Quyền NTFS

Trong bất kỳ mạng Windows nào, bạn có thể đặt quyền chia sẻ cho các ổ đĩa và thư mục. Trên mạng đó, mỗi người dùng có thể chọn chia sẻ toàn bộ ổ đĩa hoặc các thư mục riêng lẻ với mạng.

Quyền NTFS (Hệ thống tệp NT) khả dụng cho các ổ đĩa được định dạng bằng NTFS. Ưu điểm của quyền NTFS là chúng ảnh hưởng đến người dùng cục bộ cũng như người dùng mạng và chúng dựa trên quyền được cấp cho từng người dùng cá nhân khi đăng nhập Windows, bất kể người dùng đang kết nối ở đâu.

NTFS là hệ thống tệp tiêu chuẩn của Windows NT và tất cả các hệ điều hành Windows sau nó. Windows 2000 trở lên đã giới thiệu một số thay đổi sâu rộng bao gồm quyền kiểm soát các quyền được kế thừa và cách các quyền được cấu hình để chia sẻ tệp và thư mục. Bạn sử dụng các thư mục chia sẻ để cung cấp cho người dùng mạng quyền truy cập vào tài nguyên tệp.

Quản trị viên có thể sử dụng tiện ích NTFS để cung cấp quyền kiểm soát truy cập cho các tệp và thư mục, vùng chứa và đối tượng trên mạng như một kiểu bảo mật hệ thống. Được gọi là "Bộ mô tả bảo mật", thông tin này kiểm soát loại quyền truy cập nào được phép cho người dùng cá nhân và nhóm người dùng.

Cùng với chức năng bổ sung mà NTFS cung cấp, tiềm năng cho các cấu hình phức tạp có thể dẫn đến đau đầu về quản trị. Nếu bạn không hiểu rõ về các quyền khác nhau và mối quan hệ của chúng, có thể khó giải quyết vấn đề về quyền khi nó xảy ra.

Để biết mô tả cấp thấp hơn của NTFS SECURITY_DESCRIPTOR, hãy xem metafile $ Secure trong hướng dẫn này.

Metafile $ Secure

Để giúp kiểm soát quyền truy cập vào các đối tượng NTFS, NTFS phiên bản 1.2 đã giới thiệu $ SECURITY_DESCRIPTOR được nhúng vào MFT (Master File Table).

Cấu trúc của $ SECURITY_DESCRIPTOR yêu cầu thông tin cụ thể từ mỗi tệp và thư mục. Bởi vì nhiều tệp và thư mục có thông tin giống hệt nhau, các bản ghi lặp lại trong thuộc tính này được chứng minh là chiếm quá nhiều dung lượng ổ đĩa có giá trị. Ngoài ra, phải mất nhiều thời gian hơn để tìm kiếm MFT.

Để cải thiện điều này, NTFS phiên bản 3.0 đã giới thiệu một trường mới - ID bảo mật trong thuộc tính $ STANDARD_INFORMATION dưới dạng liên kết đến tệp siêu dữ liệu mới - $ Secure - để hoạt động như một cơ sở dữ liệu bảo mật của hệ thống tệp trung tâm.

File and Folder Basic NTFS Permissions

Permissions are grouped in order to make it easier to assign complimentary permissions to users. These groups are called "basic" permissions. The table below shows how permissions are assigned to basic permissions in each case.

Permissions	Basic Full Control	Basic Modify	Basic Read & Execute	Basic List Folder Contents	Basic Read	Basic Write
Travers Folder/Execute File	✔	✔	✔	✔		
List Folder/ Read Data	✔	✔	✔	✔	✔	
Read Attributes	✔	✔	✔	✔	✔	
Read Extended Attributes	✔	✔	✔	✔	✔	
Create Files/Write Data	✔	✔				✔
Create Folders/Append Data	✔	✔				✔
Write Attributes	✔	✔				✔
Write Extended Attributes	✔	✔				✔
Delete Subfolders and Files	✔					
Delete	✔	✔				
Read Permissions	✔	✔	✔	✔	✔	✔
Change Permissions	✔					
Take Ownership	✔					
Synchronize	✔	✔	✔	✔	✔	✔

File and Folder Advanced Permissions

The permissions that you can set on folders and files depend on how an object is being accessed.

The reason that these permissions are called "advanced" permissions is because they appear in the Advanced Security Settings dialog box. To get to them, you must click the Advanced button in the Properties dialog box, Security tab.

