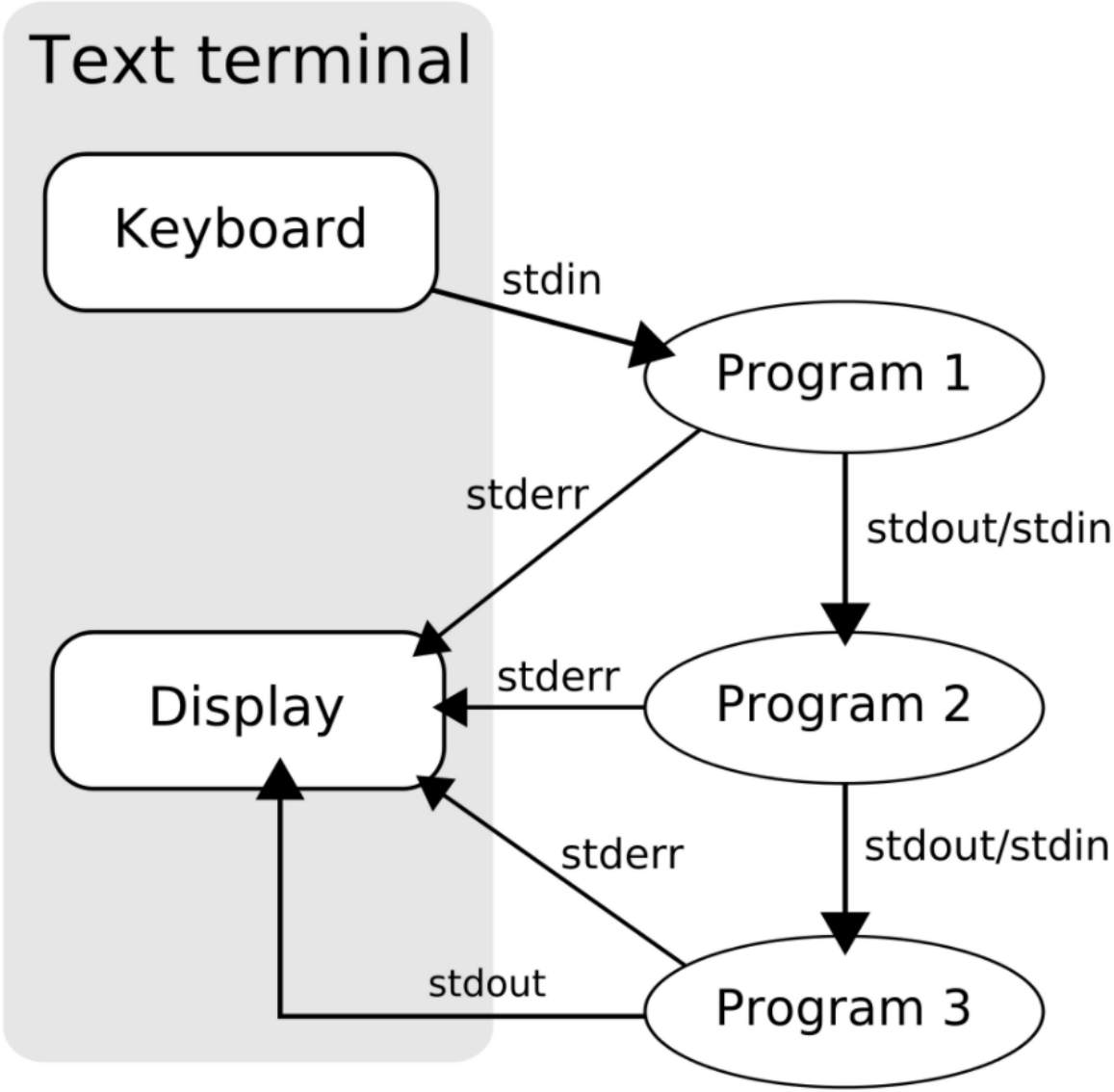
- Sử dụng "2>" để đổi hướng thông báo lỗi
- Cú pháp: lệnh 2> tập-tin
- Cách làm việc:
  - Những thông báo lỗi sẽ ghi vào tập-tin thay vì màn hình
  - Nội dung cũ trong tập-tin sẽ bị xóa
  - Trường hợp muốn giữ lại nội dung ban đầu của tập-tin và chèn thông tin lỗi vào cuối tập tin, dùng "2>>"
- Ví dụ:

```
ls -l /temp/ > t1.out 2> log.err
ls -l /etc/ >> t1.out 2>> log.err
```

Luồng và ống


Text terminal
Keyboard
#0 stdin
Process
#1 stdout
Display
#2 stderr

Luồng và ống


Text terminal
Keyboard
stdin
Program 1
stderr
stdout/stdin
Display
stderr
Program 2
stderr
stdout/stdin
stdout
Program 3

Ống lệnh (pipe)

- Cơ chế cho phép ghép các lệnh linux và lấy kết quả của lệnh trước làm đầu vào cho lệnh sau
- Cú pháp: lệnh-1 | lệnh-2 | … | lệnh-n
- Ví dụ "hiển thị từ dòng thứ 8 đến dòng thứ 10 của tập tin abc.txt": cat abc.txt | head -10 | tail -3
  - Lệnh "cat abc.txt" sẽ xuất ra nội dung abc.txt
  - Nội dung này chuyển tới "head -10": lấy 10 dòng đầu
  - 10 dòng đó chuyển cho "tail -3": cắt lấy 3 dòng cuối
- Giải thích: cat ds.txt | grep "txnam" | wc -l
- Giải thích: ls -al | grep '^d'