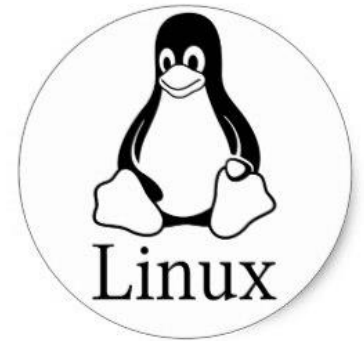


# LINUX

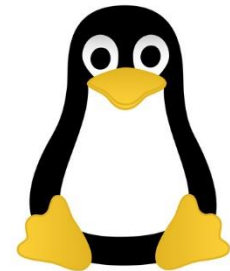


## HỆ THỐNG TẬP LINUX



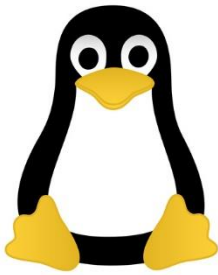
Freedom. Choices. Beautiful.

# Cấu trúc hệ thống tệp



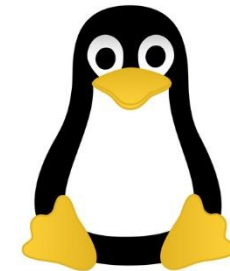
- Biểu diễn bởi một cây phân cấp thư mục và các tệp dữ liệu
  - Một thư mục dùng để tạo nhóm một tập các tệp dữ liệu
  - Một thư mục có thể chứa các thư mục con
- Thư mục gốc (/) là điểm vào đầu tiên cho cả cây thư mục

# Các thư mục thông dụng

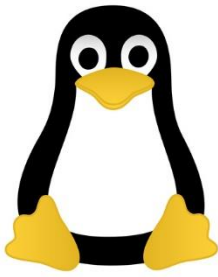


- / (thư mục gốc)
  - /bin : thư mục tập chương trình cơ bản
  - /boot : thư mục chứa hạt nhân của HĐH
  - /etc : thư mục các tệp cấu hình
  - /dev : thư mục các tệp thiết bị
  - /home : thư mục chứa dữ liệu NSD
  - /lib : thư viện hệ thống
  - /usr : thư mục ứng dụng
  - /var : thư mục dữ liệu hệ thống, thường được cập nhật

# Tệp UNIX vs. tệp Windows

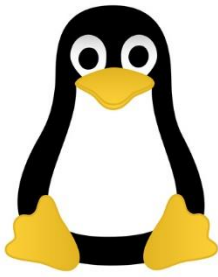


- Giống nhau
  - độ dài tối đa cho tên tệp là 255
  - Chấp nhận tất cả các kí tự để đặt tên tệp (nhưng nên tránh sử dụng các kí tự đặc biệt như \* ? [ ] & để tránh sự nhập nhằng trong câu lệnh sử dụng sau này)
- Tính đặc thù của Unix
  - Quản lý dưới một khung nhìn của tệp cho cả thư mục và các loại tài nguyên hệ thống (ngoại vi, bảng phân chương đĩa)
  - Không có khái niệm phần mở rộng của tên tệp (kí tự '.' trong tên tệp được đối xử như mọi kí tự khác)
  - Không dùng ổ đĩa logic trong cây thư mục
  - '/' được dùng thay cho '\' trong đường dẫn thư mục



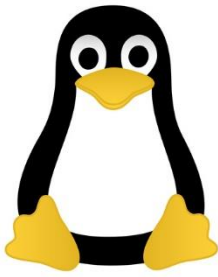
# Kiểu của tệp

- Có 3 loại tệp chính trong một cây thư mục của Unix:
  - Tệp thư mục là một thư mục trong đường dẫn phân loại (vd., /usr, /home,...)
  - Tệp thông thường là một tệp chứa dữ liệu hoặc tệp chương trình (vd., /bin/passwd, /etc/passwd, ...)
  - Tệp đặc biệt là một tệp thiết bị tương ứng với thiết bị ngoại vi hoặc các tệp tự sinh bởi HĐH (vd., /dev/hda1, /dev/cdrom, /dev/lp0, ...)



# Quản lý tệp

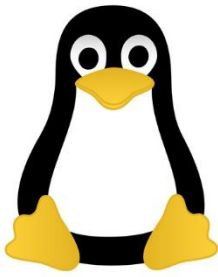
- pwd: hiển thị đường dẫn tuyệt đối của thư mục hiện tại
- cd: thay đổi vị trí thư mục hiện tại
  - \$ cd /home/tuananh ↵
  - \$ cd tuananh ↵
- ls: liệt kê các tệp trong một thư mục
  - \$ ls ↵
  - \$ ls /home/tuananh
  - \$ ls -la tuananh
    - tùy chọn -a cho phép hiển thị cả các tệp ẩn
    - tùy chọn -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)
- mkdir: tạo một thư mục rỗng
- rmdir: xóa một thư mục rỗng



