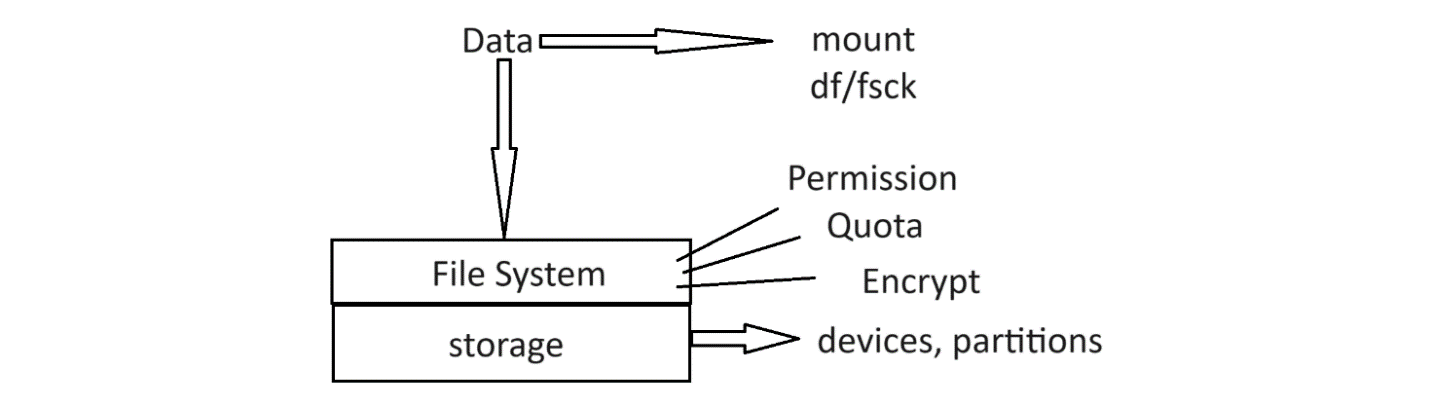
Bài 7: Partition, File System, Permission, Quota và Users



1. Partitions

***Quản lý thiết bị***

Để quản lý thiết bị ta cần hiểu tên logic của thiết bị được đặt như thế nào và công cụ để quản lý

Với các thiết bị ổ cứng SCSI, SAS, SSD thì tên logic của các ổ và phân vùng của chúng được đặt như sau: sd | a(b,c,d,...) | 1(2,3,...)

Trong đó: sd là chỉ loại ổ dạng SCSI, SAS, SSD

a(b,c,d,...) chỉ thứ tự của ổ đĩa

1(2,3,...) chỉ thứ tự phân vùng trong mỗi ổ

Các ổ cứng IDE cũ cũng có cách đặt tên logic tương tự như vậy nhưng thay sd bằng hd.

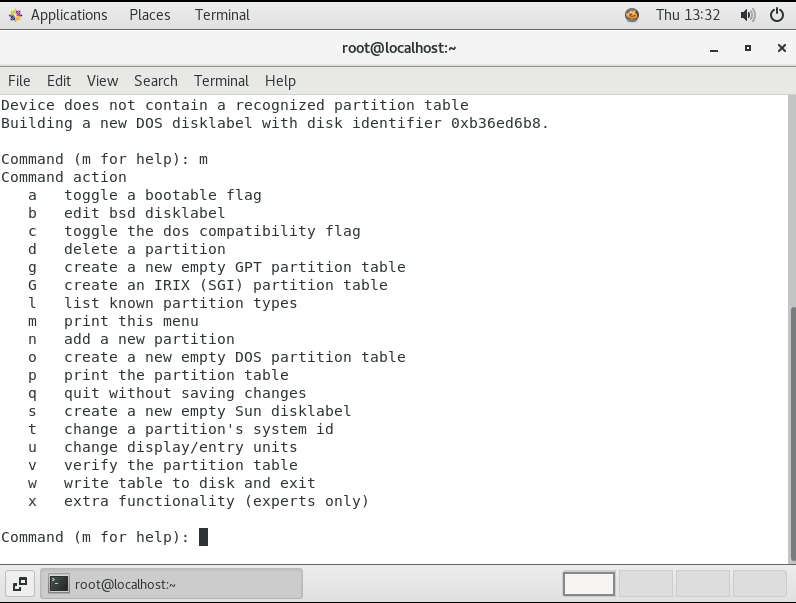
Để quản lý thiết bị thì có nhiều công cụ, với Linux thì ta thường sử dụng tool lsblk.

***Phân vùng ổ đĩa***

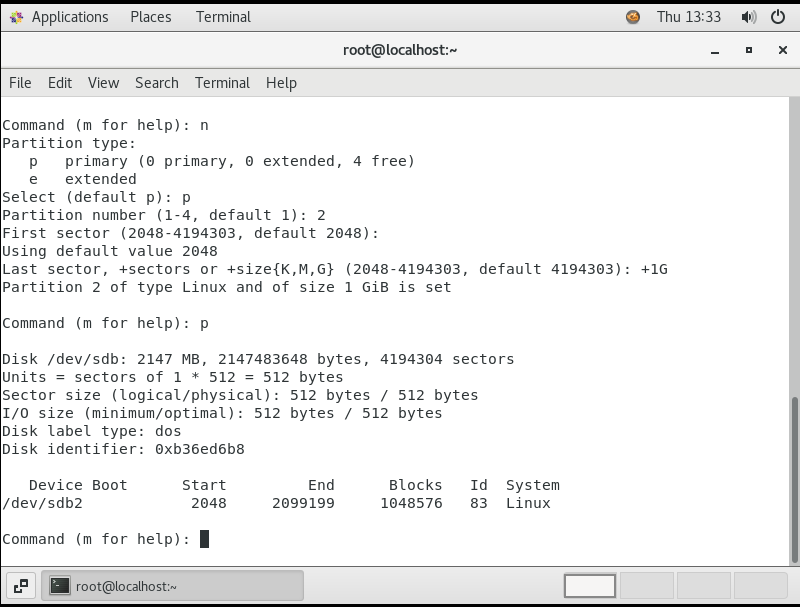
Ta có hai chuẩn để quản lý phân vùng là chuẩn MBR với công cụ fdisk và chuẩn GPT với công cụ parted. Ở đây ta chỉ làm với chuẩn MBR, tức là quản lý phân vùng các ổ đĩa truyền thống, partition giới hạn 2TB trở lại.

Giả định ta đã có một ổ đĩa được kết nối với hệ thống với tên logic là /dev/sdb, ta tiến hành phân vùng ổ đĩa đó như sau:

+ fdisk /dev/sdb



+ Sau khi nhập lệnh trên, màn hình chuyển tới trang lựa chọn của fdisk, lúc này ta gõ m để hiển thị các hướng dẫn, gõ n để bắt đầu tiến hành phân vùng



+ Sau khi gõ n, màn hình hiện lên hai lựa chọn là extend (nút e) và primary partition (nút p)

