

# GIAO TIẾP BẰNG LỆNH TRÊN LINUX

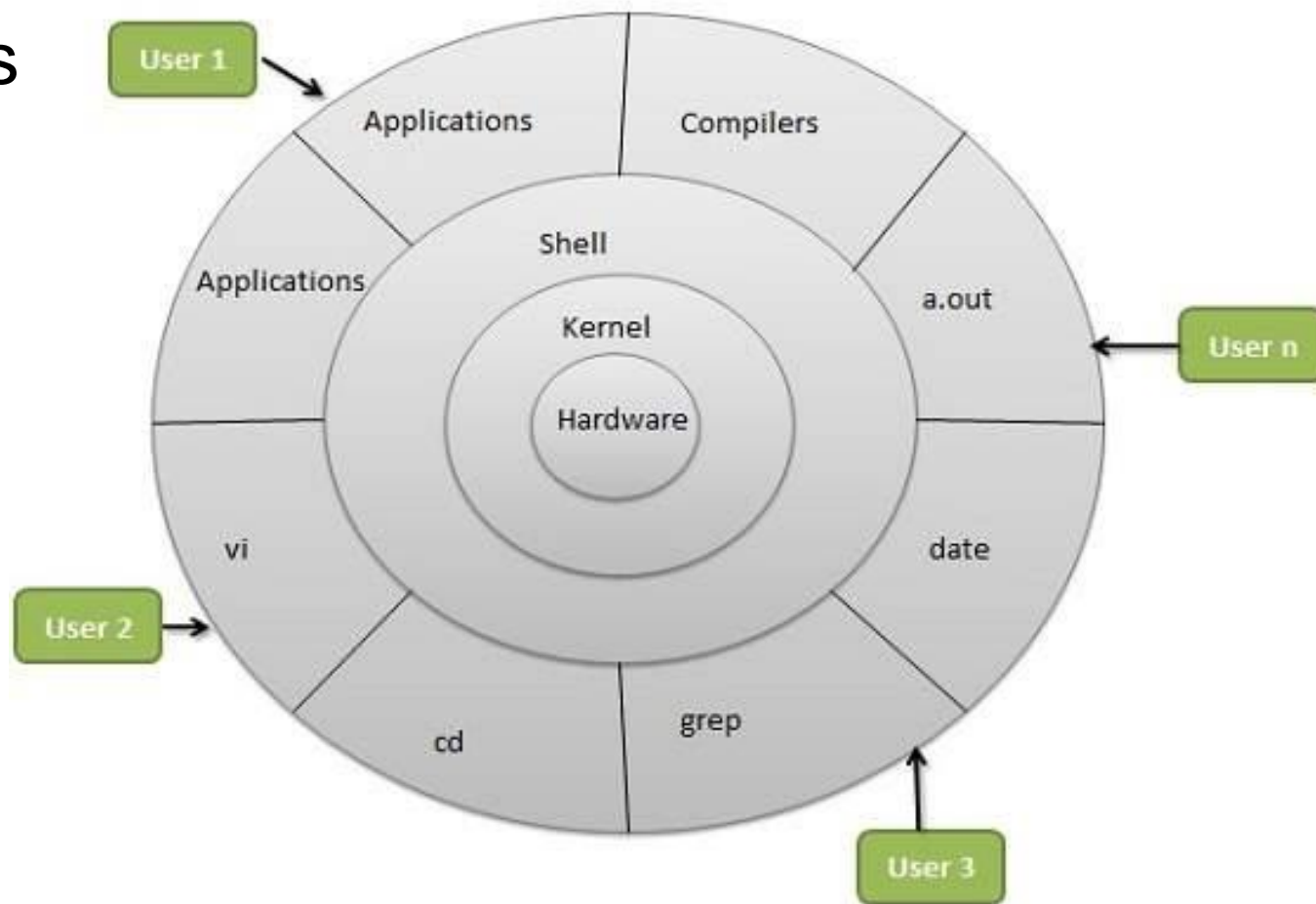


# Các câu lệnh cơ bản của Linux

- Các thành phần của linux
- Cơ bản về dấu nhắc Shell
- Cấu trúc cây thư mục linux
- Lệnh trong linux - Một số câu lệnh thông dụng
- Quản lý tập tin và thư mục
- Các đặc tính của Shell
- Một số câu lệnh hữu dụng

# Các thành phần của hệ thống linux

- Kernel
- Shell
- Applications and Utilities



# Kernel

- Nhân (còn được gọi là hệ lõi) của Linux.
- Là một bộ các mô đun chương trình có vai trò điều khiển các thành phần của máy tính, phân phối các tài nguyên cho người dùng (các quá trình người dùng).
- Là cầu nối giữa chương trình ứng dụng với phần cứng máy tính.

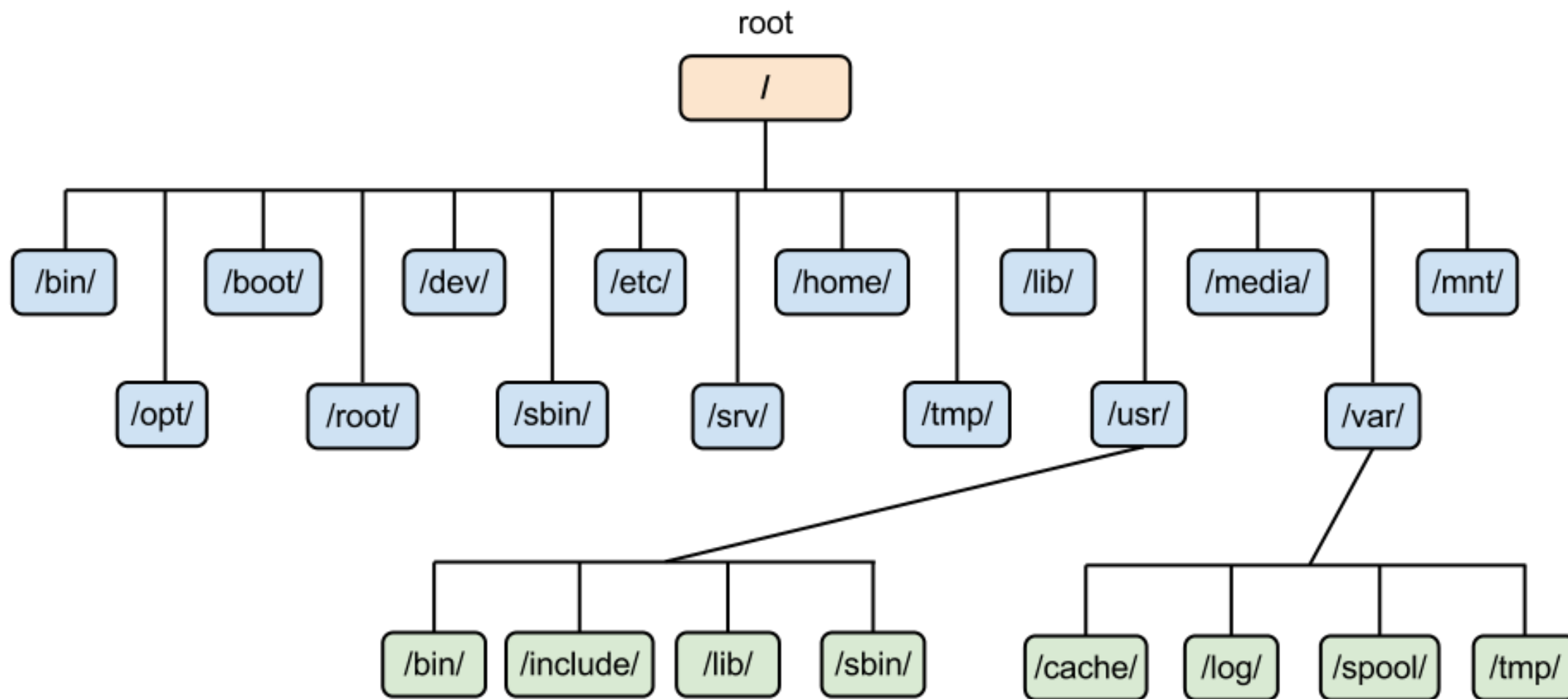
# Khái niệm Shell

- Trình thông dịch dòng lệnh.
- Giống như các ứng dụng bình thường khác.
- Tương tác trực tiếp với hệ điều hành, đóng vai trò trung gian giữa nhân linux và người dùng cũng như các chương trình ứng dụng khác.
- Cung cấp môi trường lập trình đơn giản.