The following is a list of file and folder advanced permissions with a short description for each:

- **Traverse Folder/Execute File**
  - Traverse Folder: Allows or denies moving through a restricted folder to reach files and folders beneath the restricted folder in the folder hierarchy. Traverse folder takes effect only when the group or user is not granted the "Bypass traverse checking user" right in the Group Policy snap-in. This permission does not automatically allow running program files.
  - Execute File: Allows or denies running program (executable) files.

- **List Folder/Read Data**
  - List Folder: Allows or denies viewing file names and subfolder names within the folder. List Folder only affects the contents of that folder and does not affect whether the folder you are setting the permission on will be listed.
  - Read Data: Allows or denies viewing data in files.
- **Read Attributes**
  - Allows or denies viewing the attributes of a file or folder, for example, "read-only" and "hidden".
- **Read Extended Attributes**
  - Allows or denies viewing the extended attributes of a file or folder. Extended attributes are defined by programs and may vary by program.
- **Create Files/Write Data**
  - Create Files: Allows or denies creating files within the folder.
  - Write Data: Allows or denies making changes to a file and overwriting existing content.
- **Create Folders/Append Data**
  - Create Folders: Allows or denies creating subfolders within the folder.
  - Append Data: Allows or denies making changes to the end of the file but not changing, deleting, or overwriting existing data.
- **Write Attributes**

    - Allows or denies changing the attributes of a file or folder, for example, "read-only" or "hidden".
    - The Write Attributes permission does not imply creating or deleting files or folders, it only includes the permission to make changes to the attributes of an existing file or folder.
- **Write Extended Attributes**
    - Allows or denies changing the extended attributes of a file or folder. Extended attributes are defined by programs and may vary by program.
    - The Write Extended Attributes permission does not imply creating or deleting files or folders, it only includes the permission to make changes to the extended attributes of an existing file or folder.
- **Delete Subfolders and Files**
    - Allows or denies deleting subfolders and files, even if the Delete permission has not been granted on the subfolder or file.
- **Delete**
    - Allows or denies deleting the file or folder. If you don't have Delete permission on a file or folder, you can still delete it if you have been granted Delete Subfolders and Files on the parent folder.
- **Read Permissions**
    - Allows or denies reading permissions of a file or folder.
- **Change Permissions**
    - Allows or denies changing permissions of the file or folder.
- **Take Ownership**
    - Allows or denies taking ownership of the file or folder. The owner of a file or folder can always change permissions on it, regardless of any existing permissions that protect the file or folder.
- **Synchronize**
    - Allows or denies different threads to wait on the handle for the file or folder and synchronize with another thread that may signal it. This permission applies only to multithreaded, multiprocessing programs.

LINUX

Phân quyền truy cập file bằng lệnh chmod

(Theo trang Quantrimang.com)

Bạn đã tìm thấy phiên bản Linux mà mình yêu thích, nhưng bây giờ lại thấy bối rối vì các lệnh terminal và quyền truy cập file trên Linux. Hoặc có thể bạn có một trang web được host trên máy chủ Linux và gặp phải một số vấn đề về quyền truy cập file, chỉ có thể được giải quyết bằng cách sử dụng dòng lệnh.

Bất kể bạn đang ở trong trường hợp nào, một trong những lệnh Linux cần thiết nhất để học, tuy nhỏ nhưng vô cùng mạnh mẽ được gọi là chmod. Nhưng trước khi giải thích lệnh này dùng để làm gì, ta phải tìm hiểu một chút về cách Linux xử lý việc bảo mật file.

Tìm hiểu về lệnh chmod và việc phân quyền file trong Linux

Khái niệm cơ bản về quyền đối với file trong Linux

Các hệ điều hành Linux thực sự là những hệ thống giống như Unix (tham khảo bài viết Unix/Linux là gì? để biết thêm chi tiết) và các hệ thống giống như Unix tiếp cận quyền truy cập file như sau:
Mỗi file đều có chủ sở hữu (**owner**), xác định "**user class**" (lớp người dùng) của file. Mỗi file cũng có một nhóm (**group**), xác định "**group class**" (lớp nhóm) của file. Bất kỳ người dùng hệ thống nào không phải là chủ sở hữu và không thuộc cùng một nhóm đều được xác định là thuộc lớp khác (**others**).