# Tên đặc biệt

- « . » : thư mục hiện tại
- « .. » : thư mục cha
- « ~ » : thư mục cá nhân
- « .xxx » : tệp ẩn (e.g.,  
/home/tuananh/.bashrc)

# Ví dụ



```
$ cd ~
```

```
$ pwd
```

```
/home/tuananh
```

```
$ ls -la
```

```
-rw-r--r-- 1 tuananh  user1 2451 Feb  7 07:30 .bashrc
-rw-r--r-- 1 tuananh  user1 4025 Feb 10 19:12 linux.ppt
drwxr-xr-- 2 tuananh  user1  512 Feb 10 19:12 linux
```

```
$ mkdir vanban
```

```
$ cd vanban
```

```
$ pwd
```

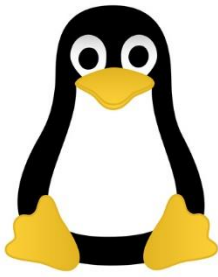
```
/home/tuananh/vanban
```

```
$ cd ..
```

```
$ pwd
```

```
$ rmdir vanban
```

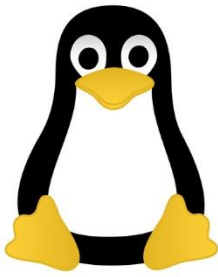




# Các siêu kí tự

- \* dùng để thay thế cho một chuỗi kí tự bất kì bao gồm cả xâu rỗng
- ? thay thế cho một kí tự bất kì
- [ ] được thay thế bởi một kí tự trong một tập kí tự cho trước
- [! ] được thay thế bởi một kí tự không có trong một tập kí tự cho trước

# Ví dụ



```
$ ls -l *. [c,h]
```

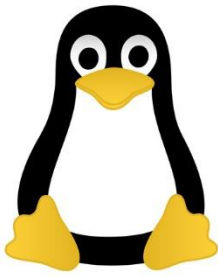
```
-rw-r--r-- 1 tuananh  user1 2451 Feb  7 07:30 myprog.c  
-rw-r--r-- 1 tuananh  user1 2451 Feb  7 07:30 myprog.h
```

```
$ ls -l *prog
```

```
drwxr-xr-- 2 tuananh  user1  512 Feb 10 19:12 c_prog  
drwxr-xr-- 2 tuananh  user1  512 Feb 10 19:12 java_prog
```

```
$ ls -l .*
```

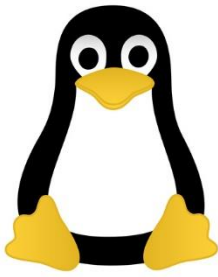
```
-rw-r--r-- 1 tuananh user1 451 Feb 7 07:30 .bashrc  
-rw-r--r-- 1 tuananh user1 225 Feb 7 07:30 .bash_profile  
-rw-r--r-- 1 tuananh user1 351 Feb 7 07:30 .bash_logout
```



# Quản lý tệp

- `$cp file1 [...] dir`
  - sao chép một hoặc nhiều tệp vào một thư mục
- `$mv file1 [...] dir`
  - di chuyển một hoặc nhiều tệp đến một thư mục
- `$rm file1 [...]`
  - xoá một hoặc nhiều tệp
- tùy chọn `-R` (recursive)
  - cho phép sao chép/di chuyển/xoá toàn bộ thư mục bao gồm cả các thư mục con

# Ví dụ



```
$ ls -l
```

```
-rw-r--r-- 1 tuanh user1 16 Feb 10 19:12 test.txt  
drwxr-xr-- 2 tuanh user1 512 Feb 10 19:14 vanban
```

```
$ cp test.txt vanban
```

```
$ ls -l vanban
```

```
-rw-r--r-- 1 tuanh user1 16 Feb 12 20:03 test.txt
```

```
$ rm -R vanban
```

```
$ ls -l
```

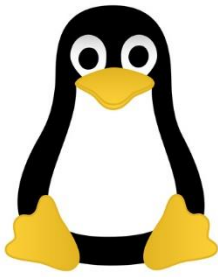
```
-rw-r--r-- 1 tuanh user1 16 Feb 10 19:12 test.txt
```

```
$ rm test.txt
```

