



1

## || Tập và thư mục

- Tập (file) là đơn vị lưu trữ nhỏ nhất trong hệ điều hành Unix/Linux
- Tập là một chuỗi byte
- Thư mục trong Unix là một loại tập đặc biệt, dùng để quản lý thông tin về các file khác

3

## || Nội dung

- 3.1. Tổng quan về hệ thống file
- 3.2. Quyền truy nhập thư mục và file
- 3.3. Thao tác với thư mục
- 3.4. Các lệnh làm việc với file
- 3.5. Sao lưu và nén dữ liệu

2

## || Tên tập/Tên thư mục

- Tập và thư mục được phân biệt qua tên
- Tên tập/thư mục có thể dài 256 ký tự, bao gồm chữ, số, gạch chân, dấu chấm, dấu cách
- Các ký tự không được sử dụng trong tên tập/thư mục: !, \*, \$, &, #, ?, ", ', `...
- Tập các tập có trong máy do Unix/Linux quản lý được gọi là "hệ thống tập"

4

## Ký hiệu chỉ nhóm tệp

- Có thể sử dụng các ký hiệu đặc biệt \* và ? để chỉ định nhóm các tệp

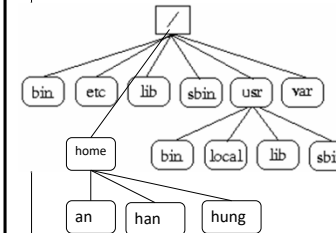
Dấu \* đại diện cho nhóm ký tự bất kỳ từ 0 đến nhiều ký tự

Dấu ? đại diện cho 1 ký tự

- Ví dụ:
  - **ab\***: Tất cả các tệp có tên bắt đầu bằng **ab**
  - **ab\*.c**: Tất cả các tệp có tên bắt đầu bằng **ab** và kết thúc bằng **.c**
  - **a?cd**: Tất cả các tệp có tên bắt đầu bằng chữ **a**, sau đó là một ký tự bất kỳ rồi kết thúc là **cd**

5

## Cấu trúc hệ thống tệp



- Một số tên thư mục đặc biệt:

- ~ : thư mục chủ của người dùng hiện hành
- " / " : Thư mục gốc
- " . " : Thư mục hiện hành
- " . . " : Thư mục cha

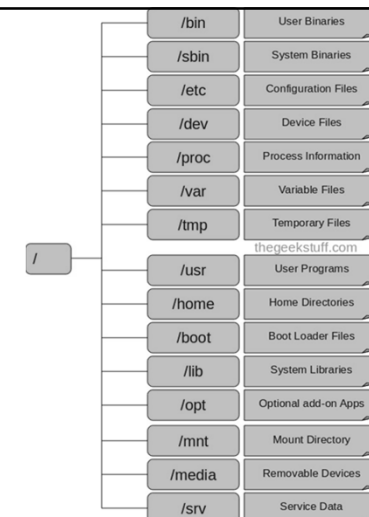
- Ví dụ:

- "usr" là thư mục con của "/" và là thư mục cha của "lib", "local" ...

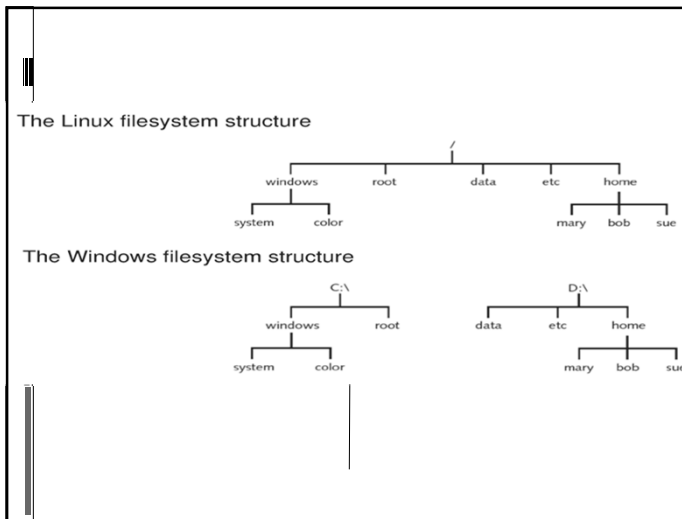
7

[ ] đại diện cho 1 ký tự trong tập hợp  
[!] không có ký tự trong tập hợp

6



8



9

## Đường dẫn tuyệt đối và tương đối

- Đường dẫn tuyệt đối bắt đầu bởi dấu "/"
- Đường dẫn tương đối dùng để chỉ cách định vị tệp/thư mục từ thư mục hiện hành

11

## Đường dẫn (path)

- Để định vị một tệp hoặc một thư mục trong hệ thống tệp, ta cần một đường dẫn
- Ví dụ:
  - Đường dẫn đến thư mục: /usr/bin
  - Đường dẫn đến tệp: /usr/bin/vi (vi là tên một hệ soạn thảo văn bản trên Unix)
- Đường dẫn có nhiều thành phần, các thành phần là tên thư mục hoặc tên tệp (thường ở vị trí cuối cùng) cách nhau bởi dấu "/"

10

## Kiến trúc bên trong hệ thống tệp

- Không bắt buộc học phần này
- Xem trong giáo trình, mục 3.1.2, 3.1.3 trang 40 đến trang 49

12

## Liên kết

- Unix có 2 kiểu liên kết
  - Liên kết cứng
  - Liên kết tượng trưng (liên kết mềm)
- Liên kết cứng cho phép chúng ta tạo thêm file một cách định vị trong hệ thống tệp (không có liên kết cứng đến thư mục)
- Liên kết mềm là một kiểu tệp đặc biệt tham chiếu đến tên một tệp hoặc thư mục khác
- Tạo liên kết: ln nguồn đích
- Tạo liên kết mềm: ln -s nguồn đích
- Tạo lk và xóa thông tin cũ: ln -sf nguồn đích

