



# FILE AND DIRECTORY MANAGEMENT



# Nội dung



1. Hệ thống file
2. Làm việc với hệ thống file
3. Phân quyền tệp tin



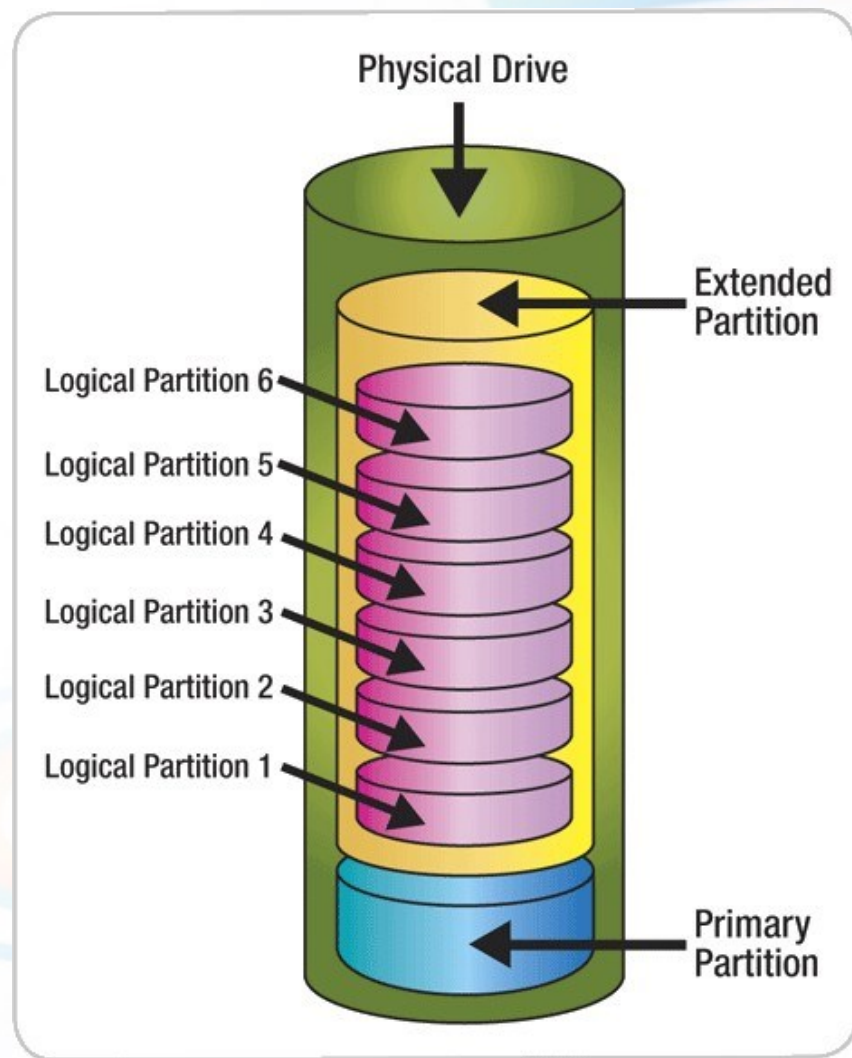


# PHẦN 1: HỆ THỐNG FILE



# Hệ thống file

- Ổ cứng (vật lý)
- Phân vùng (partition)
- Hệ thống file bố trí trên từng phần vùng
- Với linux, các phân vùng ánh xạ vào hệ thống file thống nhất
- Có những hệ thống file khác không tuân theo nguyên tắc này



# Một số hệ thống file thông dụng



Tên	Cỡ file tối đa	Cỡ phần vùng tối đa	Nhật ký thao tác	Ghi chú
Fat16	2 GiB	2 GiB	No	Đã cũ
Fat32	4 GiB	8 TiB	No	Đã cũ
NTFS	2 TiB	256 TiB	Yes	Dùng cho Windows, linux hỗ trợ đọc-ghi
ext2	2 TiB	32 TiB	No	Đã cũ
ext3	2 TiB	32 TiB	Yes	Dùng phổ biến trong đa số hệ thống linux
ext4	16 TiB	1 EiB	Yes	Mới nhất của linux
reiserFS	8 TiB	16 TiB	Yes	Đã dừng phát triển
JFS	4PiB	32PiB	Yes (metadata)	Phát triển bởi IBM
XFS	8 EiB	8 EiB	Yes (metadata)	Phát triển bởi SGI, chú trọng đến sự ổn định

Chú ý: **GiB** = Gibibyte (1024 MiB) / **TiB** = Tebibyte (1024 GiB) /  
**PiB** = Pebibyte (1024 TiB) / **EiB** = Exbibyte (1024 PiB)





# Hệ thống file của linux

- Trên linux, có thể xem các phân vùng hiện tại bằng:

`cat /proc/partitions`

- Hoặc xem thông tin chi tiết hơn (quyền root):