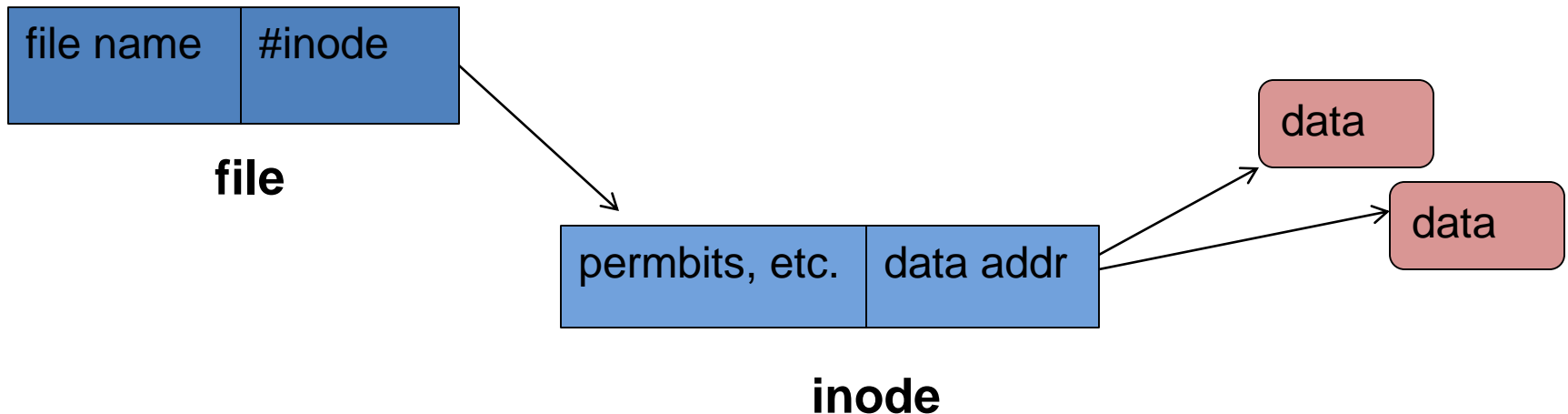
```
$ ls -l
```

```
$
```

# Inode, soft link and hard link

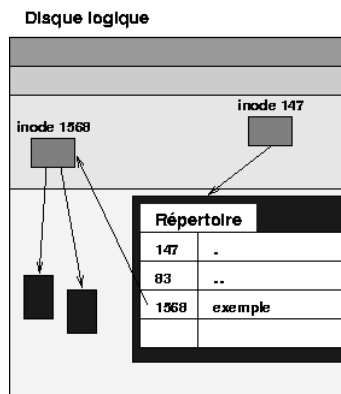
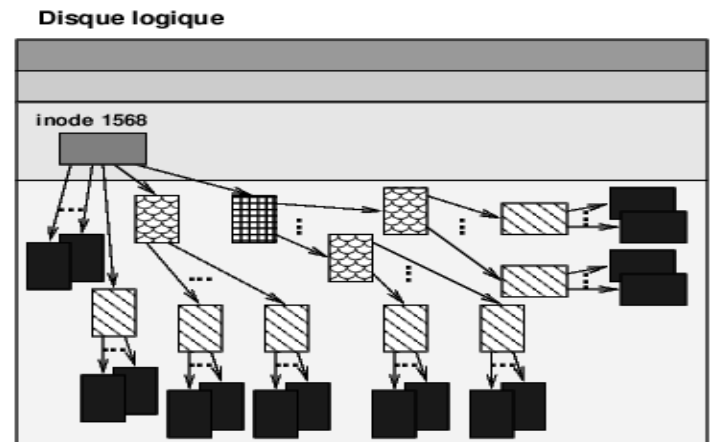


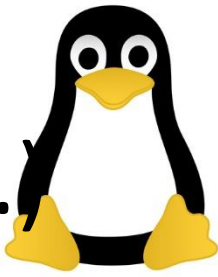
- Một file trong Unix:



# Thư mục

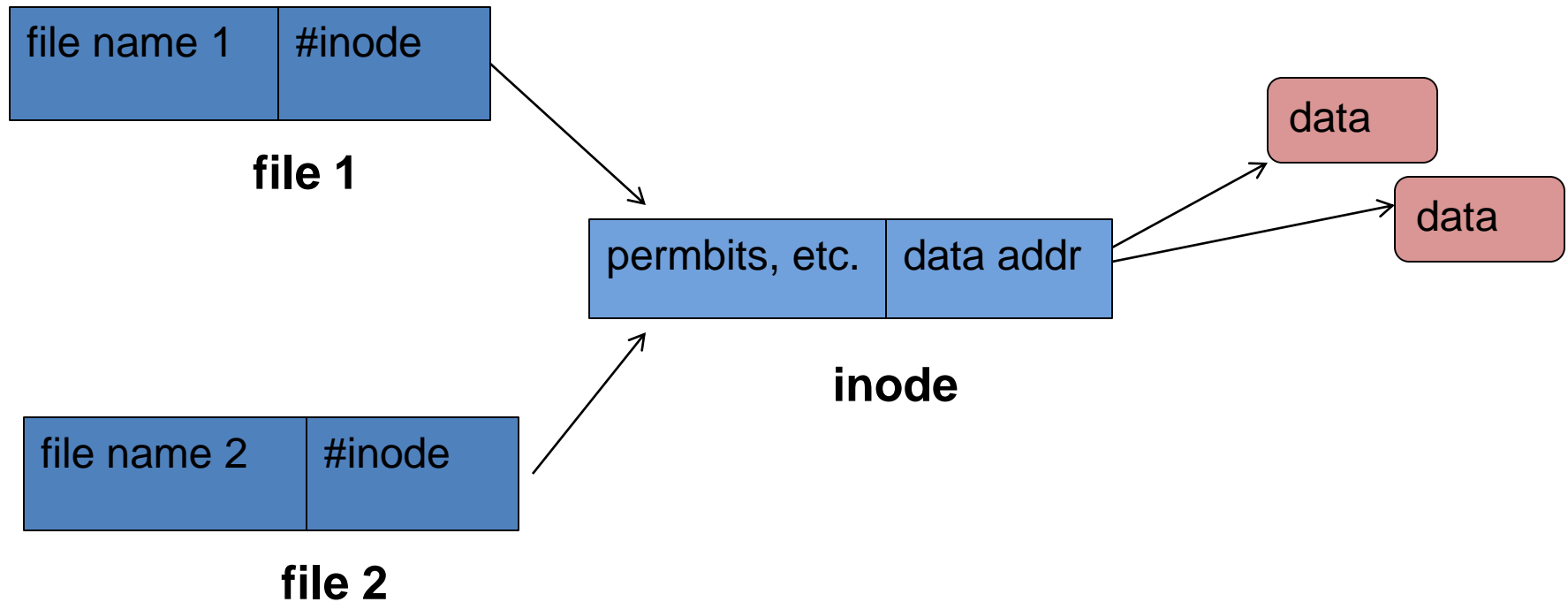
- Một thư mục là một tập với nội dung là một bảng liên kết
  - một liên kết gắn một tên tệp với một inode của hệ thống tệp