# Đăng nhập và dấu đợi lệnh của trình shell

- [root prompt]# : dấu đợi lệnh cho tài khoản root
- [normal user prompt]\$ : dấu đợi lệnh cho tài khoản thường
- Chú ý:
  - Shell linux xử lý các lệnh có phân biệt chữ thường và hoa
  - Shell mặc định của linux là "bash" (GNU Bourne-Again SHell)
- Thoát khỏi trình Shell
  - ^D (Ctrl + D)
  - exit
  - logout

# Cấu trúc cây thư mục trong linux



# Các kiểu tập tin trong linux

## 4 kiểu tập tin cơ bản:

- Tập tin bình thường(Ordinary files)
- Thư mục
- Tập tin đặc biệt (devices, sockets, pipes, ...)
- Liên kết (Symlinks)

## Quy ước đặt tên tập tin

- Tên tập tin có thể dài tới 255 ký tự
- Bất kỳ ký tự nào cũng có thể dùng để đặt tên (bao gồm cả ký tự đặc biệt)
- Tập tin/thư mục ẩn bắt đầu bằng "."
  - .bash\_history .bash\_profile .bashrc
  - .kde .gnome .mozilla



# Đường dẫn(Path)

- Đường dẫn tuyệt đối - luôn bắt đầu bằng "/"
  - Ví dụ:
    - / : Thư mục gốc (root)
    - /usr: Thư mục usr nằm trong thư mục gốc
    - /usr/local/bin
- Đường dẫn tương đối - không bao giờ bắt đầu bằng "/"
  - Ví dụ: bin ../usr/local ./bin
- Đường dẫn đặc biệt:
  - parent directory: **dấu hai chấm** ..
  - Thư mục đang làm việc: **dấu chấm** .

# Giới thiệu về sử dụng lệnh trong linux

- Dạng tổng quát của lệnh Linux có thể được viết như sau:
  - `$ <Tên lệnh> [Tùy chọn] [tham số]`
  - Tên lệnh là một dãy ký tự, không có dấu cách, biểu thị cho một lệnh của Linux hay một chương trình. Tên lệnh là bắt buộc phải có khi gõ lệnh.
    - Tùy chọn: Chính là những tham số điều khiển hoạt động của lệnh theo các trường hợp riêng. Trong Linux, tham số tùy chọn thường bắt đầu bởi dấu trừ "-" hoặc hai dấu trừ liên tiếp "--" nếu tùy chọn là một từ gồm nhiều chữ cái.
    - Tham số vị trí: thường là tên file, thư mục và thường là các đối tượng chịu sự tác động của lệnh. Khi gõ lệnh, tham số vị trí được thay bằng những đối tượng mà người dùng cần hướng tác động tới.
- Ví dụ
  - `$ ls -l`
  - `$ ls -l /home/jerry`      `$ ls -l ..`

# Một số câu lệnh thông dụng

- Các câu lệnh là các chương trình, các đoạn script, hay các lệnh có sẵn của shell
- `passwd` - Thay đổi mật khẩu đăng nhập
- `pwd` - hiển thị thư mục hiện hành đang làm việc
- `cd` - thay đổi thư mục hiện hành đang làm việc
- `ls` - liệt kê nội dung thư mục
- Để tra cứu cú pháp, chức năng của một lệnh:
  - `$ man command`

# passwd - thay đổi mật khẩu đăng nhập

```
$ passwd username  
Changing password for student  
(current) UNIX password:  
New password:  
Retype new password:
```

- Chú ý:
  - Mật khẩu phân biệt chữ thường, chữ hoa
  - Chọn một mật khẩu tốt

# **pwd - hiển thị thư mục hiện hành đang làm việc**

- Ví dụ in thư mục đang làm việc:

**(printing working directory)**

**\$ pwd**

**/home/student**

# cd - thay đổi thư mục hiện hành

- `$ cd /usr`
  - Chuyển sang thư mục: [/usr]
- `$ cd bin`
  - Chuyển sang thư mục: [/usr/bin]
- `$ cd ../../etc`
  - Chuyển sang thư mục: [/etc]
- `$ cd ~`
  - Chuyển về thư mục người dùng: [/home/student]
- `$ cd`
  - Chuyển về thư mục người dùng: [/home/student]

## cd: các tùy chọn, chức năng của lệnh

- `$ cd` : di chuyển tới thư mục đăng nhập (home directory)
- `$ cd ~` : giống như trên
- `$ cd ..` : Di chuyển ra thư mục cha (parent directory)
- `$ cd ~other` : Di chuyển vào thư mục nhà (home directory) của người dùng khác
- `$ cd path_directory` : Di chuyển vào 1 thư mục

# ls - liệt kê nội dung thư mục

- `$ ls`
  - Kết quả: Desktop dir1 dir2 dir3 dir4 dir5 file1 file2 file3 file4 homelist.txt
- `$ ls /home`
- Kết quả: student
- Liệt kê chi tiết thư mục: `$ ls -l`
- Liệt kê tất cả các nội dung trong thư mục kể cả các tệp ẩn:
  - `$ ls -a`



# Câu lệnh alias

- Thay thế 1 chuỗi bởi 1 từ
- Làm cho câu lệnh đơn giản hơn, ngắn hơn và hữu dụng hơn
- **alias** – Tạo 1 bí danh
- **unalias** – Xóa bí danh
- **\$ alias**
  - **\$ alias cp='cp -i'**
  - **\$ alias ls='ls -l --color=tty'**
  - **\$ alias lsc='ls --color=tty'**
  - **\$ alias mvi='mv -i'**
  - **\$ alias rmi='rm -i'**
- **\$ unalias ls**

# Cấu trúc lệnh

- `$ command [options] [arguments]`
- Một số chú ý về dòng lệnh:
  - các thành phần của lệnh được phân cách nhau bởi **một** khoảng trắng
  - các tùy chọn thường bắt đầu bằng "-"
  - các tùy chọn có thể kết hợp lại với bằng một dấu "-"
  - Xem, tra cứu các chức năng, cách sử dụng lệnh bằng lệnh man
    - VD `$ man ls`
  - Không phải tất cả các lệnh của linux đều theo chuẩn trên
  - Linux cho phép tùy chọn đặt sau tham số (argument)

# Xem trợ giúp một lệnh

- Để xem trợ giúp cho một lệnh: Sử dụng lệnh man
- Cú pháp: `$ man command`
- Di chuyển trong khi xem trợ giúp
  - Spacebar: next page
  - b: previous page
  - q quit
  - ctrl + z: Đưa tiến trình man xuống background
  - fg: Đưa tiến trình man về lại foreground

# Các phím điều khiển trên thiết bị cuối

- ^C: Dừng thao tác
- ^D: Kết thúc tệp
- ^\ : Kết thúc lệnh
- ^H: Xóa ngược lại 1 ký tự
- ^W: Xoá tới 1 ký tự
- ^U: Xóa ngược tới đầu dòng
- Arrow: - Di chuyển ngược/tới 1 ký tự

# Quản lý tệp/thư mục

- Create
- Copy
- Move/Rename
- Remove
- View

# mkdir - Tạo một thư mục

`mkdir [OPTION]... DIRECTORY...`

- Ví dụ:
  - `$ mkdir dir1`
  - `$ mkdir dir1 dir2`
- Option:
  - `-p`: Tạo thư mục cha nếu cần thiết
- Ví dụ:
  - `$ mkdir -p dir3/dir4`

