

## Bài 2. Lệnh

### 1. Giới thiệu

Hầu hết các distro Linux hiện nay đều cung cấp cho người dùng hai môi trường làm việc đồ họa và văn bản. Mặc dù môi trường đồ họa đã được phát triển đủ thân thiện với người dùng, nhưng với mục tiêu tiếp cận Linux như một quản trị hệ thống, tài liệu sẽ chỉ đưa ra các kiến thức liên quan đến môi trường văn bản (tức là chỉ cần sử dụng bàn phím để gõ lệnh là đủ). Cũng trong tài liệu này, nếu không có giải thích rõ ràng, mọi chỉ dẫn về hoạt động đều được thực hiện trong shell bash.

### 2. Kết cấu lệnh

Lệnh là một tập các ký tự được sắp xếp với mong muốn hệ điều hành thực hiện một yêu cầu của người dùng. Việc sắp xếp/tổ hợp các ký tự này phải tuân theo một số quy tắc đã xác định gọi là cú pháp lệnh.

Một lệnh trong Linux có thể chia thành 3 thành phần như sau:

**<command> [options] [arguments]**

#### a. Lệnh

Thường là tên file chứa mã lệnh (nhị phân hoặc shell script). Tên này hay được viết dưới dạng một động từ hoặc viết tắt của một ngữ động từ nhằm diễn giải chức năng của lệnh. Tuy nhiên do tập các lệnh trong Linux không được phát triển từ một nhóm tác giả duy nhất nên không thực sự có sự quy hoạch thống nhất cho việc đặt tên. Ngoài ra, người dùng cần chú ý đến một số đặc điểm sau về lệnh trong Linux:

- + Tên lệnh trong Linux phân biệt hoa-thường. Vì vậy khi gõ **runme** sẽ có kết quả khác với **RunME** nếu chạy được.
- + Mỗi lệnh được kết thúc bởi phím Enter.
- + Có thể nối nhiều lệnh với nhau trên một dòng bởi dấu **;** nếu chúng chạy rồi rạc hoặc bởi ký tự **|** (đọc là pipe) nếu chúng có gắn kết dữ liệu đầu vào/đầu ra với nhau. Kết thúc dòng lệnh vẫn là phím Enter.

Ví dụ:

```
cd /usr/share/doc; ls -l  
cat /etc/passwd | grep root
```

- + Có thể viết một lệnh trên nhiều dòng bằng cách sử dụng ký tự **\** ngay trước khi gõ các phím Enter (↵). Khi đó chỉ có phím Enter cuối cùng (không có **\**) là có ý nghĩa bắt đầu thi hành lệnh. Các phím Enter trước đó chỉ mang tác dụng xuống dòng, về đầu dòng. Mỗi dòng nếu chưa kết thúc nhập lệnh thường sẽ bắt đầu bởi ký tự **>** để nhắc nhở. Cách sử dụng này thường được dùng khi phải biểu diễn những lệnh dài và phức tạp.

Ví dụ:

```
./configure \↵  
> --prefix=/usr/local/myapps \↵  
> --with-ssl=enabled \↵
```

> **--with-lib=enabled.**↵

## b. Tùy chọn

Là một tập các ký tự định nghĩa từ trước (cố định sẵn trong code khi lập trình) được sử dụng với mục đích điều khiển lệnh hoạt động theo ý muốn của người gọi lệnh. Đa số các lệnh trong Linux không phân biệt thứ tự xuất hiện của tham số. Ví dụ: **ls -l -a** và **ls -a -l** là tương đương nhau. Khi sử dụng các tham số được chia thành hai loại:

- + Đơn ký tự: Đi ngay sau duy nhất một dấu trừ "-". Lệnh sẽ tự tách tất cả các ký tự sau dấu trừ để chuyển thành tham số. Ví dụ: **ls -al** tương đương với **ls -a -l**. Tùy chọn đơn ký tự thường được dùng trong các trường hợp gõ lệnh nhanh.
- + Đa ký tự: Đi ngay sau hai dấu trừ "--". Tùy chọn đa ký tự thông thường được sử dụng trong các shell script để diễn giải ý nghĩa việc sử dụng tùy chọn này của người dùng.

## c. Tham số

Là các từ/ngữ có ý nghĩa nhưng chưa xác định khi lập trình, cũng được dùng để điều khiển hoạt động của lệnh như tùy chọn.