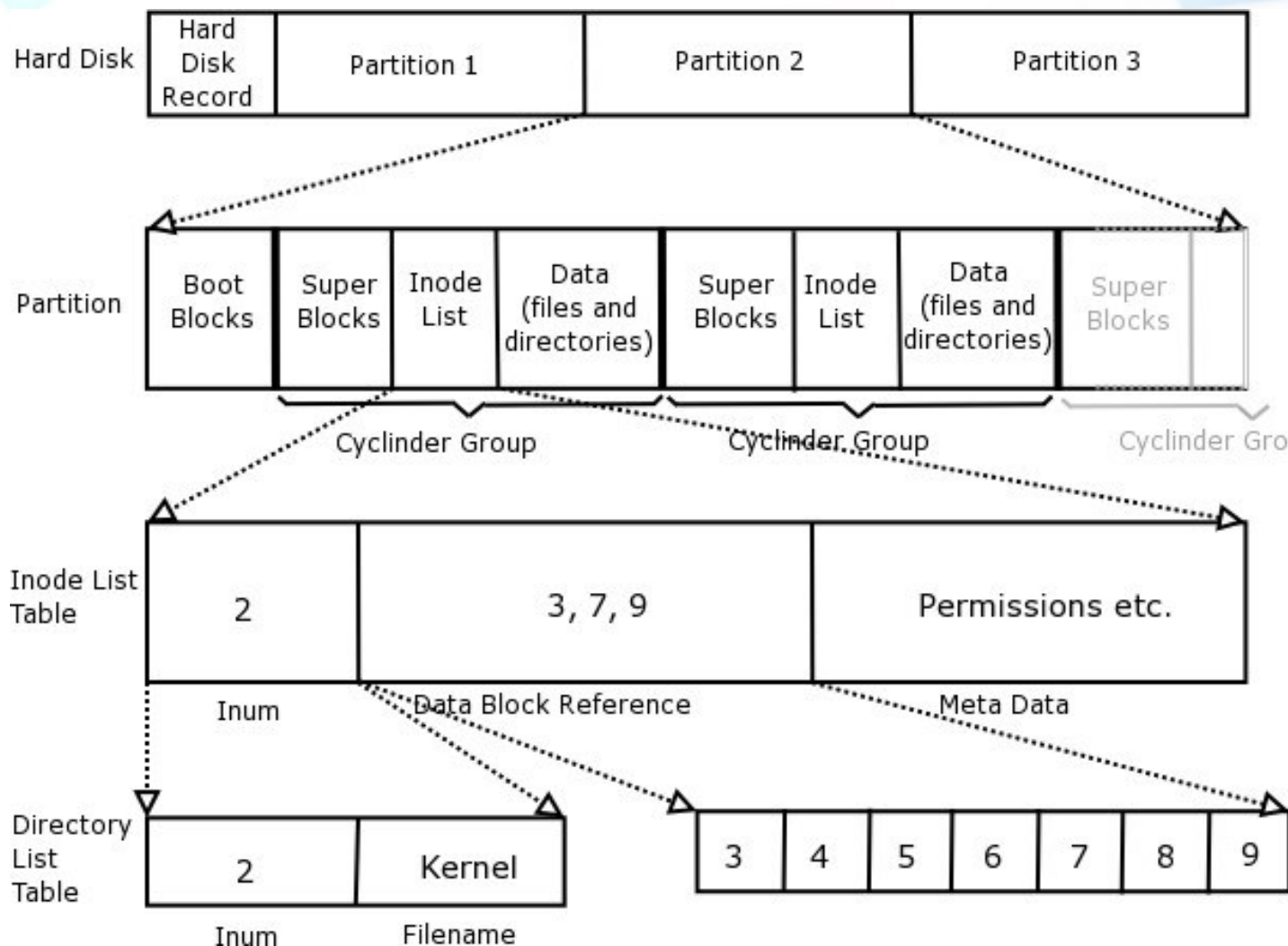
`fdisk -l`

- Một số khái niệm cần biết về cơ chế cấp phát tài nguyên của hệ thống file trên linux

- Super block
- Storage block
- i-node



# Hệ thống file của linux





# Super block và storage block

## ■ Super block:

- Là cấu trúc được tạo tại vị trí bắt đầu hệ thống file
- Lưu trữ các thông tin:
  - Thông tin về block size, free block
  - Thời gian gắn kết (mount) cuối cùng của tập tin
  - Thông tin trạng thái tập tin

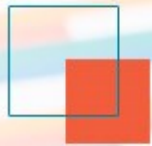
## ■ Storage block:

- Là vùng lưu dữ liệu thực sự của tập tin và thư mục
- Chia thành những data block (thường là 1024 byte)
  - Data block của tập tin lưu i-node và nội dung của tập tin
  - Data block của thư mục lưu danh sách những entry gồm i-node number, tên tập tin và những thư mục con





# i-node



- Lưu những thông tin về tập tin và thư mục được tạo trong hệ thống (nhưng không lưu tên)
- Mỗi tập tin có một i-node lưu thông tin sau:
  - Loại tập tin và quyền hạn truy cập
  - Người sở hữu tập tin
  - Kích thước và số hard link đến tập tin
  - Ngày và giờ chỉnh sửa tập tin lần cuối cùng
  - Vị trí lưu nội dung tập tin trong filesystem

*Một lỗi phổ biến với linux: hệ thống file còn chỗ trống nhưng không tạo được file mới vì hết i-node*



# Gắn kết (mount) hệ thống file



- Khi lắp một thiết bị lưu trữ mới, linux có thể tự động nhận ra thiết bị đó (!) nhưng sẽ không tự động đưa thiết bị đó vào hệ thống file
- Người dùng phải yêu cầu hệ thống ánh xạ thiết bị đó vào một thư mục nào đó trong hệ thống file
- Công việc này gọi là “mount”
- Có thể xem những thiết bị nào đã được mount vào hệ thống và chúng nằm ở đâu bằng “**mount -l**”
- Có thể gỡ thiết bị đã mount bằng lệnh:  
**umount <tên thiết bị hoặc tên đường dẫn>**



# Gắn kết (mount) hệ thống file

- Cú pháp lệnh mount:

`mount [-t type] <device> <directory>`

- Trong đó:

- -t type                      kiểu hệ thống file trên thiết bị
- device                      tên thiết bị vật lý muốn gắn kết
- directory                      tên thư mục muốn ánh xạ tới

- Có thể bỏ qua tham số <directory>

- Trong hầu hết các tình huống, không cần chỉ ra kiểu hệ thống file trên thiết bị muốn gắn kết (hệ thống tự nhận ra)



# Gắn kết (mount) hệ thống file

- Ví dụ: ta có ổ usb có dạng FAT32 đã được hệ thống phát hiện và đặt tên /dev/sdb1, muốn gắn kết ổ này thành thư mục /mnt/usb
- Các bước thực hiện như sau:  
`mkdir /mnt/usb` (nếu chưa tồn tại)  
`mount -t vfat /dev/sdb1 /mnt/usb`
- Sau khi thực hiện những lệnh này, mọi lệnh đọc/ghi vào thư mục /mnt/usb sẽ đọc/ghi vào ổ usb
- Gỡ ổ usb trên: `umount /dev/sdb1` hoặc  
`umount /mnt/usb`



# Tự động gắn kết (auto mount)



- Đôi khi việc gắn kết cần thực hiện ngay khi hệ thống khởi động (chẳng hạn như cần dữ liệu trên ổ đĩa)
- Linux lưu danh sách những thiết bị được gắn kết khi khởi động trong file “/etc/fstab”
- Có thể xem file này bằng lệnh: “`cat /etc/fstab`”
- Nếu chỉnh sửa file này sẽ thay đổi cấu hình tự động gắn kết của hệ thống (đây là cách quản trị viên hay sử dụng)
- Nếu sửa file xong muốn tự động gắn kết luôn (mà không khởi động lại máy) dùng: “`mount -a`”





# File cấu hình /etc/fstab

- Hình chụp một file /etc/fstab thông thường

```
[root@localhost root]# more /etc/fstab
LABEL=/                                /                ext3              defaults          1 1
none                                  /dev/pts          devpts            gid=5,mode=620    0 0
/dev/hda2                             /home             ext3              defaults          1 2
none                                  /proc             proc              defaults          0 0
none                                  /dev/shm          tmpfs            defaults          0 0
/dev/hda3                             swap              swap              defaults          0 0
/dev/fd0                              /mnt/floppy       auto              noauto,owner,kudzu 0 0
/dev/cdrom                            /mnt/cdrom        udf,iso9660      noauto,owner,kudzu,r
0 0 0
```

- Chứa các dòng khai báo thiết bị / phân vùng được mount tự động
- Mỗi thiết bị trên một dòng
- Mỗi dòng có 6 cột tham số



# File cấu hình /etc/fstab



- Cột 1 – **/dev/hda2** – tên phân vùng được kết gán
- Cột 2 – **/home** – ánh xạ đến thư mục /home
- Cột 3 – **ext3** – hệ thống file là ext3 (nên để auto)
- Cột 4 – **defaults** – các lựa chọn mặc định để một phân vùng hoạt động bình thường (rw, suid, dev, exec, auto, nouser, async,...)
- Cột 5 – **1/0** – có sao lưu phân vùng này khi chạy lệnh dump hay không?
- Cột 6 – **1/0** – có cần kiểm tra phân vùng này (bằng lệnh fsck) khi khởi động hay ko? (lớn hơn 1 là có)