# rmmdir - Xóa các thư mục rỗng

- Synopsis: `mdir [OPTION] ... DIRECTORY ...`
- Ví dụ:
  - `$ rmmdir dir1`
  - `$ rmmdir dir1 dir2`
- Option:
  - `-p`: Xóa tất cả thư mục trong đường dẫn.
- Ví dụ:
  - `$ rmmdir -p a/b/c` tương đương với `$ rmmdir a/b/c a/b a`
  - `$ rmmdir -p dir3/dir4`

# touch, cat - Tạo các tệp

- **touch**: Tạo một tệp tin rỗng
  - Ví dụ: **\$ touch hello.txt** : Tạo tệp tin hello.txt
- **cat**:
  - **\$ cat**: soạn thảo nội dung trên màn hình terminal
  - **\$ cat > hello.txt**: Soạn thảo nội dung trên màn hình terminal và ghi đè nội dung vào tệp tin hello.txt, tạo mới tệp tin nếu tệp tin chưa tồn tại
  - **\$ cat >> hello.txt**: Soạn thảo nội dung trên màn hình terminal và ghi nối nội dung vào tệp tin hello.txt, tạo mới tệp tin nếu tệp tin chưa tồn tại
  - **\$ cat hello.txt**: Hiển thị nội dung tệp tin hello.txt trên màn hình terminal



# rm - Xóa tập tin và thư mục

- Synopsis: `rm [OPTION] ... FILE ...`
- Ví dụ:
  - `$ rm file6`
  - `$ rm file4 file5`
- Option:
  - `-f, --force`: bỏ qua các tập tin và tùy chọn không tồn tại, không hiển thị nhắc nhở
  - `-i : interactive move`: Hiển thị lời nhắc, xác nhận của người dùng khi xóa
  - `-r`: Xóa thư mục và các nội dung bên trong một cách đệ qui
- Ví dụ:
  - `$ rm -rf dir2`
  - NEVER do as root: `"rm -rf /"`

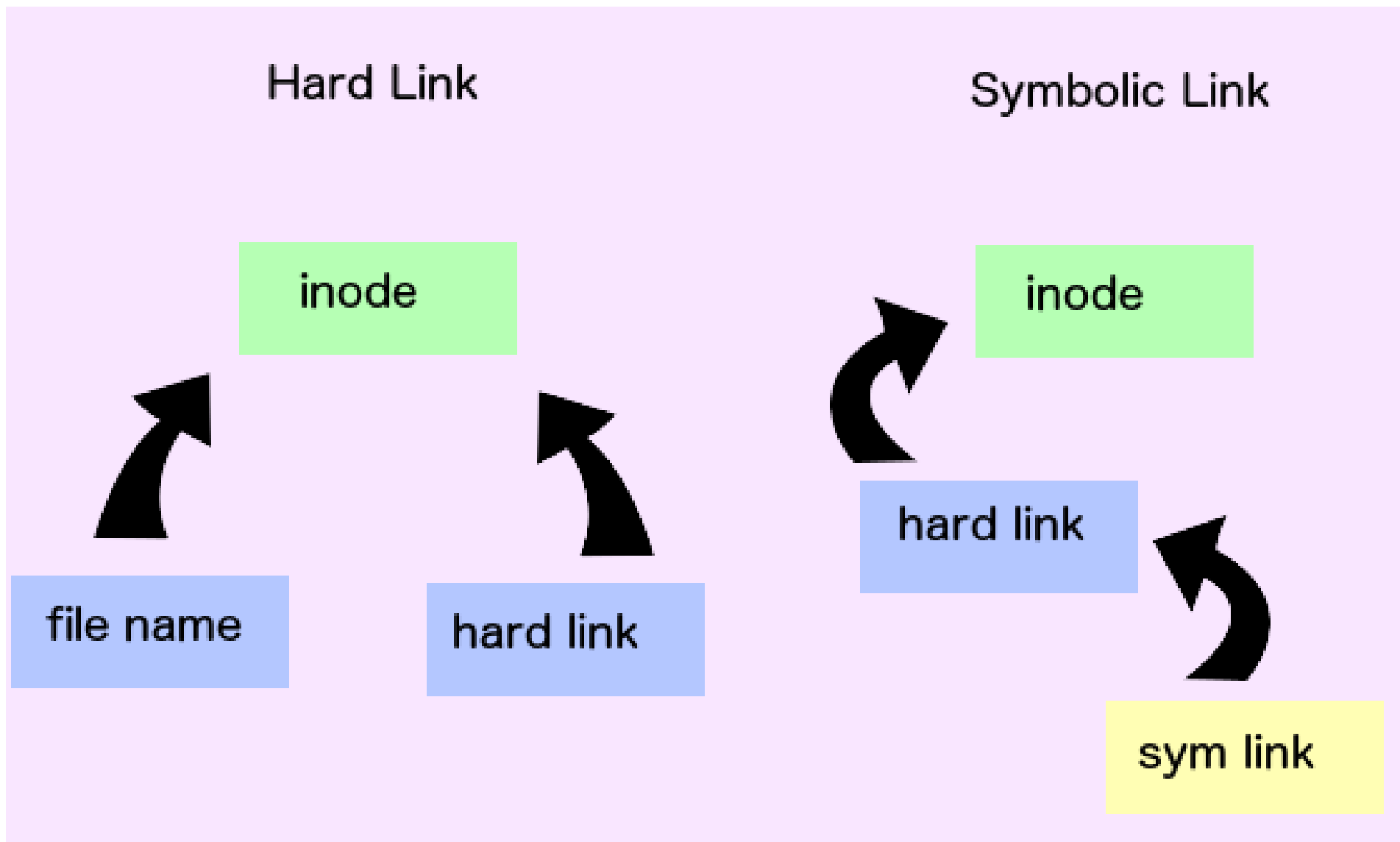
# cp - Sao chép các tập tin và thư mục

- Synosis: `cp [OPTION] ... SOURCE ... DEST ...`
- Ví dụ:
  - `$ cp file5 file6`
  - `$ cp file5 /tmp`
- Option
  - `-r,-R`: copy recursively
- Ví dụ:
  - `$ cp -r dir1 dir5`

