

CHUYÊN ĐỀ HỆ ĐIỀU HÀNH LINUX

Tuần 2: CÁC LỆNH CƠ BẢN

GVLT: NGUYỄN Thị Minh Tuyền

Cú pháp lệnh

- Cú pháp:

`command [flags] arg1 arg2 arg3 ...`

- Các thành phần cách nhau một khoảng trắng.

- Các cờ thường theo sau dấu “-” hoặc “--” (nhất là các cờ nhiều ký tự)

- Ví dụ:

- `ls -a -l -F`

- `bash --version`

- Nhiều cờ có thể dùng chung một ký tự “-”, ví dụ: `ls -al` tương đương `ls -a -l`

- Để tự động điền dòng lệnh: ấn phím <Tab>

Các lệnh cơ bản [1]

- Một số lệnh được tích hợp vào bash : "built-in"

`echo "message"` hiển thị message ra màn hình

option `-n` : hiển thị message ra màn hình và không xuống dòng

`read toto` đọc một dòng từ bàn phím, sau đó lưu vào biến `toto`.

- Ví dụ:

```
$ echo "Nhập tên của bạn:"
```

```
Nhập tên của bạn:
```

```
$ read name
```

```
Tony
```

```
$ echo "Tên bạn là: $name"
```

```
Tên bạn là: Tony
```

Các lệnh cơ bản [2]

- Đa phần các lệnh những chương trình bên ngoài: ít nhất 800 lệnh (nằm /bin, /usr/bin, etc.)

<code>cat file</code>	Hiển thị tất cả nội dung của file
<code>more file</code>	Tương tự + dừng sau mỗi màn hình
<code>ls</code>	Liệt kê nội dung của thư mục
<code>cd dir</code>	Thay đổi thư mục hiện hành
<code>pwd</code>	Hiển thị thư mục hiện hành
<code>mkdir dir</code>	Tạo một thư mục con dir
<code>rmdir dir</code>	Xoá thư mục con dir, nếu thư mục đó rỗng

Các lệnh cơ bản [3]

<code>cp f1 f2</code>	Copy file <code>f1</code> thành file <code>f2</code>
<code>cp f1 f2 ... fn rep</code>	Copy tất cả các file <code>f1, f1, ... fn</code> vào thư mục <code>rep</code>
<code>mv f1 f2</code>	Đổi tên file <code>f1</code> thành file <code>f2</code>
<code>mv f1 f2 ... fn rep</code>	Di chuyển các file <code>f1, f2, ... Fn</code> vào thư mục <code>rep</code>
<code>rm f1 ... fn</code>	Xoá các file <code>f1 ... fn</code>
<code>vi f1</code>	Mở file <code>f1</code> với trình soạn thảo <code>vi</code>
<code>touch f1</code>	Thay đổi ngày tháng cập nhật của <code>f1</code> hoặc tạo một file rỗng
<code>chmod droits f1 ... fn</code>	Thay đổi quyền sử dụng file
<code>chown f1 ... fn</code>	Thay đổi quyền sở hữu file
<code>chgrp groupe f1 ... fn</code>	Thay đổi group của các file <code>f1 .. fn</code>

Lệnh `ls`

- Xem tất cả option của lệnh: `man ls`
- Ấn phím `q` để kết thúc
- Một số option thường dùng:
 - `-l`: hiển thị nội dung ở định dạng dài (chứa thông tin như chế độ file, số lượng link, người sở hữu file, tên nhóm, ...)
 - `-a`: hiển thị nội dung thư mục kể cả các tên bắt đầu bằng một dấu `.`
 - `-li`: với mỗi file, hiển thị số serial của file (inode number)

Lệnh pwd

➤ Hiển thị thư mục hiện hành

➤ Ví dụ:

```
$pwd
```

```
/Users/tuyennguyen1
```

Lệnh cd (change directory) [1]

- Thay đổi thư mục hiện hành
- Cú pháp: `cd dir`
- Ví dụ:

```
$pwd
```

```
/Users/tuyennguyen1
```

```
$cd test
```

```
$ pwd
```

```
/Users/tuyennguyen1/test
```


Lệnh cd (change directory) [2]

- `cd ~` : chuyển về thư mục home của người dùng (macro tượng trưng cho home directory của người dùng)
- `cd ..` : về thư mục cha của thư mục hiện hành

Lệnh mkdir, rmdir

- **mkdir**: Tạo một thư mục mới

- Option `-p` để tạo thư mục cha nếu chưa tồn tại
- Option `-v` để liệt kê các thư mục đã được tạo sau khi thực hiện xong

- Ví dụ:

```
$ mkdir test
```

```
$ mkdir -p p1/p2/test
```

- **rmdir**: Xoá một thư mục rỗng

- Ví dụ:

```
$ rmdir test
```

Lệnh touch

- Nếu file đã tồn tại, cập nhật lại ngày tháng của file.
- Tạo một file rỗng với quyền mặc định.
- Ví dụ:

```
$ touch file.txt
```

Lệnh cp

- `cp source dest`: Copy file source thành file dest
- `cp f1 f2 ... fn dir`: Copy tất cả các file f1, f1, ... fn vào thư mục dir
- Option
 - `-f` ghi đè (nếu file đích không thể mở, xoá đi vào tạo file mới, không cần hỏi xác nhận và bỏ qua quyền)
 - `-i` hỏi trước khi ghi đè
 - `-r` , nếu source là một thư mục, copy toàn bộ thư mục