# File cấu hình /etc/fstab



- Các tham số ứng với cột 4 (default):
  - **rw** / **ro**: cho đọc ghi (read write) / chỉ đọc (read only)
  - **exec** / **noexec**: cho phép / không cho thực thi các file nhị phân (chẳng hạn như dùng với ổ đĩa của Windows)
  - **auto** / **noauto**: mount tự động / không tự động mount phân vùng tương ứng (khi boot hoặc gõ mount -a)
  - **nouser** / **user**: không cho phép / cho phép các user khác được mount thiết bị ngoài root
  - **async** / **sync**: không / có ghi dữ liệu lên đĩa vật lý ngay khi thực hiện các thao tác ghi dữ liệu (sync nên được dùng với các thiết bị removeable)





# PHẦN 2: LÀM VIỆC VỚI HỆ THỐNG FILE



# Hệ thống file phân cấp



- Hệ thống file trong linux có 2 loại đối tượng chính
  - Tập tin (file): vùng lưu trữ dữ liệu cơ bản
  - Thư mục (directory/folder): chỗ chứa các tập tin và các thư mục khác
- Cấu trúc thư mục theo kiểu cây phân cấp với thư mục gốc (root directory)
- Hầu hết mọi đối tượng trong hệ thống đều được ánh xạ vào hệ thống file
  - Các tài nguyên phần cứng (ổ đĩa cứng, CD, USB,...)
  - Các tiến trình trong hệ thống

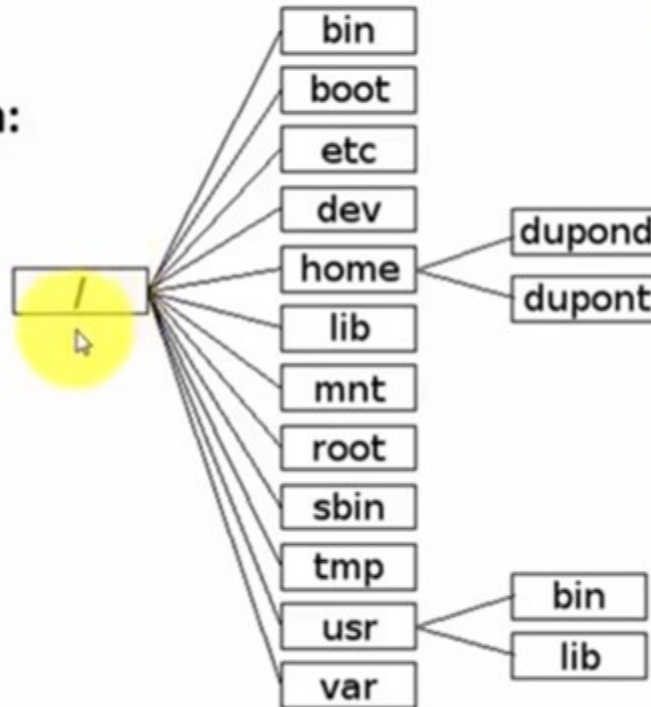




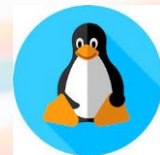
# Hệ thống file phân cấp



Thư mục gốc: /  
Các thư mục con:  
bin, boot, etc...



# Hệ thống file phân cấp



- Thư mục người dùng root: ~

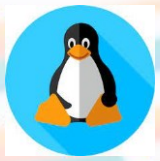
```
[root@localhost ~]# pwd
/root
[root@localhost ~]# _
```

- Thư mục gốc trên file system: /

```
[root@localhost ~]# cd /
[root@localhost /]# pwd
/
[root@localhost /]# ls
bin    dev    home   lib64  mnt    proc   run    srv    thumuc2  usr
boot  etc    lib    media  opt    root   sbin   sys    tmp      var
[root@localhost /]# _
```



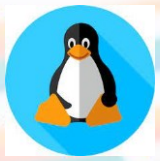
# Các thư mục thông dụng (1/2)



- `/boot` kernel và cấu hình khởi động
- `/bin` các lệnh cơ bản
- `/dev` các thiết bị
- `/etc` cấu hình hệ thống và các ứng dụng
- `/home` dữ liệu của các người dùng
- `/lib` thư viện dùng chung
- `/media` các nguồn dữ liệu (ổ CD/DVD)
- `/mnt` các nguồn dữ liệu (usb, ổ đĩa mạng,...)
- `/opt` các gói phần mềm bổ sung



# Các thư mục thông dụng (2/2)



- `/sbin` file nhị phân hệ thống
- `/srv` dữ liệu cho các dịch vụ server
- `/tmp` thư mục chứa các file tạm thời
- `/usr` các ứng dụng theo người dùng
- `/var` file dữ liệu của ứng dụng
- `/root` thư mục của tài khoản quản trị
- `/proc` thông tin về các tiến trình

***Chú ý:*** vai trò của các thư mục trên chỉ là quy ước, không phải distro nào của linux cũng tuân theo



# Các thư mục thông dụng



- **/bin** và **/sbin**: chứa các file chương trình thực thi (dạng nhị phân) và file khởi động hệ thống
- **/boot**: các file ảnh (image file) của kernel dùng cho quá trình khởi động thường đặt trong thư mục này
- **/dev**: thư mục này chứa các file thiết bị
  - Trong thế giới unix và linux các thiết bị phần cứng được xem như là file
  - Ổ đĩa cứng IDE hoặc SCSI: hda, hdb, sda, sdb,...
  - Các phân vùng trên ổ cứng: hda1, hda2,...
  - Ổ đĩa mềm là fd0, cd là cdrom, sbd là usb,...





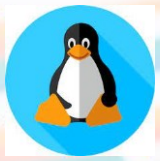
# Các thư mục thông dụng



- **/etc**: thư mục này chứa các file cấu hình toàn cục của hệ thống như quản lý người dùng, cấu hình mạng...
- **/mnt**: thư mục này chứa các thư mục gắn kết tạm thời đến các ổ đĩa hay thiết bị khác. Ta có thể thấy trong /mnt các thư mục con như cdrom (kết gán đĩa cd) hoặc floppy là thư mục kết gán với đĩa mềm
- **/tmp**: thư mục chứa các file tạm mà chương trình sử dụng chỉ trong quá trình chạy, sẽ được hệ thống dọn dẹp nếu không cần dùng đến nữa

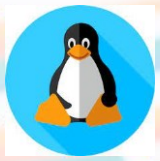


# Các thư mục thông dụng



- **/home**: thư mục này chứa các thư mục con đại diện cho mỗi user khi đăng nhập
  - Mỗi user hoàn toàn có quyền sao chép, xóa file, tạo thư mục con trong thư mục home của mình mà không ảnh hưởng đến các người dùng khác
- **/lib**: thư mục này chứa các file thư viện của hệ điều hành và ứng dụng
  - Các thư viện C và các thư viện liên kết động cần cho chương trình khi chạy và cho toàn hệ thống
  - Thư mục này tương tự thư mục system32 của Windows



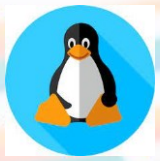


# Các thư mục thông dụng

- **/usr**: thư mục chứa các chương trình cài đặt phục vụ tất cả các user, tương tự Program Files trên Windows
  - Thư mục con **/usr/local** chứa đủ các thư mục tương tự ngoài thư mục gốc như sbin, lib, bin, ...
  - Khi nâng cấp hệ thống thì các chương trình cài đặt trong **/usr/local** vẫn giữ nguyên
- **/var**: thư mục chứa các file làm việc của ứng dụng
  - Thư mục này còn chứa log file để người quản trị xem lại khi hệ thống có lỗi



