

# Bài 1: LINUX VÀ CÁC LỆNH CƠ BẢN

## I. Lý Thuyết

### 1. Các khái niệm cơ bản

- **Users (Người dùng)**: Để có thể sử dụng được Linux, bạn phải được cấp tài khoản (account) đăng nhập vào máy Linux. Thông tin về tài khoản bao gồm tên đăng nhập (username), mật khẩu đăng nhập (password), và các quyền truy xuất tập tin và thư mục mà bạn có được dựa vào tài khoản mà bạn đăng nhập và máy.

- **Group (Nhóm)**: Các người dùng làm việc trên cùng một bộ phận hoặc đang làm việc chung trên cùng một dự án (project) có thể được đưa vào cùng một nhóm. Đây là một cách đơn giản của việc tổ chức để quản lý người dùng.

- **File (Tập tin)**: Tất cả các thông tin trên Linux được lưu giữ trong các tập tin. Các tập tin được tạo ra bởi người dùng và người chủ tập tin có quyền truy xuất, tạo, sửa đổi, thiết lập kích thước của tập tin và phân phối quyền để cho phép người dùng khác có thể truy xuất tập tin.

- **Directory (Thư mục)**: Thư mục giống như Folder trong Windows. Nó được dùng để chứa các tập tin và thư mục khác, và tạo ra cấu trúc cho hệ thống tập tin. Dưới Linux, chỉ có một cây thư mục và gốc của nó là /. Giống như tập tin, mỗi thư mục có thông tin kết hợp với nó, kích thước tối đa và những người dùng được quyền truy xuất thư mục này, ...

- **Path (Đường dẫn)**: Đường dẫn là 1 chuỗi các thư mục và có thể kết thúc bằng tên của một tập tin. Các thư mục và tên tập tin được phân cách bởi ký tự /. Ví dụ : /dir1/dir2/file là một *đường dẫn tuyệt đối* tới file được chứa trong dir2, với dir2 được chứa trong dir1, và dir1 nằm trong thư mục gốc. Ví dụ khác: ~/homework là một *đường dẫn tương đối*, tính từ thư mục đăng nhập của người dùng, vào thư mục homework.

- **Permissions (Quyền)**: Quyền là một đặc tính quan trọng của Linux. Chúng tạo ra sự bảo mật bằng cách giới hạn các hành động mà người dùng có thể thực hiện đối với tập tin và thư mục. Các quyền đọc (read), ghi (write) và thực thi (execute) điều khiển việc truy xuất tới việc truy xuất tập tin của người tạo ra nó, nhóm và các người dùng khác. Một người dùng sẽ không thể truy xuất tới tập tin của người dùng khác nếu không có đủ quyền truy xuất.

- **Process (Tiến trình)**: Khi người dùng thực thi một lệnh, Linux tạo ra một tiến trình chứa các chỉ thị lệnh. Một tiến trình còn chứa các thông tin điều khiển như thông tin người dùng thực thi lệnh, định danh duy nhất của tiến trình (PID – process id). Việc quản lý của tiến trình dựa trên PID này.

- **Shell**: Trong chế độ console, người dùng giao tiếp với máy thông qua shell (hệ vỏ). Một shell là một chương trình thường được dùng để bắt đầu một chương trình khác từ dấu nhắc của shell. Một shell được cấu hình bằng việc thiết lập các biến môi trường cho nó. Khi đăng nhập vào Linux, một shell sẽ được tự động tạo ra, và các biến môi trường mặc nhiên (default) sẽ được thiết lập. Ở đây, ta sẽ sử dụng shell BASH (Bourne Again SHell), là shell thông dụng của hầu hết các hệ thống Linux.