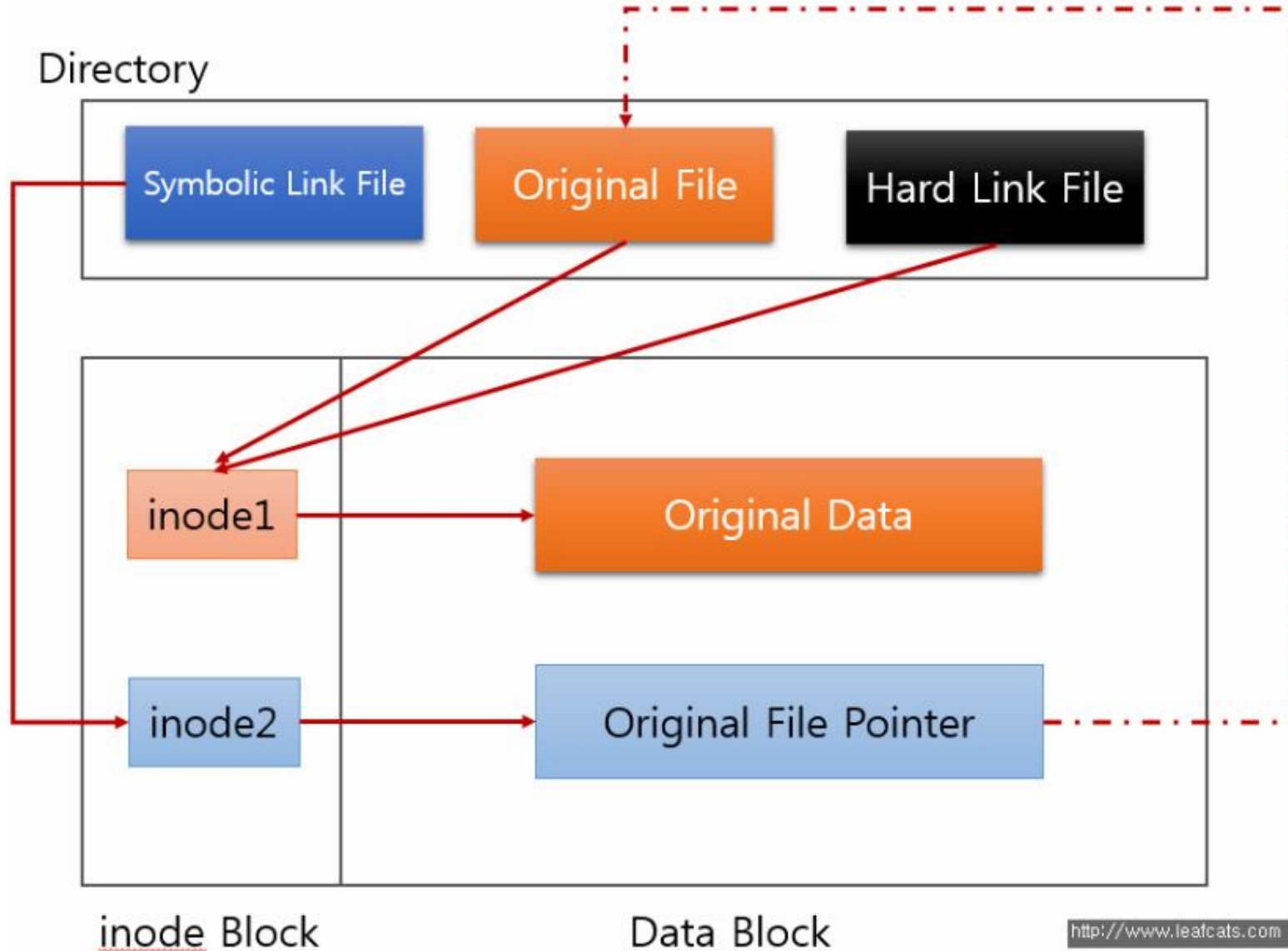
# mv - Di chuyển(đổi tên) tập tin

- Synopsis: `mv [OPTION] ... SOURCE ... DEST ...`
  - Nếu SOURCE và DEST cùng thuộc một thư mục: Đổi tên
  - Nếu SOURCE và DEST không cùng thuộc một thư mục: di chuyển
- Ví dụ:
  - `$ mv file5 file6`
  - `$ mv file6 /tmp`
  - `$ mv dir1 dir2`
- Option:
  - `-f`: force overwrite
  - `-i`: interactive mode

# Liên kết biểu tượng và liên kết vật lý



# Liên kết biểu tượng và liên kết vật lý

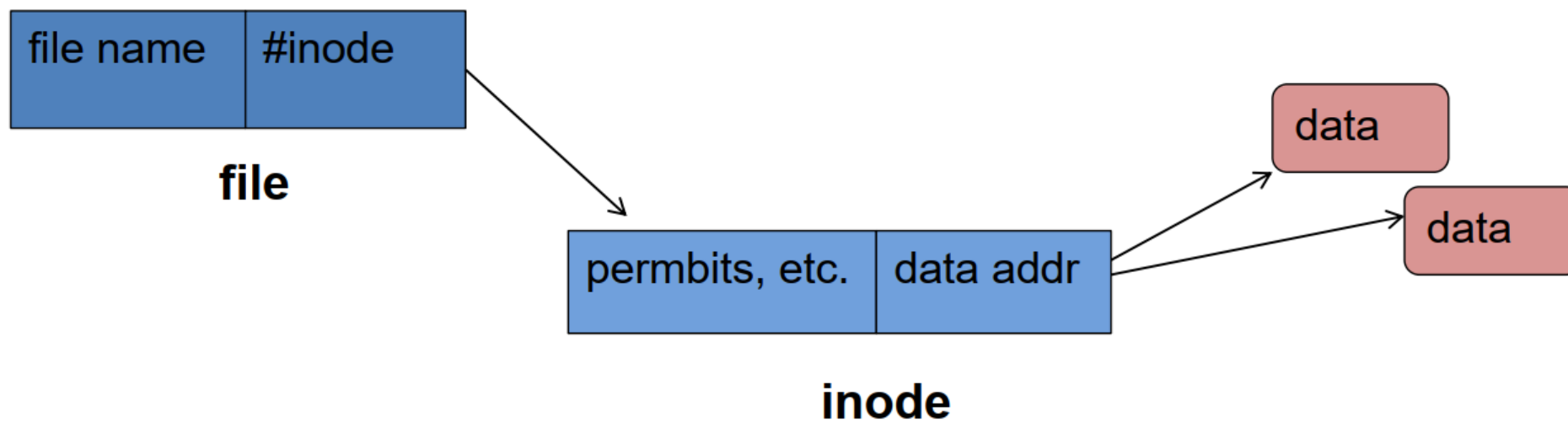


# In - Tạo một liên kết biểu tượng tới tập tin

- Synopsis:
  - `ln [OPTION] ... TARGET ... LINK_NAME`
  - `ln [OPTION] ... TARGET` : Tạo một liên kết cho TARGET trong thư mục hiện hành
  - `ln [OPTION] ... TARGET ... DIRECTORY`: Tạo một liên kết cho TARGET trong thư mục DIRECTORY
- Option:
  - `f`: Xóa tệp trong thư mục đích
  - `-s`: Tạo liên kết mềm(symbolic links) thay vì liên kết cứng(hard links)
- Ví dụ:
  - `$ ln -s /usr/local/bin`
  - `$ ln -s dir1 firstdir`

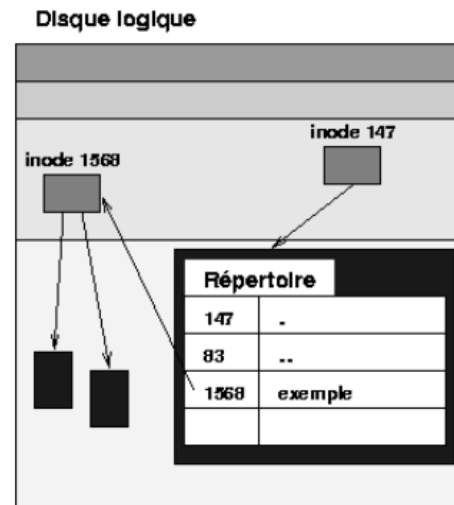
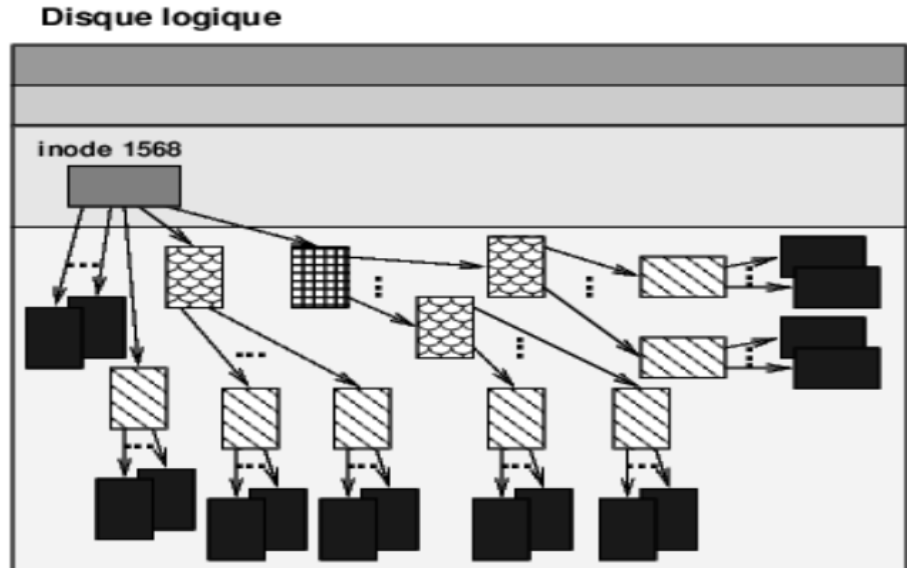
# Khái niệm inode