Tất cả các file trên những hệ thống giống như Unix đều có quyền được gán cho cả ba lớp và chúng xác định hành động nào có thể được thực hiện bởi các lớp đã nói đối với file đã cho.

Ba hành động có sẵn trên một hệ thống giống như Unix là: **read** (đọc - khả năng mở và xem nội dung của file), **write** (ghi - khả năng mở và sửa đổi nội dung của file) và **execute** (thực thi - khả năng chạy file như một chương trình thực thi).

Nói cách khác, các quyền của file xác định xem:

- Chủ sở hữu có thể đọc, viết và thực thi file không.
- Nhóm có thể đọc, viết và thực thi file.

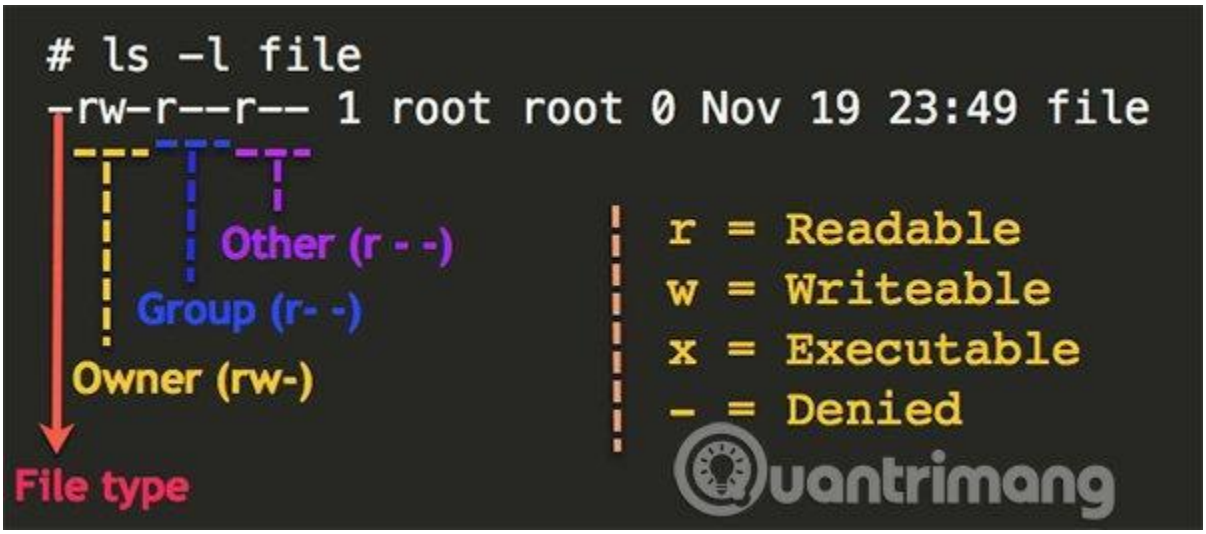
- Bất cứ ai khác có thể đọc, viết và thực thi file không.

Quyền truy cập file Linux có thể được hiển thị ở hai định dạng.

Định dạng đầu tiên được gọi là **symbolic notation** (ký hiệu tượng trưng), là một chuỗi gồm 10 ký tự: Một ký tự đại diện cho loại file và 9 ký tự đại diện cho các quyền đọc (**r**), ghi (**w**) và thực thi (**x**) của file theo thứ tự chủ sở hữu, nhóm, và những người dùng khác. Nếu không được phép, biểu tượng dấu gạch ngang (-) sẽ được sử dụng.

Ví dụ:

```
-rwxr-xr--
```



Điều này có nghĩa nó là một file thông thường với quyền đọc, ghi và thực thi cho chủ sở hữu, đọc và thực thi cho nhóm và chỉ đọc cho những người khác.

Định dạng thứ hai được gọi là **numeric notation** (ký hiệu số), là một chuỗi gồm ba chữ số, mỗi chữ số tương ứng với user, nhóm và các quyền khác. Mỗi chữ số có thể nằm trong khoảng từ **0** đến **7** và mỗi giá trị của chữ số có được bằng cách tính tổng các quyền của lớp:

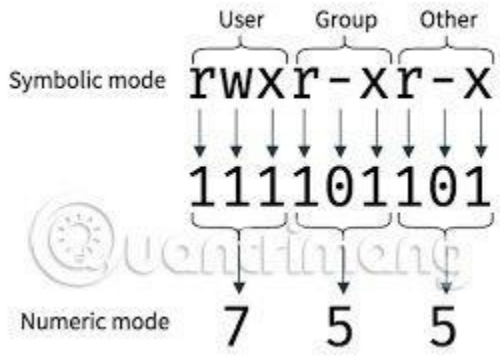
- **0** có nghĩa là không có quyền nào được cho phép.
- **+1** nếu lớp có thể thực thi file.
- **+2** nếu lớp có thể ghi vào file.
- **+4** nếu lớp có thể đọc file.