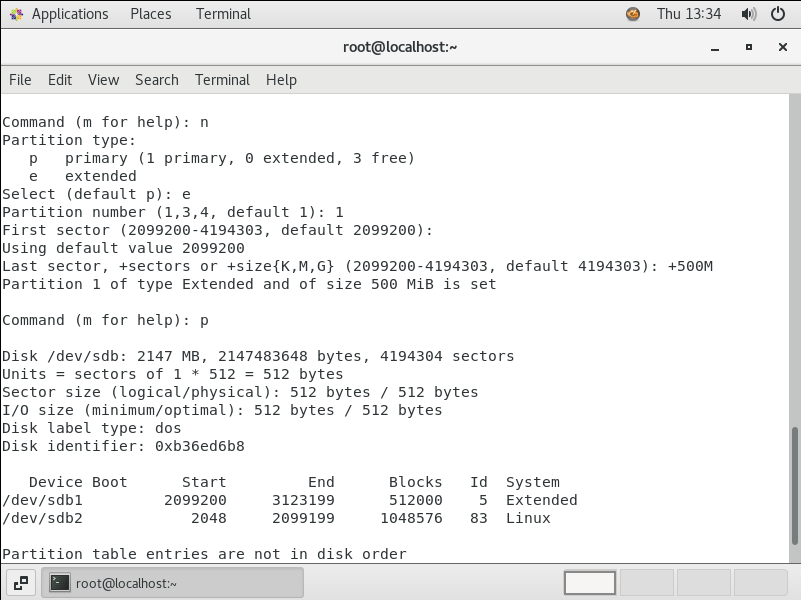
+ Tạo primary partition: p

+ Ở đây chọn partition number 2 (đây là ví dụ, có thể chọn số khác từ 1-4)

+ First sector: để default không nhập gì cả

+ Last sector: ghi +1G (nghĩa là 1GB)

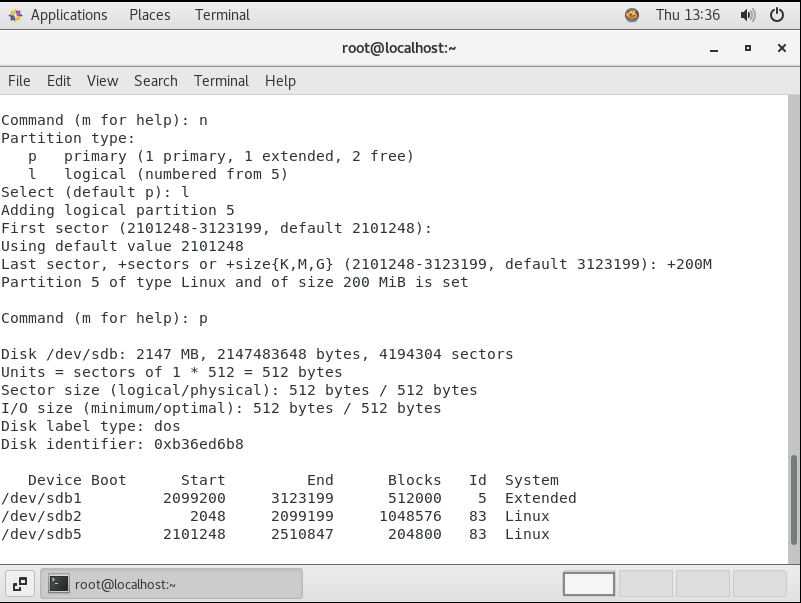
+ Sau đó là quá trình sizing partition, các bạn theo dõi toàn bộ hướng dẫn hiển thị trên màn hình máy tính của mình để làm theo. Để in ra bảng partition: p



+ Tạo extended để tạo thêm logical partitions

+ Chọn extended: e

+ Làm tương tự trên: extended partition này có dung lượng 500MB

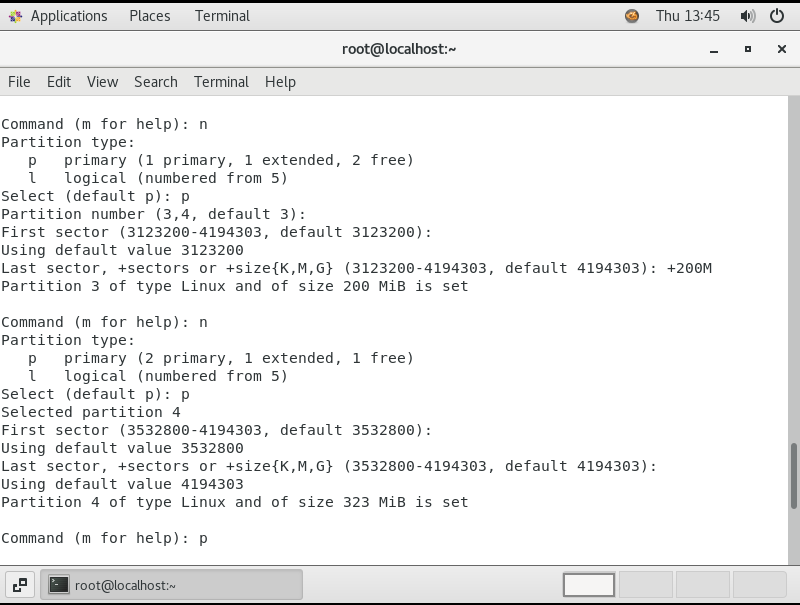


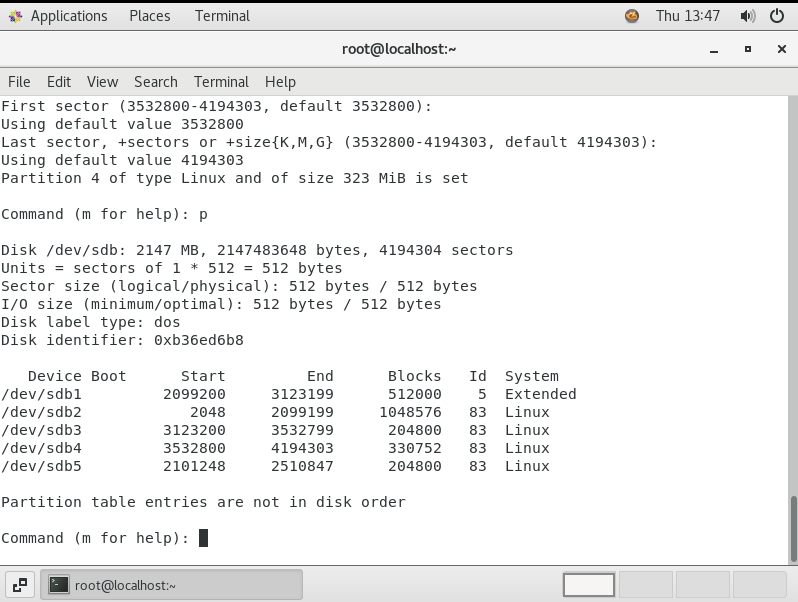
+ Tạo logical partition từ extended: l

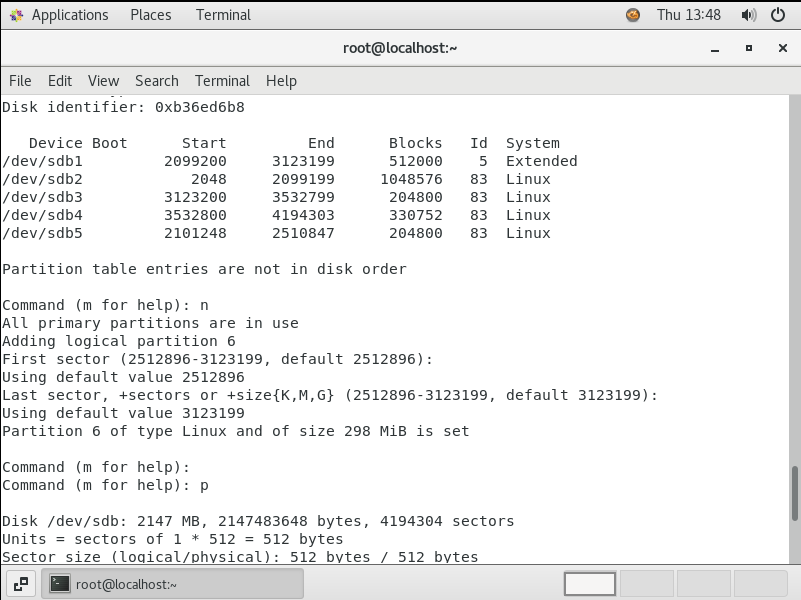
+ Các logical partitions sẽ được đánh số từ 5 trở đi, ở đây đang sizing partition số 5