### 2. Thực thi Lệnh

- **Nhập lệnh**: Để nhập lệnh, đơn giản bạn chỉ đánh vào tên của lệnh sau dấu nhắc của shell rồi nhấn Enter. Dấu nhắc của shell thường có dạng [user@host directory]\$, nó có thể được thiết lập lại, và có thể khác nhau đối với các máy khác nhau. Hầu hết các lệnh thường chấp nhận nhiều đối số (argument) hoặc lựa chọn (option) (thường được gọi là flag – cờ). Thông thường các đối số được đưa vào bằng cách sử dụng 1 hoặc 2 dấu -. Nếu một lệnh yêu cầu đối số và chúng ta không đưa vào, lệnh sẽ tự động hiển thị một mô tả ngắn về cách sử dụng các đối số kết hợp với nó. Một lệnh và các đối số thường có dạng như sau:

**command -a1 -a2**

**command --long\_argument\_name**

- **Biến môi trường PATH**: Đây là biến môi trường của shell mà cho phép các thư mục mà Linux có thể *nhìn thấy* được khi thực thi lệnh nếu đường dẫn đầy đủ của lệnh không được chỉ định rõ ràng. Biến môi trường **PATH** bao gồm 1 chuỗi tên các đường dẫn thư mục, phân cách bởi dấu ':'. Hầu hết các lệnh mà chúng ta sẽ thực hành đều nằm trong các thư mục mà đã được đưa vào biến môi trường **PATH** và có thể thực hiện đơn giản bằng cách nhập tên của nó tại dấu nhắc lệnh. Vì lý do bảo mật, thư mục hiện hành sẽ không được đưa vào biến môi trường **PATH**, do đó, để chạy một chương trình nằm trong thư mục hiện hành, chúng ta phải thêm './' vào trước tên chương trình:

**./command**

### 3. Một số lệnh cơ bản

Nhóm lệnh	Lệnh	Mục đích
Gọi sự trợ giúp	<b>command -h</b>	Hiển thị thông tin trợ giúp ngắn gọn về lệnh.
	<b>command --help</b>	Hiển thị thông tin trợ giúp ngắn gọn về lệnh.
	<b>man command</b>	Hiển thị trang trợ giúp đầy đủ của lệnh.
Liệt kê tập tin (file)	<b>ls</b>	Liệt kê nội dung của thư mục hiện hành.
	<b>ls -a</b>	Liệt kê tất cả tập tin, kể cả các tập tin có thuộc tính ẩn.
	<b>ls -l</b>	Hiển thị đầy đủ các thông tin (quyền truy cập, chủ, kích thước, ...)
	<b>ls   less</b>	
Thay đổi thư mục	<b>cd path</b>	Chuyển đến thư mục được chỉ định bởi <b>path</b> .
	<b>cd ~</b>	Chuyển về thư mục nhà.
	<b>cd -</b>	Chuyển về thư mục trước của bạn.
	<b>cd ..</b>	Chuyển về thư mục cha của thư mục hiện hành.
Quản lý tập tin và thư mục	<b>cp</b>	Cho phép tạo ra một bản sao (copy) của một tập tin hoặc thư mục: <b>cp source_path destination_path</b>
	<b>mkdir</b>	Cho phép tạo ra một thư mục mới (make directory), rỗng, tại vị trí được chỉ định: <b>mkdir directoryname</b>
	<b>mv</b>	Cho phép di chuyển (move) một tập tin từ thư mục này tới thư mục khác, có thể thực hiện việc đổi tên tập tin: <b>mv source_path destination_path</b>
	<b>rm</b>	Cho phép xóa (remove) các tập tin, dùng lệnh ' <b>rm - R</b> ' để xóa một thư mục và tất cả những gì nằm trong nó: <b>rm filename</b>
	<b>rmdir</b>	Dùng để xóa thư mục: <b>rmdir directoryname</b>
	<b>touch</b>	Tạo tập tin trống: <b>touch filename</b>
Xác định vị trí của tập tin	<b>find</b>	Tìm tập tin <b>filename</b> bắt đầu từ thư mục <b>path</b> : <b>find path -name filename</b>
	<b>locate</b>	Tìm tập tin trong cơ sở dữ liệu của nó có tên là <b>filename</b> : <b>locate filename</b>
Làm việc với tập tin văn bản	<b>cat</b>	Để xem nội dung của một tập tin văn bản ngắn, chúng ta dùng lệnh ' <b>cat</b> ' để in nó ra màn hình: <b>cat filename</b>
	<b>less</b>	Cho phép xem một tập tin dài bằng cách cuộn lên xuống bằng các phím mũi tên và các phím pageUp, pageDown. Dùng phím <b>q</b> để thoát chế độ xem: <b>less filename</b>
	<b>grep</b>	Một công cụ mạnh để tìm một chuỗi trong một tập tin văn bản. Khi lệnh ' <b>grep</b> ' tìm thấy chuỗi, nó sẽ in ra cả dòng đó lên màn hình: <b>grep string filename</b>
	<b>sort</b>	Sắp xếp các dòng trong tập tin theo thứ tự alphabet và in nội dung ra màn hình: <b>sort filename</b>
Giải nén	<b>bunzip2</b>	Giải nén một tập tin <b>bzip2 (*.bz2)</b> . Thường dùng cho các tập tin lớn: <b>bunzip2 filename.bz2</b>
	<b>gunzip</b>	Giải nén một tập tin <b>gzipped (*.gz)</b> : <b>gunzip filename.gz</b>
	<b>unzip</b>	Giải nén một tập tin PkZip hoặc WinZip ( <b>*.zip</b> ): <b>unzip filename.zip</b>
	<b>tar</b>	Nén và giải nén các tập tin <b>.tar, .tar.gz</b> : Ví dụ: <b>tar -xvf filename.tar</b> và <b>tar -xvzf filename.tar.gz</b>
Xem thông tin hệ thống	<b>date</b>	In ngày giờ hệ thống.
	<b>df -h</b>	In thông tin không gian đĩa được dùng.