Ví dụ: Trong hai lệnh **ls /usr** và **fdisk /dev/sda** các tham số được sử dụng là **/usr** và **/dev/sda** giúp hai lệnh **ls** và **fdisk** biết phải hoạt động với thư mục/đĩa cứng nào. Các thông tin này không thể được xác định từ khi lập trình.

## d. Đường dẫn

Là khái niệm dùng để giúp hệ thống định vị các file/thư mục.

### • Đường dẫn tuyệt đối (Absolute Path)

Mỗi file hay thư mục đều chỉ có một đường dẫn duy nhất tính từ thư mục gốc đến nó thông qua các thư mục con ở trung gian. Đường dẫn này gọi là đường dẫn tuyệt đối, nó là duy nhất với mỗi file/thư mục, không có hai cặp file/thư mục nào có chung đường dẫn tuyệt đối. Đường dẫn tuyệt đối luôn bắt đầu bằng thư mục gốc (ký hiệu là /) <sup>[1]</sup> và bên trong nó không có các ký tự dạng . (một dấu chấm) hoặc .. (hai dấu chấm cạnh nhau). Mỗi file/thư mục khi được truy xuất bởi nhân đều thông qua đường dẫn tuyệt đối bởi đường dẫn tuyệt đối cho phép định vị file/thư mục tại bất kỳ ngữ cảnh nào.

### • Đường dẫn tương đối (Relative Path)

Là đường dẫn không phải đường dẫn tuyệt đối. Đường dẫn này chỉ được hiểu đúng khi kết hợp với ngữ cảnh hay chính xác là thư mục hiện hành. Khi đó, hệ thống sẽ tự biến đổi từ đường dẫn tương đối sang đường dẫn tuyệt đối theo công thức sau:

$$\text{Absolute Path} = \text{Working Dir} + [ / ] + \text{Relative Path}$$

## e. Thứ tự thi hành lệnh

Độ ưu tiên khi shell thi hành lệnh là đường dẫn đến lệnh, alias và cuối cùng là tìm trong biến môi trường PATH.

Đường dẫn chỉ đến lệnh được ưu tiên hàng đầu và duy nhất. Ví dụ: khi gõ **./runme** shell sẽ lập tức thử nạp file **runme** trong thư mục hiện hành vào bộ nhớ để chạy. Nếu không tồn tại file hoặc file không thi hành được một lỗi sẽ được thông báo ra màn hình chứ shell sẽ không tiếp tục tìm kiếm lệnh khác.

Trong đa số các trường hợp, người dùng chỉ cần chỉ ra tên lệnh chứ không nhập vào đường dẫn đến lệnh. Khi đó shell sẽ tìm kiếm lệnh để thi hành theo thứ tự sau:

- + Alias: Là danh sách các bí danh dùng để thay thế các lệnh thực sự. Có thể xem/thiết lập các bí danh này bằng chính lệnh alias.
- + Các thư mục trong đường dẫn PATH.

Ví dụ:

Khi người dùng gõ lệnh `cp /etc/passwd /tmp` hệ thống trước tiên sẽ thay thế cp thành alias nó tìm được là `cp -i` sau đó nó lại tìm tại các thư mục có trong đường dẫn PATH để biết vị trí chính xác của file cp thi hành được (đường dẫn tuyệt đối `/bin/cp`) rồi dùng kết quả tìm được để nạp lệnh vào bộ nhớ. Khi đó lệnh chính xác được nạp vào bộ nhớ sẽ là `/bin/cp -i /etc/passwd /tmp`

Lưu ý: nếu lệnh cần chạy nằm trong thư mục hiện hành mà thư mục này không tồn tại trong nội dung biến môi trường PATH thì nếu chỉ gõ tên lệnh (không có đường dẫn đến lệnh) thì lệnh đó sẽ không được chạy.

## f. Sử dụng TAB

Trong nhiều trường hợp, việc bắt buộc phải gõ một đường dẫn dài là không thể tránh khỏi. Shell bash cung cấp một cơ chế tự động điền khá mạnh cho phép người dùng có khả năng phải gõ ít nhất có thể đó là việc sử dụng linh hoạt phím <TAB> trên bàn phím.

### • Ví dụ 1: Điền nhanh tên lệnh

Tại dòng lệnh gõ ký tự “s”, sau đó nhấn <TAB> một lần, trên màn hình không có thay đổi gì. Nhấn tiếp <TAB> lần thứ hai, một dòng thông báo sẽ hiện ra như sau: “Display all 257 possibilities? (y or n)” trong đó 257 là con số các lệnh bắt đầu bằng ký tự s mà người dùng có thể thi hành <sup>[2]</sup>.