13

## Phân quyền truy xuất trên tập tin và thư mục

- Các quyền truy xuất trên tập tin và thư mục
- Lệnh **chmod**
- Thay đổi đối tượng sở hữu tập tin

15

## Lệnh tạo liên kết

- ln [<tùy chọn>] <đích> [<tên liên kết>]
- Các tùy chọn:
  - -b, --backup[=CONTROL] : tạo liên kết quay trở lại cho mỗi file đích đang tồn tại.
  - -f, --force : xóa bỏ các file đích đang tồn tại.
  - -d, -F, --directory : tạo liên kết cứng đến các thư mục (tùy chọn này chỉ dành cho người dùng có quyền quản trị hệ thống). Một số phiên bản không có tùy chọn này.
  - -n, --no-dereference : một file bình thường được xem là đích liên kết từ một thư mục.
  - -i, interactive : vẫn tạo liên kết dù file đích đã bị xóa bỏ.
  - -s, --symbolic : tạo các liên kết tượng trưng.
  - --target-directory=<tên-thư-mục> : xác định thư mục mà thư mục đích là thư mục có chứa các liên kết.
  - -v, --verbose : hiển thị tên các file trước khi tạo liên kết.
  - --help : hiển thị trang trợ giúp và thoát.

14

## Phân quyền truy xuất trên tập tin và thư mục

- Quyền truy xuất một tệp/thư mục qui định nhóm người sử dụng nào được phép làm gì (thao tác) trên tệp/thư mục đó
- Các nhóm người sử dụng
  - User (người sở hữu), ký hiệu là **u**
  - Group (những người cùng nhóm), ký hiệu là **g**
  - Other (những người khác), ký hiệu là **o**
  - All (tất cả mọi người), ký hiệu là **a**

16

### Phân quyền truy xuất trên tập tin và thư mục

- Quyền truy xuất tập tin và thư mục được quy định với những thuộc tính sau:

Quyền		Tập tin	Thư mục
r	Read – đọc	Xem nội dung của tập tin, có nghĩa là dùng được lệnh <b>cat</b>	Xem nội dung của thư mục, có nghĩa là dùng được lệnh <b>ls</b> với thư mục
w	Write – ghi	Thuộc tính ghi (hiệu chỉnh nội dung)	Được tạo mới/xóa tập tin/thư mục con trong thư mục đó
x	eXecute – thực thi	Thuộc tính thực thi (chạy chương trình)	Được chuyển vào trong thư mục bằng lệnh <b>cd</b>
-	None	Không có quyền	Không có quyền

17

### Ý nghĩa các cột của lệnh ls -l

- Lệnh **ls -l** giúp xem thông tin đầy đủ / chi tiết về nội dung của một thư mục
- Cột 1: thông tin phân quyền (10 ký tự)
- Cột 2: số liên kết đến thư mục / tập đó
- Cột 3: chủ sở hữu
- Cột 4: nhóm sở hữu
- Cột 5: kích thước
- Cột 6: ngày cập nhật gần nhất
- Cột 7: tên file / thư mục

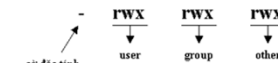
19

- Trình bày nội dung các tập tin và thư mục được thiết lập quyền (*set permission*) trong thư mục cá nhân (Home Directory) của người dùng tên là **ngochan**

```
ngochan@ubuntu:~$ ls -l
total 80
-rw-rw-r-- 1 ngochan ngochan 90 2012-03-12 16:45 baitho
-rw-r--r-- 1 wiki ngochan 60 2012-03-12 11:49 cadao
drwxr-xr-x 2 ngochan ngochan 4096 2012-02-28 16:40 Desktop
drwxr-xr-x 2 ngochan ngochan 4096 2012-03-05 12:28 Documents
drwxr-xr-x 2 ngochan ngochan 4096 2012-02-08 15:25 Downloads
-rw-r--r-- 1 ngochan ngochan 179 2012-02-07 23:11 examples.desktop
-rwxrwxr-x 1 ngochan ngochan 7195 2012-03-20 15:12 hello
-rwxr--r-- 1 ngochan ngochan 116 2012-03-20 15:39 hello.c
drwxrwxr-x 3 ngochan ngochan 4096 2012-02-28 17:22 Linux
drwxr-xr-x 2 ngochan ngochan 4096 2012-02-08 15:25 Music
-rw-rw-r-- 1 hana ngochan 30 2012-03-12 11:12 ngochanNote.txt
drwxr-xr-x 2 ngochan ngochan 4096 2012-02-08 15:25 Pictures
drwxr-xr-x 2 ngochan ngochan 4096 2012-02-08 15:25 Public
drwxr-xr-x 2 ngochan ngochan 4096 2012-02-08 15:25 Templates
drwxrwxr-x 2 ngochan ngochan 4096 2012-03-12 11:43 testinhome
drwxrwxr-x 2 ngochan ngochan 4096 2012-03-12 16:43 testinnyhome
drwxrwxr-x 2 ngochan ngochan 4096 2012-02-11 23:06 ubuntu.dig
```

18

Chú ý đến các thuộc tính sau:



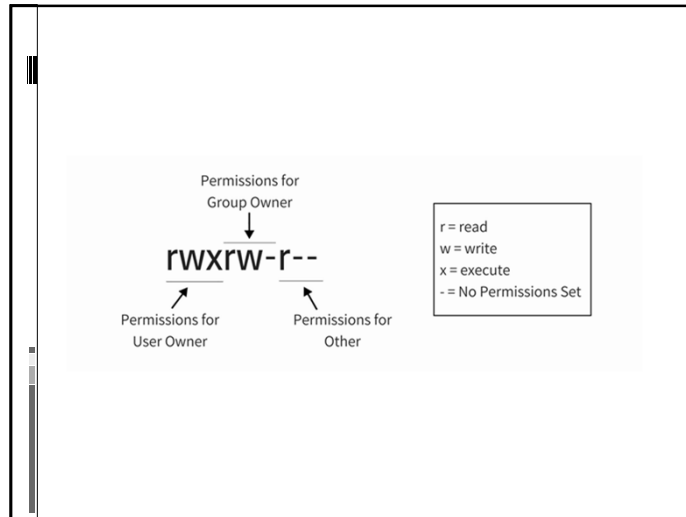
- Cờ đầu tiên (cờ đặc tính) chỉ loại file:

-	File thông thường
d	Thư mục
p	Kiểu đường ống pipe
s	Socket
l	Link (liên kết hay shortcut)

```
-rw-rw-r--
-rw-r--r--
drwxr-xr-x
drwxr-xr-x
drwxr-xr-x
-rw-r--r--
```

- Có 3 đối tượng chính là **owner**, **group**, **other** và mỗi đối tượng có 3 quyền cụ thể **read**, **write**, **execute** với file
- Thiết lập (thay đổi) thuộc tính truy nhập cho tập tin và thư mục bằng lệnh **chmod** → Phải thực hiện bằng quyền của **ROOT** hoặc **chủ sở hữu**

20



21

### Ví dụ:

```
bibo@ubuntu: /home/ngochan
ngochan@ubuntu:~$ cat > apple
Ngoc Han's Apple.

ngochan@ubuntu:~$ ls -l apple
-rw-rw-r-- 1 ngochan ngochan 19 2012-03-25 14:04 apple
ngochan@ubuntu:~$ chmod o-r apple
ngochan@ubuntu:~$ ls -l apple
-rw-rw---- 1 ngochan ngochan 19 2012-03-25 14:04 apple
ngochan@ubuntu:~$ chmod g-w apple
ngochan@ubuntu:~$ ls -l apple
-rw-r----- 1 ngochan ngochan 19 2012-03-25 14:04 apple
ngochan@ubuntu:~$ su bibo
Password:
bibo@ubuntu:/home/ngochan$ cat apple
cat: apple: Permission denied
bibo@ubuntu:/home/ngochan$
```

23

### Tài khoản

- Khi tạo tài khoản trong Ubuntu, thì hệ thống tự động tạo ra một nhóm trùng tên với tài khoản đó

22

- Khi người dùng tạo ra một tập tin (hoặc thư mục) ⇒ người dùng là chủ sở hữu (**Owner**)
  - Mặc định quyền được thiết lập với file là **read - write ~ `rw`**
  - Mặc định quyền được thiết lập với thư mục là **read write execute ~ `rwx`**
- Nếu muốn chia sẻ với người khác, cần thiết lập quyền cho nhóm – (**Group**)
- Ví dụ:

```
ngochan@ubuntu: ~
ngochan@ubuntu:~$ sudo chgrp boss apple
[sudo] password for ngochan:
ngochan@ubuntu:~$ ls -l apple
-rw-rw---- 1 ngochan boss 19 2012-03-25 14:04 apple
ngochan@ubuntu:~$
```

24

### Thay đổi quyền truy nhập với lệnh **chmod**

- Qua lệnh **chmod** ta có thể thay đổi quyền **r-w-x** của đối tượng như *user, group, other hay all* đối với tập tin (hoặc thư mục).
- Quyền truy nhập tập tin còn phụ thuộc vào thư mục chứa nó. Ví dụ:
- Lệnh chmod có thể thực hiện theo cách xác lập tương đối hoặc tuyệt đối

25

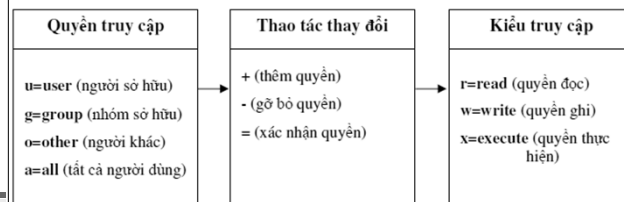
### Xác lập tuyệt đối

- Thông tin phân quyền có các vị trí rwx là cố định: Có thì xuất hiện chữ, không có thì là dấu -
- Vị trí xuất hiện chữ ta đặt là 1, không có quyền ta đặt là 0
- Một file abc có thông tin phân quyền là rw-rw-r--
- Tương đương với quyền 110110100  $\Leftrightarrow$  664
- Ta muốn chuyển để quyền là: rwxrwxrwx  $\Leftrightarrow$  777
- Thực hiện lệnh
- `chmod 777 abc`

27

### Cách xác lập tương đối

- Cách xác lập tương đối là dễ nhớ theo ý nghĩa của nội dung các mod và chỉ những thay đổi thực sự mới được biểu diễn trong lệnh



- Ví dụ:
  - `chmod g+w test`
  - `chmod o-rx test`
- Chủ sở hữu hoặc root có thể thực hiện lệnh chmod này

26

### Thay đổi quyền sở hữu tập tin

- Thay đổi người sở hữu:
  - `chown <tên người sở hữu> <tên tệp>`
  - Luôn phải là root hoặc tương đương
 

```
# chown username_mới tên_tệp
$ sudo chown username_mới tên_tệp
```
- Thay đổi nhóm:
  - `chgrp <tên nhóm> <tên tệp>`

28

## || Kịch bản cho việc phân quyền

- Sudoer: ngochan / hoặc root
- Standard user: dev1
- Standard user: dev2
- dev1 & dev2 thuộc nhóm 56pml và chuyển sang nhóm này sau khi đăng nhập (lệnh newgrp)
- chmod g+w dev1
- dev1 tạo file -> dev2 vẫn xóa được

29

## || User dev1

```
dev1@ubuntu:~$ cat > devldiary
Nhat ky cua dev1
dev1@ubuntu:~$ ls -l devldiary
-rw-rw-r-- 1 dev1 dev1 17 2014-10-26 14:32 devldiary
dev1@ubuntu:~$ chmod og-rx devldiary
dev1@ubuntu:~$ ls -l devldiary
-rw----- 1 dev1 dev1 17 2014-10-26 14:32 devldiary
dev1@ubuntu:~$ ls -l
total 0
dev1@ubuntu:~$
```