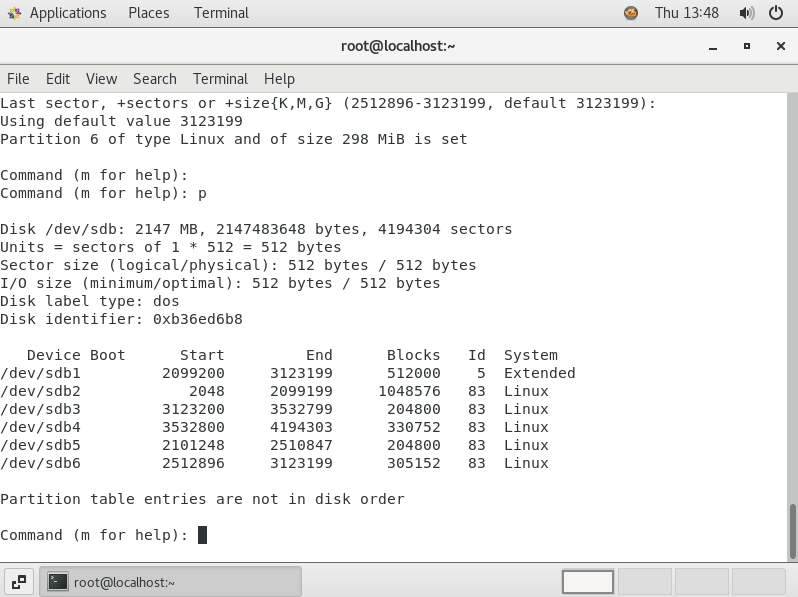
+ Làm tương tự trên, partition này có dung lượng 200MB

Các phân vùng còn lại làm như các bước dưới đây

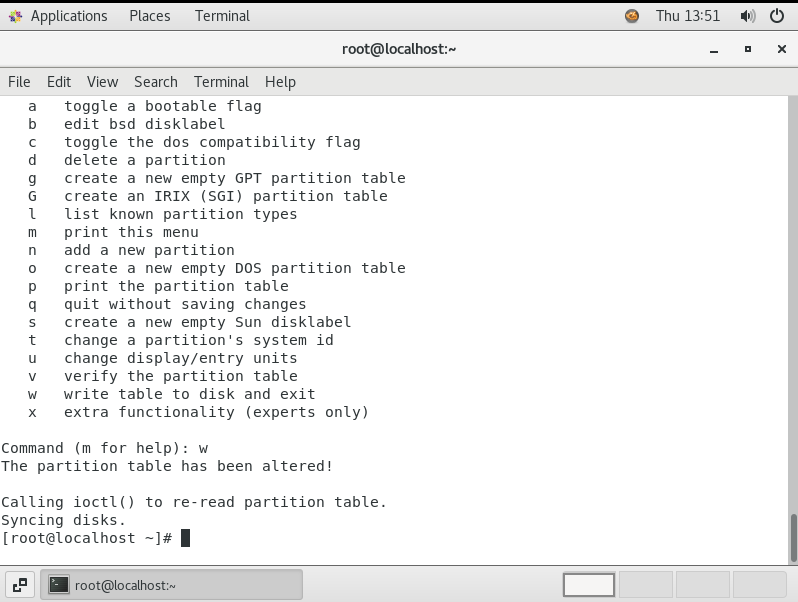




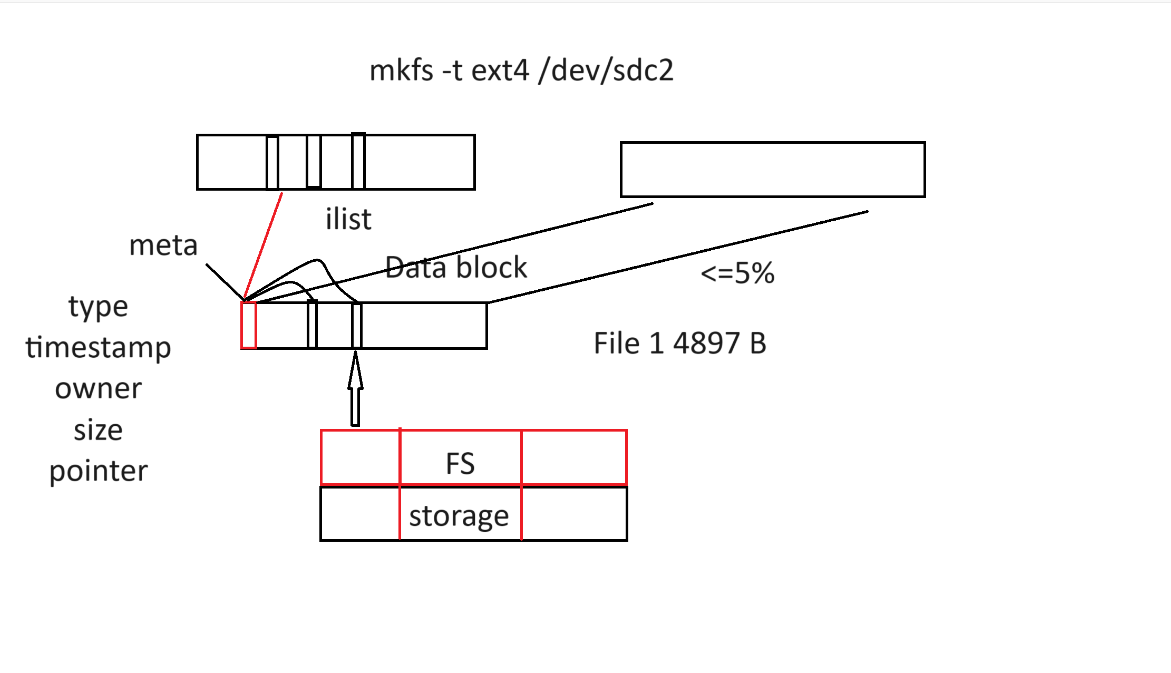




+ Sau khi phân vùng xong, tiến hành ghi bảng phân vùng xuống ổ đĩa và thoát: w



2. File system

****

Sau khi tạo ra các phân vùng để quản lý thiết bị thì ta cần phải có File System trên từng phân vùng đó.

Mỗi FS sẽ có hai phần dữ liệu tương ứng là meta và data

+Meta: đây là một không gian lưu trữ thông tin, cấu trúc của FS, các thông tin thuộc tính của một file như là type, timestamp, owner, size, pointer. Dung lượng của meta chiếm nhiều nhất là 5% dung lượng của cả FS.

+ Datablock: là không gian lưu trữ dữ liệu

!Pointer là con trỏ dữ liệu, nó sẽ trỏ đến các datablock lưu trữ dữ liệu của file tương ứng.

Và bên trong meta cũng chia ra làm các không gian lưu trữ nhỏ hơn, gọi là inodes. Mỗi không gian inode sẽ lưu trữ thông tin thuộc tính của file mà inode đó trỏ pointer đến. Các inodes được đánh số từ 1, cách đánh số này được lưu trong một danh sách gọi là ilist.