	<b>free</b>	In thông tin bộ nhớ được dùng.
	<b>history</b>	Hiển thị các lệnh được thực hiện bởi tài khoản hiện tại.
	<b>hostname</b>	In tên của máy cục bộ (host).
	<b>pwd</b>	In đường dẫn đến thư mục làm việc hiện hành.
	<b>rwho -a</b>	Liệt kê tất cả người dùng đã đăng nhập vào network.
	<b>uptime</b>	In thời gian kể từ lần reboot gần nhất.
	<b>who</b>	Liệt kê tất cả người dùng đã đăng nhập vào máy.
	<b>whoami</b>	In tên người dùng hiện hành.
Các lệnh dùng theo dõi tiến trình	<b>ps</b>	Liệt kê các tiến trình đang kích hoạt bởi người dùng và PID của các tiến trình đó.
	<b>ps -aux</b>	Liệt kê các tiến trình đang kích hoạt cùng với tên của người dùng là chủ tiến trình.
	<b>top</b>	Hiển thị danh sách các tiến trình đang kích hoạt, danh sách này được cập nhật liên tục.
	<b>command &amp;</b>	Chạy <b>command</b> trong nền.
	<b>fg</b>	Đẩy một tiến trình nền hoặc bị dừng lên bề mặt trở lại.
	<b>bg</b>	Chuyển một tiến trình vào nền. Có thể thực hiện tương tự với Ctrl-z.
	<b>kill pid</b>	Thúc đẩy tiến trình kết thúc. Đầu tiên phải xác định <b>pid</b> của tiến trình cần hủy với lệnh <b>ps</b> .
	<b>killall -9 name</b>	Hủy tiến trình với <b>name</b> chỉ định.
	<b>nice level program</b>	Chạy <b>program</b> với cấp ưu tiên ngược <b>level</b> . Cấp <b>nice</b> càng cao, chương trình càng có mức ưu tiên thấp.