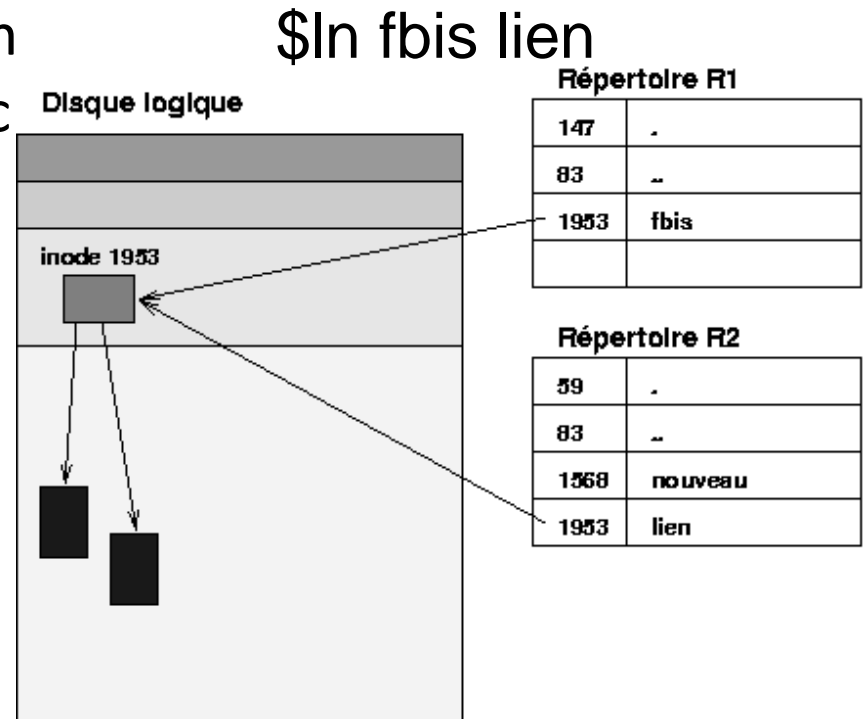
# Inode, soft link and hard link (cont.)

- Liên kết vật lý (hard link)

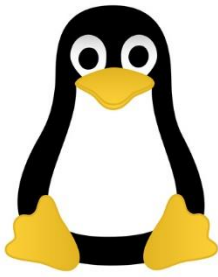


# Liên kết vật lý (2)

- Một liên kết vật lý là một quan hệ giữa tên tệp trong thư mục với một inode
- Có thể có nhiều liên kết vật lý đến cùng một inode
- Lệnh **ln** cho phép tạo một liên kết vật lý đến một inode (tệp) đã tồn tại
  - tệp mới chia sẻ cùng inode và khối dữ liệu của tệp ban đầu