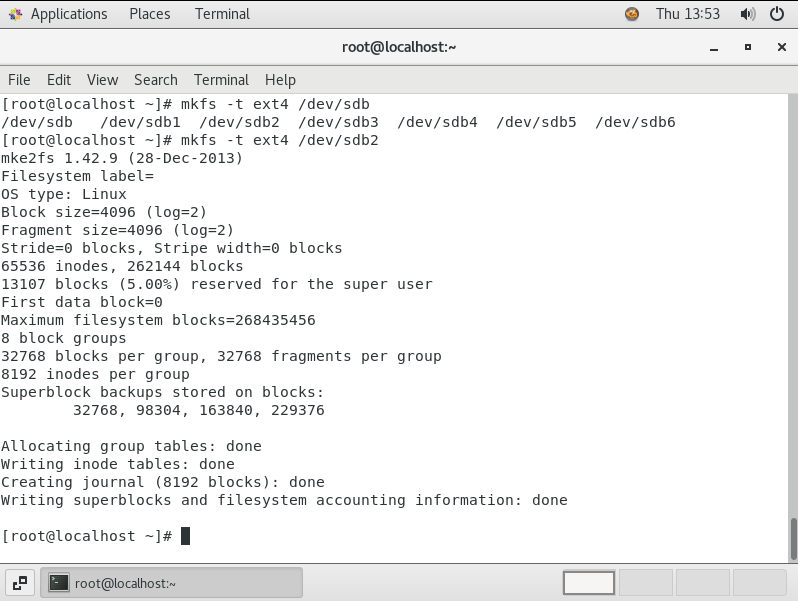
Ta hình dung meta giống như là một mục lục của cuốn sách và datablock là các trang sách. Dữ liệu meta lại được backup trong một số datablock ở trong data, những block này được gọi là superblock.

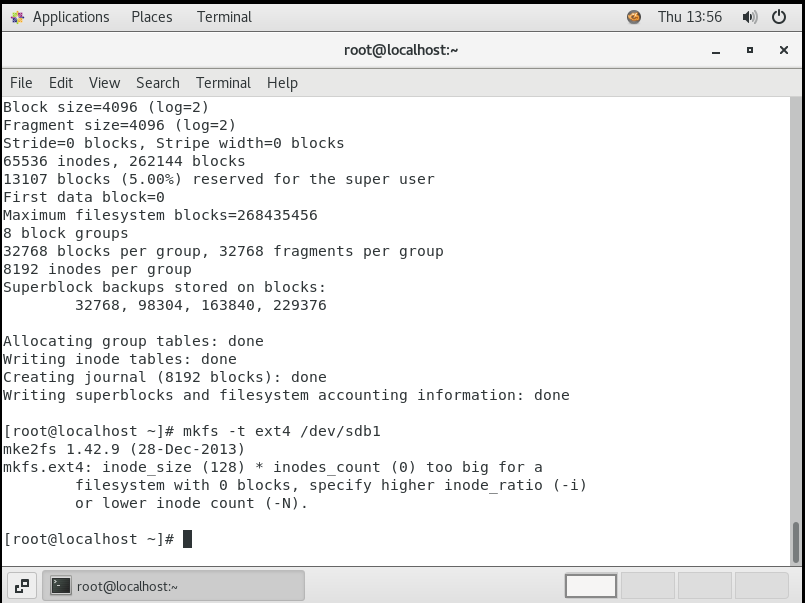
Để quản lý và tạo FS trên Linux, ta sử dụng công cụ mkfs như sau

Ví dụ: Ta có một phân vùng /dev/sdb2 nào đó trên hệ thống vừa mới được phân vùng, lúc này ta cần phải tạo FS cho phân vùng đó để thực hiện việc ghi dữ liệu lên phân vùng

mkfs –t ext4 /dev/sdb2

Với extended partition sẽ không thể tạo FS. Bởi vì các phân vùng này không được phép sử dụng thiết bị để lưu dữ liệu, mà phụ trách việc cấp phát tài nguyên cho các logical partitions để lưu dữ liệu. Bên dưới là ví dụ:





Có thể thấy hệ thống thông báo rằng “quá lowns cho một filesystem với 0 blocks dữ liệu”, bởi vì extended sinh ra không phải để lưu dữ liệu mà là để phân chia tài nguyên bộ nhớ cho các logical partitions.

Như vậy để có thể sử dụng một thiết bị để ghi, lưu trữ dữ liệu, ta cần phải có FS trên mỗi phân vùng của thiết bị đó.

Các lệnh để moniter FS:

df –h/-i (xem size/xem inodes)

du –sh (xem size thư mục)

...

***Quản lý files và thư mục***

!Bản chất file và thư mục đều là các files, vì bản chất chúng đều được định nghĩa bởi các inodes và blocks dữ liệu. Sự khác nhau nằm ở cấu trúc dữ liệu của chúng.

+ Thư mục quan trọng nhất trên hệ thống và cần ưu tiên backup đầu điên đó chính là /etc. Lý do là vì thư mục này lưu toàn bộ cấu hình hiện trạng của hệ thống, nếu có sự cố xảy ra, ta vẫn có thể khôi phục lại hệ thống về hiện trạng nếu ta có backup của /etc.

+ /bin và /sbin: hai thư mục này chứa những chương trình, file chạy, các câu lệnh cho người dùng sử dụng. /bin sẽ chữa những câu lệnh hầu hết người dùng có thể chạy được, còn /sbin thì chứa những câu lệnh chỉ root mới có quyền chạy.

+ /var: Thư mục chứa những files, thư mục có nội dung thay đổi, không cố định (log)

+ /proc, /tmp: các thư mục không có thiết bị nào đứng sau. /proc chứa các files thông tin về CPU, RAM, còn /tmp chứa các file tạm ứng dụng tạo ra khi chạy.

+ Files type:

-: regular file

d: directory

l: link

c: special file

s: sockets

p: Named pipe

b: block device

+ Tìm kiếm files: which, whereis, locate và find.

3. Permission

Trên Linux có 3 quyền cơ bản là các quyền read, write và execute

|  |  |  |
| --- | --- | --- |
| Read | r | 4 |
| Write | w | 2 |
| Execute | x | 1 |

Tại sao là các giá trị 1, 2, 4? Lần lượt 3 quyền r, w, x sẽ được thể hiện trên các bits phân quyền, r:100, w:010, x:001.

Quyền của users, groups đối với file, thư mục sẽ được phân theo 3 nhóm sử dụng sau:

+ Owner: người dùng sở hữu file, ký hiệu u

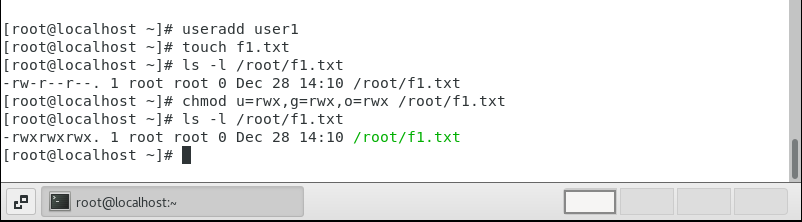
+ Group: nhóm người dùng sở hữu file, ký hiệu g

+ Others: người dùng và nhóm người dùng khác không sở hữu file, ký hiệu o

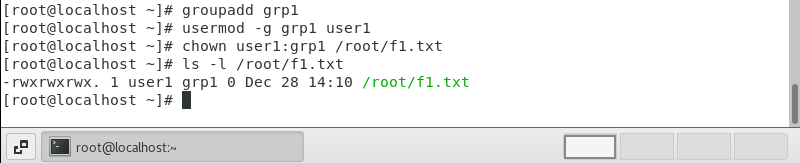
Như vậy tổng cộng ta cần 9 bits để phân 3 quyền cơ bản cho 3 nhóm đối tượng sử dụng trên hệ thống.