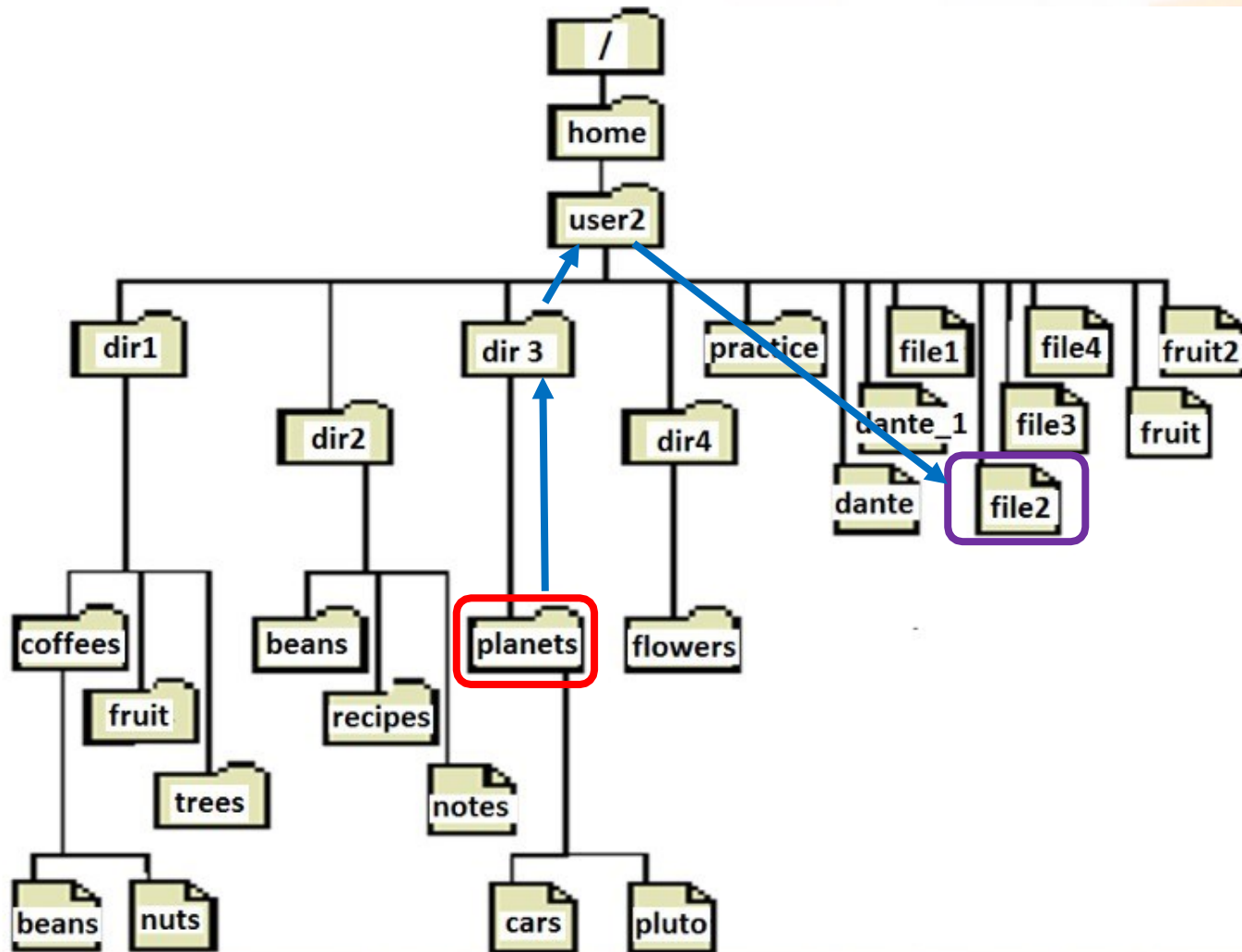
# Đường dẫn



- Đường dẫn (path) đến một vị trí (file/folder) nào đó: có thể hiểu như thứ tự các “bước” để tìm được file/folder đó
  - Định vị file/folder bằng cách chỉ ra vị trí của nó
  - Viết liên tiếp các “bước”, ngăn cách với nhau bởi dấu /
- Đường dẫn tuyệt đối (absolute path): đi từ thư mục gốc đến vị trí cần thiết
- Đường dẫn tương đối (relative path): đi từ thư mục hiện tại đến vị trí đó



# Đường dẫn





# Đường dẫn

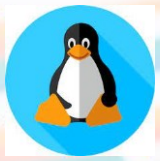


- Vị trí hiện đang làm việc là thư mục “planets”, đường dẫn tuyệt đối đến vị trí hiện tại sẽ là:  
`/home/user2/dir3/planets`
- Tập tin “file2”, đường dẫn tuyệt đối sẽ là:  
`/home/user2/file2`
- Đường dẫn (tương đối) từ vị trí hiện tại đến “file2”:  
`../../file2`

Tên file/folder trong linux phân biệt chữ hoa/thường



# Đường dẫn



```
[root@localhost ~]# cd /etc/sysconfig
[root@localhost sysconfig]# ls
anaconda  ebttables-config  iptables-config  man-db          nftables.conf  sshd
console   firewallld        irqbalance       modules          rsyslog        sshd-permitrootlogin
cpupower  grub              kdump            network          run-parts
crond     ip6tables-config  kernel           network-scripts  selinux
```

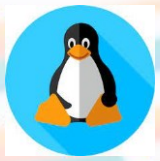
```
[root@localhost sysconfig]# cd console
[root@localhost console]# pwd
/etc/sysconfig/console
```

absolute path?

relative path?