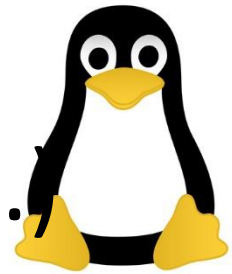


# Liên kết vật lý (3)

- Số liên kết vật lý đến một inode có thể được xem bằng lệnh `ls -l`

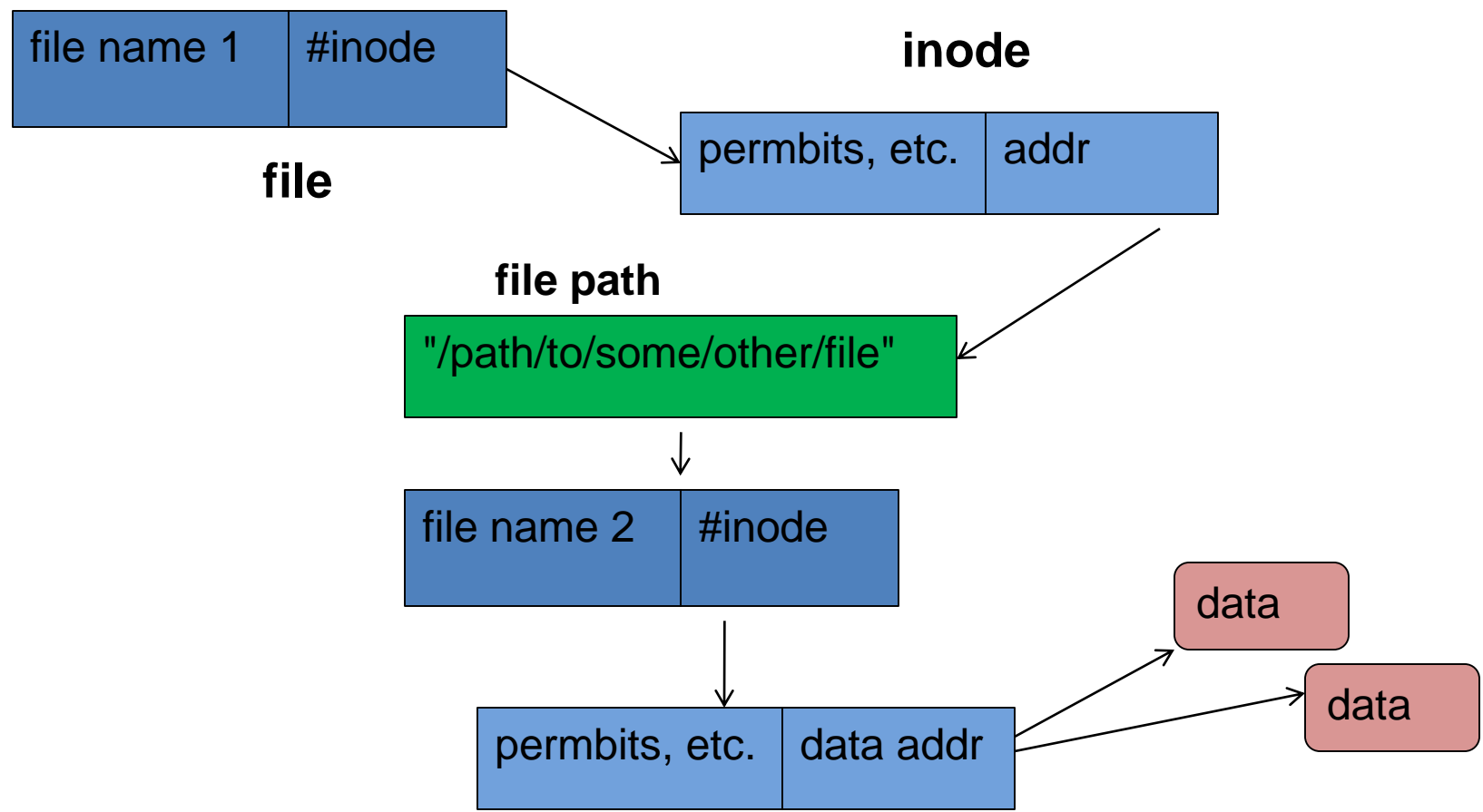
```
$ ls -l
-rw-rw-r-- 1 tuanh user1 0 Nov 12 15:19 file
drwxr-xr-x 2 tuanh user1 4096 Dec 14 17:50 dir
```

- Tại sao với một thư mục luôn có ít nhất 2 liên kết vật lý?
- Xóa một tệp (lệnh `rm`) đồng nghĩa với xóa một liên kết
  - Nếu là liên kết vật lý cuối cùng trỏ đến inode được xóa thì các khối liên quan đến inode cũng được xóa theo

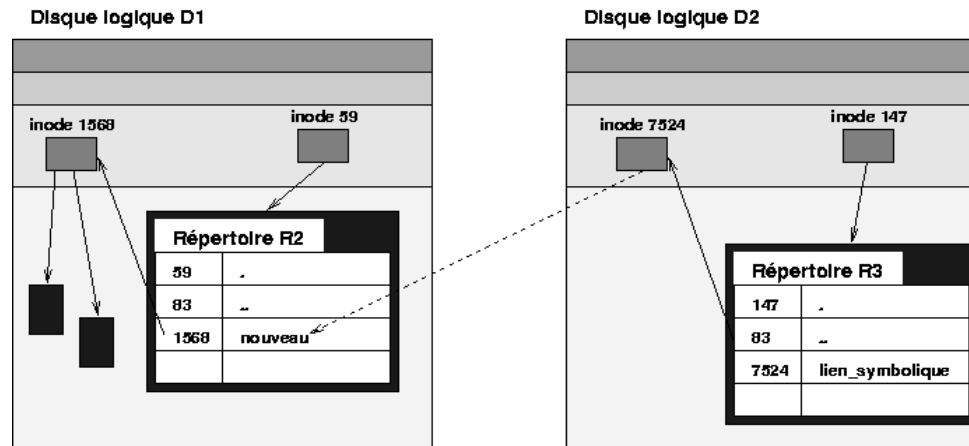


# Inode, soft link and hard link (cont.)

- Liên kết biểu tượng (soft link)