Lệnh mv

- `mv f1 f2`: Đổi tên file `f1` thành file `f2`
- `mv dir1 dir2`: Đổi tên thư mục `dir1` thành `dir2`
- `mv f1 f2 ... fn dir`: Di chuyển các file `f1`, `f2`, ... `fn` vào thư mục `dir`
- `rm f1 ... fn`: Xóa các file `f1` ... `fn`
- `-r` : xóa cả thư mục và file

Lệnh ln

- Tạo liên kết (link), có thể xem như một shortcut trong windows

- Ví dụ:

```
$ ln -s dir1 firstdir
```

```
$ ln -f /tmp/test.txt
```

- Option :

- -s : tạo symbolic link,
- -f : Nếu file đích đã tồn tại, unlink nó để link lại.
- -F : Nếu file đích đã tồn tại, xoá nó đi và tạo link mới.

Các quyền

Truy cập	File	Thư mục
Đọc	Đọc nội dung file	Hiển thị nội dung của thư mục
Ghi	Thay đổi nội dung file	Thay đổi nội dung của thư mục (thêm, xoá, di chuyển file, thư mục con)
Thực thi	Thực thi file	Truy cập vào nội dung file trong thư mục

Lệnh chmod

- Cú pháp:

`chmod mode file ...`

- Trong đó :

`mode = mask bảo vệ hoặc`

`mode = <u|g|o><+|-><r|w|x>`

Tài liệu

- Đa phần các lệnh được tài liệu hoá.
- Để xem hướng dẫn cho các lệnh built-in : `help`
- Ví dụ: `help echo`
- Để xem hướng dẫn các lệnh ngoài : `man` hoặc `info`
- Ví dụ: `man ls`
`info gcc`
- Những lệnh `help`, `man` hay `info` tự bản thân nó cũng được tài liệu hoá:
`help help`
`man man`
`info info`

ĐIỀU HƯỚNG, PIPE & FILTER

Điều hướng (redirection)

- Shell cho phép điều hướng dòng dữ liệu vào kênh vào chuẩn(stdin) từ một file

`commande arguments < fichier`

→ Đọc một file, ghi ra màn hình

Đối với kênh ra chuẩn (stdout) :

`commande arguments > fichier`

→ Đọc từ bàn phím, ghi ra màn hình

- Ta có thể kết hợp:

`commande arguments < f1 > f2`

Các biến thể

>	Tạo một file nếu file không tồn tại; nếu không xoá hoặc thất bại, tùy vào cấu hình
>	Tạo hoặc xoá (bắt buộc)
>>	Thêm vào cuối file nếu file tồn tại hoặc thất bại (append)
0< 1>	Tương tự < và >
2>	Điều hướng ra đầu ra lỗi (stderr)
<< <<<	"here document"

Pipe

- Một vùng nhớ cho phép hai tiến trình của cùng một máy giao tiếp đồng bộ.
- Cho phép nối đầu ra của một lệnh trở thành đầu vào của một lệnh khác, thông qua bộ đệm quản lý bởi hệ thống.
- Ký hiệu « | »: được đặt giữa hai lệnh; đầu ra của lệnh thứ nhất là đầu vào của lệnh thứ hai.

com1 arguments | com2 arguments

- Ta có thể kết hợp :
ls -l | sort -r | head -5
ls -l | grep "tuyen"

Bộ lọc (filter)

- Một số lệnh được lọc :
 - Lắng nghe đầu vào chuẩn
 - Hiển thị trên đầu ra chuẩn
- `cat` (không có tên)
 - ..`..` Copy mỗi dòng sau khi gõ RC ..
 - `^D` kết thúc file
- Một số bộ lọc khác : `sort`, `uniq`, `tr`, `head`, `tail`, `cut`, `paste`, `join`

Script

Script

- ▶ Script = file chứa lệnh, được thực thi bởi shell
- ▶ Phải báo cho hệ thống shell nào sẽ được sử dụng:
 - ▶ Dòng đầu tiên của file phải chứa ký hiệu `#!`
 - ▶ `#! /dir/interpreter`
- ▶ Ví dụ : `/bin/bash` hoặc `/bin/csh` hoặc `/usr/bin/python`
- ▶ Script phải được chuyển sang quyền thực thi : `chmod +x file`

Script bash

➤ Ví dụ : hello.sh

```
#!/bin/bash  
# My first script  
echo "Hello world!"  
exit 0
```

➤ Chạy file script :

```
$ chmod +x hello.sh # thực hiện một lần duy nhất  
$ ./hello.sh  
Hello world!
```

Bản chất của một file

- ▶ **Lệnh file** : nhận biết được bản chất của file theo nội dung của nó ;

```
$ file hello.sh
```

```
hello.sh: Bourne-Again shell script, text executable
```

- ▶ **Tìm các script trong thư mục hiện hành** :

```
$ file * | grep script
```

Mã kết thúc

- Khi một lệnh hoặc một script kết thúc, nó sẽ trả về một mã kết thúc :
 `exit [n]` Trong ngôn ngữ C : `exit (n);`
- Ý nghĩa : 0 thành công
 1...255 thất bại
- Ta có thể lấy mã này với \$? :
 \$./hello.sh
 Hello world!
 \$ echo \$?
 0
 \$ false
 \$ echo \$?
 1
 \$ true
 \$ echo \$?
 0