- Một file trong linux



# Thư mục

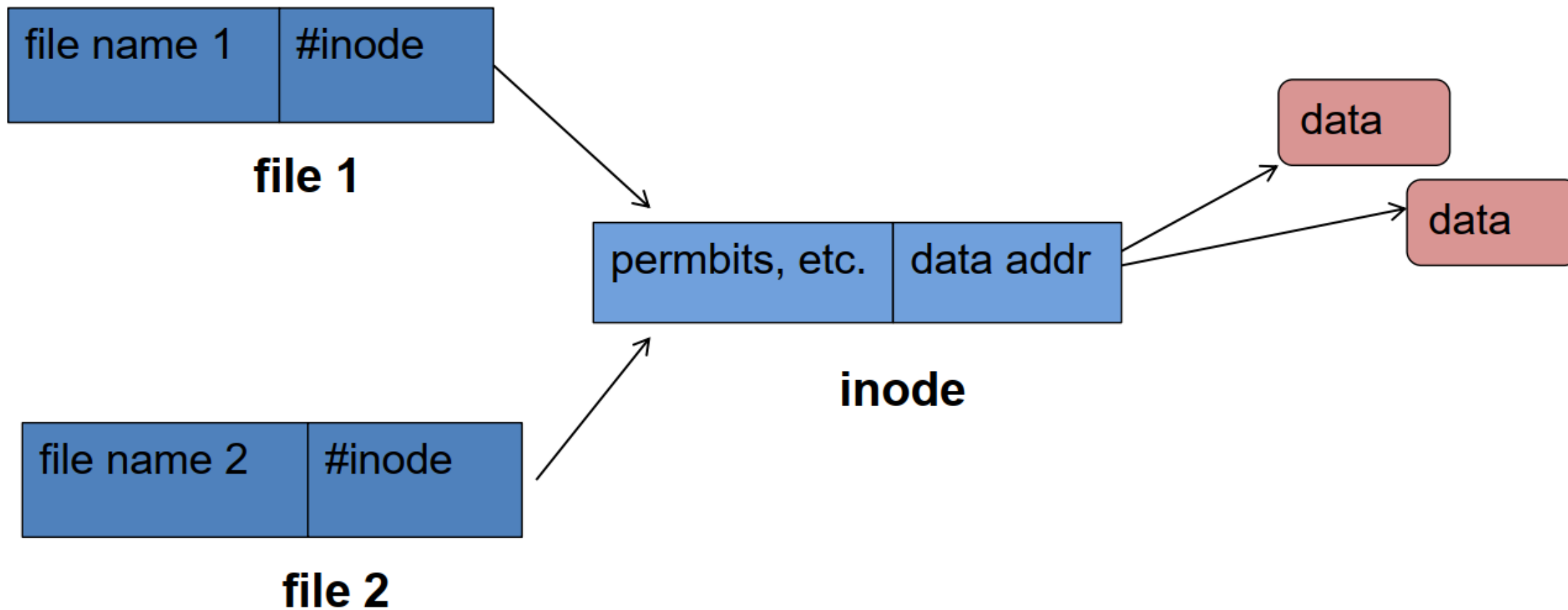
- Một thư mục là một tập với nội dung là một bảng liên kết
  - Một liên kết gắn một tên tệp với một inode của hệ thống tệp





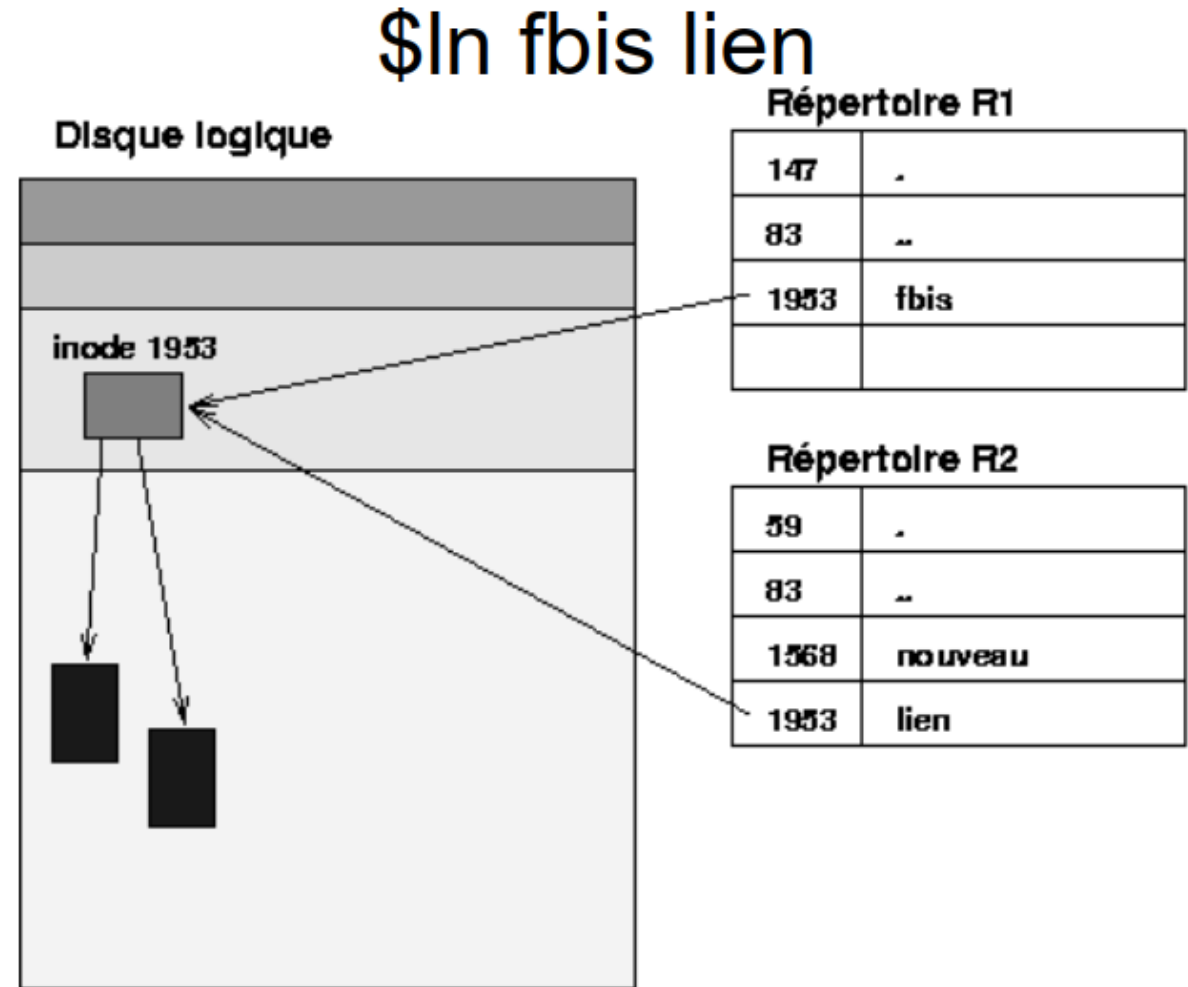
# Inode, soft link and hard link

- 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 (2)

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

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

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

# Liên kết biểu tượng và liên kết vật lý

- Không tạo liên kết vật lý cho thư mục
- Liên kết biểu tượng không cập nhật nếu di chuyển liên kết hoặc file nguồn thì có thể sẽ không truy cập đến file nguồn thông qua liên kết biểu tượng được.
- Liên kết vật lý luôn tham chiếu đến nguồn ngay cả khi nguồn bị di chuyển hay xóa.
- Chúng ta có thể phân biệt rõ một file liên kết biểu tượng và file gốc tương ứng bằng lệnh ls
- Điều gì sẽ xảy ra khi một người xóa đi file gốc trong hai trường hợp liên kết vật lý và liên kết biểu tượng.

