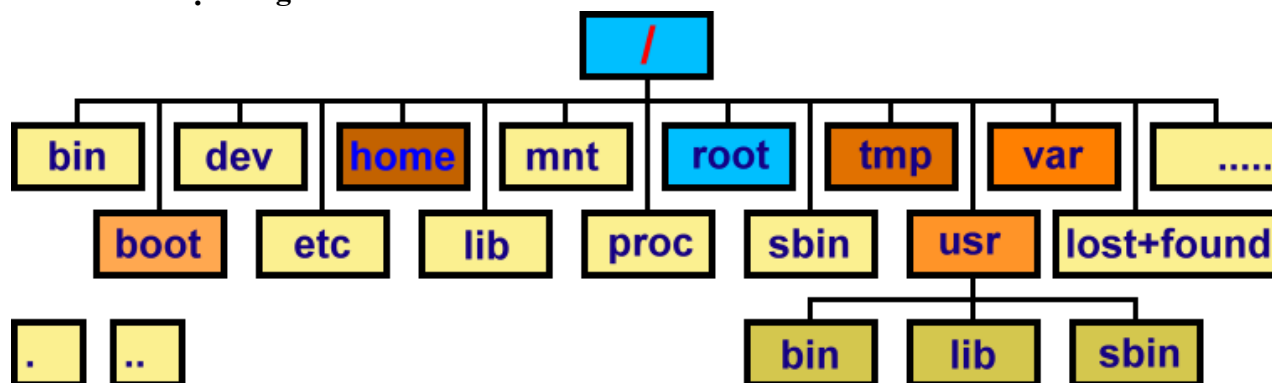




Quản lý hệ thống File **Linux**

1. Cấu trúc hệ thống file



/ : thư mục gốc gốc của cấu trúc cây thư mục Linux, trong đó các file là các nút lá

/bin : thư mục chứa các lệnh thực thi cơ bản ((binary applications))

/sbin: thư mục chứa các lệnh quản trị hệ thống (system binaries)

/boot : thư mục chứa kernel và boot loader

/etc : thư mục chứa các tập tin cấu hình hệ thống

/sys : thư mục lưu các tập tin của hệ thống (system files)

/dev : thư mục các file thiết bị (device files), các thiết bị phần cứng được xem là các file

/root : thư mục home của người quản trị hệ thống root

/home : mỗi thư mục con trong thư mục này là Home Directory của các users

/lib : thư mục thư viện chia sẻ của hệ thống (system libraries) ~ như *C:\Windows\System32*

/usr : thư mục chứa những tập tin của các ứng dụng chính đã được cài đặt cho mọi người dùng (all users), thư mục */usr/local* ~ như *C:\Program Files* chứa các chương trình của người dùng được cài từ mã nguồn

/var : thư mục lưu tập tin ghi dữ liệu biến đổi (variable files) như các tập ghi nhật ký (log file)

/tmp : thư mục chứa các file tạm thời trong lúc chạy chương trình (temporary files). Một điểm lưu ý là các users đều có quyền write trên /tmp. Một user hoàn toàn có khả năng tạo ra hàng loạt các file có kích thước lớn trong /tmp để làm đầy partition /

/mnt : thư mục này được dùng để gắn các hệ thống tập tin tạm thời (mounted filesystems)

/media : thư mục này được dùng để tạo ra các tập tin gắn (loaded) tạm thời được hệ thống tạo ra khi một thiết bị lưu động (removable media) được cắm vào như đĩa CDs, máy ảnh kỹ thuật số...

/opt : thư mục chứa các phần mềm ứng dụng (optional applications) được cài đặt thêm.

/lost+found : thư mục lưu các tập tin không có thư mục cha mà được tìm thấy dưới thư mục gốc / sau khi thực hiện lệnh kiểm tra hệ thống tập tin (fsck)

/proc : thư mục đặc biệt để lưu các thông số của kernel.