31

## || User ngochan -> root

```
root@ubuntu:/home# tail -4 /etc/group
admin:x:110:ngochan
ftp:x:111:ftp
dev1:x:1001:
dev2:x:1004:
root@ubuntu:/home# groupadd 56pml
root@ubuntu:/home# usermod -g 56pml dev1
root@ubuntu:/home# usermod -g 56pml dev2
root@ubuntu:/home# ls -l dev1
drwxr-xr-x 2 dev1 56pml 4096 2014-10-26 14:32 dev1
root@ubuntu:/home# chmod g+w dev1
root@ubuntu:/home# ls -l dev*
drwxrwxr-x 2 dev1 56pml 4096 2014-10-26 14:32 dev1
drwxr-xr-x 2 dev2 56pml 4096 2014-10-26 14:28 dev2
root@ubuntu:/home#
```

30

## || User dev2

```
dev2@ubuntu:~$ newgrp 54pml
dev2@ubuntu:~$ cd /home/dev1
dev2@ubuntu:/home/dev1$ ls
devldiary
dev2@ubuntu:/home/dev1$ cat devldiary
Nhat ky cua dev1
dev2@ubuntu:/home/dev1$ ls
devldiary
dev2@ubuntu:/home/dev1$ rm devldiary
rm: remove write-protected regular file `devldiary'? y
dev2@ubuntu:/home/dev1$ ls -l
total 0
dev2@ubuntu:/home/dev1$
```

32



### Các lệnh thao tác với tệp tin và thư mục

Lệnh	Tác dụng
<code>pwd</code>	Hiện tên thư mục đang làm việc
<code>cd</code>	di chuyển sang thư mục «/home/người_dùng»
<code>cd ~/Desktop</code>	di chuyển sang thư mục «/home/người_dùng/Desktop»
<code>cd ..</code>	di chuyển sang thư mục cha (ngay trên thư mục hiện hành)
<code>cd /usr/apt</code>	di chuyển sang thư mục «/usr/apt»

33

### Các lệnh thao tác với tệp tin và thư mục

Lệnh	Tác dụng
<code>cp file1 file2</code>	chép file1 sang file2
<code>cp file thumuc</code>	chép file vào thư mục «thumuc»
<code>cp -r thumuc1 thumuc2</code>	chép toàn bộ nội dung của thư mục «thumuc1» sang thư mục «thumuc2»
<code>mv file1 file2</code>	đổi tên tệp tin file1 thành tên file2
<code>mv thumuc1 thumuc2</code>	đổi tên thumuc1 thành thumuc2
<code>mv file thumuc</code>	chuyển tệp tin file vào thư mục
<code>mv file1 thumuc/file2</code>	chuyển file1 vào thư mục thumuc đồng thời đổi tên tệp tin thành file2

35

### Các lệnh thao tác với tệp tin và thư mục

Lệnh	Tác dụng
<code>ls -l Thumuc</code>	liệt kê danh mục tệp tin trong <i>Thumuc</i> một cách chi tiết
<code>ls -a</code>	liệt kê tất cả các tệp tin, kể cả các tệp tin ẩn (thường có tên bắt đầu bằng một dấu chấm)
<code>ls -d</code>	liệt kê tên các thư mục nằm trong thư mục hiện hành
<code>ls -t</code>	xếp lại các tệp tin theo ngày được tạo ra, bắt đầu bằng những tệp tin mới nhất
<code>ls -S</code>	xếp lại các tệp tin theo kích thước, từ to nhất đến nhỏ nhất
<code>ls -l   more</code>	liệt kê theo từng trang một, nhờ tiện ích «more»
<code>ls -1</code>	liệt kê theo dạng 1 cột

34

### Các lệnh thao tác với tệp tin và thư mục

Lệnh	Tác dụng
<code>mkdir thumuc</code>	tạo ra thư mục thumuc
<code>mkdir -p thumuc1/thumuc2</code>	tạo ra thư mục cha thumuc1 và thư mục con thumuc2 cùng lúc
<code>rm file</code>	xóa bỏ tệp tin file trong thư mục hiện hành
<code>rmdir thumuc</code>	xóa bỏ thư mục trống mang tên thumuc
<code>rm -rf thumuc</code>	xóa bỏ thư mục mang tên thumuc với tất cả các tệp tin trong đó (recursive force)
<code>ln -s file liênkết</code>	tạo ra một liên kết mang tên <i>liênkết</i> đến tệp tin file (nối tắt)

36

## Một số lệnh quan trọng liên quan đến file & thư mục

37

- Một số tùy chọn :
  - b : cho phép chỉ ra kiểu file mà không đưa kèm tên file
  - f **tên-file** : cho phép hiển thị kiểu của các file có tên trùng với nội dung trên mỗi dòng trong file tên-file. Để kiểm tra thiết bị vào chuẩn, dùng dấu “-”
  - z : xem kiểu của file nén

39

## Lệnh file

- Lệnh file giúp xác định & in ra kiểu thông tin chứa trong file. Kiểu file có thể thuộc một trong 3 dạng sau:
  - text : dạng văn bản thông thường, chỉ chứa các mã ký tự ASCII
  - executable : dạng file nhị phân khả thi
  - data : thường là dạng file chứa mã nhị phân và không thể in ra được
- Cú pháp:
 

```
file [tùy-chọn] [-f file] ...
```

38

## Lệnh nl

- Lệnh nl giúp đánh số thứ tự các dòng trong file.
- Cú pháp:
 

```
$ nl [tùy-chọn] <file>
```
- Lệnh này đưa nội dung file ra thiết bị ra chuẩn, với số thứ tự của dòng được thêm vào. Nếu không có file, hoặc khi file là dấu “-” thì đọc nội dung từ thiết bị vào chuẩn.