Nói cách khác, ý nghĩa của từng giá trị chữ số là:

- **0**: Không được phép thực hiện bất kỳ quyền nào
- **1**: Thực thi
- **2**: Viết
- **3**: Viết và thực thi
- **4**: Đọc
- **5**: Đọc và thực thi

- **6**: Đọc và viết
- **7**: Đọc, viết và thực thi

Vì vậy, ở ví dụ trên, **-rwxr-xr--** sẽ là **754** trong ký hiệu số. Đó là quyền đối với file Linux một cách ngắn gọn.



Chmod là gì?

Trên các hệ thống giống như Unix, chmod là một lệnh cấp hệ thống, viết tắt của "**change mode**" và cho phép bạn thay đổi cài đặt quyền của file theo cách thủ công.
Đừng nhầm lẫn với **chown.** Đó là một lệnh cấp hệ thống khác trên những hệ thống giống như Unix, viết tắt của "**change owner**" và cho phép bạn gán quyền sở hữu một file cho người dùng khác, hoặc **chgrp,** viết tắt của "**change group**" và gán file cho một nhóm khác. Đây là những lệnh quan trọng cần biết, nhưng không được sử dụng phổ biến như **chmod.**

Chmod 644 có nghĩa là gì?

Việc đặt quyền của file thành 644 cho phép chủ sở hữu có thể truy cập và sửa đổi file theo cách họ muốn, trong khi mọi người dùng khác chỉ có thể truy cập mà không thể sửa đổi và không ai có thể thực thi file ngay cả chủ sở hữu. Đây là cài đặt lý tưởng cho những file có thể truy cập công khai vì nó duy trì cân bằng giữa sự linh hoạt và tính bảo mật.

Chmod 755 có nghĩa là gì?

Đặt quyền của file thành 755 về cơ bản giống như 644, ngoại trừ mọi người đều có quyền thực thi. Quyền này chủ yếu được sử dụng cho các thư mục có thể truy cập công khai, vì cần có quyền thực thi để thực hiện thay đổi đối với thư mục.

Chmod 555 có nghĩa là gì?

Việc đặt quyền của file thành 555 làm cho file không thể bị sửa đổi bởi bất kỳ ai, ngoại trừ superuser (siêu người dùng) của hệ thống. Quyền này không thường được sử dụng như 644, nhưng việc biết về nó vẫn rất quan trọng, vì cài đặt quyền chỉ đọc ngăn ngừa các thay đổi ngẫu nhiên và/hoặc giả mạo.

Chmod 777 có nghĩa là gì?

Đặt quyền truy cập file thành 777 cho phép mọi người có thể làm bất cứ điều gì họ muốn với file. Đây là một rủi ro bảo mật rất lớn, đặc biệt là trên các máy chủ web! Theo nghĩa đen, bất cứ ai cũng có thể truy cập file, sửa đổi theo cách họ muốn và thực thi nó trên hệ thống. Bạn có thể tưởng tượng thiệt hại tiềm tàng nếu một kẻ lừa đảo nhúng tay vào file này.

Cách sử dụng Chmod trên Linux

Lệnh chmod có định dạng đơn giản:

```
chmod [permissions] [file]
```

Quyền có thể được cung cấp trong ký hiệu số, đây là định dạng tốt nhất để sử dụng khi bạn muốn gán quyền cụ thể cho tất cả các lớp:

```
chmod 644 example.txt
```

Quyền cũng có thể được cung cấp trong ký hiệu tượng trưng, rất hữu ích khi bạn chỉ muốn sửa đổi các quyền của một lớp cụ thể. Ví dụ:

- chmod u=rwx example.txt
- chmod g=rw example.txt
- chmod o=rw example.txt

Bạn có thể sửa đổi quyền cho nhiều lớp, chẳng hạn như ví dụ này cho chủ sở hữu quyền đọc/ghi/thực thi nhưng nhóm và các người dùng khác chỉ có quyền đọc/thực thi:

```
chmod u=rwx,g=rw,o=rw example.txt
```

Khi gán cùng một quyền cho nhiều lớp, bạn có thể kết hợp chúng:

```
chmod u=rwx,go=rw example.txt
```

Nhưng lợi ích của việc sử dụng ký hiệu tượng trưng sẽ được thấy rõ khi bạn chỉ muốn thêm hoặc xóa quyền cho một hành động cụ thể đối với một lớp.