Hệ thống thư mục /proc

Thư mục **/proc** là một thư mục quan trọng, đóng vai trò sống còn đối với hệ thống Linux. Thông tin chứa trong **/proc** được tạo ra một cách tự động dựa trên các quá trình startup và shutdown của hệ thống hay thay đổi theo thời gian thực của file system. Thông tin của hệ thống, các tiến trình, các tham số của hệ thống đều thể hiện trong thư mục này

Ví dụ: Mở trình soạn thảo **vi** sau đó bấm Ctrl-Z để **vi** chạy trong background

```
# vi /tmp/test
```

```
[1]+ Stopped          vi /tmp/test
```

```
# ps -ef | grep vi
```

```
root  23900 23872 0 15:52 pts/0  00:00:00 vi /tmp/test
```

Kiểm tra thư mục **/proc** thấy thư mục có PID **23900**

```
# ls -l /proc | grep 23900
```

```
dr-xr-xr-x 6 root    root      0 Apr 15 15:52 23900
```

cd vào thư mục **23900** sẽ thấy các trạng thái của **vi** đang chạy, đóng trình soạn thảo **vi**, thư mục **23900** sẽ được xóa bỏ

Trên Linux giá trị maximum của Process ID được thiết lập trong file **/proc/sys/kernel/pid_max**

```
# cat /proc/sys/kernel/pid_max
```

```
32768
```

Hiển thị cấu trúc cây thư mục

```
# tree -L 1 /
```

Xem thông tin dung lượng thư mục đã dùng (du : disk usage)

```
# du -sh /boot
```

```
# du -h --max-depth=1 / | sort -n -r
```

File Linux vs Windows

Giống nhau

Độ dài tối đa cho tên file là 255

Chấp nhận tất cả các ký tự để đặt tên file (nhưng nên tránh sử dụng các ký tự đặc biệt như * ? [] & để tránh sự nhập nhằng trong câu lệnh sử dụng sau này)

Tính đặc thù của Linux

- Quản lý tất cả dưới dạng file cho cả thư mục và các loại tài nguyên hệ thống (ổ cứng, thiết bị ngoại vi...)

- Không có khái niệm phần mở rộng của tên file (ký tự '.' trong tên file được đối xử như mọi ký tự khác)

- Không dùng ổ đĩa logic trong cây thư mục

- '/' được dùng thay cho '\' trong đường dẫn thư mục

Đường dẫn và thư mục đặc biệt

Truy cập file và thư mục cần dùng các đường dẫn

Đường dẫn có thể có gốc từ các thư mục đặc biệt

/ : thư mục gốc

~ : thư mục home của user đang login

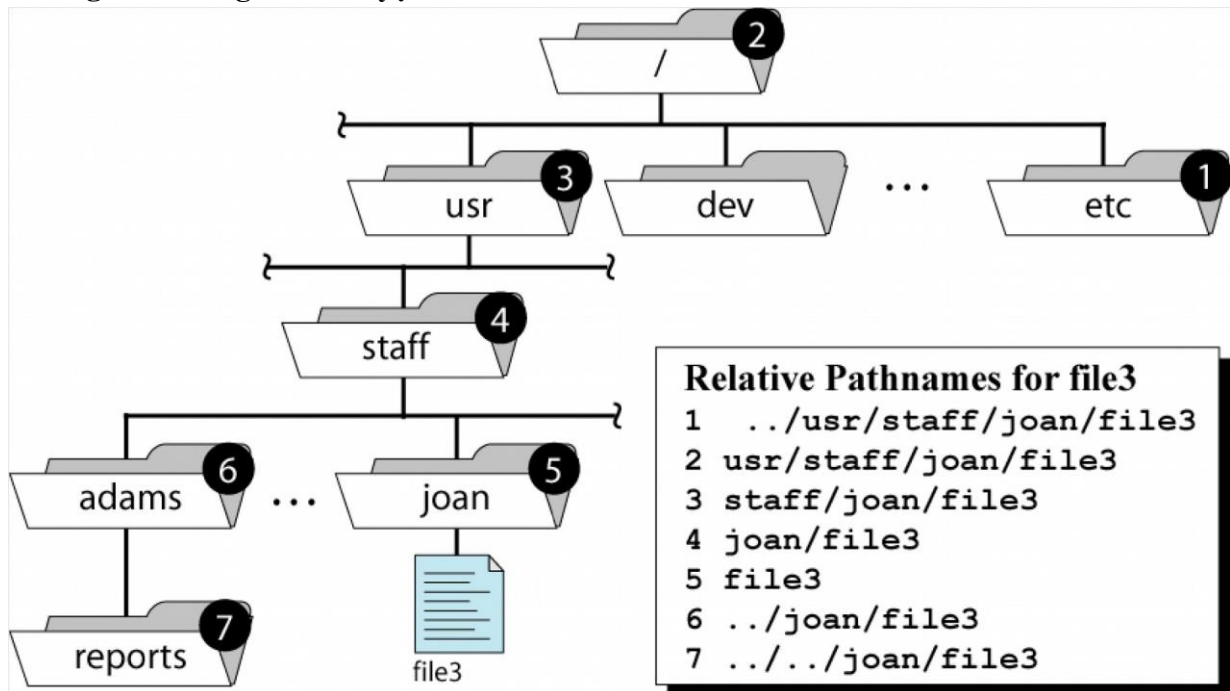
~user : thư mục home của người dùng có tên user