# ls - Xem nội dung của thư mục

- **ls [OPTION] FILE**

- -a: liệt kê tất cả, bao gồm cả tệp ẩn
- -l: liệt kê dưới định dạng dài
- -R: liệt kê đệ quy, cả các thư mục con

```
$ 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
```

Số liên kết đến Inode

# 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 (ký tự)
  - l : link
  - m : phần bộ nhớ trong dùng chung
  - p : đường ống

# Câu lệnh hiển thị

- **echo**: Hiển thị lại nội dung trong câu lệnh trên màn hình
  - `$ echo This is a test --> Enter`
  - Kết quả: This is a test
- **cat**: Hiển thị nội dung của tập tin trên màn hình
  - `$ cat /etc/passwd`
- Lệnh **more** hoặc **less**: Cho phép mở một tập tin lớn trên màn hình để xem theo từng trang (Sử dụng phím pgup, pgdn)
  - Khác với more, lệnh less không nạp một lúc toàn bộ tập tin vào bộ nhớ
  - `$ more /etc/passwd`
  - `$ less /etc/group`

# Câu lệnh hiển thị

- Lệnh **head**: Dùng để xem những dòng đầu của tập tin (mặc định là 10 dòng)
  - Synopsis: **head [option] ... file**
  - Các tùy chọn:
    - -n: Số dòng đầu tiên của mỗi tệp. VD: **\$ head -n 20 /ect/passwd**
    - -c: Số byte đầu tiên của mỗi tệp
    - -v luôn in tiêu đề xác định tên tệp. VD: **\$ head -n 20 -v /ect/passwd**
    - -q: Không in tiêu đề xác định tên tệp
- Lệnh **tail**: Dùng để xem những dòng cuối của tập tin
  - Synopsis: **tail [option] ... file**
    - ....
    - -f: Tiếp tục đọc tập tin cho đến khi CTRL + C



# Ký tự đại diện

- Khi chỉ ra tên của tập tin hay thư mục thì có thể chỉ một cách không rõ ràng
- Ký tự đại diện có thể dùng để chỉ ra một phần hay tất cả tên
  - \* tương ứng bất kỳ chuỗi ký tự nào, bao gồm ký rỗng
  - ? tương ứng với 1 ký tự bất kỳ
  - [...] tương ứng bất kỳ ký tự nào nằm trong cặp ngoặc vuông đóng
  - [!], [^] không tương ứng với bất kỳ ký tự nào
  - \ giải phóng các ký tự đặc biệt \*, ?, )

## Ví dụ ký tự đại diện

- *Is a\** : liệt kê tất cả tệp bắt đầu a
- *Is a?.txt* : liệt kê tất cả tệp dưới dạng a?.txt, với ? là bất kỳ ký tự nào
- *Is [aei]\** : liệt kê tất cả tệp bắt đầu với a, e, hoặc i
- *Is [a-d]\** : liệt kê tất cả tệp bắt đầu từ a tới d
- *Is [!A-D]\** : liệt kê tất cả tệp không bắt đầu từ A tới D

# Chức năng tự động hoàn thành

- Nhấn phím<Tab> để thực hiện chức năng hoàn thành trên văn bản
  - username, hostname, command or filename completion
  - list all possible completions
- Ví dụ:
  - `$ cp<Tab>`
    - cp cpp
    - cpio cproto
  - `$ cd dir<Tab>`
    - dir1 dir2 dir3 dir4 dir5
  - `$ cd ~s<Tab>`
    - ~shutdown ~squid ~student ~sync
  - `$ <Tab><Tab>`
    - Display all 1977 possibilities? (y or n)

# Xem lịch sử các lệnh đã sử dụng

- Các lệnh đã từng sử dụng được lưu trữ trong tập tin "~/.bash\_history"
  - ^P hoặc UP (điều hướng lên): Xem lệnh trước
  - ^N hoặc DOWN (điều hướng xuống): Xem lệnh sau
- Lệnh history: Hiển thị các lệnh đã sử dụng
  - ```
$ history  
1 pwd  
2 cd /  
3 cd  
4 ls  
5 ls -a  
6 mkdir dir1
```
- Để sử dụng một lệnh trong danh sách: **\$ !n**
  - VD: thực hiện lệnh **pwd**: **\$ !1** --> Enter

# Số nhập/xuất chuẩn

- Số nhập chuẩn (stdin): 0
- Số xuất chuẩn (stdout): 1
- Số lỗi chuẩn (stderr): 2

# Sử dụng điều hướng nhập xuất

- < điều hướng nhập
- > điều hướng xuất (overwrite)
- >> điều hướng xuất (append)
- 2> điều hướng lỗi (overwrite)
- 2>> điều hướng lỗi(append)

# Sử dụng điều hướng nhập xuất

| Lệnh           | Miêu tả                                                                               |
|----------------|---------------------------------------------------------------------------------------|
| pgm > file     | Đầu ra của pgm được điều hướng tới file.                                              |
| pgm < file     | Chương trình pgm đọc input từ file.                                                   |
| pgm >><br>file | Đầu ra của pgm được gán tới file.                                                     |
| n > file       | Đầu ra từ stream với dấu hiệu n được điều hướng tới file.                             |
| n >> file      | Đầu ra từ stream với dấu hiệu n được gán tới file.                                    |
| n >& m         | Sáp nhập đầu ra từ stream n với stream m.                                             |
| n <& m         | Sáp nhập input từ stream n với stream m.                                              |
| << tag         | input tiêu chuẩn xuất phát từ đây qua thẻ tiếp theo tại phần đầu của dòng.            |
|                | Nhận đầu ra từ một chương trình hoặc một tiến trình và gửi nó cho chương trình khác.. |

## Ví dụ

- `$ ls -la /home > homelist.txt`
  - Liệt kê tất cả các tệp trong thư mục home vào tệp homelist.txt
- `$ cat > file1 < homelist.txt`
  - Nhập nội dung của homelist.txt vào tệp file1, ghi đè nếu file1 đã có nội dung
- `$ ls -l ~/dir1 >> homelist.txt`
  - Bổ sung nội dung các tệp trong ~/dir1 vào tệp homelist.txt
- `$ ls -lR /home 2>/dev/null`
  - Liệt kê tất cả các tệp trong /home (kể cả thư mục con) ra màn hình, điều hướng lỗi vào /dev/null
- `$ ls -lR /home > homelist.txt 2>&1`
  - Liệt kê tất cả các tệp trong /home (kể cả thư mục con) vào tệp homelist.txt, điều hướng lỗi ra màn hình (1: số xuất chuẩn)



# Ổng dẫn

- Kết quả của câu lệnh trước sẽ là dữ liệu nhập vào cho câu lệnh sau
  - "|" - ký hiệu ổng dẫn
  - command1 | command2 | command3 ...
- Ví dụ:
  - \$ ls -lR | more

# Lưu lại phiên làm việc

- Lệnh **script** - tạo một tập tin lưu nội dung phiên làm việc
- Synopsis: **script [-a] [file]**
  - -a: append the output to file or typescript
- Ví dụ:

```
$ script
```

```
Script started, file is typescript
```

```
$ pwd
```

```
$ /home
```

```
$ cd /
```

```
$ exit
```

```
Script done, file is typescript
```