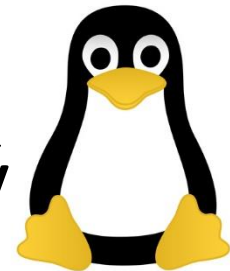


# Liên kết biểu tượng



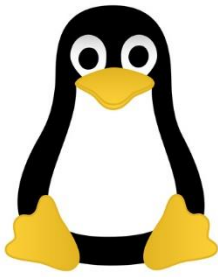
- In -s R2/nouveau R3/lien\_symbolique
  - khi tạo liên kết biểu tượng (tùy chọn -s) một inode mới được tạo ra
  - inode này chứa tên (dạng tuyệt đối hay tương đối) của phần tử được trỏ tới

# Liên kết biểu tượng vs. Liên kết vật lý

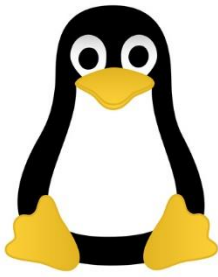


- Liên kết biểu tượng cho phép tránh được các hạn chế về mặt dung lượng của thiết bị lưu trữ
  - Một liên kết vật lý luôn luôn chiếm một số lượng inode nhất định
- Chúng ta có thể phân biệt rõ một file được tạo ra và file gốc trong một liên kết biểu tượng
  - Điều gì sẽ xảy ra khi một người xóa đi file gốc ?

# Ví dụ



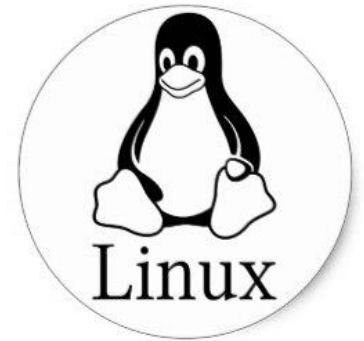
```
$ ls -l
-rw-r--r-- 1 tuanh user1 8 Feb 10 1:12 test.txt
$ ln test.txt link1
$ ln -s test.txt link2
$ ls -l link*
-rw-r--r-- 2 tuanh user1 16 Feb 10 1:12 link1
lrw-r--r-- 1 tuanh user1 16 Feb 10 1:13 link2->test.txt
```



# Các kiểu tệp

- Các ký hiệu dưới đây được sử dụng để biểu diễn các kiểu tệp
  - - : tệp thông thường
  - d : thư mục
  - b : tệp đặc biệt (block)
  - c : tệp đặc biệt (karakter)
  - l : link
  - m : phần bộ nhớ trong dùng chung
  - p : đường ống

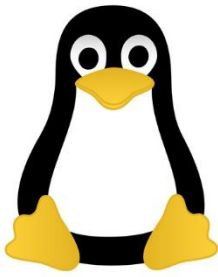
# LINUX



Phân quyền truy cập tệp

Freedom. Choices. Beautiful.

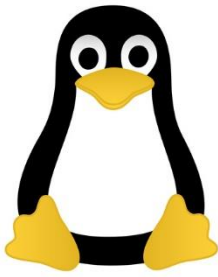
# Khái niệm người sử dụng



- Tạo một người sử dụng
  - Tên
  - Mật khẩu
  - home của người sử dụng (/home/tên)
  - Nhóm (một người sử dụng có thể thuộc một hoặc nhiều nhóm, tuy nhiên cần phải xác định một nhóm chính)
  - Tất cả các thông tin về người sử dụng được lưu trong file: /etc/passwd

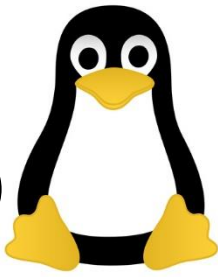


# Nhóm người sử dụng



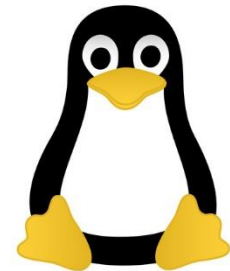
- Mỗi người sử dụng có thể thuộc về một hoặc nhiều nhóm
  - Một nhóm = tên nhóm + danh sách các thành viên
  - Khả năng chia sẻ các file giữa những người sử dụng trong cùng một nhóm.
  - Danh sách các nhóm được lưu trữ trong file: `/etc/group`
  - root có khả năng tạo ra các nhóm bổ xung, ngoài các nhóm mà hệ điều hành đã ngầm định