. : thư mục hiện hành

.. : thư mục cha thư mục hiện hành



Đường dẫn tương đối và tuyệt đối



Đường dẫn để định vị một file hoặc một thư mục trong hệ thống file

Đường dẫn tuyệt đối là đường dẫn bắt đầu từ thư mục gốc `/`

Đường dẫn tương đối dùng để chỉ cách định vị file hay thư mục từ *thư mục hiện hành*

Câu lệnh ls

`# ls [-options] [pathname]`

- ♣ Tùy chọn `-l` cho phép hiển thị thuộc tính cho mỗi tệp (kiểu, quyền, liên kết, chủ sở hữu, nhóm sở hữu, kích thước, ngày sửa đổi)
- ♣ Tùy chọn `-a` cho phép hiển thị cả các tệp ẩn
- ♣ Tùy chọn `-i` cho phép hiển thị chỉ số inode
- ♣ Tùy chọn `-h` in ra kích thước dễ đọc
- ♣ Tùy chọn `-t` liệt kê theo thứ tự thời gian file (giảm dần)

```
$ ls -al
total 94
drwxr-xr-x  2 john doc   512 Jul 10 22:25 .
drwxr-xr-x  4 bin  bin  1024 Jul  8 11:48 ..
-rw-r--r--  1 john doc   136 Jul  8 14:46 .exerc
-rw-r--r--  1 john doc   833 Jul  8 14:51 .profile
-rw-rw-rw-  1 john doc 31273 Jul 10 22:25 ch1
-rw-rw-rw-  1 john doc    0 Jul 10 21:57 ch2
```

Type	# of Links	Owner	Group	Size (in bytes)	Modification Date and Time	Name
drwxr-xr-x	2	john	doc	512	Jul 10 22:25	.
drwxr-xr-x	4	bin	bin	1024	Jul 8 11:48	..
-rw-r--r--	1	john	doc	136	Jul 8 14:46	.exerc
-rw-r--r--	1	john	doc	833	Jul 8 14:51	.profile
-rw-rw-rw-	1	john	doc	31273	Jul 10 22:25	ch1
-rw-rw-rw-	1	john	doc	0	Jul 10 21:57	ch2

7 trường thuộc tính cho mỗi file:



- Tập quyền truy xuất (permission)
- Số liên kết đến file (link count): File: số hard link; Thư mục: số thư mục liên kết đến (một thư mục luôn có ít nhất 2 liên kết)
- User sở hữu file - owner
- Nhóm user sở hữu file - group
- Kích thước file (tính theo byte)
- Ngày giờ tạo file
- Tên file

Các ký hiệu kiểu file

- ♣ -: file thông thường
- ♣ d : file thư mục
- ♣ b : file đặc biệt (block)
- ♣ c : file đặc biệt (ký tự)
- ♣ l : file liên kết