+ Các câu lệnh gán quyền và thay đổi quyền sở hữu: chmod và chown

Ví dụ: Thay đổi quyền của các nhóm sử dụng đối với file /root/f1.txt



Hoặc thay đổi người dùng, nhóm người dùng sở hữu của /root/f1.txt



***Umask*** – con số quyết định một file hay thưc mục được tạo ra sẽ có quyền gì. Ta có thể xem umask bằng lệnh: umask. Hoặc thay đổi giá trị umask: umask <value>

Giá trị umask có 4 con số, và thường là 0022 trên mọi hệ thống. Để giải thích về 4 con số, ta bắt đầu từ 3 số cuối trước.

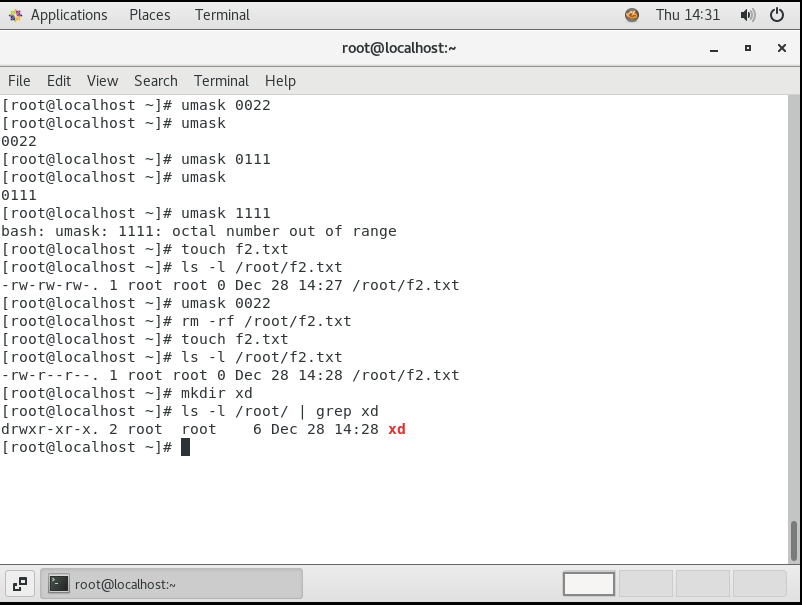
Hãy lưu ý là, với umask là 0022, các file khi vừa được tạo ra sẽ có 9 bits phân quyền là rw-r--r--, còn các thư mục sẽ là rwxr-xr-x.

Nghĩa là giá trị quyền sẽ lần lượt là 644 với file và 755 với thư mục.

Suy ra ta có công thức:

Với file: permissions = 666 – umask

Với thư mục: permissions = 777 – umask



Có thể thấy rằng, nếu ta đặt umask với chữ số đầu tiên khác 0, hệ thống sẽ báo “number out of range”. Ở đây số 0 tượng trưng cho các quyền đặc biệt trên hệ thống.

***Quyền đặc biệt***

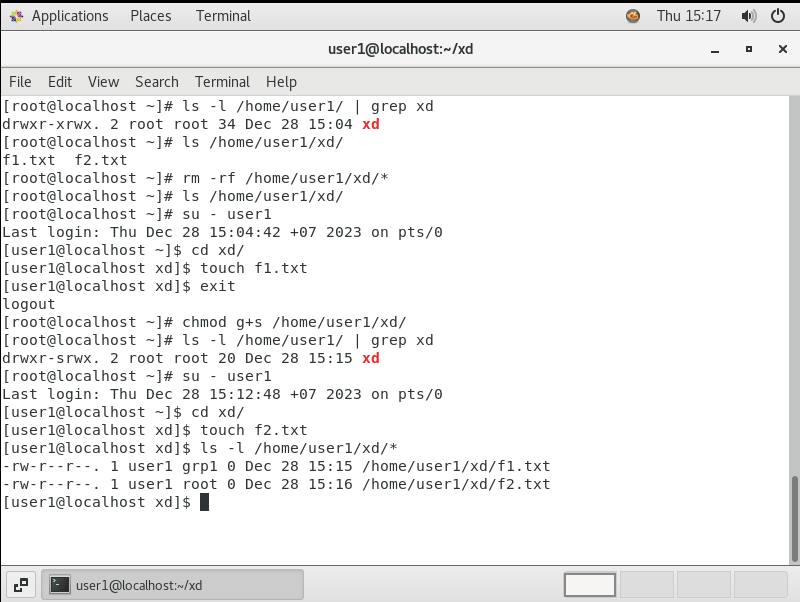
Ngoài 3 quyền cơ bản thì trên Linux còn có thêm 3 quyền đặc biệt nữa. Đó lần lượt là SUID (set userid), SGID (set groupid) và Sticky Bit.

+ SUID: Cấp quyền thực thi như là chủ sở hữu đối với 1 file cho tất cả các users có quyền thực thi với file đó trên hệ thống. Ví dụ user root là sở hữu của file f1.sh và có quyền thực thi file f1.sh, khi thêm quyền SUID cho f1.sh, thì 1 usera nào đó có quyền thực thi với f1.sh, cũng có thể chạy f1.sh như là người dùng root chạy.

Cú pháp cấp quyền: chmod u+s <file\_name>

+ SGID: Quyền này được cấp cho thực mục. Nếu 1 thư mục được cấp quyền này, thì các files trong thư mục được tạo ra khi đó sẽ có nhóm sở hữu là nhóm của thư mục cha, chứ không phải nhóm sở hữu của người dùng hiện đang tạo file. Ví dụ nhóm người dùng root là sở hữu của thư mục folder, thư mục này được cấp quyền SGID, một usera thuộc nhóm grp1 có quyền rwx với folder, tạo 1 file f1.txt trong folder, thì file này có nhóm sở hữu là root chứ không phải là grp1

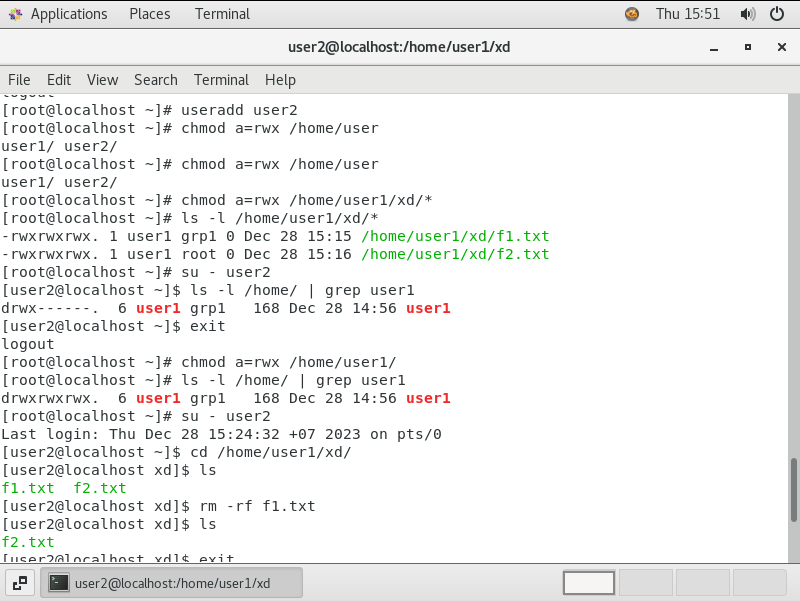
Cú pháp cấp quyền: chmod g+s <directory\_name>



