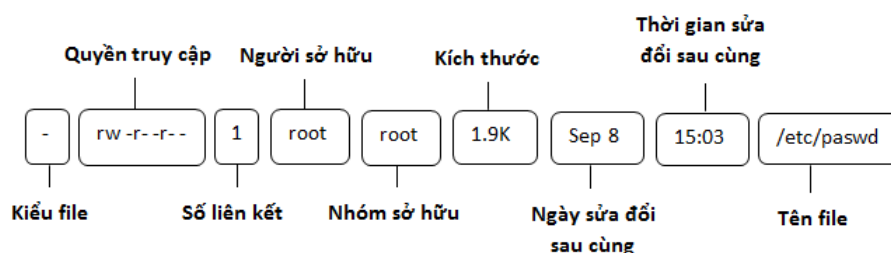


## Phần 1: Quyền truy cập file và thư mục

Trong Linux, mỗi file và thư mục được sở hữu bởi một người dùng và một nhóm người dùng nào đó. Xem thông tin của file:

Ví dụ: `student@linux ~ $ ls -l /etc/passwd`

Kết quả: `-rw-r--r-- 1 root root 1,9K Sep 8 15:03 /etc/passwd`



Trong quyền truy cập file:

- 3 ký tự đầu thể hiện quyền truy cập cho người sở hữu
- 3 ký tự tiếp theo thể hiện quyền truy cập đối với nhóm người sở hữu
- 3 ký tự cuối cùng thể hiện quyền truy cập đối với những người dùng và nhóm người dùng khác

Có 3 loại quyền truy cập trên file và thư mục:

Quyền	Trên file	Trên thư mục
r (đọc)	Đọc nội dung của file (cat)	Đọc nội dung của thư mục (ls)
w (ghi)	Thay đổi nội dung của file	Tạo mới file trong thư mục (touch)
x (thực thi)	Sử dụng fileName như một lệnh trong linux	Thay đổi thư mục (cd)

Ví dụ về quyền truy cập:

`student@linux $ ls -k file1.txt`

`-r- -r- -r- - 1 student student 15 Sep 14 17:22 file1.txt`

`student@linux $ cat » file1.txt`

`bash: file1.txt: Permission denied (Lý do: Không có quyền write)`

Để thay đổi quyền truy cập đối với file và thư mục, dùng lệnh **chmod**

Cú pháp: `chmod [options] files`

**chmod** chấp nhận 2 dạng tùy chọn (options):

- Các quyền truy cập được xác định dưới dạng chuỗi ký tự: u(user), g(group), o(other), a(all), w(write), x(execute), +(thêm quyền), -(xóa quyền), =(gán quyền)

Ví dụ: `student@linux ~ $ chmod ug = rw,o-rw,a-x *.txt`

Câu lệnh trên cài đặt quyền cho tất cả các file kết thúc bằng `*.txt` thành `rw-rw----`

- Các quyền truy cập được xác định bởi một chuỗi gồm 3 số bát phân. Ánh xạ các quyền vào các số bát phân được thể hiện như sau:

Quyền	Số bát phân	Số nhị phân
---	0	000
--x	1	001
-w-	2	010
-wx	3	011
r--	4	100
r-x	5	101
rw-	6	110
rwX	7	111

Linux sử dụng bộ ba số bát phân viết liền nhau lần lượt biểu diễn quyền truy cập của người sở hữu, nhóm người sở hữu tệp tin/thư mục, những người dùng và nhóm người dùng khác

Ví dụ: `student@linux ~ $ chmod 600 private.txt`

Câu lệnh trên cài đặt quyền cho file `private.txt` là `rw-----`

(?) Trên máy, tài khoản `student`, đặt quyền truy cập cho tệp tin `private.txt` sao cho tất cả các người dùng khác đều có thể đọc, ghi được.

## Phần 2: Quản lý người dùng trong Linux

### Quản lý người dùng

Linux là hệ điều hành đa người dùng, nghĩa là có thể có nhiều người truy cập và sử dụng một máy tính cài Linux. Mỗi người dùng có tên truy cập và mật khẩu riêng với những quyền hạn nhất định.