Dùng lệnh `file filename` để biết `filename` là loại file gì

Thuộc tính size và size on disk

```
# echo hello > hello.txt
```

```
# ls -l hello.txt
```

```
-rw-r--r-- 1 root root 6 Apr  9 16:09 hello.txt
```

```
# du -h hello.txt
```

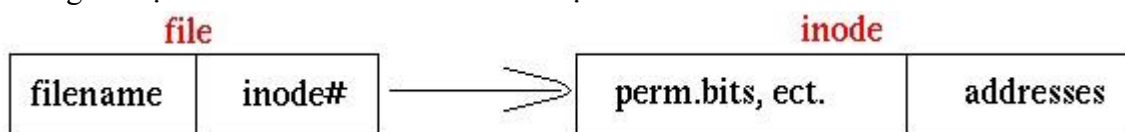
```
4.0K hello.txt
```

2. File và inode

File là khối dữ liệu được lưu trữ liên tục (hoặc không liên tục → dữ liệu bị phân mảnh) trên thiết bị lưu trữ. Về phía người sử dụng sẽ nhận dạng một file dựa theo tên của file.

Inode là một cấu trúc dữ liệu trong các hệ thống File truyền thống như UFS/EXT3/EXT4 của các Hệ điều hành Unix/Linux. **Inode lưu trữ tất cả các thông tin thuộc tính file trừ tên file và nội dung file**

Cấu trúc một **inode** chứa: thông tin thuộc tính của file (*chủ sở hữu, quyền truyền cập...*) và địa chỉ vùng dữ liệu của file nằm ở đâu trên thiết bị lưu trữ.



Một file luôn có 2 phần **filename** và **chỉ số inode**

Người sử dụng tên thân thiện **filename**, còn hệ điều hành chỉ làm việc với chỉ số **inode**.

Mỗi **inode** xác định bởi 1 con số, lệnh `ls -i` để biết chỉ số inode của file, `ls -l` để biết thông tin file mà inode chứa

Thư mục là loại file đặc biệt chứa danh sách các liên kết, mỗi liên kết chứa tên đối tượng (file, folder, link...) và inode tương ứng với đối tượng đó.

```
# df -i
```

```
# for i in $(seq 1 5000000); do echo testing >> $i.txt; done
```

```
# rm -f *
```



No space left on device

/bin/rm: Argument list too long

Which is the fastest method to delete files in Linux

RM Command Is not capable of deleting large number of files

Find Command with -exec 14 Minutes for half a million files

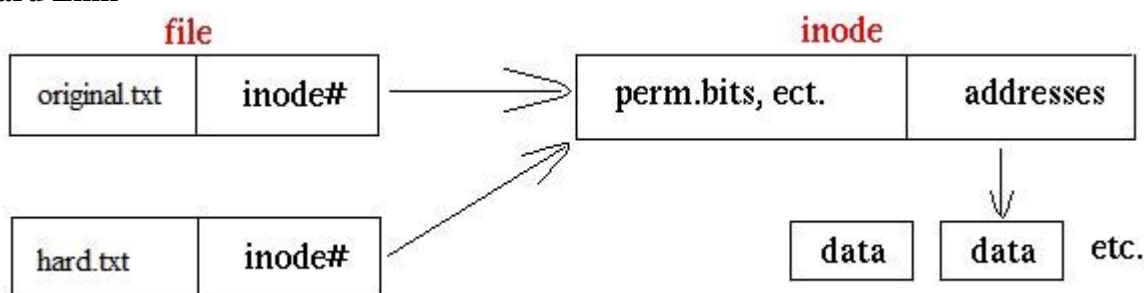
Find Command with -delete 5 Minutes for half a million files