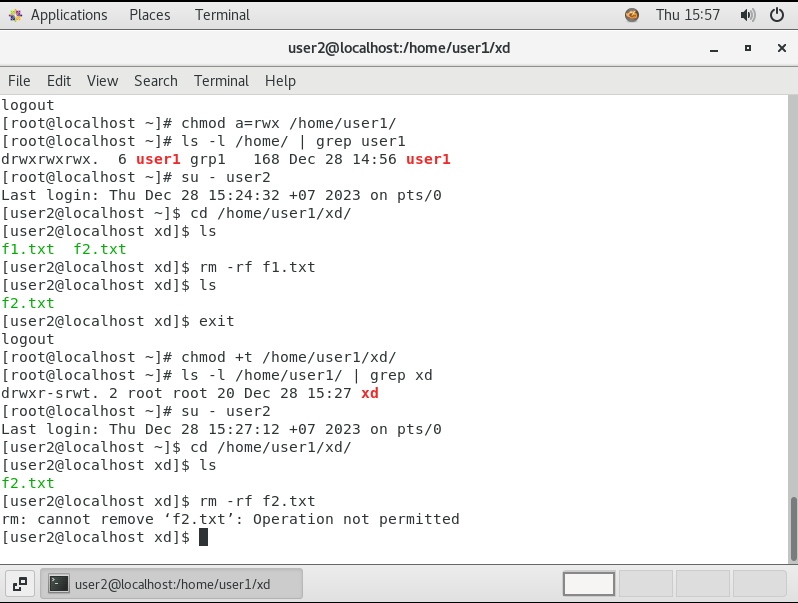
+ Sticky Bit: Khi một thư mục được đặt quyền sticky bit, chỉ người sở hữu của tệp tin đó mới có thể xóa hoặc di chuyển tệp tin khỏi thư mục.

Xem ví dụ sau:

Đã có sẵn user1, với 2 file đều thuộc sở hữu của nó từ ví dụ trên. Cấp quyền rwx cho tất cả users đối với 2 file này. Thêm user2 để tiến hành xóa file khi chưa thiết lập sticky bit.



Có thể thấy mặc dù f1.txt không thuộc sở hữu của user2, nhưng user2 vẫn có thể xóa file này. Lúc này, ta tiến hành thêm quyền Sticky Bit cho thư mục xd.



Có thể thấy user2 không thể xóa được file f2.txt dù có toàn bộ các quyền rwx đối với nó.

4. Quota

Quota là một tính năng cho phép ta quản lý, phân chia hạn ngạch cho người dùng trên hệ thống. Cụ thể, tính năng có thể được thiết lập trên từng thiết bị lưu trữ kết nối với máy chủ, trên bản thân ổ cứng của máy chủ. Để enable quota chúng ta sẽ sử dụng một câu lệnh quen thuộc là lệnh mount.

***Khái niệm:***

+ Hard limit: Đây là giới hạn khi mà người sử dụng chạm đến thì ngay lập tức sẽ không thể chiếm dụng tài nguyên trên thiết bị đã được thiết lập quota nữa.

+Soft limit: Đây là giới hạn khi mà người dùng chạm đến thì sẽ đưa ra cảnh báo về việc hạn ngạch sử dụng của người dùng sắp hết, cần cân nhắc trong việc sử dụng

Và khi người dùng chạm ngưỡng soft limit thì sẽ có một khoảng thời gian được định ra cho người dùng tiếp tục sử dụng, đó gọi là grace time

+ Grace time: Là khoảng thời gian đếm ngược, cho phép người dùng chiếm dụng tài nguyên trong giới hạn của hard limit. Nếu grace time kết thúc thì dù cho chưa chạm tới hard limit, người dùng cũng không thể chiếm dụng thêm tài nguyên được nữa.

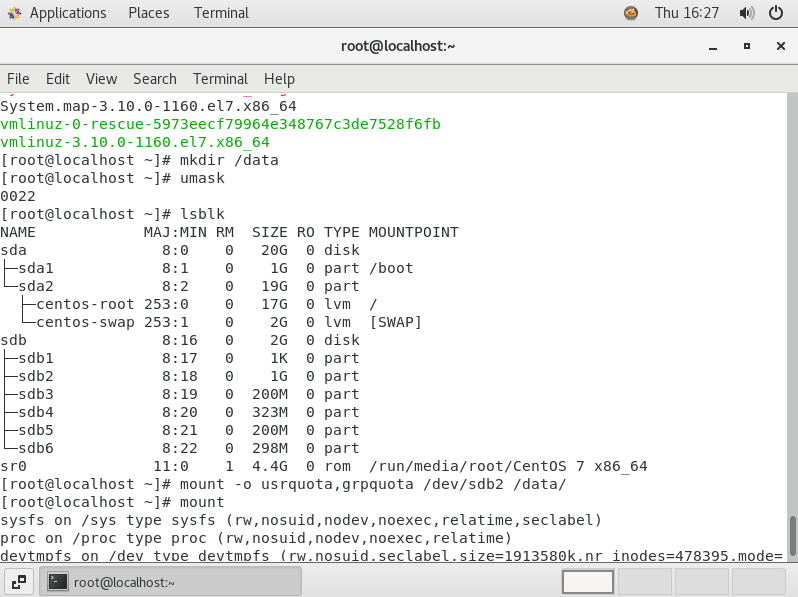
***Thiết lập quota:***

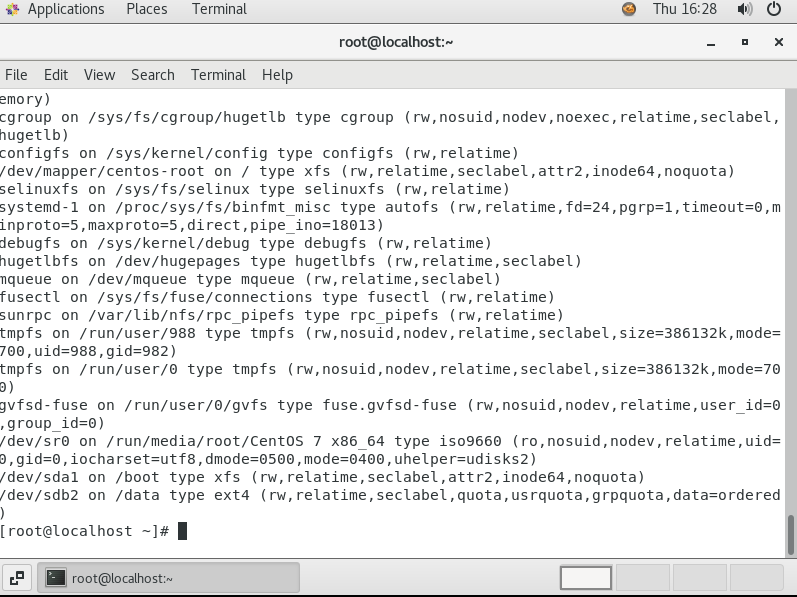
Bước 1: Enable quota

+ Sử dụng lệnh mount một thiết bị nào đó, chẳng hạn /dev/sdb2 đến một mountpoint nào đó /data, để thiết lập quota trên thiết bị đó.

mount -o usrquota,grpquota /dev/sdb2 /data

+ Để xem tính năng quota đã enabled chưa, dùng lệnh: mount





Bước 2: Tạo file lưu thông tin quota người dùng

Có 2 file lưu thông tin là: aquota.user – thông tiin người dùng và aquota.group – thông tin nhóm người dùng.