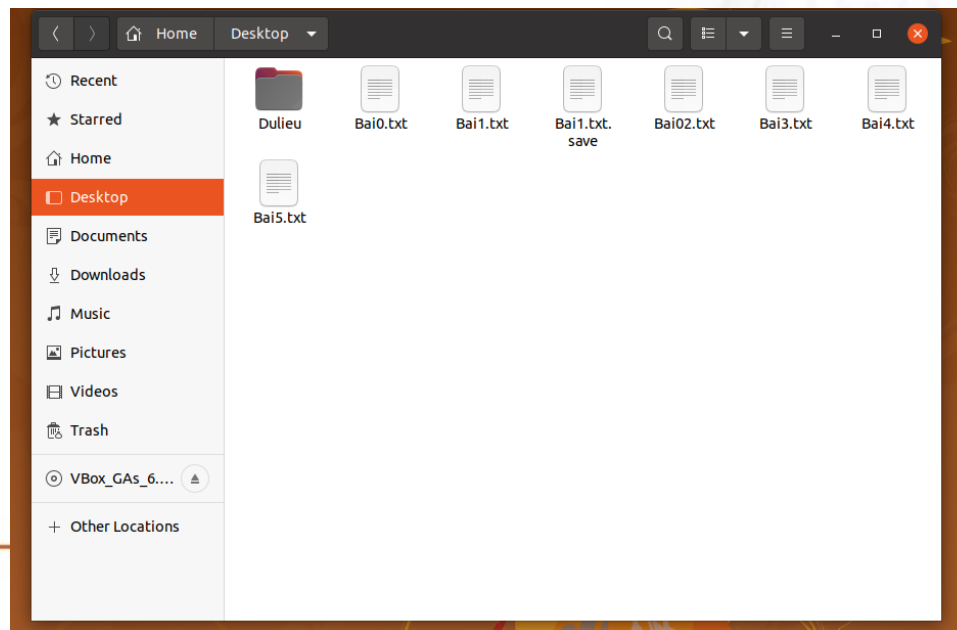
# Làm việc với thư mục



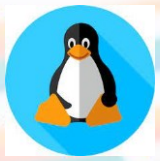
- Khi một người dùng login vào hệ thống, linux gán cho hiện làm việc đó một vị trí làm việc trên hệ thống file (thường là thư mục của người dùng)

```
linux2021@linux2021-VirtualBox: ~/Desktop
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

linux2021@linux2021-VirtualBox:~/Desktop$
```



# Làm việc với thư mục

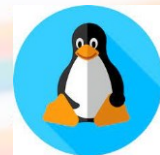


## Các lệnh làm việc với thư mục

Lệnh	Viết tắt	Ý nghĩa
<b>pwd</b>	print work director	Hiển thị đường dẫn đầy đủ của thư mục hiện tại
<b>ls</b>	list	Hiển thị nội dung của thư mục
<b>cd</b>	change directory	Chuyển đến thư mục mà với đường dẫn được cung cấp
<b>mkdir</b>	make directory	Tạo một thư mục với tên là folder
<b>rmdir</b>	remove directories	Xóa thư mục trống



# Lệnh pwd



Hiển thị đường dẫn đầy đủ của thư mục hiện tại (pwd-print work directory)

```
linux2021@linux2021-VirtualBox:~/Desktop$ pwd  
/home/linux2021/Desktop
```





# Lệnh ls



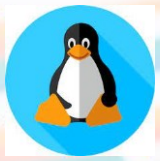
- Liệt kê nội dung của một thư mục
- Cú pháp:

**ls [tùy chọn] [thư mục]**

- Một số tùy chọn:
  - 1: Mỗi file/thư mục trên một dòng
  - x hiển thị trên nhiều cột
  - l hiển thị chi tiết các thông tin của tập tin
  - a hiển thị tất cả các tập tin kể cả tập tin ẩn
- Nếu không chỉ tên thư mục, thì lệnh sẽ liệt kê các file trong thư mục hiện tại
- “ls \*/\*” ?



# Lệnh cd

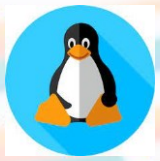


- Thay đổi thư mục làm việc mặc định
- Cú pháp:

Lệnh	Ý nghĩa
<code>cd &lt;Đường dẫn&gt;</code>	Chuyển đến thư mục mà với đường dẫn được cung cấp (cd-change directory)
<code>cd /</code>	Chuyển đến thư mục gốc của ổ cứng (còn gọi là thư mục root)
<code>cd</code> hoặc <code>cd ~</code>	Chuyển đến thư mục home của người dùng (thư mục đăng nhập)
<code>cd ..</code>	Chuyển đến thư mục cha của thư mục hiện hành
<code>cd -</code> <code>cd ../</code>	Chuyển về thư mục trước đó



# Lệnh mkdir



- Tạo thư mục
- Cú pháp:

`mkdir [tùy chọn] <thư mục 1> ... <thư mục n>`

- Một số tùy chọn:

-p  $f_1/f_2/.../f_n$ : tạo một thư mục chứa nhiều thư mục con với điều kiện  $f_1$  đã tồn tại, trong đó  $f_n$  thuộc  $f_{n-1}$ , ...,  $f_2$  thuộc  $f_1$

```
[root@localhost home]# mkdir f1
[root@localhost home]# mkdir f2 f3
[root@localhost home]# mkdir -p f1/f11/f111
[root@localhost home]# ls
63cntt  f2  Genisys  SU0  SU2  SU4  Tin2  Tin62  Tommy
f1      f3  Skyteam  SU1  SU3  Tin1  Tin3  Tin63
[root@localhost home]# cd /home/f1
[root@localhost f1]# ls
f11
```



# Lệnh rmdir

- Xóa thư mục rỗng

- Cú pháp:

`rmdir [tùy chọn] <thư mục 1> ... <thư mục n>`

- Một số tùy chọn:

`-p f1/f2/.../fn`: xóa nhiều thư mục con

Xóa thư mục:

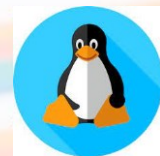
```
linux2021@linux2021-VirtualBox:~/Desktop$ cd Tep1
linux2021@linux2021-VirtualBox:~/Desktop/Tep1$ cd Tep2
linux2021@linux2021-VirtualBox:~/Desktop/Tep1/Tep2$ rmdir Tep3
linux2021@linux2021-VirtualBox:~/Desktop/Tep1/Tep2$
```

Xóa nhiều thư mục con:

```
linux2021@linux2021-VirtualBox:~/Desktop$ rmdir -p Tep1/Tep2/Tep3
```



# Các lệnh làm việc với tệp và thư mục



Lệnh	Viết tắt	Ý nghĩa
<b>cp</b>	copy	Sao chép tệp tin / thư mục
<b>rm</b>	remove	Xóa tệp tin và thư mục
<b>mv</b>	move	Đổi tên hoặc dịch chuyển tệp tin/thư mục





# Lệnh cp



- Sao chép tập tin / thư mục

- Cú pháp:

`cp [tùy chọn] <nguồn> <đích>`

- Một số tùy chọn:

- -f ghi đè không cần hỏi (force)
- -i hỏi trước khi ghi đè (interactive)
- -r sao chép toàn bộ thư mục kể cả con



# Lệnh cp



- Copy thư mục:

```
[root@localhost home]# ls ThumucA
LKA.txt
[root@localhost home]# cp -r thumuc1 ThumucA
[root@localhost home]# ls ThumucA
LKA.txt  thumuc1
[root@localhost home]#
```

- Copy tệp vào tệp:

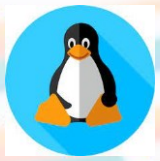
```
[root@localhost home]# cat TepA.txt
1. Nguyen Van An
2. Minh Man Ung
[root@localhost home]# cat TepB.txt
Nguyen Minh Anh
[root@localhost home]# cp -r TepA.txt TepB.txt
cp: overwrite 'TepB.txt'? y
[root@localhost home]# cat TepB.txt
1. Nguyen Van An
2. Minh Man Ung
[root@localhost home]# cat TepA.txt
1. Nguyen Van An
2. Minh Man Ung
[root@localhost home]#
```

- Copy tệp vào thư mục:

```
[root@localhost ~]# ls Thumuc1
[root@localhost ~]# cp Tep2 Thumuc1
[root@localhost ~]# ls Thumuc1
Tep2
[root@localhost ~]#
```



# Lệnh rm



- Xóa tập tin và thư mục (**rm-remove**)

- Cú pháp:

**rm [tùy chọn] <tập tin>**

- Một số tùy chọn:

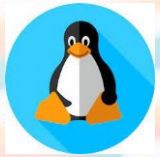
- -f xoá không cần hỏi
- -i hỏi trước khi xoá
- -r xoá toàn bộ thư mục kể cả con

- Lưu ý:

**KHÔNG dùng lệnh: **rm -rf /****



# Lệnh rm

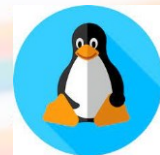


```
[root@localhost thumuc1]# ls
tep1.txt tep2.txt tepA.txt tepB.txt TepB.txt thumuc2 tm11
[root@localhost thumuc1]# rm TepB.txt
rm: remove regular empty file 'TepB.txt'? y
[root@localhost thumuc1]# ls
tep1.txt tep2.txt tepA.txt tepB.txt thumuc2 tm11
[root@localhost thumuc1]# _
```

```
[root@localhost home]# rm -r f2
rm: remove directory 'f2'? y
[root@localhost home]# _
```



# Lệnh mv



- Đổi tên hoặc dịch chuyển tập tin (**mv-move**)

- Cú pháp:

**mv [tùy chọn] <nguồn> <đích>**

- Một số tùy chọn :

- -f ghi đè không cần hỏi (force)
- -i hỏi trước khi ghi đè (interactive)





# Lệnh mv



- Di chuyển:

```
[root@localhost home]# ls
63cntt  home      lkmem     SU1  SU4  TepA.txt  thumuc2  Tin1  Tin62  Tommy
f3      K63_SU1  Skyteam   SU2  Tep1  TepB.txt  Thumuc2  Tin2  Tin63
Genisys Lienket  SU0       SU3  Tep2  thumuc1  ThumucA  Tin3  tm12
[root@localhost home]# ls ThumucA
LKA.txt  thumuc1
[root@localhost home]# mv TepA.txt ThumucA
[root@localhost home]# ls ThumucA
LKA.txt  TepA.txt  thumuc1
[root@localhost home]#
```

- Sửa tên:

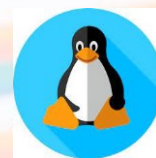
```
[root@localhost ~]# ls
anaconda-ks.cfg  Tep2  Tep4  Tep6  TepB.txt  Thumuc2  Thumuc4  Thumuc5
Tep1             Tep3  Tep5  TepA.txt  Thumuc1  Thumuc3  Thumuc5
[root@localhost ~]# mv Tep4 Tep4A
[root@localhost ~]# ls
anaconda-ks.cfg  Tep2  Tep4A  Tep6  TepB.txt  Thumuc2  Thumuc4  Thumuc5
Tep1             Tep3  Tep5  TepA.txt  Thumuc1  Thumuc3  Thumuc5
[root@localhost ~]#
```



# PHẦN 3: PHÂN QUYỀN TẬP TIN



# Cấu trúc tệp tin



Phân loại file

các quyền truy cập cho chủ

Các quyền truy cập cho nhóm chủ

Các quyền truy cập cho người dùng bên ngoài

Kích thước của file

Thời điểm file được tạo ra

1 /home/user # ls -l

2 total 36

3	drwxr-xr-x	3	root	root	4096	2006-06-29 04:21	.
4	drwxr-xr-x	7	root	root	4096	2006-06-23 02:13	..
5	-rwxr-xr-x	1	root	root	6096	2006-06-22 09:26	functions
6	-rw-r--r--	2	anon	users	651	2006-06-23 05:23	hardlink
7	-rw-r--r--	2	anon	users	651	2006-06-23 05:23	mark.txt
8	drwxr-xr-x	2	root	root	4096	2006-06-22 09:27	mydir
9	brw-rw----	1	root	disk	8, 192	2005-05-24 08:09	sdm
10	-rwsr-sr-x	1	root	root	6096	2006-06-22 09:29	share
11	lrwxrwxrwx	1	root	root	9	2006-06-22 09:28	softlink -> functions
12	-rw-r--r--	1	root	root	0	2006-06-29 04:21	zerobyte.txt

Tổng số liên kết  
đến cùng 1 file

Tên nhóm chủ của tệp tin

Tài khoản cá nhân chủ tệp tin

Tên tệp tin

Kiểm tra thuộc tính của các tệp tin bằng lệnh ls -l



# Kiểu tệp tin



Kiểu tập tin	Quyền tập tin	Số liên kết	Chủ nhân	Tên nhóm	Kích thước (byte)	Thời điểm sửa đổi sau cùng	Tên tập tin
	-rw-r--r--	1	chris	weather	207	Feb 20 11:55	mydata

## Kí tự đầu: Xác định kiểu tệp tin

- : Tệp tin thông thường (ordinary file)

d : Tệp tin thư mục (directory file)

l : Liên kết

c : Special file hoặc file thiết bị ký tự (character device file)

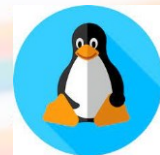
s : Socket

p : Named pipe

b : File thiết bị khối (block device file)



# Quyền người dùng

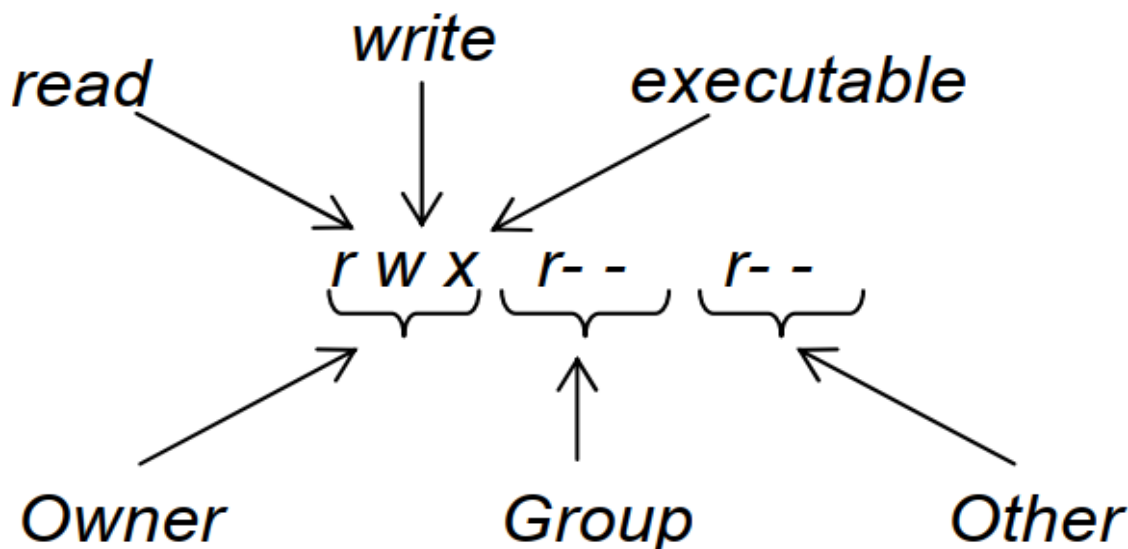


## Kiểu người dùng trên file:

- + **u**: Người sở hữu file (owner user) là người tạo ra file, nhóm
- + **g**: Người dùng file (group) là những người cùng nhóm với người sở hữu
- + **o**: Những người dùng khác (other)
- + **a**: all, tất cả user (u, g và o)