Ví dụ, lệnh sau thêm quyền thực thi cho chủ sở hữu file:

```
chmod u+x example.txt
```

Và lệnh này loại bỏ quyền ghi và thực thi cho người dùng khác:

```
chmod o-wx example.txt
```

Cuối cùng, nếu bạn muốn áp dụng một nhóm quyền cụ thể cho tất cả các file và mục trong một thư mục cụ thể (nghĩa là một chmod đệ quy), hãy sử dụng tùy chọn **-R** và nhắm mục tiêu tới một thư mục:

```
chmod -R 755 example_directory
```

Mặc dù lệnh chmod thoạt nhìn có vẻ hơi kỳ lạ, nhưng nó thực sự khá đơn giản và hoàn toàn hợp lý. Nếu bạn hiểu những điều trên, về cơ bản bạn đã thành thạo chmod!

Các lệnh như chmod, chown và chgrp chỉ là phần nổi của tảng băng Linux. Nếu bạn là người hoàn toàn mới với hệ điều hành này, bạn nên kiểm tra các lệnh cơ bản cho người mới sử dụng Linux cũng như những lệnh Linux không bao giờ nên chạy.
Bên cạnh đó, bạn cũng nên xem các meo và thủ thuật cho người dùng Ubuntu. Bạn sẽ nắm được mọi thứ cần biết để bắt đầu làm quen và cảm thấy thoải mái trên hệ điều hành này.

Chúc bạn thành công!

- Cách giới hạn quyền truy cập vào lệnh su trong Linux
- Cách dùng lệnh htop theo dõi các tiến trình hệ thống theo thời gian thực
- 15 lệnh Tar nên thử trong Linux
- Liệt kê tên thiết bị, thông tin ổ đĩa và phân vùng trong Linux với lsblk
- 14 lệnh Linux thú vị trong Terminal
- Các lệnh Shell cơ bản trong Linux

Các lệnh chown command trong linux và cách sử dụng chúng

Trong hệ điều hành Linux, mỗi file được đặt trong một nhóm sở hữu – group ownership và đặt bởi một chủ sở hữu. Chown là chữ viết tắt của “change owner” – đổi chủ sở hữu. Như tên gọi, chown command được dùng để thay đổi chủ sở hữu và nhóm chủ sở hữu của file, thư mục, links nếu bạn đang có quyền superuser của hệ thống Unix. Bài viết này sẽ hướng dẫn bạn cách sử dụng chúng.

Nếu một người dùng bình thường muốn chỉnh sửa file, superuser có thể dùng lệnh chown command này để thay đổi ownership và cho phép họ chỉnh sửa

Xem thông tin sở hữu

Đầu tiên, bạn cần đăng nhập VPS bằng SSH. Nếu bạn chưa biết cách dùng, hãy xem qua **bài hướng dẫn này**, nó có tất cả thông tin cần thiết.

Trước khi sử dụng lệnh chown, bạn cần biết thông tin tin sử hữu nhóm và user sở hữu trước đã. Để lấy các thông tin này, bạn có thể dùng lệnh **cd** rồi chuyển tới thư mục cần biết.

Ví dụ, nếu đường dẫn của file là **/tmp/TestUnix**, bạn có thể chuyển tới thư mục này bằng lệnh:

```
cd /tmp/TestUnix
```

Tại đây liệt kê danh sách file trong thư mục bằng lệnh sau:

```
ls -l
```

Trong bài hướng dẫn này, ví dụ có một file có tên **chownSample.txt** trong thư mục này. Output của lệnh trên sẽ có kết quả như sau:

```
-rw-r--r-- 1 root root 0 Feb 20 17:35 chownSample.txt
```

Phần đầu tiên **-rw-r–r–**, đại diện cho file permission (quyền sỡ hữu của file). Còn lại, thông tin đầu tiên *root* là user ownership, *root thứ 2* là group owener. Vậy **chownSample.txt** sở hữu bởi user root, và user này thuộc về nhóm có tên root.