+ Tạo files bằng cách: quotacheck –cug /data

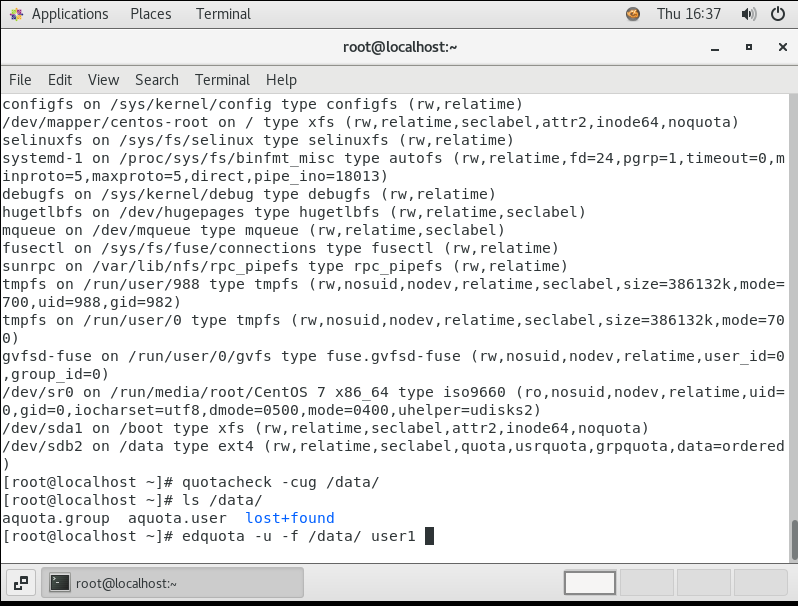
Trường hợp hệ thống báo lỗi không tạo được, tắt SELinux: setenforce 0

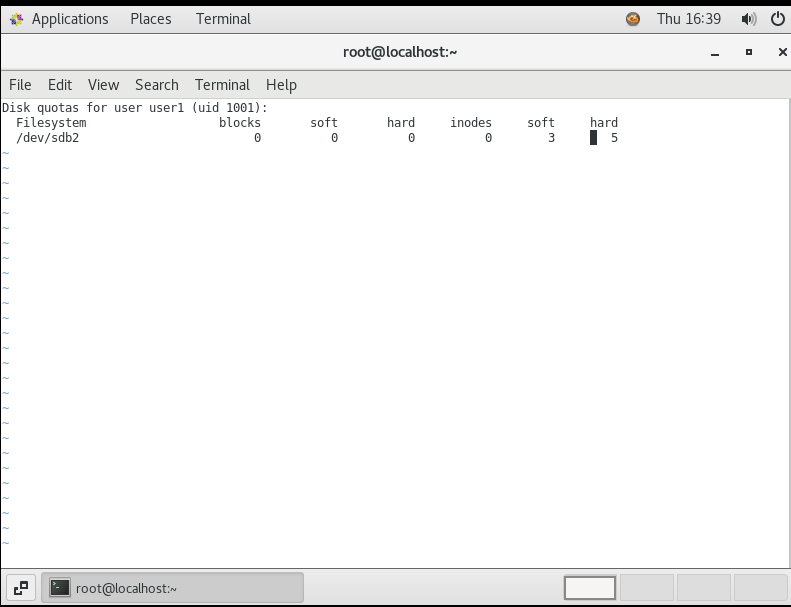
Bước 3: Thiết lập quota:

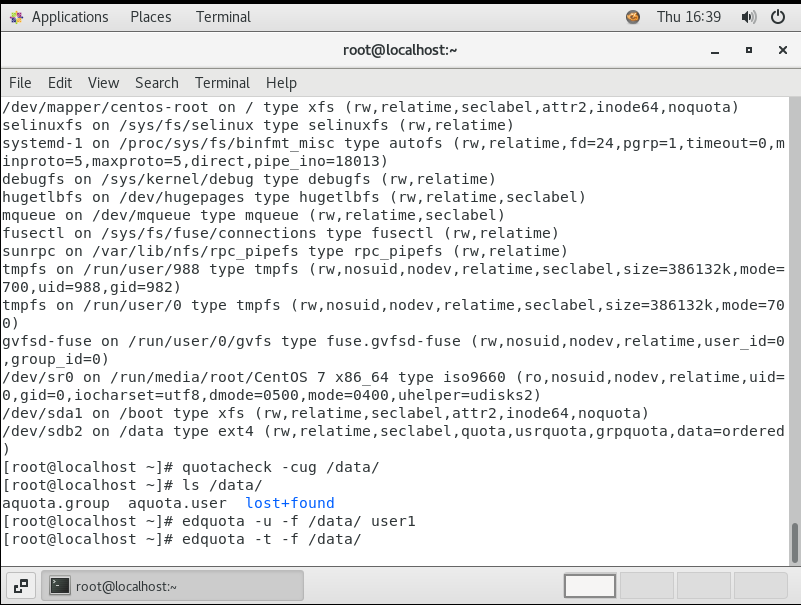
Ví dụ: user1: softlimit=3 inodes, hardlimit=5 inodes, grace time=2 minutes

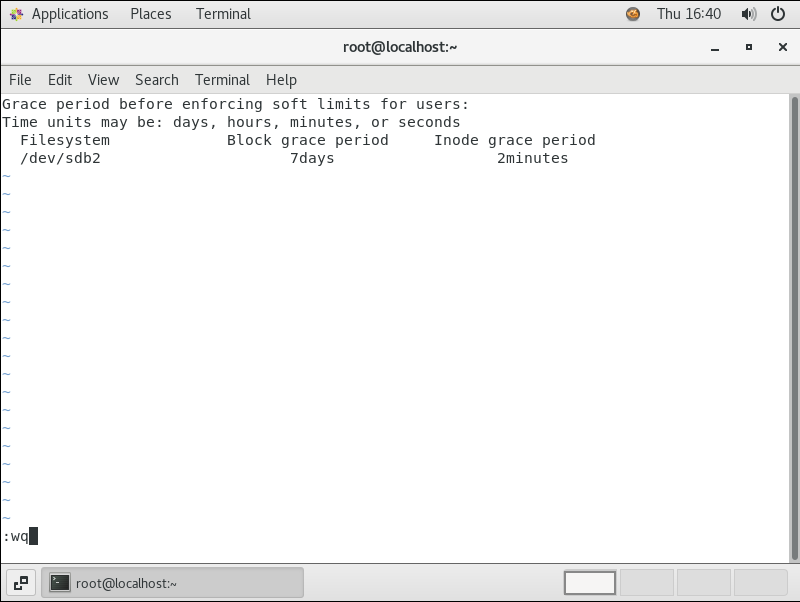
+ Thiết lập các giới hạn: edquota –u –f /data user1

