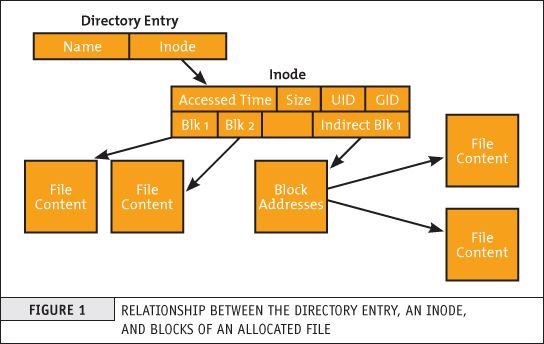
Lệnh về tập tin và thư mục

|  |  |
| --- | --- |
| Câu lệnh | Chú thích |
| # cd /home | Vào thư mục “/home” |
| # cd .. | Trở về thư mục trước 1 bậc |
| # cd ../.. | Trở về thư mục trước 2 bậc |
| # cd | Trở về thư mục home |
| # cd ~tên user | Đến thư mục home của user |
| # cd – | Quay lại thư mục trước đó |
| # cp file1 file2 | Copy file 1 thành file 2 |
| # cp dir/\* . | Copy toàn bộ file trong thư mục bao gồm cả thư mục hiện tại |
| # cp -a /tmp/dir1 . | Copy 1 thư mục bao gồm cả thư mục hiện tại |
| # cp -a dir1 dir2 | Copy thư mục 1 thành thư mục 2 |
| # iconv -l | Liệt kê danh sách Encodings |
| # iconv -f fromEncoding -t toEncoding inputFile > outputFile | Chuyển đổi mã hóa kí tự từ dạng này sang dạng khác |
| # ln -s file1 lnk1 | Tạo một liên kết tượng trưng tới tập tin hoặc thư mục |
| # ln file1 lnk1 | Tạo một liên kết vật lý đến tập tin hoặc thư mục |
| # ls | Xem danh sách tập tin và thư mục có trong thư mục cần xem |
| # ls -f | Như lệnh ls nhưng bao gồm cả tập tin ẩn |
| # ls -l | Như lệnh ls nhưng xem đc chi tiết của tập tin và thư mục |
| # ls \*[0-9]\* | Liệt kê tập tin và thư mục có chứa chữ số |
| # mkdir dir1 | Tạo thư mục |
| # mkdir dir1 dir2 | Tạo 2 thư mục dir1 dir2 cùng lúc |
| # mkdir -p /tmp/dir1/dir2 | Tạo 1 cây thư mục |
| # mv dir1 new\_dir | Di chuyển thư mục 1 và đổi tên thành thư mục mới |
| # pwd | Xem đường dẫn của thư mục hiện tại |
| # rm file1 | Xóa tập tin 1 nhưng cần xác nhận |
| # rm dir1 | Xóa thư mục 1 nhưng cần xác nhận |
| # rm -f file1 | Xóa tập tin 1 ngay lập tức (Cẩn thận khi dùng) |
| # rm -rf dir1 | Xóa thư mục 1 ngay lập tức (Cẩn thận khi dùng) |
| # touch -t 0712250000 file1 | Sửa thời gian của tập tin |

Lệnh tìm kiếm

|  |  |
| --- | --- |
| Câu lệnh | Chú thích |
| # find / -name file1/dir1 | Tìm tập tin hoặc thư mục trong thư mục gốc của hệ thống. Bắt đầu tìm kiếm từ thư mục “/” |
| # find / -user user1 | Tìm tập tin và thư mục của user 1 trong thư mục gốc của hệ thống. Bắt đầu tìm kiếm từ thư mục “/” |
| # find /home/user1 -name \\*.bin | Tìm trong thư mục “/home/user1” như tập tin đuôi “.bin” |
| # find /usr/bin -type f -atime +100 | Tìm binary file không được sử dụng trong vòng 100 ngày |
| # find /usr/bin -type f -mtime -10 | Tìm tập tin được tạo ra hoặc thay đổi trong vòng 10 ngày trở lại |
| # find / -name \*.rpm -exec chmod 755 ‘{}’ \; | Tìm những tập tin có đuôi “rpm” và cấp quyền |
| # find / -xdev -name \\*.rpm | Tìm những tập tin có đuôi “rpm” nhưng không tìm ở các phân vùng gắn ngoài như USB,CD/DVD,… |
| # whereis halt | Hiển thị toàn bộ vị trí liên quan của tập tin (đường dẫn, nguồn,…) |
| # which halt | Hiển thị đường dẫn của tập tin thực thi |

 Inode

Trong Linux, dữ liệu của các file được chia thành các block. Có nhiều cách tổ chức để liên kết các khối dữ liệu trong một file với nhau, một trong các cách đó là dùng chỉ mục (indexed allocation).

Trong một inode có các metadata sau:

Dung lượng file tính bằng bytes.

Device ID : id của thiết bị lưu file.

User ID : id chủ sở hữu của file.

Group ID: id nhóm của chủ sở hữu file.

File mode : gồm kiểu file và cách thức truy cập file.

Timestamps: các mốc thời gian khi: bản thân inode bị thay đổi (ctime, inode change time), nội dung file thay đổi (mtime, modification time) và lần truy cập mới nhất (atime, access time).

