

## Bài 2: Sử dụng dòng lệnh trong Linux

Nguyễn Minh Hải

Ngày 13 tháng 9 năm 2016

# Nội dung

- 1 Giới thiệu
  - Giới thiệu về dòng lệnh
- 2 Cấu trúc và các nhóm lệnh cơ bản
- 3 Tổ chức thư mục trong Linux
- 4 Wildcards
- 5 Đường dẫn cứng và mềm

# Nội dung

- 1 Giới thiệu
  - Giới thiệu về dòng lệnh
- 2 Cấu trúc và các nhóm lệnh cơ bản
- 3 Tổ chức thư mục trong Linux
- 4 Wildcards
- 5 Đường dẫn cứng và mềm

# Nội dung

- 1 Giới thiệu
  - Giới thiệu về dòng lệnh
- 2 Cấu trúc và các nhóm lệnh cơ bản
- 3 Tổ chức thư mục trong Linux
- 4 Wildcards
- 5 Đường dẫn cứng và mềm

# Giao diện đồ họa và dòng lệnh

*"Graphical User Interfaces make easy tasks easy, while Command Line Interfaces make difficult tasks possible"*

- Giao diện đồ họa phổ biến với người dùng phổ thông, trực quan và dễ dùng
- Giao diện dòng lệnh ra đời từ rất lâu, rất phổ biến với lập trình viên và người quản trị máy chủ
- Giao diện dòng lệnh cho phép chúng ta "ra lệnh" và "nói chuyện" trực tiếp với hệ điều hành
- Có thể thực hiện nhiều công việc phức tạp một cách dễ dàng
- Khó học vì phải nhớ nhiều, hiểu cấu trúc HĐH và đòi hỏi tư duy lập trình

**Câu hỏi 1.** Tìm hiểu thêm một số công việc phức tạp mà dòng lệnh cho phép chúng ta làm.

# Khái niệm và chương trình

- *Command line* là khái niệm chỉ việc tương tác với HĐH thông qua việc gõ lệnh (bằng bàn phím)
- *Shell* là một chương trình nhận những lệnh từ bàn phím và chuyển chúng đến HĐH để thực thi
- Phần lớn các HĐH Linux cung cấp chương trình shell có tên là *bash* (Bourne Again SHell) (\*)
- Các shell khác là *csh* (C Shell), *tcsh* (T C Shell), *ksh* (Korn Shell)
- Trong GUI, chúng ta dùng các chương trình để tương tác với *shell* là các *terminal emulator* như *Konsole* và *Gnome-terminal* thường được gọi chung là *terminal*

**Câu hỏi 1.** Tại sao lại đặt tên là bash, và nó liên hệ thế nào với sh của unix.

**Câu hỏi 2.** Đọc thêm bài giới thiệu về Knome và Gnome.

# Nội dung

- 1 Giới thiệu
  - Giới thiệu về dòng lệnh
- 2 Cấu trúc và các nhóm lệnh cơ bản
- 3 Tổ chức thư mục trong Linux
- 4 Wildcards
- 5 Đường dẫn cứng và mềm

# Cấu trúc chung của lệnh

## Cấu trúc chung của lệnh (command)

tên lệnh [luật lựa chọn] ... [đối tượng] ...

command-name [option] ... [object] ...

## Các thành phần

- Tên lệnh
- Luật lựa chọn dùng để tùy chỉnh phương thức hoạt động, đầu ra, ... của lệnh; viết dưới dạng đầy đủ hoặc viết tắt, ví dụ *-h* hoặc *-human-readable*; một lệnh có thể kèm theo nhiều luật lựa chọn
- Đối tượng bị tác động (tệp, địa chỉ ip, chuỗi kí tự, ...)
- Ví dụ: `ls -l /etc/bin`



# Các nhóm lệnh cơ bản đầu tiên trong dòng lệnh

Các nhóm lệnh:

- Tra cứu lệnh
- Thư mục
- Tập tin
- Nội dung tập tin

## Nhóm tra cứu lệnh (1)

- Chủ yếu về lệnh **man** và một số lệnh khác như **whereis** và **whatis**
- **man** là lệnh hướng dẫn cách dùng lệnh khác. Các chức năng của lệnh được phân thành các nhóm sau (tương ứng với các phần trong mô tả của **man**); mỗi lệnh có thể có nhiều hơn một nhóm chức năng:
  - các chương trình thực thi hoặc lệnh shell
  - lệnh gọi hệ thống
  - lệnh gọi thư viện
  - các tệp tin đặc biệt
  - định dạng tệp tin và quy ước
  - trò chơi
  - các thể loại khác (gồm các gói macro và convention)
  - lệnh quản lí người dùng
  - định tuyến nhân (kernel routines)