# Quyền truy cập tập tin

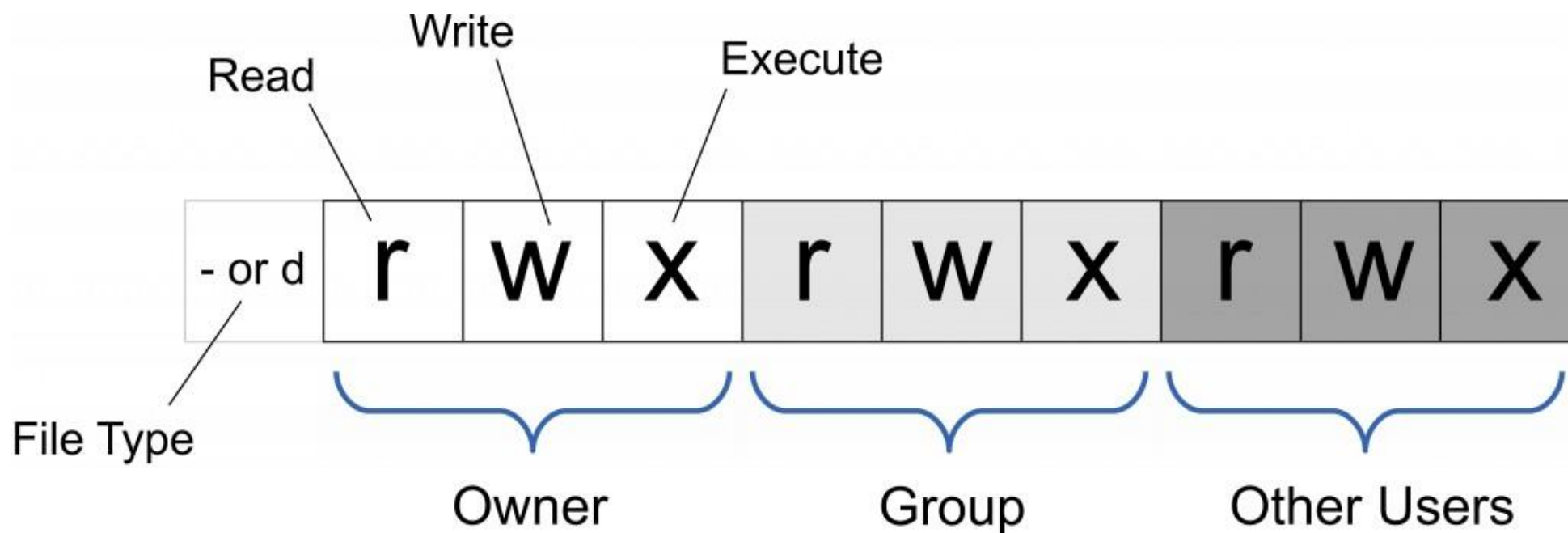
- Người sở hữu và quyền trên tập tin (ownership and file permissions)
- Cách trình bày các quyền (permissions represent)
- Thay đổi quyền (changing permissions)

# Người sở hữu và quyền trên tập tin

- Tất cả tập tin và thư mục đều được sở hữu bởi chính người tạo ra nó
- Quyền truy cập tập tin được chia ra thành từng nhóm
  - **User** - người sở hữu của tập tin/thư mục đó
  - **Group** - Nhóm mà người sở hữu thuộc về
  - **Others** - Người dùng khác
- Để xem quyền trên tập tin dùng lệnh "**ls -l**"

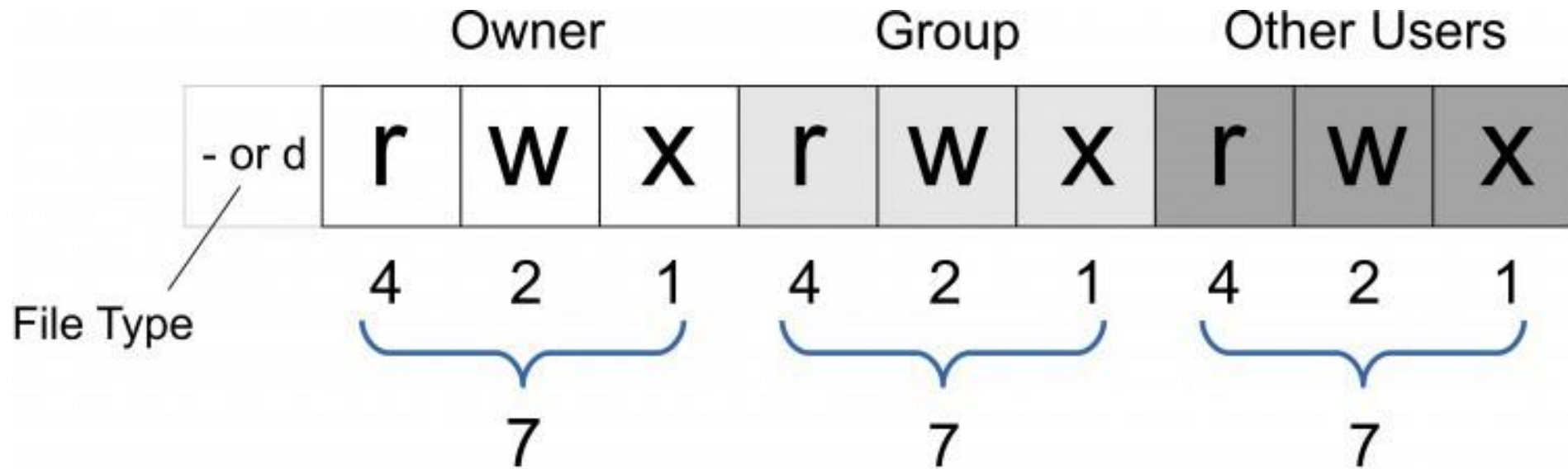
# Cách trình bày các quyền – dạng chữ

- r: readable
- w: writeable
- x: executable
- - : quyền không được cấp



# Cách trình bày các quyền – dạng số

- Nếu quyền không được cấp: giá trị 0
- Nếu quyền không được cấp: giá trị như hình bên dưới
- Ví dụ:  $r-e\ rwx\ --x \leftrightarrow (4+0+2)(4+2+1)(0+0+1) = 671$



# Đối tượng cấp quyền và các thao tác trên quyền

- Đối tượng cấp quyền
  - **u** user, người dùng sở hữu tập tin
  - **g** group, nhóm mà người dùng thuộc về
  - **o** others, những người khác (không phải là người sở hữu và nhóm người sở hữu thuộc về)
  - **a** all, tất cả mọi người (u, g, and o)
- Các thao tác trên quyền
  - **+** thêm quyền
  - **-** bỏ quyền
  - **=** gán quyền

# chmod - thay đổi quyền

- **Synopsis:** **chmod [OPTION] .. MODE .. FILE**
- **-R:** thay đổi quyền trên thư mục một cách đệ quy
- Ví dụ cách sử dụng chmod:
  - **g+w** : thêm quyền ghi cho nhóm
  - **o-rwx** : xóa tất cả quyền của người dùng khác
  - **u+x** : thêm quyền chạy cho người dùng
  - **a+rw** : thêm quyền r và w cho tất cả mọi người
  - **ug+r** : thêm quyền đọc cho người dùng và nhóm
  - **g=rx** : gán nhóm chỉ có quyền đọc và chạy (not write)



## Ví dụ cách dùng chmod

- `chmod u-x header.php`
- `chmod -R a+rw lecture`
- `chmod u=rwx,go=r myhome.jpg`
- `chmod 644 dirlist.txt`
- `chmod 755 myprogram`

# Một số quyền đặc biệt

- Trong linux, có 3 quyền đặc biệt:
  - **setuid** - thường dùng cho các ứng dụng, quyền này chỉ ra rằng ứng dụng sẽ chạy với quyền của người sở hữu nó và không chạy với quyền của người chạy chương trình. nó được chỉ ra bởi ký tự s tại vị trí x của người sở hữu tập tin. Nếu người sở hữu tập tin không có quyền x, ký tự S hoa dùng để phản ánh điều này.
  - **setgid** - thường dùng cho các ứng dụng, quyền này chỉ ra rằng ứng dụng sẽ chạy với quyền của nhóm sở hữu nó mà không chạy với quyền của nhóm của người dùng đang chạy ứng dụng. nó được chỉ ra bởi ký tự s tại vị trí x của quyền nhóm. Nếu nhóm của người sở hữu tập tin không có quyền x, ký tự S hoa dùng để phản ánh điều này.
  - **sticky bit** - thường dùng cho các thư mục, bit này chỉ ra rằng các tập tin được tạo trong thư mục này chỉ có thể xóa bởi chính người dùng đã tạo ra nó. nó được chỉ ra bằng ký tự t tại vị trí x của đối tượng người dùng khác, nếu đối tượng người dùng khác không có quyền x, ký tự T dùng để phản ánh điều này

