Điều khiển truy cập là gì? (khải niệm): kiếm trên mạng

Các mô hình điều khiển truy cập: DAC, MAC, Role Base AC, Rule Base AC (Giới thiệu qua): tài liệu FPT

Subject, object, access right: 4.2

Cấu trúc của AC: figure 4.2: P.119

# AC trong linux:

## File và thư mục trong UNIX

Tất cả các loại tập tin UNIX được quản lý bởi hệ điều hành bằng inodes. Một inode (nút chỉ mục) là một cấu trúc điều khiển chứa thông tin quan trọng cần thiết cho một tệp cụ thể. Một số tên tệp có thể được liên kết với một inode đơn, nhưng một inode hoạt động được liên kết với chính xác một tệp và mỗi tệp được điều khiển bởi chính xác một inode. Các thuộc tính của tệp cũng như các quyền của nó và các thông tin điều khiển khác được lưu trữ trong inode. Trên đĩa, có một bảng inode hoặc danh sách inode, chứa các inode của tất cả các tệp trong hệ thống tệp. Khi một tập tin được mở, inode của nó được đưa vào bộ nhớ chính và được lưu trữ trong một bảng inode bộ nhớ cư

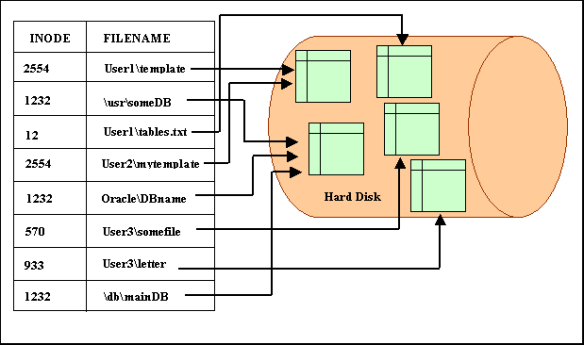
**inode** là một cấu trúc dữ liệu chứa các metadata của mỗi file, thư mục trong các hệ thống file Linux. Trong một inode có các metadata sau:

* Dung lượng file tính bằng bytes.
* [**Device ID**](http://en.wikipedia.org/w/index.php?title=Device_ID&action=edit&redlink=1) : mã số thiết bị lưu file.
* [**User ID**](http://en.wikipedia.org/wiki/User_identifier_%28Unix%29) : mã số chủ nhân của file.
* [**Group ID**](http://en.wikipedia.org/wiki/Group_identifier_%28Unix%29).: mã số nhóm của chủ file.
* [**File**](http://en.wikipedia.org/wiki/File_system_permissions) [***mode***](http://en.wikipedia.org/wiki/File_system_permissions) : gồm kiểu file và các quyền truy cập file (permissions).
* Hệ thống phụ và các cờ hạn chế quyền truy cập file.
* [**Timestamps:**](http://en.wikipedia.org/wiki/Timestamp) các mốc thời gian khi: bản thân inode bị thay đổi (ctime), nội dung file thay đổi (mtime) và lần truy cập mới nhất (atime).
* [**Link count**](http://en.wikipedia.org/wiki/Reference_counting) : số lượng [**hard links**](http://en.wikipedia.org/wiki/Hard_link) trỏ đến inode.
* Các con trỏ (từ 11-15 con trỏ) chỉ đến các blocks trên ổ cứng dùng lưu nội dung file. Theo các con trỏ này mới biết file nằm ở đâu để đọc nội dung.

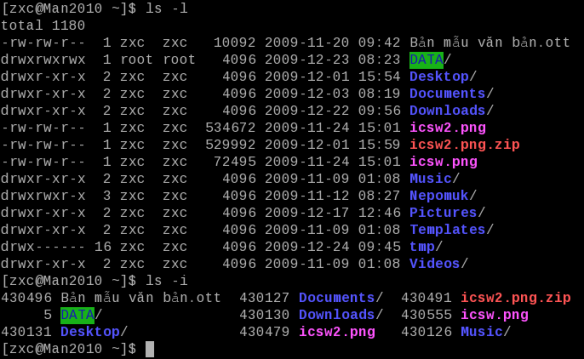
Có hai chú ý trong nội dung inode:

1. Inode không chứa tên file, thư mục.
2. **Các con trỏ là thành phần quan trọng nhất: nó cho biết địa chỉ các block lưu nội dung file và tìm đến các block đó có thể truy cập được nội dung file.**

Mỗi inode có một **id inode**, tạo thành một **bảng inode** ghi ở một khu vực riêng trên ổ cứng.