## Nhóm tra cứu lệnh (2)

- Tìm hiểu các lệnh: **man** \$command, **man** \$configfile, **man** \$daemon, **man** -k string, **apropos** string, **man** \$section \$file, **man** man, **whatis**, **whereis**

**Câu hỏi 1.** Dùng lệnh **man** để tìm hiểu kĩ về lệnh **ls**.

**Câu hỏi 2.** Bên cạnh **man** còn có lệnh tra cứu khác như **info**, hoặc luật lựa chọn *--help*. Tìm hiểu cách dùng của các lệnh, luật lựa chọn này. **info** có điểm khác quan trọng nào so với **man**.

# Nhóm lệnh thư mục (1)

## Đường dẫn

- Hai loại đường dẫn
  - Đường dẫn tuyệt đối: đường dẫn ghi địa chỉ bắt đầu từ địa chỉ root
  - Đường dẫn tương đối: đường dẫn bắt đầu từ thư mục hiện tại hoặc từ thư mục người dùng (các ký hiệu `..` `~`)
- Ví dụ

```
[user_name@machine_name] $mkdir playground
```

```
[user_name@machine_name] $cd playground
```

```
[user_name@machine_name] ~/playground$mkdir dir1
```

Thư mục `dir1` có

- đường dẫn tuyệt đối `/home/user_name/playground/dir1`
- đường dẫn tương đối `/dir1` hoặc `~/playground/dir1`

## Nhóm lệnh thư mục (2)

Tìm hiểu các lệnh

- **pwd** (print working directory) Xem thư mục hiện tại
- **cd** (change directory) chuyển thư mục (chuyển vị trí)
- **ls** (list directory) liệt kê nội dung thư mục
- **mkdir** (make directory) tạo thư mục mới
- **rmdir** (remove directory contents) xóa thư mục
- **pushd** và **popd** quản lý ngăn xếp đường dẫn

Câu hỏi 1. Tìm hiểu về **pushd** và **popd**.

Câu hỏi 2. Xem thêm về luật lựa chọn **-l** và **-a** của **ls**

# Nhóm lệnh về tệp tin

## Nhận xét

Trong HĐH Linux mọi thứ từ ổ đĩa, thư mục, tệp tin, luồng dữ liệu, ... đều là tệp tin

- Các lệnh tìm hiểu trong phần này
  - **file** xem thông tin của tệp tin
  - **touch** tạo tệp tin
  - **rm** gỡ bỏ tệp tin
  - **cp** sao chép tệp tin
  - **mv** di chuyển tệp tin (có thể dùng để đổi tên)
  - **rename** đổi tên tệp tin

**Câu hỏi 1.** Các lệnh trên có tác dụng cho thư mục không?

**Câu hỏi 2.** Có thể dùng **mv** để đổi tên thư mục như thế nào?

# Nhóm lệnh về nội dung tệp tin

- Các lệnh cơ bản
  - **head** xem phần đầu nội dung tệp
  - **tail** xem phần sau nội dung tệp
  - **cat** ghi dữ liệu, ví dụ **cat > filename**
  - **tac** xem dữ liệu sao chép theo thứ tự ngược lại
  - Các lệnh xem nội dung khác như: **more**, **less**, **strings**

**Câu hỏi 1.** Tại sao dùng lệnh **cat** ta có thể xem nội dung của một tệp tin?

**Câu hỏi 2.** Lệnh **less** là phiên bản cải tiến của **more**, điểm cải tiến ở đây là gì?

# Nội dung

- 1 Giới thiệu
  - Giới thiệu về dòng lệnh
- 2 Cấu trúc và các nhóm lệnh cơ bản
- 3 Tổ chức thư mục trong Linux
- 4 Wildcards
- 5 Đường dẫn cứng và mềm



# Cây tệp Linux (1)

## Nhận xét

Các tệp trong Linux được cấu trúc dạng cây thứ bậc (File system Hierachy Standard(FHS))