Chown cho Files – đổi quyền sở hữu của file

Để thay đổi owner của file (chủ sở hữu của file), lệnh cơ bản sẽ như sau:

```
chown user filename(s)
```

Lấy ví dụ file trên là **chownSample.txt**, chúng tôi thay đổi từ user sở hữu là root sang một user khác có tên là *whales.* Thực hiện lệnh như sa:

```
chown whales chownSample.txt
```

Để kiểm tra thay đổi có được thực thi, bạn có thể dùng lại lệnh **ls -l**. Output sẽ hiện lên như sau:

```
-rw-r--r-- 1 whales root 0 Feb 20 17:45 chownSample.txt
```

Lệnh trên có thể chỉnh một chút để thay đổi quyền group owner, như sau:

```
chown user[:group] filename(s)
```

Lấy ví dụ trên nếu bạn muốn đổi group owner của chownSample.txt thành group *aquatic*, vậy lệnh sẽ cần được thực thi như sau:

```
chown whales:aquatic chownSample.txt
```

Để kiểm tra file đã được đổi thành công chưa, bạn dùng lại lệnh **ls -l**. Kết quả sẽ như sau:

```
-rw-r--r-- 1 whales aquatic 0 Feb 20 17:50 chownSample.txt
```

Nếu chỉ có group cần thay đổi, mà giữ nguyên owner, thì bạn gõ lệnh sau như sau:

```
chown :aquatic chownSample.txt
```

Chown có chức năng tương tự như lệnh chgrp khi bạn không đưa ra thông tin owner.

Tóm lại, cấu trúc của lệnh chown command với các tùy chọn là:

```
chown [OPTIONS] [USER] [:GROUP] filename(s)
```

Chown cho thư mục – đổi ownership của thư mục

Chown cũng có thể áp dụng cho thư mục. Thư mục này chỉ chứa files hoặc thư mục hoặc cả hai.

Lấy ví dụ chúng tôi có thư mục TestUnix, với các quyền khi dùng lệnh **ls -l** liệt kê có kết quả như sau:

```
drwxr-xr-x 2 root root 4096 Feb 20 17:35 TestUnix
```

Như bạn thấy đoạn đầu **drwxr-xr-x**, đại diện của việc phân quyền thư mục. Còn phần hai có 2 chữ root là đại diện cho quyền sở hữu. *Root đầu tiên* là thông tin

user sở hữu và *root thứ hai* là thông tin nhóm sở hữu. TestUnix trong ví dụ này vì vậy có chủ sở hữu là root và thuộc về nhóm sở hữu là root.

Giống với files, bạn có thể thay chủ sở hữu cho thư mục này. Bạn thực hiện lệnh như sau để đổi chủ sở hữu của thư mục này thành *whales*:

```
chown whales /TestUnix
```

Để đổi group sở hữu, dùng lệnh:

```
chown :aquatic /TestUnix
```

Để thay đổi cả owner và group owner, bạn dùng lệnh sau:

```
chown whales:aquatic /TestUnix
```

Lệnh này cũng dùng được cho nhiều file và thư mục. Bạn thao tác như cấu trúc bên dưới:

```
chown [OPTIONS] [USER][:GROUP] file1 file2
```

Ví dụ của lệnh trên là:

```
chown whales:aquatic /tmp/TestUnix/chownSample.txt /tmp/TestUnix
```

Chown cho Links

Chown command có thể được dùng trên symbolic link và soft link. Symbolic link là liên kết tham chiếu tới vị trí file gốc vật lý đã tồn tại. Lệnh ln được dùng để tạo soft links. Ví dụ như file **chownSample.txt**, chúng tôi tạo symbolic link bằng lệnh sau::

```
ln -s chownSample.txt symlink
```

Để xác nhậ chủ sở hữu và nhóm chủ sở hữu, chúng tôi lại dùng lệnh **ls -l**. Lệnh này sẽ xuất kết quả như bên dưới:

```
-rw-r--r--  1 root root  0 Feb 19 22:01 chownSample.txt
lrwxr-xr-x  1 root root 5 Feb 19  7 22:01 symlink -> chownSample.txt
```