Chú ý là trong inode không có tên file. Tên file được lưu trong thư mục cùng với số inode của nó (xem lệnh ls -i dưới đây).

Khi truy cập đến một file, hệ điều hành từ tên file tìm ra số inode rồi dùng số inode để đọc nội dung inode, từ đó theo các con trỏ để đọc nội dung file.

Trong hình trên, lệnh **ls -l** liệt kê nội dung của thư mục /home/zxc. Lệnh thứ hai **ls -i** liệt kê các số inode của các đối tượng trong thư mục: thư mục DATA có số inode là 5, file icsw.png có số inode là 430555.

Lệnh **stat <path/filename>** cho biết chi tiết về nội dung inode:

**III. Tên file và hard link, soft link**

Trong một hệ thống file, **mỗi file có một và chỉ một inode, mỗi inode cũng chỉ có một số inode duy nhất**. Nhưng **một file có thể có nhiều tên file** tùy theo số hard link trỏ đến nó.

Một **hard link** là một liên kết (link) **trỏ đến vị trí lưu một file trên ổ cứng**:

* Nếu đổi tên, xóa hoặc di chuyển file gốc sang thư mục khác, hard link vẫn mở được file đó vì nó vẫn trỏ đến vị trí lưu file cố định trên ổ cứng.
* Tên hard link có thể khác tên file gốc, hard link có thể nằm trong một thư mục khác với thư mục của file gốc. Vì vậy **một file có thể có nhiều tên file nằm ở các thư mục khác nhau**. Khi truy cập vào hard link (ví dụ nhấn chuột) sẽ truy cập đến file (mở hoặc chạy).
* Nếu đồng thời mở một file từ các hard link và tên file gốc, khi sửa ở một bản, các bản khác cũng sẽ thay đổi theo sau khi refresh hoặc reload vì thực chất là sửa trên cùng một file.
* Nếu xóa hard link hoặc xóa tên file gốc nhưng **còn một hard link, file vẫn không bị xóa**. **File chỉ bị xóa khi không còn cái gì trỏ đến vị trí lưu nó**. Như vậy **muốn xóa một file, phải xóa tên file và tất cả các hard link của nó**.
* Hard link không tạo được với thư mục và không tạo được với file nằm trên một partition khác.
* Hard link được tạo bởi lệnh **ln**: **ln <path/tên file> <tên hard link>.**

**Soft link** (còn gọi là Symbolic link hoặc symlink) là một liên kết **tạo một đường dẫn khác đến thư mục hoặc file gốc**.

* Ví dụ file gốc ***passwd*** có đường dẫn là ***/etc/passwd***. Trong thư mục /home/zxc tạo một soft link đặt tên là “mật khẩu” trỏ đến file đó. Như vậy đường dẫn mới đến file ***/etc/passwd*** là ***/home/zxc/mật khẩu***. Khi truy cập đến một trong hai đường dẫn trên đều là truy cập đến file passwd.
* Nếu đổi tên, xóa hoặc chuyển file gốc sang thư mục khác thì soft link mất tác dụng, không truy cập được đến file đó nữa. Khác với hard link, khi xóa file có soft link, file bị xóa thật.
* Có thể tạo soft link với thư mục và file nằm trên partition khác.
* Soft link tạo bằng lệnh **ln -s <path/tên file> <tên soft link>**

**Tóm lại thì tên file, hardlink, softlink giống và khác nhau cái gì?**

Tên file giống như tên khai sinh và tên các hardlink giống như các bí danh (tên gọi khác). Chúng đều **tham chiếu trực tiếp** **tới một số inode** cụ thể và từ đó tới inode trỏ tới các block đang lưu file trên ổ cứng. Đi từ tên file hay hardlink đều thông qua số inode để đến cùng một chỗ trên ổ cứng.

Softlink **không tham chiếu trực tiếp đến số inode mà tham chiếu đến “cấp trung gian” là tên file** (kèm theo đường dẫn ở một thư mục cụ thể), từ tên file mới đến số inode rồi inode để truy cập vào file. Vì vậy nếu “cấp trung gian” là tên file thay đổi, file bị di chuyển hoặc xóa là softlink “bơ vơ” không truy cập được vào nội dung file nữa.

Hardlink chỉ tạo được với file nằm trên cùng một partition, không tạo được với thư mục hoặc với file nằm trên partition khác.

Softlink tạo được với thư mục và tạo được với thư mục, file nằm trên partition khác.

**Thư mục:**

Thư mục trong linux tổ chức theo cây phân cấp. Thư mục là 1 tệp chứa danh sách tệp và con trỏ tới các inode liên quan.