## Cây tệp Linux (2)

Một số thư mục cần quan tâm

- Thư mục chủ (root /) đây là thư mục chứa mọi thư mục khác, là thư mục gốc của cả HĐH
- Thư mục khởi động /boot chứa các tệp tin cần thiết khi khởi động máy tính
- Thư mục binary /bin và /sbin đây là thư mục chứa các mã nguồn đã được biên dịch
- Thư mục thư viện /lib chứa các thư viện được dùng bởi các tệp tin nhị phân, trong /bin và /sbin (vai trò giống như DLL trong Windows)
- Thư mục /etc
- Thư mục dữ liệu người dùng thông thường /home
- Tìm hiểu các thư mục con của / như /root, /srv, /media, /mnt, /tmp

## Cây tệp Linux (3)

### Bài tập

1. Tìm cách tạo ra một thông điệp hiển thị khi khởi động hệ điều hành Linux của bạn.

# Nội dung

- 1 Giới thiệu
  - Giới thiệu về dòng lệnh
- 2 Cấu trúc và các nhóm lệnh cơ bản
- 3 Tổ chức thư mục trong Linux
- 4 Wildcards
- 5 Đường dẫn cứng và mềm

# Wildcards (1)

## Định nghĩa

Đây là những ký tự đặc biệt để mô tả "mẫu" tên gọi

## Tác dụng

- Ta có thể mô tả một lớp tên gọi (của tệp, thư mục) bằng wildcard

## Bảng ký hiệu

Ký hiệu	Ý nghĩa
*	chuỗi ký tự bất kỳ
?	1 ký tự đơn bất kỳ
[các ký tự]	một ký tự trong các ký tự liệt kê
[! các ký tự]	một ký tự khác với tất cả các ký tự liệt kê
[:class:]	một ký tự thuộc lớp ký tự class

## Wildcards (2)

Một số lớp ký tự

[alnum:]	chữ cái và chữ số
[alpha:]	chữ cái
[digit:]	chữ số
[lower:]	chữ cái in thường
[upper:]	chữ cái in hoa
[space:]	dấu trắng (dấu cách, tab, xuống dòng, ...)

# Wildcards (3)

## Ví dụ

- Liệt kê các tệp (thư mục) bắt đầu bằng chữ *b* và tận cùng bằng đuôi *.txt* như sau,

```
[user_name@machine_name] /$ls b*.txt
```

- Liệt kê các tệp (thư mục) mà tên có 7 chữ cái trong đó 4 chữ cái đầu là *Data*

```
[user_name@machine_name] /$ls Data???
```

- Liệt kê các tệp (thư mục) mà tên có chữ cái đầu là *a*, *b* hoặc *c*

```
[user_name@machine_name] /$ls [abc]*
```

- Liệt kê các tệp (thư mục) mà tên có chữ cái in hoa mở đầu

```
[user_name@machine_name] /$ls [[:upper:]]*
```

# Nội dung

- 1 Giới thiệu
  - Giới thiệu về dòng lệnh
- 2 Cấu trúc và các nhóm lệnh cơ bản
- 3 Tổ chức thư mục trong Linux
- 4 Wildcards
- 5 Đường dẫn cứng và mềm



# Khái niệm

## Định nghĩa

Đường dẫn cứng (hard link) là một tệp liên kết đến inode của tệp khác nằm trên hệ thống tệp

## Định nghĩa

Đường dẫn mềm (symbolic link hoặc soft link) là một loại tệp ghi lại địa chỉ (tương đối hoặc tuyệt đối) của một thư mục hoặc một tệp

**Câu hỏi 1.** Chỉ ra 2 hạn chế của đường dẫn cứng so với đường dẫn mềm.

## Ví dụ

Tiếp từ ví dụ nhóm lệnh thư mục

```
[user_name@machine_name] ~/playground$touch hello
```

```
[user_name@machine_name] ~/playground$ln hello h1
```

```
[user_name@machine_name] ~/playground$file h1
```

```
[user_name@machine_name] ~/playground$cd dir1
```

```
[user_name@machine_name] ~/playground/dir1$ln -s ../hello h2
```

- Câu hỏi 1.** Dùng lệnh `ls -li` trong thư mục `playground` và nhận xét về cột kết quả đầu tiên (ghi số inode\*) của `hello.txt` và `h1`.
- Câu hỏi 2.** Nếu xóa đường link đi thì tệp nó trở đến thế nào.
- Câu hỏi 3.** Nếu xóa hay di chuyển tệp nguồn thì đường link (cứng và mềm) sẽ như thế nào.