# Cách thiết lập userid, groupid, sticky bit

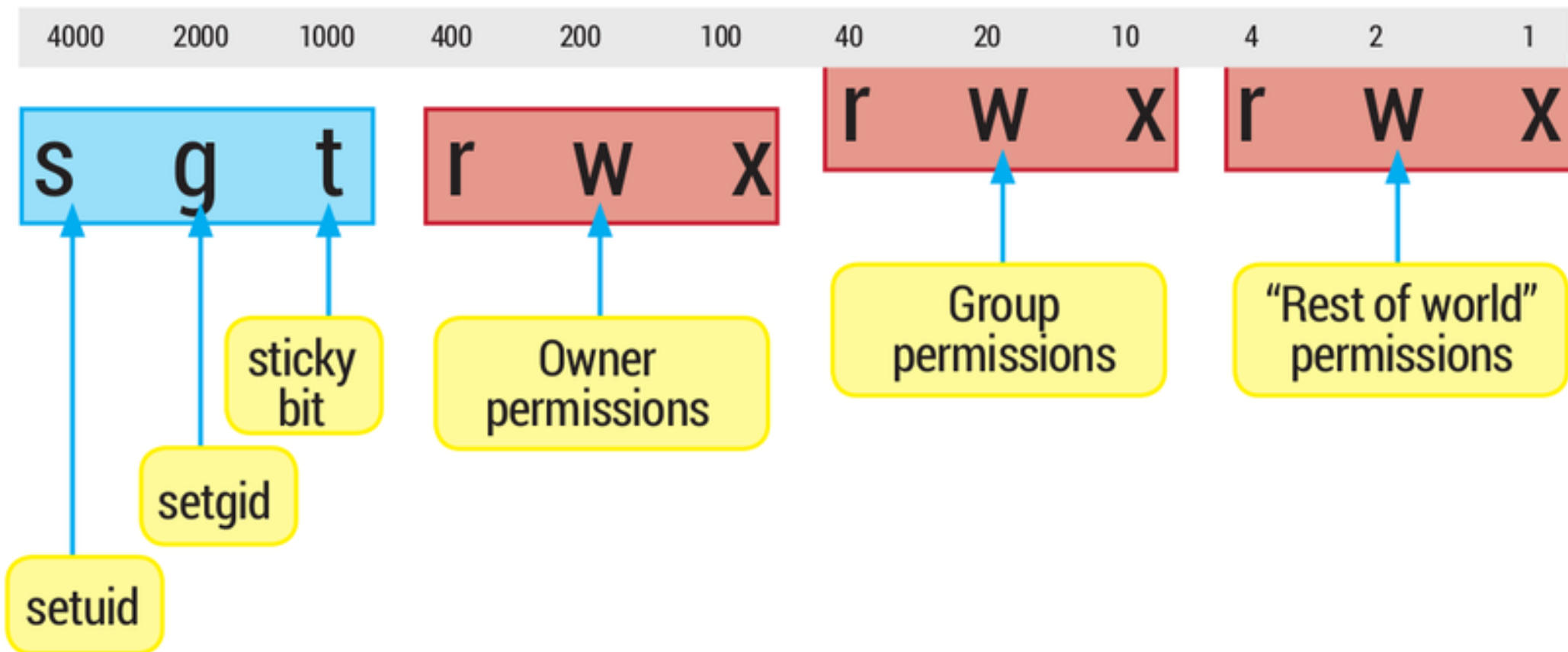


## *Working with SUID, SGID, and Sticky Bit*

| Permission | Numerical Value | Relative Value | On Files                                            | On Directories                                            |
|------------|-----------------|----------------|-----------------------------------------------------|-----------------------------------------------------------|
| SUID       | 4               | u+s            | User executes file with permissions of file owner.  | No meaning.                                               |
| SGID       | 2               | g+s            | User executes file with permissions of group owner. | File created in directory gets the same group owner.      |
| Sticky bit | 1               | +t             | No meaning.                                         | Users are prevented from deleting files from other users. |

# Cách thiết lập các quyền dạng số

Octal values



## Thể hiện ký tự

|        |                                                                                                                                                                                         |
|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SUID   | <p><b>-rws-----</b> quyền chạy và SUID của owner được thiết lập</p> <p><b>-r-S-----</b> SUID được thiết lập, nhưng quyền chạy của onwer thì không</p>                                   |
| SGID   | <p><b>-rwxrws---</b> quyền chạy và SGID của nhóm được thiết lập</p> <p><b>-rwxr-S---</b> SGID được thiết lập, quyền chạy của nhóm thì không</p>                                         |
| Sticky | <p><b>-rwxrwxrwt</b> quyền chạy và sticky bit của người dùng khác được thiết lập</p> <p><b>-rwxrwxr-T</b> sticky bit được thiết lập, nhưng quyền chạy của người dùng khác thì không</p> |

# Thay đổi người sở hữu và nhóm

- `$ chown [-R] <user> <files>`
  - Thay đổi người sở hữu của file
- `$ chgrp <group> <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

# Một số lệnh hữu dụng khác

- Nén và giải nén
  - gzip/gunzip/zcat, bzip2/bunzip2/bzcat, tar
- Hiển thị/xử lý văn bản
  - head, tail, cut, sort, uniq, tr
- Thao tác với tập tin
  - file, dd, du, df, sync, find, grep
- Một số lệnh shell
  - date, dirname, basename, id, whoami, who, logname,
  - uname, hostname, tee, nohup, su, wc
- Một số lệnh khác
  - which, whereis, whatis, locate
- diff : hiển thị sự khác nhau giữa 2 tập tin.
  - Ex: diff file1 file2 : tìm sự khác nhau giữa file1 & file2.

# gzip/gunzip/zcat - Nén và giải nén tập tin

- Synopsis: `gzip [OPTION] ... [NAME]`
- Option:
  - `-d`: decompress
  - `-c`: write output to stdout, keep original files unchanged
  - `-1`: fastest (less compression)
  - `-9`: slowest (best compression)
- `gunzip = gzip -d`
- `zcat = gunzip -c`
- Ví dụ:
  - `$ gzip file1` : Nén file1 --> kết quả là tệp nén: file1.gz
  - `$ gunzip -c file1.gz` : Giải nén tệp file1.gz



# tar - Nén giải nén thư mục

- Synosis: `tar [Options] ... [<file>, ...] ... [<thư-mục>, ...]`
- Option
  - `-c` : Tạo một tệp lưu trữ mới
  - `-x` : Giải nén các tệp từ tệp lưu trữ
  - `-z` : Sử dụng gzip để nén file
  - `-j` : Sử dụng bzip để nén file
  - `-f filename`: thông báo với tiện ích tar rằng file được tạo có tên là *filename*
  - `v` : Hiển thị danh sách các file trong tiến trình thêm vào tệp lưu trữ
- Ví dụ:
  - `$ tar -czvf dir1.tar.gz dir1/`
  - `$ tar -czvf dir2.tar.gz dir2/`
  - `$ tar -zxvf dir3.tar.gz`

# find - tìm kiếm tập tin

- Synopsis: `find [PATH] ... [EXPRESSION]`
- Ví dụ:
  - `$ find ./ -name "*.txt"`
    - Tìm kiếm các tập tin có phần mở rộng .txt trong thư mục hiện hành
  - `$ find /usr/local -type f :`
    - Tìm kiếm các tập là các file thông thường trong thư mục /usr/local
  - `$ find /usr/local -type d`
    - Tìm kiếm các tập là các thư mục trong thư mục /usr/local
  - `$ find ./ -perm 755 -type f`
    - Tìm kiếm các tập là các file thông thường trong thư mục hiện hành có quyền truy cập 755
  - `$ find / -size 50M`
    - Tìm kiếm các tập trong thư mục /, có kích thước 50M
  - `$ find / -size +50M -size -100M`
    - Tìm kiếm file có dung lượng lớn hơn 50M và nhỏ hơn 100M

# grep - tìm kiếm bên trong nội dung các tập tin

- Synopsis: `grep [OPTION] ... PATTERN ... [FILE]`
- Option
  - `-i`: ignore case distinctions
  - `-n`: prefix line number when output
  - `-r, -R`: recursively through directories
  - `-v`: invert matching, select non-matching lines
- Ví dụ:
  - `$ grep -n nobody /etc/passwd` : Tìm kiếm những dòng có từ nobody trong tệp /etc/passwd, in ra màn hình kết quả và số thứ tự của dòng
  - `$ grep false /etc/passwd`
  - `$ grep -r LANG /etc/X11`