## Quyền truy cập

- + **r**: Quyền đọc (read), mã quyền là 4
- + **w**: Quyền ghi (write), mã quyền là 2
- + **x**: Quyền chạy (execute), mã quyền là 1





# Liên kết



- Linux cho phép tạo ra các liên kết file nhờ đó ta có thể có nhiều tên file tại các thư mục khác nhau tham chiếu tới cùng một file vật lý
- Có hai loại liên kết:
  - Hard link
  - Symbolic link (còn gọi là soft link)



# Tên tập tin



- Độ dài tên file tối đa 255 ký tự, linux hỗ trợ khá tốt việc sử dụng unicode trong tên file, nhưng nhiều phần mềm hỗ trợ không tốt lắm
- Có thể sử dụng bất kỳ ký tự nào (kể cả ký tự đặc biệt – thường bị cấm trên các hệ thống khác)

"very ? long - file + name.test"

- Tập tin / thư mục ẩn được bắt đầu bằng một dấu "."  
    .bash\_history   .bash\_profile   .bashrc  
    .desktop/       .kde/           .mozilla/



# Phân quyền truy cập

- Phân quyền file cho nhóm người dùng (**Change Group**)
- Cú pháp:

`chgrp <Group_name> <File_name>`

```
-rwxr-xr-x. 1 root    root          35 Oct 28 16:07 TepA.txt
```

```
[root@localhost home]# chgrp Linux63CNTT TepA.txt
[root@localhost home]# ls -l
total 8
drwx-----. 2 63cntt  63cntt          62 Oct 25 15:47 63cntt
drwxr-xr-x.  2 root    root           6 Oct 27 14:50 f3
drwx-----. 2      1007      1011       62 Oct 26 15:34 Genisys
drwx-----. 2 K63_SU1 K63_SU1       62 Oct 29 23:19 K63_SU1
drwx-----. 2      1008 Tommy          62 Oct 26 15:37 Skyteam
drwx-----. 2 SU3      SU3           62 Oct 26 10:21 SU0
drwx-----. 2 Tin62    SU0           62 Oct 26 10:00 SU1
drwx-----. 2 SU0      SU3           62 Oct 26 10:01 SU2
drwx-----. 2 SU3      SU3           62 Oct 26 10:31 SU3
drwx-----. 2 SU4      SU4           62 Oct 26 10:33 SU4
-rwxr-xr-x.  1 root    Linux63CNTT 35 Oct 28 16:07 TepA.txt
```

# Phân quyền truy cập

- Phân quyền file cho người dùng (**chown-Change Owner**)
- Cú pháp:

`chown <User_name> <File_Name>`

```
-rwxr-xr-x. 1 root Linux63CNTT 35 Oct 28 16:07 TepA.txt
```

```
[root@localhost home]# chown K63_SU1 TepA.txt
[root@localhost home]# ls -l
total 8
drwx-----. 2 63cntt 63cntt      62 Oct 25 15:47 63cntt
drwxr-xr-x. 2 root    root        6 Oct 27 14:58 f3
drwx-----. 2      1007      1011  62 Oct 26 15:34 Genisys
drwx-----. 2 K63_SU1 K63_SU1    62 Oct 29 23:19 K63_SU1
drwx-----. 2      1008 Tommy    62 Oct 26 15:37 Skyteam
drwx-----. 2 SU3      SU3        62 Oct 26 10:21 SU0
drwx-----. 2 Tin62    SU0        62 Oct 26 10:00 SU1
drwx-----. 2 SU0      SU3        62 Oct 26 10:01 SU2
drwx-----. 2 SU3      SU3        62 Oct 26 10:31 SU3
drwx-----. 2 SU4      SU4        62 Oct 26 10:33 SU4
-rwxr-xr-x. 1 K63_SU1 Linux63CNTT 35 Oct 28 16:07 TepA.txt
```

# Phân quyền truy cập

- Phân quyền file cho người dùng, nhóm
- Cú pháp:

`chown <User_name>.<Group_name> <File Name>`

`chown <User_name>:<Group_name> <File Name>`

```
[root@localhost Thumuc1]# ls -l
total 0
-rw-r--r--. 1 root root 0 Nov  5 08:53 Tep2
-rw-r--r--. 1 root root 0 Nov  1 22:57 Tep4A
[root@localhost Thumuc1]# chown Tommy:Tommy Tep2
[root@localhost Thumuc1]# ls -l
total 0
-rw-r--r--. 1 Tommy Tommy 0 Nov  5 08:53 Tep2
-rw-r--r--. 1 root root 0 Nov  1 22:57 Tep4A
[root@localhost Thumuc1]#
```





# Thay đổi quyền truy cập

- Thay đổi chế độ truy cập quyền (**chmod-Change Mode**) của một hoặc nhiều tệp.
- Chỉ một chủ sở hữu của tệp hoặc người dùng có đặc quyền có thể thay đổi chế độ truy cập
  - Mode (Chế độ)
    - Có thể là số hoặc biểu thức ở dạng cho phép ai chọn mã
    - Mode ứng với quyền của user/group/other
- Tác vụ trên quyền truy cập:
  - **+**: Dùng để thêm quyền
  - **-**: Dùng để giảm quyền
  - **=**: Dùng để gán quyền

Cú pháp:

**chmod [op] <Mode> <File Name>**

Trong đó, op -R: Thay đổi cả trong thư mục



# Thay đổi quyền truy cập



a) Đặt quyền bằng ký hiệu quyền

Cú pháp:

`chmod kiểu_người_dùng+quyền <tên_file>`

`chmod kiểu_người_dùng-quyền <tên_file>`

`chmod kiểu_người_dùng=quyền <tên_file>`

Chú ý: Nếu muốn gán quyền cho tất cả các người dùng ta có thể dùng ký tự a (all) hoặc để trống tại kiểu\_người\_dùng.

```
[root@localhost homel]# chmod +x TepA.txt
[root@localhost homel]# ls -l
total 4
drwx-----. 2 63cntt 63cntt 62 Oct 25 15:47 63cntt
drwxr-xr-x. 2 root root 6 Oct 27 14:50 f3
drwx-----. 2 1007 1011 62 Oct 26 15:34 Genisys
drwx-----. 2 1008 Tommy 62 Oct 26 15:37 Skyteam
drwx-----. 2 SV3 SV3 62 Oct 26 10:21 SV0
drwx-----. 2 Tin62 SV0 62 Oct 26 10:00 SV1
drwx-----. 2 SV0 SV3 62 Oct 26 10:01 SV2
drwx-----. 2 SV3 SV3 62 Oct 26 10:31 SV3
drwx-----. 2 SV4 SV4 62 Oct 26 10:33 SV4
-rwxr-xr-x. 1 root root 35 Oct 28 16:07 TepA.txt
```



# Thay đổi quyền truy cập



## b) Đặt quyền tuyệt đối bằng mã nhị phân

- Do mỗi kiểu người dùng có 3 quyền lần lượt là r, w và x nên quyền tuyệt đối của một người dùng gồm 3 bit:

+ Nếu tại vị trí có giá trị 0 thì quyền tại đó bị hạn chế

+ Nếu có giá trị 1 thì quyền tại đó là được phép.