# Các lệnh quản lý user và group



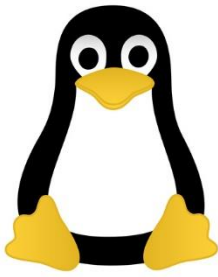
- Thêm user: `sudo adduser username`
- Xóa user: `sudo deluser username`
- Khóa hoặc hủy khóa user:  
`sudo passwd -l username`  
`sudo passwd -u username`
- Thêm/Xóa nhóm:  
`sudo addgroup groupname`  
`sudo delgroup groupname`
- Thêm một người vào một nhóm:  
`sudo adduser username groupname`
- Ví dụ thêm một người vào group sudo:  
`usermod -aG sudo username`

# Các quyền



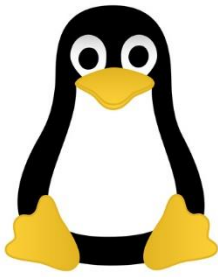
- Mỗi file luôn thuộc về một người sử dụng và một nhóm xác định
  - Người tạo ra file hoặc thư mục sẽ là người sở hữu, nhóm chứa người tạo ra file hoặc thư mục sẽ là nhóm sở hữu đối với file/thư mục.
- Sự phân quyền cho phép xác định rõ các quyền mà người sử dụng có đối với một file hoặc một thư mục.

# Quyền truy cập



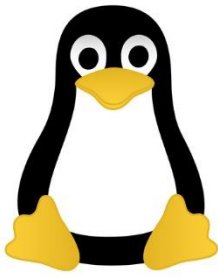
- **r** : đọc
  - Cho phép hiển thị nội dung của file hoặc thư mục
- **w** : ghi
  - Cho phép thay đổi nội dung của file
  - Cho phép thêm hoặc xóa các file trong một thư mục
- **x** : thực thi
  - Cho phép thực thi file dưới dạng một chương trình
  - Cho phép chuyển đến thư mục cần truy cập

# Các nhóm người sử dụng



- Có 3 nhóm người sử dụng đối với 1 file/ thư mục:
  - **u** (người sở hữu) : người sở hữu duy nhất của file
  - **g** (groupe) : những người sử dụng thuộc nhóm chứa file
  - **o** (others) : những người sử dụng khác, không phải là người sở hữu file cũng như không thuộc nhóm chứa file.
- Mỗi nhóm người sử dụng sẽ có một tập các quyền (r, w, x) xác định.

# Ví dụ



```
$ ls -l
```

```
----rw-rw- 1 tuanh user1 16 Feb 10 19:12 test1.txt  
-rw-rw-rw- 1 tuanh user1 16 Feb 10 19:12 test2.txt  
drw-r--r-- 2 tuanh user1 512 Feb 10 19:14 vanban
```

```
$ whoami
```

```
tuanh
```

```
$ cat test1.txt
```

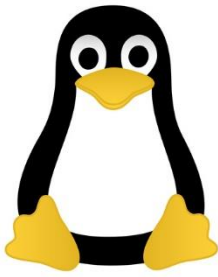
```
cat: test1.txt: Permission denied
```

```
$ cat test2.txt
```

```
Un fichier de test
```

```
$ cp test2.txt vanban
```

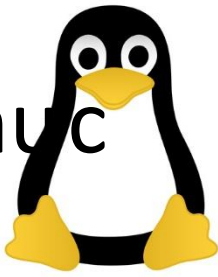
```
cp: vanban: Permission denied
```



# Các lưu ý

- Để có thể thêm các file, cần phải có quyền « w » đối với thư mục
- Để có thể xóa, thay đổi nội dung hoặc di chuyển 1 file, người sử dụng cũng cần phải có quyền « w » đối với thư mục
- Việc xóa một file còn phụ thuộc vào quyền đối với thư mục chứa file đó
- Để bảo mật các dữ liệu, người sở hữu file thậm chí có thể bỏ cả quyền đọc « r » đối với tất cả mọi người sử dụng khác.
- Để hạn chế quá trình truy cập vào hệ thống file, người sử dụng có thể bỏ quyền thực thi (x) đối với thư mục gốc của hệ thống file.