Link count : số lượng hard links trỏ đến inode. Các con trỏ chỉ đến các blocks trên ổ cứng dùng lưu nội dung file. Các con trỏ cho biết file nằm ở đâu để đọc nội dung

Inode xác định file và thuộc tính của nó (các thuộc tính đã liệt kê ở trên). Mỗi Inode được xác định bởi 1 con số duy nhất trong hệ thống tệp tin.

INODE là:

Inode là một cấu trúc dữ liệu trong hệ thống tệp truyền thống của các họ Unix ví dụ như UFS hoặc EXT3. Inode lưu trữ thông tin về 1 tệp thông thường, thư mục, hay những đối tượng khác của hệ thống tệp tin.

Có hai chú ý trong nội dung inode:

Inode không chứa tên file, thư mục.

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.

**Hard links** là các liên kết cấp thấp ( low-level links) mà hệ thống sử dụng để tạo các thành phần của chính hệ thống file, chẳng hạn như file và thư mục. Liên kết cứng sẽ tạo một liên kết trong cùng hệ thống tập tin với 2 inode entry tương ứng trỏ đến cùng một nội dung vật lý (cùng số inode vì chúng trỏ đến cùng dữ liệu).

Tất cả các hệ thống tệp tin dựa trên thư mục phải có ít nhất một liên kết cứng (link counts từ 1 trở lên) cung cấp tên gốc cho mỗi tệp tin.

Lệnh tạo liên kết cứng như sau: ln [file nguồn] [file đích]

Khi sử dụng lệnh rm để xóa file thì làm giảm đi một hard link. Khi số lượng hard link giảm còn 0 thì không thể truy cập tới nội dung của file được nữa.

Nội dung trong file nguồn vẫn còn vì khi xóa file file đích hệ thống chỉ xóa đi số link count trong inode của file đi 1.

**Symbolic links** là một file đặc biệt trỏ đến một file hoặc thư mục khác - được gọi là **target**. Khi được tạo, một symbolic links có thể được sử dụng thay cho target file. Nó có thể có một tên độc nhất, và được đặt trong bất kỳ thư mục nào. Nhiều symbolic links thậm chí có thể được tạo cho cùng một target file, cho phép truy cập target bằng nhiều tên khác nhau.

Tạo Symbolic link cho một file:

ln -s [target file] [Symbolic filename]

Lệnh ln trong Linux tạo ra liên kết giữa symbolic link file và file gốc.

-s – lựa chọn để tạo Symbolic Links.

[target file] – tên của file gốc hoặc nơi bạn muốn tạo file link tới

[Symbolic filename] – tên của file symbolic link.

Symbolic link không chứa bản sao dữ liệu của target file. Nó tương tự như một shortcut trong Microsoft Windows: nếu bạn xóa một symbolic link, target sẽ không bị ảnh hưởng. Vì chỉ đơn thuần là một shortcut, symbolic link không dùng đến inode entry. Nó sẽ tạo ra một inode mới và nội dung của inode này trỏ đến tên tập tin gốc.

Ngoài ra, nếu target của một symbolic link bị xóa, di chuyển hoặc đổi tên, symbolic link không được cập nhật. Khi điều này xảy ra, liên kết tượng trưng được gọi là "broken" hoặc "orphaned" và sẽ không còn hoạt động như một liên kết.

Xóa file file nguồn thì nội dung của file file đích sẽ không còn.

Nội dung của file đích không hiển thị được vì file đích trỏ đến một tập tin khác, mà tập tin này không tồn tại.

Tạo Symbolic Link Linux cho thư mục:

ln -s [Specific file/directory] [symlink name]

|  |  |
| --- | --- |
| Command Switch | Mô tả |
| –backup[=CONTROL] | backup từng file gốc |
| -d, -F, –directory | superuser được cho phép hard link |
| -f, –force | file đích bị xóa |
| -I, –interactive | thông báo trước khi xóa file đích |
| -L, –logical | chọn file gốc là symbolic links |
| -n, –non-dereference | symbolic links tới thư mục được xem như là files |
| -P, –physical | tạo hard links trực tiếp tới symbolic links |
| -r, –relative | tạo symbolic links tương đối tới link location |
| -s, –symbol | tạo symbolic links thay vì hard links |
| -S, –suffix=SUFFIX | ghi đè lên backup suffix thông thường |
| -v, –verbose | in ra tên của từng linked file |

Làm thế nào để đổi hoặc xóa Symbolic Link Linux:

Bạn có thể xóa link đã gắn với file hoặc thư mục bằng lệnh unlink hoặc rm

unlink [symlink to remove] or rm [symlink name]

| Hard links | Symbolic links |
| --- | --- |
| Chỉ liên kết được tới file, không liên kết được tới thư mục | Có thể liên kết được tới thư mục |
| Không tham chiếu được tới file trên ổ đĩa khác | Có thể tham chiếu tới file/thư mục khác ổ đĩa |
| Liên kết tới một file vẫn còn ngay cả khi file đó đã được di chuyển | Liên kết không còn tham chiếu được nữa nếu file được di chuyển |
| Được liên kết với inode tham chiếu vật lý trên ổ cứng nơi chứa file | Liên kết tham chiếu tên file/thư mục trừu tượng mà không phải địa chỉ vật lý. Chúng được cung cấp inode riêng của mình |
| Có thể làm việc với mọi ứng dụng | Một số ứng dụng không cho phép symbolic link |