- Dãy liên tiếp gồm 9 bit hay 3 số ở hệ bát phân chính là tập quyền phân cho cả ba kiểu người dùng.

Octal	0	1	2	3	4	5	6	7
Binary	000	001	010	011	100	101	110	111

3    2    7 (octal)  
↓   ↓   ↓  
011 010 111 (binary)

U: -wx

G: -w-

O: rwx

Read= 4, write=2, execute=1



# Thay đổi quyền truy cập

```
[root@localhost home]# chmod 755 TepA.txt
[root@localhost home]# ls -l
total 4
drwx-----. 2 63cntt 63cntt 62 Oct 25 15:47 63cntt
drwxr-xr-x. 2 root root 6 Oct 27 14:50 f3
drwx-----. 2 1007 1011 62 Oct 26 15:34 Genisys
drwx-----. 2 1008 Tommy 62 Oct 26 15:37 Skyteam
drwx-----. 2 SV3 SV3 62 Oct 26 10:21 SV0
drwx-----. 2 Tin62 SV0 62 Oct 26 10:00 SV1
drwx-----. 2 SV0 SV3 62 Oct 26 10:01 SV2
drwx-----. 2 SV3 SV3 62 Oct 26 10:31 SV3
drwx-----. 2 SV4 SV4 62 Oct 26 10:33 SV4
-rwxr-xr-x. 1 root root 35 Oct 28 16:07 TepA.txt
```

```
[root@localhost home]# chmod 544 TepA.txt
[root@localhost home]# ls -l
total 4
drwx-----. 2 63cntt 63cntt 62 Oct 25 15:47 63cntt
drwxr-xr-x. 2 root root 6 Oct 27 14:50 f3
drwx-----. 2 1007 1011 62 Oct 26 15:34 Genisys
drwx-----. 2 1008 Tommy 62 Oct 26 15:37 Skyteam
drwx-----. 2 SV3 SV3 62 Oct 26 10:21 SV0
drwx-----. 2 Tin62 SV0 62 Oct 26 10:00 SV1
drwx-----. 2 SV0 SV3 62 Oct 26 10:01 SV2
drwx-----. 2 SV3 SV3 62 Oct 26 10:31 SV3
drwx-----. 2 SV4 SV4 62 Oct 26 10:33 SV4
-r-xr--r--. 1 root root 35 Oct 28 16:07 TepA.txt
```



# Thay đổi quyền truy cập



## Ví dụ:

- g+w
- o-rwx
- +x
- a+rw
- ug+r
- o=x

Thêm quyền ghi cho group

Loại bỏ tất cả các quyền của others

Thêm quyền thực thi cho tất cả

Thêm quyền đọc và ghi cho tất cả

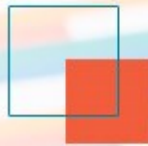
Thêm quyền đọc cho user và group

Chỉ cho phép thực thi với others





# Thay đổi quyền truy cập



Hãy giải thích các câu lệnh dưới đây

```
chmod -x *.php
chmod -R ug+rw lecture
chmod u=rwx,ug=r desktop.jpg
chmod 644 homelist.txt
chmod 755 myprogram
chmod 777 /tmp/tmp
chmod -R 777
```



# Tạo liên kết



## a) Liên kết cứng (Hard Link)

- Các file liên kết (**In-link**) cùng tham chiếu tới một file vật lý và cùng có một chỉ số inode, có cùng số liên kết, cùng kích cỡ...

### - Ưu điểm:

- + Việc thay đổi nội dung trong một file liên kết cũng chính là thay đổi nội dung của các file liên kết còn lại.

- + Nếu xóa đi một trong các liên kết thì nội dung file vật lý không mất đi, ta vẫn có thể truy nhập vào file đó thông qua các liên kết khác.

- **Nhược điểm:** Không thể tạo liên kết cứng cho một file gốc mà liên kết đó và file gốc thuộc hai hệ thống file khác nhau.

### Cú pháp:

**In <Tên file gốc> <Tên file liên kết>**



# Tạo liên kết



## a) Liên kết cứng (Hard Link)

- Tạo liên kết hai file TepA.txt và LK.txt trên cùng thư mục. Hai file này tham chiếu tới cùng một file vật lý.

```
[root@localhost Lienket]# ls -l
total 4
-rw-r--r--. 1 root root 10 Oct 30 16:02 TepA.txt
[root@localhost Lienket]# ln TepA.txt LK.txt
[root@localhost Lienket]# ls -l
total 8
-rw-r--r--. 2 root root 10 Oct 30 16:02 LK.txt
-rw-r--r--. 2 root root 10 Oct 30 16:02 TepA.txt
[root@localhost Lienket]# _
```



# Tạo liên kết



## a) Liên kết cứng (Hard Link)

- Tạo liên kết hai file TepA.txt trên thư mục /home/Lienket và LKA.txt trên thư mục /home/ThumucA. Hai file này tham chiếu tới cùng một file vật lý.

+ Trên thư mục /home/Lienket

```
[root@localhost Lienket]# mkdir /home/ThumucA
[root@localhost Lienket]# ln TepA.txt /home/ThumucA/LKA.txt
[root@localhost Lienket]# ls -l
total 8
-rw-r--r--. 3 root root 10 Oct 30 16:02 LK.txt
-rw-r--r--. 3 root root 10 Oct 30 16:02 TepA.txt
[root@localhost Lienket]# _
```

+ Trên thư mục /home/ThumucA

```
[root@localhost Lienket]# cd /home/ThumucA
[root@localhost ThumucA]# ls -l
total 4
-rw-r--r--. 3 root root 10 Oct 30 16:02 LKA.txt
[root@localhost ThumucA]# _
```



# Tạo liên kết



## b) *Liên kết mềm (Symbolic Link)*

- Lưu đường dẫn đến file mà nó liên kết tới.
- Chỉ chứa thông tin về vị trí của file được liên kết.
- Để xóa hẳn một file, chỉ cần xóa tất cả các liên kết cứng tới nó. Tuy nhiên nếu tồn tại một liên kết mềm tới một file đã bị xóa thì liên kết này sẽ không thể truy nhập được file.
- Dùng liên kết mềm để tạo liên kết tới một thư mục.

Cú pháp:

`ln -s <Tên file gốc> <Tên file liên kết>`





# Tạo liên kết

## b) Liên kết mềm (Symbolic Link)\

Tạo file liên kết tên là TepB cho file Tep1

```
[root@localhost ~]# ln -s /home/lkmem/Tep1.txt TepB.txt
[root@localhost ~]# ls -l
total 8
-rw-----. 1 root root 1135 Oct 25 15:48 anaconda-ks.cfg
-rw-r--r--. 1 root root  10 Oct 31 21:33 TepA.txt
lrwxrwxrwx. 1 root root  20 Oct 31 21:35 TepB.txt -> /home/lkmem/Tep1.txt
[root@localhost ~]# cd /home/lkmem
[root@localhost lkmem]# ls -l
total 8
-rw-r--r--. 1 root root 24 Oct 31 10:22 Tep1.txt
-rw-r--r--. 1 root root 10 Oct 31 10:22 Tep2.txt
[root@localhost lkmem]# _
```