Perl 1 Minute for half a million files

RSYNC with -delete 2 Minute 56 seconds for half a million files

3. Hard link và Soft link

Hard Link



Hard link là tên gọi khác cho một file đang tồn tại trên file system. Một file có thể có nhiều Hard link, mỗi Hard link cũng có thể có nhiều Hard link khác. **Không thể tạo Hard link cho một thư mục hay cho một file không nằm cùng phân vùng với Hard link** (Mỗi phân vùng có một bảng inode riêng)

Tạo file gốc

```
# echo 'This is a original file' > original.txt
```

Tạo hard link **ln {target-filename} {hardlink-filename}**

```
# ln original.txt hard.txt
```

Liệt kê số hard link đến một inode

```
# ls -li original.txt hard.txt
```

```
2195850 -rw-r--r-- 2 root root 18 Jan 8 17:43 original.txt
```

```
2195850 -rw-r--r-- 2 root root 18 Jan 8 17:43 hard.txt
```

Tìm kiếm số hard link của một file

```
# find / -xdev -samefile original.txt
```

```
# find / -xdev -inum 2195850
```

Tìm xóa file và tất cả hard link của nó

```
# find / -xdev -samefile original.txt | xargs rm
```

Khi sử dụng lệnh **rm** để xóa một file → Hệ thống giảm đi một hard link của file đó, khi số lượng hard link giảm về 0 thì không thể truy cập tới nội dung của file (nội dung file vẫn tồn tại trên thiết bị lưu trữ). Dữ liệu của file chỉ thực sự mất khi vị trí của nó bị các file mới ghi đè, → có thể khôi phục file dữ liệu vừa bị xóa nếu dữ liệu đó chưa bị ghi đè.

```
# ls -li /sbin/mkfs*
```

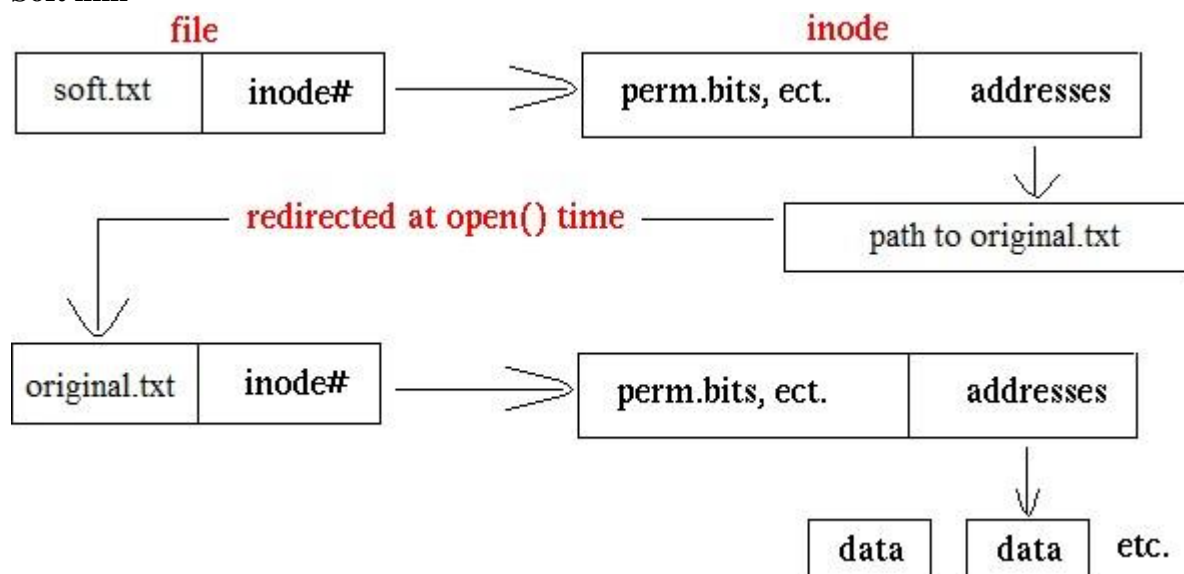
```
524471 -rwxr-xr-x 1 root root 7240 Nov 8 2012 /sbin/mkfs
```

```
524329 -rwxr-xr-x 1 root root 18248 Nov 8 2012 /sbin/mkfs.cramfs
```



```
524555 -rwxr-xr-x 3 root root 47300 May 9 2013 /sbin/mkfs.ext2
524555 -rwxr-xr-x 3 root root 47300 May 9 2013 /sbin/mkfs.ext3
524492 -rwxr-xr-x 3 root root 29880 Jan 27 2010 /sbin/mkfs.msdos
524492 -rwxr-xr-x 3 root root 29880 Jan 27 2010 /sbin/mkfs.vfat
# find / -xdev -samefile /sbin/mkfs.vfat
/sbin/mkfs.vfat
/sbin/mkdosfs
/sbin/mkfs.msdos
```