40

#### ■ Các tùy chọn:

- b, --body-numbering-STYLE: sử dụng kiểu STYLE cho việc đánh thứ tự các dòng trong nội dung file. Có các kiểu STYLE sau:
  - a : đánh số tất cả các dòng kể cả dòng trống
  - t : chỉ đánh số các dòng không trống
  - n : không đánh số dòng
- d : sử dụng CC để đánh số trang logic
- f : sử dụng kiểu STYLE để đánh số các dòng cho footer line
- h : sử dụng kiểu STYLE để đánh số các dòng cho header line

41

#### ■ wc: Các tùy chọn

<b>-c</b>	đưa ra số ký tự trong file
<b>-l</b>	đưa ra số dòng trong file
<b>-L</b>	đưa ra chiều dài của dòng dài nhất trong file
<b>-w</b>	đưa ra số từ trong file

#### ■ Ví dụ:

```
ngochan@ubuntu:~$ wc /etc/passwd
28   34 1204 /etc/passwd
```

43

#### ■ Lệnh wc

- Lệnh wc dùng để đếm số ký tự, số từ, hay số dòng trong một file.
- Cú pháp lệnh:
 

```
wc [tùy-chọn] [file]...
```
- Lệnh hiển thị ra số lượng dòng, số lượng từ, số lượng ký tự trong mỗi file.
  - Nếu nhiều hơn 1 file được chỉ ra thì có thêm một dòng tính tổng.
  - Nếu không có tên file thì lệnh sẽ đọc và đếm trên thiết bị vào chuẩn.

42

#### ■ Kết hợp wc với các lệnh khác

- wc kết hợp với ls để xác định số file/thư mục con có trong một thư mục:

```
$ ls -l | wc -l
8
```

- wc kết hợp với lệnh cat để biết số tài khoản:

```
$ cat /etc/passwd | wc -l
```

44

## || Lệnh uniq

- Lệnh **uniq** giúp loại bỏ những dòng thông tin bị trùng lặp trong một file.
- Cú pháp:  
**uniq [tùy-chọn] [input] [output]**
- Lệnh **uniq** sẽ loại bỏ các dòng trùng lặp kề nhau từ **input** và giữ lại một dòng duy nhất trong số các dòng trùng lặp rồi đưa ra **output**

45

## || Lệnh sort

- Lệnh **sort** : sắp xếp các dòng của file văn bản  
**\$ sort [tùy-chọn] [file] ...**
- Các tùy chọn:
  - f : sắp xếp không phân biệt chữ hoa chữ thường
  - n : sắp xếp theo chữ số
  - r : sắp xếp ngược
  - b : bỏ qua các dấu cách ở đầu mỗi trường
  - d : sắp xếp chỉ dựa vào các ký tự trong bảng chữ cái và chữ số (ký tự, chữ số, dấu cách)
  - t x : ký tự x được sử dụng làm dấu phân cách giữa các trường
  - u : xóa các dòng trùng nhau

47

- Các tùy chọn:
  - c, --count : đếm và hiển thị số lần xuất hiện của các dòng trong file
  - d : hiển thị lên màn hình dòng bị trùng lặp
  - u : hiển thị nội dung file sau khi xóa bỏ toàn bộ các dòng bị trùng lặp không giữ lại một dòng nào
  - i : hiển thị nội dung file sau khi xóa bỏ các dòng trùng lặp và chỉ giữ lại duy nhất một dòng có nội dung bị trùng lặp
  - D : hiển thị tất cả các dòng trùng lặp lên màn hình

46

## || Ví dụ

```
file carnet.txt
maurice:29:0298334432:Crozon
marcel:13:0466342233:Marseille
robert:75:0144234452:Paris
yvonne:92:0133444335:Palaiseau
```

```
$ sort -n -t : +1 -2 carnet.txt
```

Thực hiện quá trình sắp xếp từ trường thứ 2 đến trường thứ 3 (các trường được đánh số từ 0)

```
$ sort -t : +3 -4 +0 -1 carnet.txt
```

Thực hiện quá trình sắp xếp đầu tiên dựa trên trường thứ 4, sau đó thực hiện tiếp quá trình sắp xếp dựa trên trường thứ nhất

48

## || Lệnh diff

- Lệnh diff giúp so sánh từng dòng của hai file
- Cú pháp:  
\$ **diff** [tùy-chọn] <file1> <file2>
- Trong những trường hợp đơn giản, lệnh diff còn so sánh nội dung 2 thư mục theo thứ tự từ a-z

49

## || Lệnh grep

- Lệnh này giúp in ra các dòng phù hợp với mẫu cần tìm
  - Cú pháp:  
**grep** [tùy-chọn] <mẫu-lọc> [file]
  - Lệnh grep có thể dùng để lọc đầu ra của một lệnh khác với cú pháp:  
<lệnh> | **grep** <mẫu-lọc>
- VD: Tìm trong file /etc/passwd thông tin về người dùng ngochan -> hiển thị ra dòng chứa tên ngochan
- ```
$ grep ngochan /etc/passwd
```
- ```
$ grep 1004 /etc/passwd
```

51

## || Các tùy chọn:

- a : xem tất cả các file ở dạng văn bản và so sánh theo từng dòng
- b : bỏ qua sự thay đổi về số lượng của ký tự trống
- B : bỏ qua mọi sự thay đổi mà chỉ chèn hoặc xóa các dòng trống.
- brief : chỉ thông báo khi có sự khác nhau mà không đưa ra chi tiết nội dung khác nhau
- d : tìm ra sự khác biệt nhỏ (có thể làm chậm lệnh diff)
- exclude-from=file : khi so sánh thư mục, bỏ qua các file và các thư mục con có tên phù hợp với mẫu có trong file.
- i : so sánh không phân biệt chữ hoa, chữ thường
- r : thực hiện so sánh đệ quy trên thư mục
- s : thông báo khi hai file là giống nhau
- y: hiển thị 2 file cạnh nhau để dễ phân biệt sự khác nhau