Có 2 kết quả. Một là file vật lý và 2 là file symbolic link. Để ví dụ, chúng tôi sẽ muốn đổi ownership của file symlink này, lệnh sẽ như sau:

```
chown whales symlink
```

Lệnh ở trên thay đổi ownership của file **chownSample.txt.** Khi thực hiện lệnh **ls -l** ta sẽ thấy kết quả như sau:

```
-rw-r--r--  1 whales root  0 Feb 19 22:01 chownSample.txt
lrwxr-xr-x  1 root root 5 Feb 19  7 22:01 symlink -> chownSample.txt
```

Nếu bạn muốn thực sự đổi ownership của symbolic vậy bạn cần dùng option **-h**. Lệnh này như sau:

```
chown -h whales symlink
```

Tại đây nếu bạn sử dụng command **ls -l,** vậy output sẽ như sau:

```
-rw-r--r--  1 whales root  0 Feb 19 22:01 chownSample.txt
lrwxr-xr-x  1 whales root  5 Feb 19 7 22:01 symlink -> chownSample.txt
```

Sử dụng chown đệ quy (recursive)

Chown command áp dụng lên thư mục, tuy nhiên, nếu dùng thông thường thì không áp dụng được cho file và thư mục con bên trong thư mục áp dụng. Vậy để thay đổi ownership cho toàn bộ thư mục con và file bên trong thư mục đó, chúng ta cần thực hiện lệnh một cách đệ quy.

Rất đơn giản, chúng ta chỉ cần thêm tùy chọn **-R** khi chạy lệnh, kết quả sẽ như sau:

```
chown -R [USER][:GROUP] Directory
```

Nếu ạn có thư mục TestUnix với nhiều thư mục con, lệnh trên sẽ thay đổi user owner của toàn bộ thư mục con đó và thư mục chính sang user *whales*.

```
chown -R whales /TestUnix
```

Tóm lại

Vậy thôi, giờ bạn đã biết cơ bản về chown command. Unix system có trang hướng dẫn chi tiết cho từng command. Nó có thể giúp bạn sử dụng lệnh này điêu luyện nhất và tận dụng mọi khả năng của nó. Bạn lấy tài liệu hướng dẫn này bằng lệnh **man chown**. Chúng tôi hy vọng bài viết này hữu ích và giúp bạn quản lý tốt file VPS hiệu quả, an toàn, và chuyên nghiệp nhất. Chúc lập trình vui vẻ!

Lệnh chgrp

Các file và người dùng thuộc vào các nhóm, đây là cách truy cập file thuận tiện cho nhiều người dùng. Khi đăng nhập, mặc định sẽ là thành viên của một nhóm được thiết lập khi người dùng root tạo tài khoản người dùng. Cho phép một người dùng thuộc nhiều nhóm khác nhau, nhưng mỗi lần đăng nhập chỉ là thành viên của một nhóm. Để thay đổi quyền sở hữu nhóm của các tập tin và thư mục. Cấu trúc lệnh này là:

```
chgrp [tùy chọn] [nhóm] file
```

Các tuỳ chọn của lệnh là:

- -c : hiển thị thông báo chỉ với các file mà lệnh làm thay đổi sở hữu
- -f : Bỏ qua hầu hết các lỗi
- -R : Thực hiện thay đổi quyền sở hữu đối với thư mục và file theo đệ quy
- --help: Hiển thị thông báo trợ giúp.

Thay đổi chủ sở hữu của một tập tin hoặc thư mục:

```
chgrp group file
```

Thay đổi đệ quy chủ sở hữu của một thư mục và tất cả nội dung bên trong của nó:

```
chgrp -R group folder
```

Thay đổi chủ sở hữu của một liên kết tượng trưng:

```
chgrp -h user symlink
```

Ví dụ: Thay đổi quyền sở hữu nhóm tệp bằng cách sử dụng lệnh `chgrp`.

```
[dang@test1]$ touch file1
[dang@test1]$ ls -l file1
-rw-r--r-- 1 dang dang 0 Jan 19 21:45 file1
[dang@test1]$chgrp root file1
ls -l file1
-rw-r--r-- 1 dang root 0 Jan 19 21:45 file1
```

Đối với chown và chgrp chúng ta cần lưu ý:

Lưu ý: Khi change qua user hay group nào, thì user dùng lệnh phải có quyền trên user/group đó, hoặc chạy lệnh từ root.