# Một số quyền đặc biệt đối với các file thực thi



- set-uid: -rw**S** --- ---
  - Chương trình được chạy dưới quyền của người sở hữu
- set-gid: - --- rw**S** ---
  - Chương trình được chạy bởi các người sử dụng thuộc cùng nhóm với người sở hữu
- bit sticky
  - Chương trình chỉ được cấp phát bộ nhớ trong 1 lần



# Ví dụ



```
$ ls -l /etc/passwd
```

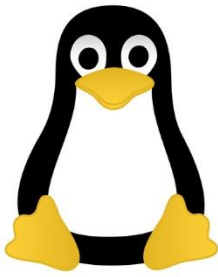
```
-rw-rw---- 1 root  root  568 Feb 10 19:12 passwd
```

```
$ ls -l /bin/passwd
```

```
-rwsrws--x 1 root  root 3634 Feb 10 19:12 passwd
```

- Khi một người sử dụng thông thường gọi lệnh `/bin/passwd`, xem như người đó được « mượn » quyền root để thay đổi mật khẩu trong file `/etc/passwd`

# Thay đổi quyền truy cập (1)



**\$chmod <mode> <files>**

set_uid	set-gid	sticky	user	group	other
			rwX	--X	--X
1	1	0	111	001	001
	6		7	1	1

```
$ chmod 6711 test
```

```
$ ls -l test
```

```
-rws--s--x 1 tuanh user1 Mar 10 10:20 test
```

```
$ chmod 711 test
```

```
$ ls -l test
```

```
-rwx--x--x 1 tuanh user1 Mar 10 10:20 test
```

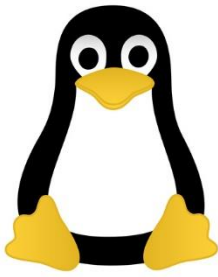
# Thay đổi quyền truy nhập (2)



`$chmod <qui><operation><droits> <files>`

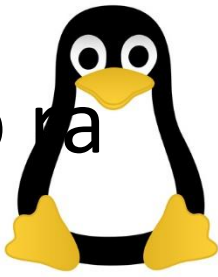
- Qui = u | g | o | a (all)
- Operation
  - + (thêm 1 hoặc 1 số quyền vào tập các quyền file đã có)
  - - (bỏ 1 hoặc 1 số quyền khỏi tập các quyền file đã có)
  - = (gán mới 1 hoặc 1 số quyền cho file)
- Quyền = r | w | x | s

# Ví dụ



```
$ ls -l test.txt
-rw-rw-r-- 1 tuanh user1 150 Mar 19 19:12 test.txt
$ chmod o+w test.txt
$ ls -l test.txt
-rw-rw-rw- 1 tuanh user1 150 Mar 19 19:12 test.txt
$ chmod a-rw test.txt
$ ls -l test.txt
----- 1 tuanh user1 150 Mar 19 19:12 test.txt
$ cat test.txt
cat: test.txt: Permission denied
```

# Định nghĩa các quyền ngầm định khi tạo ra 1 file



- Các quyền ngầm định của 1 file khi tạo ra có thể được xác định bằng lệnh `umask`

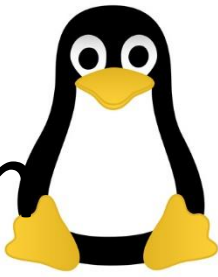
**`$umask`**

**`022`**

- Số 0 có nghĩa là quyền của người sử dụng không bị hạn chế (rwx)
- Số 2 có nghĩa là quyền ghi (w) bị hạn chế (r-x).

**`$umask 022`**

# Thay đổi người sở hữu và nhóm

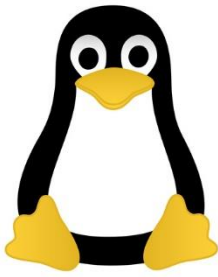


`$chown [-R] <utilisateur> <files>`

- Thay đổi người sở hữu của file

`$chgrp <groupe> <files>`

- Thay đổi nhóm của file
- Có thể sử dụng tùy chọn `-R` để lặp lại việc thực hiện các lệnh (ví dụ thực hiện việc thay đổi quyền sở hữu hoặc nhóm của mọi file trong cùng một thư mục)
- Các lệnh trên chỉ dành cho những người sử dụng có quyền root



# Tìm kiếm file

\$ find tên\_thư\_mục expressions

- Cho phép tìm kiếm các file trong một thư mục (ngầm định là trong thư mục hiện tại) với một số điều kiện hoặc các lệnh thực thi trên tập các file tìm được.
- Các điều kiện
  - Tên : -name tên
  - Quyền truy cập : -perm quyền\_truy\_cập
  - Kiểu : -type d/f/...
  - Kích thước : -size N
  - Thời gian : -atime N, -mtime N, -ctime N
- Các lệnh thực thi trên tập các file tìm được
  - -print
  - -exec câu\_lệnh



# Ví dụ

- `$find /usr -name toto -print`
  - Tìm kiếm file tên là toto trong thư mục /usr (bao gồm cả các thư mục con của /usr)
- `$find /usr -name " *.c " -print`
  - Đưa ra danh sách các file kết thúc bằng « .c »
- `$find / -mtime 3 -print`
  - Tìm tất cả các file có thay đổi trong 3 ngày gần đây
- `$find / -size 2000 -print`
  - Tìm tất cả các file có kích thước lớn hơn 1 GB (= 2000 block 512 KB)
- `$find / -type f -user olivier -perm 755 -print`
  - Tìm tất cả các file thuộc về người sử dụng olivier, đồng thời có quyền truy cập là 755