50

## || Các tùy chọn:

- G : xem mẫu lọc như một biểu thức thông thường, điều này là ngầm định
  - E : xem mẫu lọc như là một biểu thức mở rộng
  - F : xem mẫu như là một danh sách các xâu cố định, được phân ra bởi các dòng mới.
- Ngoài lệnh grep còn có lệnh egrep (tương tự lệnh grep -E) và lệnh fgrep (tương tự lệnh grep -F)
- A NUM : đưa ra NUM dòng nội dung tiếp theo dòng có chứa mẫu

52

### || Các tùy chọn (tiếp):

- B NUM : đưa ra NUM dòng nội dung trước dòng có chứa mẫu
- C [NUM] : hiển thị NUM dòng (mặc định là 2 dòng) nội dung
- b : hiển thị địa chỉ tương đối trong file đầu vào trước mỗi dòng được đưa ra.
- c : đếm số dòng tương ứng chứa mẫu trong file đầu vào thay cho việc hiển thị các dòng chứa mẫu.
- d ACTION : nếu đầu vào là một thư mục, sử dụng ACTION để xử lý nó. Mặc định ACTION là read.
- f file : lấy mẫu từ file, mỗi mẫu trên một dòng

53

### || Các ký hiệu biểu diễn mẫu lọc

Ký hiệu	Ý nghĩa
^	Bắt đầu một dòng
\$	Kết thúc dòng
.	Thay cho một ký tự đơn
[xy]	Chọn 1 trong tập ký tự đưa ra
[^xy]	Chọn 1 ký tự không thuộc tập ký tự đưa ra

55

### || Các tùy chọn (tiếp)

- H : đưa ra tên file trên mỗi dòng chứa mẫu tương ứng.
- h : không hiển thị tên file kèm theo dòng chứa mẫu trong trường hợp tìm nhiều file
- i : hiển thị các dòng chứa mẫu không phân biệt chữ hoa chữ thường
- l : đưa ra tên các file trùng với mẫu lọc
- n : thêm số thứ tự của dòng chứa mẫu trong file.
- r : đọc tất cả các file chứa trong thư mục (đệ quy)
- s : bỏ qua thông báo lỗi file không đọc được hoặc không tồn tại
- v : hiển thị các dòng không chứa mẫu
- w : chỉ hiển thị những dòng chứa mẫu lọc là một từ trọn vẹn
- x : chỉ hiển thị những dòng mà nội dung trùng hoàn toàn với mẫu lọc

54

### || Lệnh cut – xác định các cột

```
$ cut -options [files]
```

Tùy chọn:

- c<Số ký tự> xác định các ký tự
- f<số trường> xác định các trường
- d<dấu phân cách>

VD:

```
$ cut -c5 file # hiển thị cột ký tự thứ 5
$ cut -c5-10 file # hiển thị ký tự thứ 5 đến thứ 10
$ cut -d: -f1 /etc/passwd # hiển thị tên tất cả người dùng của hệ thống
```

56

## Lệnh tr

Thay thế một chuỗi bằng một chuỗi khác có cùng độ dài

Ví dụ:

```
$ tr ":" "#" < canet.txt
$ tr "[a-z]" "[A-Z]" < canet.txt

$ cat /etc/passwd | tr ":" "#"
```

57

## Lưu các thay đổi vào tập tin

```
$ sed 's/text/replace/' file > newfile
```

```
$ mv newfile file
```

Tuy nhiên, chúng ta chỉ cần 1 lệnh sau để thực hiện việc trên:

```
$ sed -i 's/text/replace/' file
```

59

## Lệnh sed

- **sed**: stream editor
  - ứng dụng phổ biến: thay thế văn bản
- sed** có thể được dùng để thay thế những xuất hiện của 1 chuỗi với 1 chuỗi khác trong 1 văn bản.

```
$ sed 's/pattern/replace_string/' file
```

hoặc

```
$ cat file | sed 's/pattern/replace_string/'
```

58

## Thay thế tất cả xuất hiện của mẫu

```
$ sed 's/pattern/replace_string/g' file
```

## Thay thế từ xuất hiện thứ N của mẫu cho đến cuối văn bản

```
$ echo thisthisthisthis | sed 's/this/THIS/3g'
thisthisTHISTHIS
```

## Thay thế xuất hiện thứ N của mẫu trong văn bản, sử dụng dạng /N

```
$ echo thisthisthisthis | sed 's/this/THIS/2'
thisTHISthisthis
```

60

## || Lệnh find

- Lệnh find cho phép tìm file trong một thư mục
- Cú pháp:  
Lưu ý: tùy chọn của lệnh là **từ** chứ không phải **ký tự**  
**\$ find [đường-dẫn] [biểu-thức]**  
biểu-thức có thể có các dạng sau:  
(expr)  
!expr hoặc -not expr  
expr1 -a expr2 hoặc expr1 -and expr2  
expr1 -o expr2 hoặc expr1 -or expr2  
expr1, expr2

61

- Ví dụ: Tìm file có tên là adduser.conf trong thư mục /etc

\$ find /etc -name adduser.conf

Linux sẽ đưa ra được địa chỉ tuyệt đối của file

Thường dùng:

find /etc -name adduser.conf 2> /dev/null

để bỏ đi các dòng báo lỗi

63

## || Lệnh find – một số tham số

- Lệnh find cho phép tìm kiếm một hay nhiều tập tin trong một cây thư mục.