Nhấn ký tự “n” để tiếp tục, sau đó nhấn tiếp ký tự “h” rồi nhấn <TAB> hai lần. Lần này shell sẽ hiện ra tất cả các lệnh bắt đầu bằng hai ký tự “sh”. Tiếp tục nhấn ký tự “i” sau đó nhấn <TAB> một lần, shell sẽ điền đủ lệnh “shift” vào dòng lệnh và thêm một ký tự cách.

### • Ví dụ 2: Điền nhanh một đường dẫn

Trong cửa sổ lệnh, gõ lệnh “cd” có dấu cách phía sau và gõ thêm hai ký tự “/u” rồi nhấn <TAB>, khi đó shell sẽ tự điền nốt đường dẫn thành “/usr”. Tiếp tục gõ thêm ba ký tự “/sh” rồi nhấn <TAB>, đường dẫn được hoàn thiện thành “/usr/share”.

### • Lưu ý khi dùng <TAB>

Việc sử dụng <TAB> chỉ đem lại hiệu quả về mặt thời gian khi nhấn <TAB> lúc số ký tự cần thiết đã được nhập, tránh gõ <TAB> quá sớm hoặc quá muộn.

## 3. Sử dụng hệ thống tài liệu trợ giúp

### a. Sử dụng tùy chọn --help

### b. Sử dụng man

### c. Sử dụng info

## 4. Sử dụng Lịch sử lệnh

- Dùng lệnh history
- File .bash\_history

- Các cú pháp của ! (!abc, !123, !\$)

## 5. Metacharacter

- Các ký tự \* ? {} / \$ ' ' " "

## 6. Thực hành

- Bài 1: Tìm cách gõ lệnh

Hãy tìm ra ít nhất 5 cách để có được cửa sổ lệnh để có thể gõ lệnh.

- Bài 2: Làm quen với lệnh

Hãy đọc tài liệu man của các lệnh sau và “**chỉ nêu chức năng chính**” của các lệnh đó:

- cd, pwd, ls, mkdir, rmdir, cat, touch, rm, cp, mv
- grep, cut, join, head, tail, wc, sed, tr, tee
- tar, gzip, bzip2
- which, whatis, apropos

- Bài 3: Tạo cấu trúc thư mục đơn giản

Hãy sử dụng những lệnh đã biết ở bài 2 để tạo cấu trúc thư mục sau bằng số lệnh ít nhất có thể, số lần gõ bàn phím ít nhất có thể:

```

/-
|-/lab
|   |-/data
|   |   |- file1.txt
|   |   |- file2.txt
|   |-/project

```

Lưu ý: Các file được tạo có nội dung được chép từ file `/etc/passwd`.

- Bài 4: Xóa, di chuyển, sao chép file và thư mục

Hãy sao chép thư mục `/lab` ở bài 3 thành thư mục `/backup` (lưu ý thư mục `/backup` chưa tồn tại) sau đó tạo thêm hai file có nội dung giống như hai file trước có tên là `/lab/project/more1.txt` và `/lab/data/more2.txt`. Lại sao chép thư mục `/lab` ở trên thành thư mục `/backup` (lưu ý lúc này đã có thư mục này) với yêu cầu là không phải trả lời các câu hỏi có ghi đề không mà ghi đề trực tiếp không hỏi, sau đó xóa thư mục `/lab`.

- Bài 5: Tập hợp, nén, giải nén thư mục

Tạo file `/root/backup.tar.bz2` là bản backup dạng nén của thư mục backup trong bài 4. Sau đó giải nén file này thành thư mục `/lab` với cấu trúc thư mục như `/backup`.

- Bài 6: Lọc dữ liệu dạng đơn giản

Sử dụng lệnh `ls` để liệt kê tất cả các file/thư mục trong hệ thống file tính từ thư mục gốc `/` sau đó dùng lệnh `grep` để lọc ra tất cả các file có phần đuôi là `".txt"`

Cũng yêu cầu trên, làm lại nhưng lọc kết quả là các file có phần đầu là `"dat"`.

---

[1] Tuy nhiên trong một số tài liệu đường dẫn bắt đầu bằng ký tự `~` (thư mục cá nhân của người dùng) vẫn được tính là đường dẫn tuyệt đối.

[2] Con số này có thể khác nhau trên các máy.