## II. Nội dung bài thực hành số 1

### 1. Tạo cây thư mục

Tạo cây thư mục như sau:

```

home
├── dsl
│   └── CTH
│       ├── user1
│       └── user2

```

Sử dụng lệnh **mkdir** để tạo thư mục con:

```

dsl@box:~$ pwd
/home/dsl
dsl@box:~$ mkdir CTH
dsl@box:~$ cd CTH
dsl@box:~/CTH$ mkdir user1
dsl@box:~/CTH$ mkdir user2
dsl@box:~/CTH$

```

Kiểm tra thư mục hiện tại

Tạo thư mục CTH

Tạo 2 thư mục user1/user2

### 2. Tạo tập tin

Lần lượt tạo các tập tin **test1.c**, **test2.c** nằm trong thư mục **user1** - tập tin **test3.c**, **test4.c** nằm trong thư mục **user2**

Để tạo file bạn có 2 cách, cách thứ nhất là tạo file rỗng bằng lệnh **touch**:

```
$touch test1.c
```

Tương tự ta tạo các file: **test2.c**, **test3.c**, **test4.c**

```

dsl@box: ~/CTH$ mkdir user1
dsl@box:~/CTH$ mkdir user2
dsl@box:~/CTH$ cd user1
dsl@box:~/CTH/user1$ touch test1.c
dsl@box:~/CTH/user1$ touch test2.c
dsl@box:~/CTH/user1$ cd ..
dsl@box:~/CTH$ cd user2
dsl@box:~/CTH/user2$ touch test3.c
dsl@box:~/CTH/user2$ touch test4.c
dsl@box:~/CTH/user2$ ls -l
-rw-rw-r-- 1 dsl staff 0 Jan 31 22:04 test3.c
-rw-rw-r-- 1 dsl staff 0 Jan 31 22:04 test4.c
dsl@box:~/CTH/user2$

```

Như bạn thấy kích thước các file được tạo ra bởi lệnh **touch** là 0 bytes. Bạn có thể dùng trình soạn thảo **vi** để bổ sung cho file sau này.

Cách thứ 2 là dùng lệnh **cat** với định hướng đầu ra là tên file như ví dụ sau:

```

dsl@box:~/user2$ cat >otherfile
Hello Linux
dsl@box:~/user2$ ls -l otherfile
-rw-rw-r-- 1 dsl staff 12 Feb 22 20:01 otherfile
dsl@box:~/user2$

```

Lệnh **cat** chuyển hướng cho phép bạn nhập vào nội dung cho file và kết thúc khi bạn nhấn phím **Ctrl+D**

### 3. Sao chép tập tin và thư mục

- Sao chép tập tin từ thư **test3.c** mục **user2** sang **user1**

- Kiểm tra tập tin trong **user1** và **user2**

```

dsl@box:~/user2$ cp test3.c ../user1
dsl@box:~/user2$
dsl@box:~/user2$ ls -l
-rw-rw-r-- 1 dsl staff user2 0 Feb 22 20:07 test3.c
-rw-rw-r-- 1 dsl staff user2 0 Feb 22 20:07 test4.c
dsl@box:~/user2$ ls -l ../user1
-rw-rw-r-- 1 dsl staff user1 0 Feb 22 20:06 test1.c
-rw-rw-r-- 1 dsl staff user1 0 Feb 22 20:06 test2.c
-rw-rw-r-- 1 dsl staff user1 0 Feb 22 20:07 test3.c
dsl@box:~/user2$

```

Muốn sao chép nhiều file bạn có thể dùng các kí tự đại diện \*,? hay liệt kê một danh sách các file cần sao chép. Ví dụ, lệnh sau đây sẽ chép file **test3.c**, **test4.c** vào thư mục **/user1**

**\$cp test3.c test4.c ../user1**

Nếu dùng kí tự đại diện bạn có thể sao chép như sau:

**\$cp \*.c /user1**

Nếu muốn sao chép toàn bộ cây thư mục (bao gồm file và thư mục con) bạn sử dụng tùy chọn **-R**. Ví dụ để sao chép toàn bộ thư mục **/mydata** vào thư mục **/tmp** bạn gọi **cp** như sau:

**\$cp -R /mydata /tmp**

### 4. Di chuyển file và thư mục

Bạn dùng lệnh **mv** để di chuyển hoặc đổi tên file. Trong Linux đổi tên file cũng tương tự như di chuyển file. Ví dụ:

-Di chuyển **test4.c** từ **user2** sang **user1**

-Kiểm tra những tập tin trong **user1** và **user2**

```
dsl@box:~/user2$ mv test4.c ../user1
dsl@box:~/user2$
dsl@box:~/user2$ ls -l
-rw-rw-r-- 1 dsl staff 0 Feb 22 20:07 test3.c
dsl@box:~/user2$
dsl@box:~/user2$ ls -l ../user1
-rw-rw-r-- 1 dsl staff 0 Feb 22 20:06 test1.c
-rw-rw-r-- 1 dsl staff 0 Feb 22 20:06 test2.c
-rw-rw-r-- 1 dsl staff 0 Feb 22 20:07 test3.c
-rw-rw-r-- 1 dsl staff 0 Feb 22 20:07 test4.c
dsl@box:~/user2$
```

user2

user1

Đổi tên **test4.c** trong thư mục hiện hành thành **test4.doc**

**\$mv test4.c test4.doc**

Đổi di chuyển các file **.doc** và **.c** vào thư mục **/tmp**

**\$mv \*.doc \*.c /tmp**

Nếu bạn muốn chuyển **user1** trong thư mục hiện hành vào **user2** với tên mới là **NewDir** bạn gọi **mv**

**\$mv user1/ user2/NewDir**

#### 4. Nén, giải nén

Nén thành tập tin **.tar**:

- Nén tập tin **test4.c** trong thư mục **user2** thành tập tin **test4.c.tar**

- Liệt kê danh sách các file trong thư mục **user2**

- Xóa tập tin **test4.c**

- Giải nén tập tin **test4.c.tar**

```
dsl@box:~$ cd user2
dsl@box:~/user2$ ls
test4.c
dsl@box:~/user2$ tar -cvf test4.c.tar test4.c
test4.c
dsl@box:~/user2$ ls
test4.c      test4.c.tar
dsl@box:~/user2$ rm test4.c
dsl@box:~/user2$ ls
test4.c.tar
dsl@box:~/user2$ tar -xvf test4.c.tar
test4.c
dsl@box:~/user2$ ls
test4.c      test4.c.tar
dsl@box:~/user2$
```

Nén -> test4.c.tar

Kiểm tra trong thư mục user2

Xóa test4.c

Giải nén test4.c.tar

Kiểm tra trong user2

Ngoài ra các bạn cũng có thể nén **thư mục** với cách thức tương tự như trên.

#### 5. Xóa tập tin, thư mục

Lệnh **rm**, **rmdir** để xóa tập tin hoặc thư mục

\* Chú ý: lệnh **rmdir** dùng để xóa thư mục rỗng, nếu muốn xóa thư mục có chứa thư mục con hoặc tập tin thì thêm tùy chọn **-r** sau lệnh **rm**. Ví dụ:

- Xóa tập tin **test1.c** trong thư mục **user1**

- Xóa tập tin **test4.c** trong thư mục **user2**

- Xóa thư mục **user2** (rỗng)

- Xóa thư mục **user1** (không rỗng)

```
/home/dsl/CTH
dsl@box:~/CTH$ rm user1/test1.c
dsl@box:~/CTH$ rm user2/test4.c
dsl@box:~/CTH$ rm -r user1
dsl@box:~/CTH$ rmdir user2
dsl@box:~/CTH$
```

### III. Bài Tập Thêm

Tạo cây thư mục như sau:

\*HDH

    \*\*DTH

        \*\*\*Linux

        \*\*\*Windows

    \*\*CTH

        \*\*\*Linux

        \*\*\*Windows

- 1.Tạo tập tin thuchanh.txt trong thư mục HDH/CTH/Linux
- 2.Copy tập tin vừa tạo sang thư mục HDH/CTH/Linux đổi tên thành luyentap.txt
- 3.Nén thư mục HDH/CTH thành tập tin nen.tar
- 4.Copy tập tin nén sang thư mục HDH/DTH/Windows
- 5.Giải nén tập tin