Tìm theo tên file: **find <path> -name <filename>**

Tìm theo tên người sở hữu: **find <path> -user <username>**

Tìm theo số i-node của tập tin: **find <path> -inum <number>**

62

## || Các tùy chọn lệnh

- daystart : đo thời gian (-amin, -atime, -cmin, -ctime, -mmin, -mtime)
- depth : thực hiện tìm kiếm từ nội dung bên trong thư mục trước (mặc định việc tìm kiếm tại gốc cây thư mục có chứa file cần tìm)
- follow : tùy chọn này chỉ áp dụng cho thư mục -> liên kết tượng trưng có trong một thư mục liên kết được chỉ ra.
- amin n : tìm file được truy cập trong n phút trước
- atime n : tìm file được truy cập trong n ngày trước
- cmin n : trạng thái của file được thay đổi n phút trước đây
- ctime n : trạng thái của file được thay đổi n ngày trước đây
- mmin n : dữ liệu của file được sửa đổi lần cuối vào n phút trước đây
- mtime n : dữ liệu của file được sửa đổi lần cuối vào n ngày trước đây

64



- empty : file rỗng và hoặc là thư mục hoặc là file bình thường
- fstype kiểu : file thuộc hệ thống file với kiểu
- gid n : chỉ số nhóm của file là n
- group nhóm : file thuộc quyền sở hữu của nhóm
- links n : file có n liên kết
- iname mẫu : tìm kiếm file có tên là mẫu
- type kiểu : tìm các file thuộc kiểu với kiểu nhận các giá trị: b , c , d , p , s , f , l , s
- uid : chỉ số người sở hữu file là n
- user tên-người : file được sở hữu bởi người dùng tên-người

65

- Lưu ý khi dùng more
  - Ấn phím «Enter» để xuống 1 dòng
  - Ấn phím «Space» để sang thêm 1 trang
  - Ấn phím «q» để thoát.
  - Ấn «F» để tiến về trước một trang,
  - Ấn «B» để lùi lại một trang
  - Ấn dấu xược xuôi «/» và thêm string để tìm chuỗi string trong file (thành /string)
- Lưu ý khi dùng pico
  - Soạn thảo file mới: pico tênfile
  - Lưu file: Ctrl + O
  - Ctrl + X để thoát
    - Chương trình sẽ hỏi có lưu không, có – Y, không – N
    - Chương trình hỏi lưu với tên file nào? Giữ nguyên – Lưu, Đổi tên – Save As...

67

### Nội dung các tập tin

cat file	Hiển thị nội dung của tập tin file trên màn hình ở dạng mã ASCII
more file	Hiển thị nội dung của tập tin file trên màn hình theo chế độ từng trang
less file	«less» giống như «more», nhưng dùng thêm phím [Pg Down]
head -n file	Hiển thị n dòng đầu tiên của tập tin file
tail -n file	Hiển thị n dòng cuối cùng của file
vi file	soạn tập tin file dùng trình soạn vi
pico file	soạn tập tin file dùng trình soạn thảo pico
command > file	ghi kết quả của lệnh command vào tập tin file
command >> file	bổ sung kết quả của lệnh command ở phần cuối của tập tin file

66

### Soạn thảo dùng vi

- **Bắt đầu vào xem file -> Chế độ lệnh**
  - Di chuyển: h j k l
  - Xoá: dd (nhiều dòng 4dd) dw x
- **Chuyển sang chế độ soạn thảo:**
  - Gõ 1 trong 6 phím: A a O o I i
  - Gõ tiếp bình thường
- **Lưu bài: :w**
- **Thoát :wq (thoát có lưu). :q! (thoát không lưu)**
- **Chuyển đổi từ soạn thảo -> lệnh Ấn phím Esc**
- **Copy: yy (nhiều dòng 5yy)**
- **Paste: p**

68

vim 8.1  
vi 1st, 06

SC

normal mode

toggle case

goto mark

vi / vim graphical cheat sheet

external filter	@> play macro	# prev ident	\$ eol	% goto match	^ "soft" bol	& repeat	* next ident	C begin sentence	) end sentence	"soft" bol down	+ next line						
1	2	3	4	5	6	7	8	9	0	"hard" bol	- prev line	= autoformat					
Q ex mode	W WORD	E end WORD	R replace till	T back till	Y yank line	U undo line	I insert at bol	O open above	P paste before	{ begin parag.	}	end parag.					
Q record macro	W next word	E end word	R replace char	T till	Y yank	U undo	I insert mode	O open below	P paste after	[ misc	]	misc					
A append at eol	S subst line	D delete to eol	F "back" find ch	G col/ goto ln	H screen top	J join lines	K help	L screen bottom	ex cmd line	" reg. spec	" goto mk. bol	\ no use					
z append	S subst char	d delete	f find char	g extra ch	h ←	j ↓	k ↑	l →	repeat 2 U/T/t/f	repeat 2 U/T/t/f	goto mk. bol	\ no use					
Z quit	X back space	C change to eol	V visual lines	B prev WORD	N (find)	M screen mid	< un-indent	> indent	? find (rev)								
Z extra ch	X delete char	c change	V visual mode	b prev word	n next (find)	m set mark	reverse > U/T/t/f	repeat cmd	/ find								

motion

command

operator

extra

q

ords:

ORDS:

moves the cursor, or defines the range for an operator  
direct action command,

if red, it enters insert mode

requires a motion afterwards,  
operates between cursor &  
destination

special functions,  
requires extra input  
commands with a dot need  
a char argument afterwards

l = beginning of line, eol = end of line,  
k = mark, yank = copy

quuq[[]]o[] b[] B[]

quuq[[]]o[] b[] B[]

quuq[[]]o[] b[] B[]

Main command line commands ('ex'):

:w (save), :q (quit), :q! (quit w/o saving)

:f (open file f)

:%s/y1/z1 (replace 'y' by 'z' filewide),

:h (help in vim), :nw (new file in vim),

Other important commands:

CTRL-R: redo (vim)

CTRL-Y: yk: page up/down,

CTRL-E-Y: scroll line up/down,

CTRL-V: block-visual mode (vim only)

Visual mode:

Move around with operator to act  
on selected region (vim only)

Notes:

(1) use "x before a yank/paste/del command to use that register ('clipboard') (x=a-z, e.g.: 'y\$' to copy rest of line to reg 'a')

(2) type in a number before any action to repeat it that number of times (e.g.: 3p, 4w, 5i, 4d)

(3) duplicate operator to act on current line (dd = delete line, >> = indent line)

(4) ZZ to save & quit, ZQ to quit w/o saving

(5) zt: scroll cursor to top,

zb: bottom, zz: center

(6) gg: top of file (vim only),

gf: open file under cursor (vim only)