Cơ sở dữ liệu người dùng trong Linux được đặt trong file `/etc/passwd`. Mỗi người dùng nằm trên một dòng gồm các trường cách nhau bởi dấu ':', thông tin của người dùng gồm: Tên tài khoản, mật khẩu, UID, GID, mô tả, Home Directory, Shell

Ví dụ: `student@linux ~ $ tail -1 /etc/passwd`

Kết quả: `student:x:1001:1001:student,,,:/home/student:/bin/bash`

- Tên người dùng: Tối đa 8 ký tự, phân biệt hoa thường
- Mật khẩu: Mã hóa thay bằng chữ x và được đặt trong file `/etc/shadow`

- UID (user identification): Định danh người dùng, giúp hệ thống phân biệt giữa các người dùng khác nhau.
- Người dùng root có UID là 0
- Các UID từ 1-99 sử dụng cho các tài khoản hệ thống
- UID của nhóm người dùng thường sử dụng giá trị bắt đầu từ 100
- GID (group identification): Định danh nhóm mà người dùng thuộc vào. Mặc định GID trùng với UID
- Mô tả: Chứa các mô tả về tài khoản
- Home Directory: Thư mục home của từng user
- Shell: Tên chương trình sẽ thực thi ngay khi user login vào. Không có Shell người dùng không thể login. Mặc định trên linux dùng bash shell

Dòng đầu tiên của tập tin */etc/passwd* mô tả thông tin cho tài khoản root, tiếp theo là các tài khoản hệ thống, cuối cùng là các tài khoản người dùng.

### Hiển thị thông tin người dùng

- Để hiển thị tên người dùng hiện tại, dùng lệnh **whoami**  
 Ví dụ: *student@linux ~ \$ whoami*  
 Kết quả: *student@linux ~ \$ student*
- Để hiển thị thông tin của những người đã đăng nhập hệ thống, dùng lệnh **who**.  
 Ví dụ: *student@linux ~ \$ who*  
 Kết quả:  
     *student tty8 2015-09-16 09:00 (:0)*  
     *student pts/0 2015-09-16 09:25 (:0)*
- Để hiển thị thông tin người dùng hiện tại đang đăng nhập hệ thống, dùng lệnh **who am i**  
 Ví dụ: *student@linux ~ \$ who am i*  
 Kết quả: *student pts/0 2015-09-14 09:25 (:0)*
- Để hiển thị những người đã đăng nhập vào hệ thống và hoạt động của người đó, dùng lệnh **w**  
 Ví dụ: *student@linux ~ \$ w*
- Để biết thông tin chi tiết về người dùng đang đăng nhập hệ thống, ta dùng lệnh **id**  
 Ví dụ: *student@linux ~ \$ id*  
 Kết quả: *uid=1001(student) gid=1001(student) groups=1001(student)*

### Chuyển đổi người dùng

- Để chuyển đổi người dùng trên cửa sổ lệnh, sử dụng lệnh **su** (điều kiện: Phải biết *password* người dùng muốn chuyển đổi). Muốn quay về người dùng ban đầu dùng lệnh **exit**.

Cú pháp: *su userName*

Ví dụ: *student@linux ~ \$ su computer*

Kết quả: *computer@linux /home/student \$*

- Để chuyển đổi sang tài khoản root, dùng lệnh *su root*, hoặc *su*  
Trường hợp đang đăng nhập bằng tài khoản *root*, có thể chuyển sang bất kỳ tài khoản người dùng nào mà không cần biết *password*.
- Để chuyển đổi người dùng kèm theo môi trường người dùng của tài khoản đích dùng lệnh **su** –

Cú pháp: *su - userName*

Ví dụ: *student@linux ~ \$ su - computer*

Kết quả: *computer@linux ~ \$*

- Để chuyển sang tài khoản root với môi trường của người dùng root dùng lệnh *su - root* hoặc *su -*
- Để chạy các chương trình với quyền của root mà không cần biết password của tài khoản root, dùng lệnh *sudo* (Superuser Do)

Cú pháp: *sudo statement*

Ví dụ: *student@linux ~ \$ sudo su -*

Chú ý: Chỉ những người dùng thuộc nhóm *sudo* mới có quyền chạy lệnh **sudo**

## Quản lý người dùng

Cần sử dụng quyền root để quản lý người dùng.

*Lưu ý: Nếu như người dùng không đăng nhập bằng quyền root mà đăng nhập bằng tài khoản người dùng trong nhóm admin thì cần sử dụng thêm từ khóa **sudo** phía trước mỗi lệnh thực thi*

- Để thêm người dùng, dùng lệnh **useradd**

Cú pháp: *useradd [option] userName*

Trong đó, [Option]:

- *-p password*: Đặt password cho người dùng
- *-m*: Tạo thư mục home của người dùng
- *-d /home/directory*: Cài đặt tên đường dẫn home
- *-c 'description'*: Cài đặt mô tả
- *-u uid*: cài đặt UID, mặc định, lấy số UID tiếp theo để gán cho user
- *-g group*: Cài đặt nhóm khởi tạo
- *-G groupName*: Tên nhóm mà người dùng thuộc vào
- *-e*: Ngày hết hạn

Ví dụ:

*root@linux ~ # useradd -m -d /home/com01 -c "computer 01" com01*

*root@linux ~ # tail -1 /etc/passwd*

Kết quả: *com01:x:529:529:computer 01:/home/com01:/bin/bash*

- Để xóa người dùng, dùng lệnh **userdel**  
Cú pháp: `userdel [-r] userName`  
Trong đó, sử dụng tùy chọn `-r` nếu muốn xóa đường dẫn *home* của người dùng  
Ví dụ: `root@linux ~ # userdel -r com01`
- Để sửa đổi thông tin của người dùng trong tệp `/etc/passwd`, dùng lệnh **usermod**  
Cú pháp: `usermod [option] <tên tài khoản>`  
Trong đó, các tùy chọn `[option]` giống với lệnh `useradd`  
Ví dụ:  
`root@linux ~ # useradd -m -d /home/com01 -c "computer 01" com01`  
`root@linux ~ # tail -1 /etc/passwd`  
`root@linux ~ # usermod -c "computer 01 test" com01`  
`root@linux ~ # tail -1 /etc/passwd`  
Kết quả: `com01:x:529:529:computer 01 test:/home/com01:/bin/bash`

### Quản lý mật khẩu người dùng

- Mật khẩu của người dùng được mã hóa và lưu trong tệp tin `/etc/shadow`. Tệp tin này chỉ có thể đọc bởi tài khoản *root*.  
Ví dụ: `root@linux ~ # tail -4 /etc/shadow`
- Để thay đổi mật khẩu của người dùng, dùng lệnh **passwd**  
Cú pháp: `passwd [userName]`
- Trường hợp không có `[userName]` thì câu lệnh `passwd` cho phép thay đổi mật khẩu của người dùng hiện tại.  
Ví dụ: `root@linux ~ # passwd com01`

## Phần 3: Quản lý tiến trình trong Linux

### Hiển thị thông tin tiến trình

Để liệt kê các tiến trình đang thực thi, dùng lệnh **ps** (*process status*)

Cú pháp: `ps [option]`

Trong đó, các option là:

- `-a`: hiển thị các tiến trình của user được liên kết tới tty
- `-e` (`-A`): hiển thị thông tin về mỗi tiến trình
- `-f`: hiển thị PID của tiến trình cha và thời điểm bắt đầu
- `-l`: tương tự như `-f`
- `x`: hiển thị các tiến trình ngoại trừ các tiến trình là controlling tty (Ví dụ: `/sbin/mingetty tty*`)
- `u`: dạng hiển thị hướng đến người dùng

Lệnh *ps* có thể kết hợp với lệnh *grep* để tìm kiếm một tiến trình đang chạy.

Ví dụ: *student@linux* \$ *ps aux* → Liệt kê tất cả các tiến trình.

Khi đó, các thông số của tiến trình sẽ được liệt kê gồm có: Chủ nhân của tiến trình (owner), mã số nhận diện tiến trình (PID), thời gian hiện sử dụng CPU (%CPU), mức chiếm dụng bộ nhớ của tiến trình (%MEM), trạng thái tiến trình (STAT) và các thông tin khác.

Một số trạng thái của tiến trình thường gặp: R-đang thi hành, S-đang bị đóng, Z-ngừng thi hành, W-không đủ bộ nhớ, ...

Ngoài ra, có thể dùng lệnh **top** để xem các thông số liên quan đến các tiến trình, thông tin sử dụng tài nguyên của các tiến trình đó.

Cú pháp: *top* [*option*]

Trong đó, các *option* là:

- -u: Xem những tiến trình đang hoạt động dưới một tài khoản nào đó.
- -p <PID>: Xem một tiến trình thông qua PID của tiến trình đó.
- -c: Hiển thị đầy đủ dòng lệnh thay vì hiển thị tên lệnh tạo tiến trình
- -d <time>: Thời gian tải lại các hiển thị liên quan đến lệnh top. Giá trị được tính theo giây. Mặc định là 5s.

Có thể dùng thêm lệnh *pgrep* để xem PID của một tiến trình trên hệ thống. Ví dụ, khi mở một tệp tin soạn thảo bằng *emacs*, bạn sẽ tìm kiếm PID của tiến trình này như sau:

Ví dụ: *student@linux* \$ *pgrep gedit*

? Liệt kê các tiến trình

? Lưu kết quả của lệnh *top* vào tệp tin *topOutput.txt*?

? Mở một tệp tin bằng *gedit*, kiểm tra xem tiến trình này có hoạt động hay không?