Soft link



Tạo soft link **ln -s {target-filename} {symbolic-filename}**

```
# ln -s original.txt soft.txt
```

```
# ls -l
```

```
-rw-r--r-- 2 root root 24 May 28 17:10 hard.txt
```

```
-rw-r--r-- 2 root root 24 May 28 17:10 original.txt
```

```
lrwxrwxrwx 1 root root 12 May 28 17:10 soft.txt -> original.txt
```

```
# rm -f original.txt
```

```
# cat soft.txt
```

```
cat: soft.txt: No such file or directory
```

```
# cat hard.txt
```

```
This is a original file
```

≡

Để sửa nội dung từ file link *soft.txt*, phải có quyền sửa trên cả *original.txt* và *soft.txt*

Tạo file liên kết trên windows, chạy Command Prompt với Run as administrator

- Tạo hard link: *mklink /h hard.txt original.txt*
- Tạo soft link: *mklink soft.txt original.txt*



4. Biến môi trường

Xem các biến môi trường

```
# env
```

Biến môi trường **PATH**

```
# echo $PATH
```

```
/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/root/bin
```

Khai báo biến môi trường với lệnh **export**

```
# export name=giangnt
```

```
# env | grep name
```

```
# vim whatday.sh
```

```
#!/bin/bash
```

```
echo "Today is $today"
```

```
today=Friday
```

```
echo "Today is $today"
```

```
# today=Thursday
```

```
# sh whatday.sh
```

```
Today is
```

```
Today is Friday
```

```
# echo $today
```

```
Thursday
```

Giá trị Thursday của biến **today** trong login Shell không có tác dụng trong script **whatday.sh** .

Khi script thi hành thì biến **today** vẫn chưa được định nghĩa cho đến khi nó nhận giá trị là

Friday. Khi script kết thúc, trở về login Shell, biến **today** vẫn giữ giá trị là Thursday .

Nếu thêm lệnh xuất biến môi trường **export**

```
# export today=Thursday
```

```
# sh whatday.sh
```

```
Today is Thursday
```

```
Today is Friday
```

```
# echo $today
```

```
Thursday
```

≡ Khai báo các biến môi trường toàn cục trong file **/etc/environment**



5. Lệnh cơ bản quản lý File

pwd: hiển thị đường dẫn thư mục hiện hành

cd: thay đổi vị trí thư mục hiện hành

mkdir: tạo thư mục mới

cp: sao chép một hoặc nhiều tệp vào một thư mục

mv: di chuyển một hoặc nhiều tệp đến một thư mục

rm: xóa file

file: xem thông tin loại file

cat: đọc nội dung một hay nhiều file (-n hiển thị số dòng)

more: xem từng dòng, từng trang

less: tương tự lệnh **more**, nhưng cho phép tìm kiếm và lùi dòng bằng phím mũi tên

head -n tên_file: hiển thị n dòng đầu tiên của 1 file

tail -n tên_file: hiển thị n dòng cuối cùng của 1 file

wc tên_file: hiển thị số dòng, số từ, số ký tự trong 1 file thông qua các tùy chọn -l, -w, -c

grep: tìm kiếm trong file hoặc trong đầu vào chuẩn các dòng có chứa các ký tự hoặc chuỗi thỏa mãn một số điều kiện nào đó.

lsof: list open files

seq 100 > number.txt

head -10 number.txt | tail -5