For a graphical vi/vim tutorial & more tips, go to [www.viemu.com](http://www.viemu.com) - home of ViEmu, vi/vim emulation for Microsoft Visual Studio

69

## Lưu ý khi dùng cat

cat file: lệnh xem file đơn giản

cat > file: Lệnh tạo file mới và soạn thảo (đơn giản). Ctrl + D để kết thúc soạn thảo và lưu vào file

cat > file: Nếu file đã có thì sẽ xóa nội dung cũ và ghi đè bằng nội dung mới

cat >> file: Ghi nối thêm nội dung vào file

70

## Nén và sao lưu các file

- Lưu trữ các tệp:
  - tar cvzf <tên tệp lưu trữ> <thư mục / nhóm tệp>
- Lấy các tệp từ lưu trữ:
  - tar xvzf <tên tệp lưu trữ> <thư mục / nhóm tệp>
- Nén và giải nén tệp:
  - Nén: zip / gzip <tệp / nhóm tệp>
  - Giải nén: unzip / gunzip <tệp / nhóm tệp>

71

## Định hướng lại vào/ra

- Unix/Linux có 3 dòng nhập/xuất chuẩn (I/O) là
  - STDIN: dòng nhập chuẩn
  - STDOUT: dòng xuất chuẩn
  - STDERR: dòng xuất lỗi chuẩn
  - Các mô tả file lần lượt là 0, 1 và 2

72

### || Định hướng lại đầu vào - STDIN

- STDIN
  - chỉ các dòng nhập chuẩn nói chung
  - thường là từ bàn phím.
  - Để dữ liệu đầu vào là một file thì ta dùng dấu "<".
- Ví dụ: nếu ta dùng lệnh cat mà không có tham số
  - khi ta gõ gì nó sẽ hiển thị ra cái đó, hay nói đúng hơn sẽ hiển thị lại những gì ta nhập vào từ input chuẩn.
  - Nếu cần hiển thị file /etc/passwd, ngoài cách truyền thống là:
 

```
$ cat /etc/passwd
```

    - ta có thể sử dụng:
 

```
$ cat < /etc/passwd
```
    - hoặc:
 

```
$ cat 0< /etc/passwd
```

73

- Đến đây ta có thể kết hợp sử dụng song song STDIN và STDOUT để làm thao tác copy file.
- Ví dụ ta muốn backup file /etc/passwd thì ta có thể làm như sau:
 

```
$ cat < /etc/passwd > ~/passwd.bak
```
- Lệnh này tương đương với lệnh:
 

```
$ cp /etc/passwd ~/passwd.bak
```

75

### || Định hướng lại đầu ra chuẩn - STDOUT

- STDOUT
  - là các dòng xuất chuẩn nói chung
  - thường là xuất ra màn hình, ra cửa sổ console hoặc terminal.
  - Để dữ liệu đầu ra được ghi vào một file thì ta sử dụng dấu ">".
- Ví dụ: ta muốn danh sách các file trong một thư mục được ghi vào file *dir.txt*
  - sử dụng lệnh sau:
 

```
$ ls -al > dir.txt
```
  - hoặc:
 

```
$ ls -al 1> dir.txt
```

74

### || Định hướng lại file-chẩn-đoán-sai chuẩn STDERR

- STDERR là dòng xuất lỗi chuẩn nói chung và nó cũng thường xuất trực tiếp ra màn hình, console hay terminal.
- Cú pháp tương tự như STDOUT, tức là sử dụng ">" để xuất ra file và ">>" để nối vào một file đã có (chưa có thì hệ thống sẽ tự tạo ra).

76

Tuy nhiên điểm khác biệt là bạn phải chỉ rõ số 2, tức là “2>” hoặc “2>>”. Lí do là vì chỉ có 1 dòng xuất chuẩn và 1 dòng nhập chuẩn cho mỗi process mà thông thường hệ thống chỉ định là STDOUT và STDIN còn STDERR thì có thể ghi vào một file đã có nào đó.

Ví dụ: Ghi thông báo lỗi khi ghi đè file...

```
mv -i file folder/  
/dev/null: “thùng không đáy”, black hole.
```

77

File chuẩn đoán sai có định danh là số 2, nên nếu muốn đổi hướng đầu ra của file báo lỗi, ta dùng 2> tenfile trong lệnh

VD: \$lệnh 2> fileGhiLoi

2>> fileGhiLoi để nối đuôi & ghi log

Nếu có quan tâm đến lỗi để sửa

Nếu không quan tâm lỗi là gì, mà chỉ muốn giấu lỗi, ta có thể đẩy vào /dev/null

79

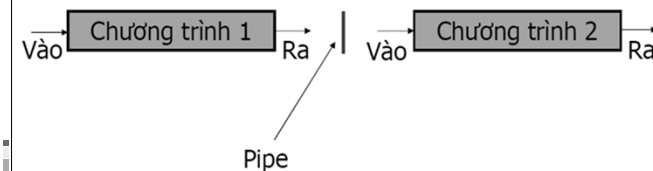
## || Ví dụ

- Cho chuyển thông báo lỗi
- Một câu lệnh có thông báo lỗi ra màn hình (cat /etc/pass). Có thể chuyển thông báo lỗi đó vào trong file.
- `cat /etc/pass 2> err_report`

78

## || Cơ chế pipe – đường ống lệnh

- <lệnh 1> | <lệnh 2> | ... | <lệnh n>



- Ví dụ: Đếm xem trong hệ thống có bao nhiêu người sử dụng, dùng lệnh:

```
$ cat /etc/passwd | wc -l
```

80

## || Các vấn đề cần nhớ

- Tập, thư mục, hệ thống tập, liên kết
- Quyền truy cập tập và thư mục
- Một số lệnh cơ bản thường dùng
- **Định hướng lại vào/ra và pipe (quan trọng)**

81

## || Thực hành trên phòng máy

- Thử nghiệm các lệnh với tập và thư mục
- Thay đổi quyền truy cập
- Copy, nén, giải nén, lưu trữ tập
- Định hướng lại vào/ra và pipe

82