### Liệt kê các tiến trình

Để liệt kê các tiến trình theo dạng cây, dùng lệnh **pstree**

Cú pháp: *pstree* [*option*]

Trong đó, các *option* là:

- -p: hiển thị PID
- -h: tô đậm những tiến trình hiện hành và những tiến trình con cháu của tiến trình hiện hành
- -a : chỉ ra tham số dòng lệnh. Nếu dòng lệnh của một quá trình được tráo đổi ra bên ngoài, nó được đưa vào trong dấu ngoặc đơn.
- -c : không thể thu gọn các cây con đồng nhất. Mặc định, các cây con sẽ được thu gọn khi có thể
- -H : giống như tùy chọn -h, nhưng quá trình con của quá trình hiện thời không có màu sáng trắng

- -l : hiển thị dòng dài
- -n : sắp xếp các quá trình cùng một tổ tiên theo chỉ số quá trình thay cho sắp xếp theo tên

### Các chế độ chạy của tiến trình

Để quản lí các chế độ chạy của một tiến trình, có thể dùng các lệnh **&** hoặc **Ctrl C**, **Ctrl Z**, **fg**, **bg**

- **&**: Cho tiến trình hoạt động ở trạng thái nền (*background*)  
Ví dụ: `student@linux $ ls -l -R/ > /home/student/list.txt &` → ứng dụng `ls` sẽ chạy nền bên dưới  
Hoặc:  
Ví dụ: `student@linux $ emacs &` → ứng dụng `emacs` sẽ chạy nền, khi đó người sử dụng có thể dùng terminal để thực hiện các lệnh khác.
- **Ctrl C**: Kết thúc tiến trình đang thực thi, sau khi ấn **Ctrl C**, có thể dùng lệnh `jobs` để hiển thị trạng thái của tiến trình đang chạy
- **Ctrl Z**: Tạm ngừng tiến trình đang thực thi sau đó, có thể dùng các lệnh `fg`, `bg` để tiếp tục:
  - `bg`: tiếp tục tiến trình vừa tạm ngừng ở trạng thái nền (*background*)
  - `fg`: tiếp tục tiến trình vừa tạm ngừng ở trạng thái hiện (*foreground*)

### Dừng một tiến trình

Lệnh **kill** thường được sử dụng để ngừng thi hành một tiến trình.

Cú pháp: `kill [option] <PID>`

Trong đó:

- option là một lựa chọn:
  - `-s` : xác định tín hiệu được gửi. Tín hiệu có thể là số hoặc tên của tín hiệu. Dưới đây là một số tín hiệu hay dùng:
    - \* `SIGHUP(1)`: Hangup (gọi lại tiến trình)
    - \* `SIGINT(2)`: Interrupt (Ngắt từ bàn phím **Ctrl**)
    - \* `SIGKILL(9)`: Hủy tiến trình ngay lập tức
    - \* `SIGTERM(15)`: Terminate – Kết thúc tiến trình, nhưng cho phép xóa các tệp tin tạm
  - `-p`: lệnh `kill` sẽ chỉ đưa ra chỉ số của quá trình mà không gửi một tín hiệu nào.
  - `-l` : hiển thị danh sách các tín hiệu mà lệnh `kill` có thể gửi đến các quá trình (các tín hiệu này có trong file `/usr/include/Linux/signal.h`)
- **PID**: mã số nhận diện tiến trình muốn dừng

Lệnh `kill` có thể gửi bất kỳ tín hiệu signal nào tới một tiến trình, nhưng theo mặc định nó gửi tín hiệu 15, `TERM` (là tín hiệu kết thúc chương trình). Super-user mới

có quyền dừng tất cả các tiến trình, còn người sử dụng chỉ được dừng các tiến trình của mình.

**Độ ưu tiên của một tiến trình** Để chạy chương trình với một độ ưu tiên nào đó, dùng lệnh **nice** Cú pháp: *nice -n <độ ưu tiên> <chương trình>*

Trong đó, độ ưu tiên từ -20 (độ ưu tiên cao nhất) đến 19 (ưu tiên thấp nhất), độ ưu tiên mặc định là 0.

Ví dụ: *student@linux \$ nice -n 12 abcd*

Để thay đổi độ ưu tiên của một tiến trình dùng lệnh *renice*

Cú pháp: *renice <độ ưu tiên> [option]*

Hoặc: *renice <độ ưu tiên> <pid>*

Trong đó, option là:

- -g : thay đổi quyền ưu tiên theo nhóm người dùng
- -p : thay đổi quyền ưu tiên theo chỉ số của quá trình
- -u : thay đổi quyền ưu tiên theo tên người dùng

Ví dụ: *student@linux \$ renice 1 3456*

Ví dụ: *student@linux \$renice +1 987 -u daemon root -p 32* → Lệnh trên sẽ thay đổi mức độ ưu tiên của quá trình có chỉ số là 987 và 32, và tất cả các quá trình do người dùng daemon và root sở hữu.

Chú ý: người dùng bình thường không thể thay đổi độ ưu tiên nhỏ hơn 0.

? Hãy thay đổi độ ưu tiên của một tiến trình bất kỳ trên hệ thống.