+ Thiết lập grace time: edquota –t –f /data







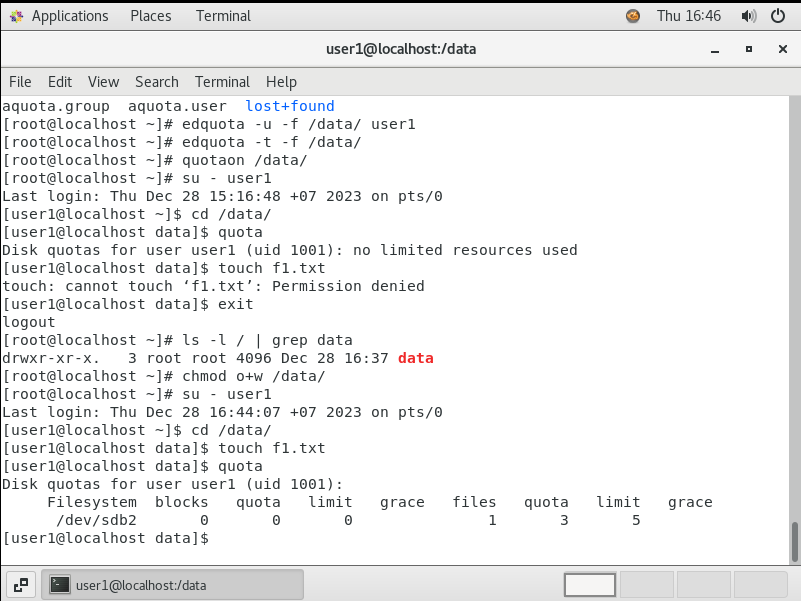


Sửa và lưu file thao tác như trình soạn thảo vi

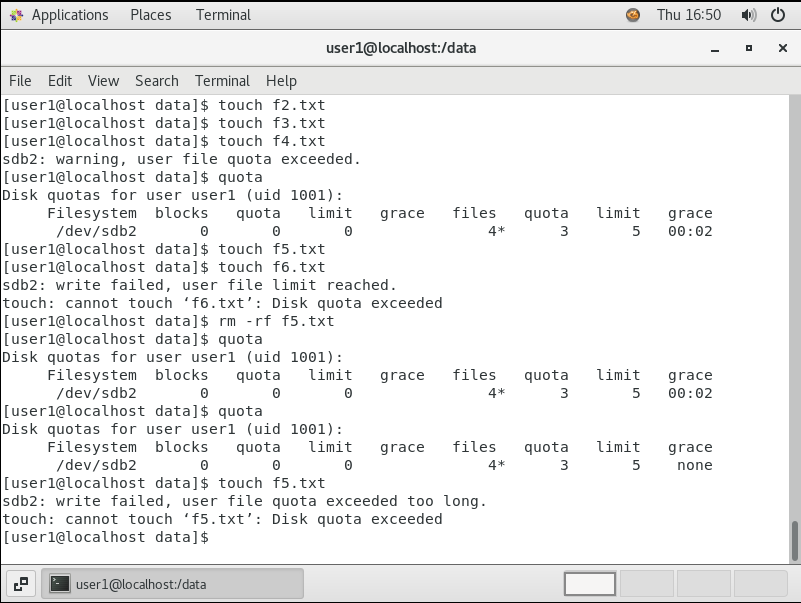
Bước 4: Turn on quota

+ Dùng lệnh: quotaon /data

+ Để xem hạn ngạch đã sử dụng, cd vào thư mục đã thiết lập rồi dùng lệnh: quota



Tiến hành thử các giới hạn và grace time



5. Users và Groups:

***Khái niệm***

Trên Linux người dùng cơ bản chia làm 2 nhóm sau:

+ Super user: Đây là người dùng toàn quyền, thường có tên là root với UID=0, có toàn quyền với tất cả các file trên hệ thống. Có thư mục home là /root.

+ Users: Là những người dùng có quyền giới hạn trên hệ thống, mỗi người dùng sẽ có thư mục home nằm trong /home

Nhóm người dùng có quyền giới hạn cũng chia ra làm hai nhóm:

+ Special users: Nhóm bị giới hạn nhất, thậm chí không có quyền đăng nhập vào hệ thống vì môi trường shell của chúng được thiết lập là nologin. Với các hệ thống cũ, nền tảng sysv, UID (và cả GID) của nhóm này sẽ đánh số từ 1-499. Với các hệ thống mới, nền tảng systemd thì từ 1-999.

+ Normal users: Những users có quyền giới hạn trong phạm vi thư mục sở hữu nhưng vẫn có quyền đăng nhập vào hệ thống ví có thể thiết lập được môi trường shell không phải là nologin. Với các hệ thống cũ, nền tảng sysv, UID (và cả GID) của nhóm này từ 500 trở lên. Với các hệ thống mới, nền tảng systemd thì từ 1000 trở lên.

***Thuộc tính users và groups***

Mỗi users sẽ có các thuộc tính sau:

+ Username

+ Password

+ Shell

+ Group

+ Desc

+ home

+ UID

Các thuộc tính này lưu trong file /etc/passwd

Groups:

+ Group name

+ Members

+ GID

Các thuộc tính này lưu trong file /etc/group

Riêng thuộc tính Password (dưới dạng mã hóa), số ngày expire password, ngày tạo password,... được lưu ở file /etc/shadow và chỉ user toàn quyền mới đọc được file này.

***Quản lý users và groups***

+ Tạo user/group: useradd/groupadd <user\_name>/<groupname>

+Thay đổi thuộc tính user: usermod/groupmod <options>

+ Xóa user/group: userdel/groupdel <username>/<group\_name>

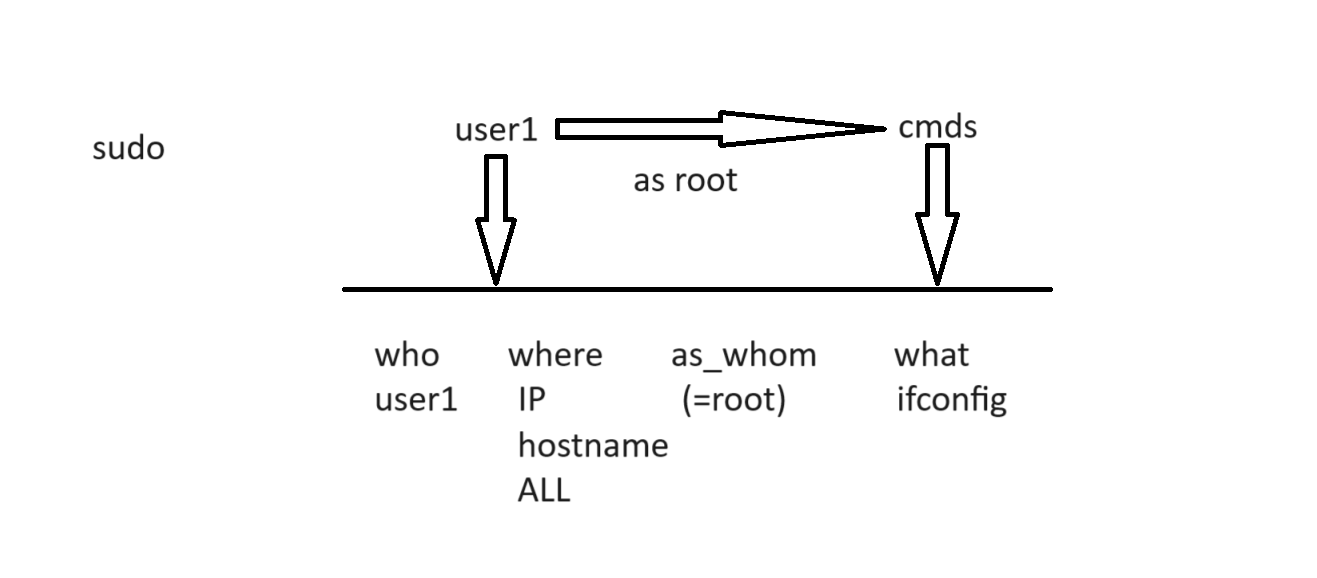
+ Đổi mật khẩu: passwd <user\_name>

+ Chuyển sang user khác: su <user\_name>, hoặc su - <user\_name>

***Quyền sudo***

Đây là quyền do root cấp cho users khác để thực hiện một số thao tác nhất định trên hệ thống.

Mô hình ví dụ về việc cấp quyền sudo cho user1 để cấu hình mạng bằng tool ifconfig.



Để cấp quyền sudo cho user1 thì ta cần khai báo cấp quyền trong file /etc/sudoers như sau:

user1 ALL=(root) /sbin/ifconfig

Như thế này, mỗi lần đầu tiên dùng lệnh sudo ifconfig trong 1 phiên sẽ phải nhập mật khẩu. Nếu không muốn nhập mật khẩu, khai bảo lại như sau:

user1 ALL=(root) NOPASSWD /sbin/ifconfig

Ngoài ra, có thể cấp quyền sudo cho một nhóm user được chỉ định bằng alias, và có thể cấp quyền sudo để thực hiện một nhóm các lệnh chứ không chỉ 1 lệnh.

Chi tiết xem nội dung file /etc/sudoers sẽ hướng dẫn đầy đủ ở các comments trong đó.