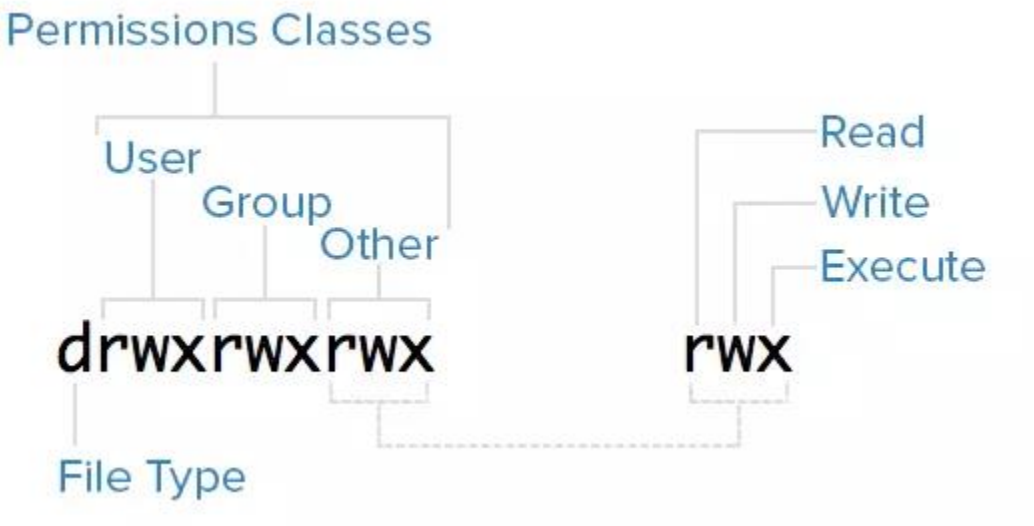
CHALLENGE 2: LẬP TRÌNH LINUX/STICKY BIT

I. Khái niệm phân quyền file, thư mục trong linux

1.1. Các cờ phân quyền r, w, x

Cờ phân quyền	Dạng số	File	Thư mục
r	4	Cho phép mở file và đọc	Cho phép liệt kê danh sách file và thư mục trong đó
w	2	Cho phép sửa đổi nội dung file	Cho phép thêm, xóa, đổi tên các file trong thư mục
x	1	Cho phép thực thi file	Cho phép truy cập (cd) đến thư mục



1.2. Các lệnh chown, chmod

a. chmod (change mode)

Dùng để thay đổi quyền của một thư mục hay 1 file trên Linux:

chmod [options] mode [mode] file1 file2 file3 ....

Danh sách các option:

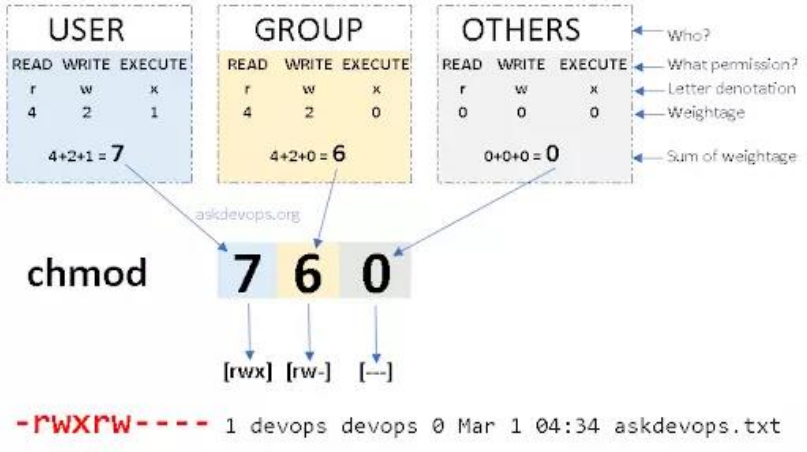
-R, --recursive: áp dụng cho tất cả các file và folder bên trong

-f (force), --silent, --quiet: set quyền trong cả trường hợp xảy ra lỗi

--reference: Cấp quyền cho file giống như file được chỉ định: **chmod --reference=file1 file2**

Lệnh trên sẽ set quyền của file2 giống như file1

- Phân quyền bằng số: **chmod <permissions-number> <filename>**



Permissions -number về cơ bản sẽ có 3 chữ số một số với ý nghĩa số thứ nhất là quyền của user, số thứ 2 là quyền của group, số thứ 3 là quyền của other

Số	Ký hiệu	Ý nghĩa
0	---	Không có quyền
1	--x	Thực thi
2	-w-	Ghi
3	-wx	Thực thi + Ghi
4	r--	Đọc
5	r-x	Đọc + Thực thi
6	rw-	Đọc + Ghi
7	rwx	Đọc + Ghi + Thực thi

- Phân quyền bằng ký tự: **chmod [OPTIONS] [ugoa…][-+=]perms…[,…] FILE...**
  - Tham số u, g, o, a đại diện cho nhóm đối tượng sẽ được xử lý quyền

Ký hiệu	Ý nghĩa
u	user
g	group
o	other
a	tất cả

- Tham số [-+=] dùng để thiết lập quyền được xóa, thêm hay thiết lập mới

Ký hiệu	-	+	=
Ý nghĩa	Xóa quyền	Thêm quyền	Gán lại quyền

b. Thay đổi owner và group

- Thay đổi quyền sở hữu của một file, thư mục. Để thay đổi được phải dùng quyền sudo:
  chown [OPTIONS] USER[:GROUP] FILE(s)

- Một số options của lệnh chmod:
  -R, recursive: Áp dụng thay đổi với tất cả các file và folder bên trong nó
  --reference: chỉnh sửa owner và group cho file giống như file được chỉ định
  chown --reference=FILE1 FILE2
  Lệnh trên sẽ set owner và group của File1 cho File2

- Ngoài ra có thể sử dụng UID Và GID thay cho user và group
  chown UID:GID FILE(s)

c. Cờ phân quyền đặc biệt: setuid, setgid, sticky_bit

1. Setuid (SUID)

- Khi một file được set quyền SUID khi có một người dùng khác chạy file thực thi đó thì nó sẽ được chạy dưới quyền của chủ sở hữu file đấy

```
nhidb@ubuntu:~/Desktop/thuctap$ ls -l /usr/bin/passwd
-rwsr-xr-x 1 root root 68208 May 28  2020 /usr/bin/passwd
```

- Thêm quyền SUID cho một file: **chmod u+s <tên_file>**
- Xóa quyền SUID: **chmod u-s <tên_file>**

2. Setgid (SGID)

- Tương tự như SUID, một file thực thi được set quyền này thì nó sẽ chạy dưới quyền của Group sở hữu file đấy

```
nhidb@ubuntu:~/Desktop/thuctap$ ls -l /usr/bin/crontab
-rwxr-sr-x 1 root crontab 43720 Feb 14  2020 /usr/bin/crontab
```

- Khi một user thuộc group sỡ hữu Folder có quyền SGID tạo một File/Foler mới bên trong nó thì mặc định group sỡ hữu folder ấy là folder cha của nó thay vì primary group như mặc định
- Thêm quyền SGID: **chmod g+s <tên_file>**
- Xóa quyền SGID: **chmod g-s <tên_file>**

3. Sticky bit

- Được sử dụng trong các thư mục chia sẻ để người dùng này không thể xóa, đổi tên thư mục của người kia. Chỉ owner và root mới có quyền đổi tên cũng như xóa thư mục khi nó đã được set cờ sticky bit

```
nhidb@ubuntu:~/Desktop/thuctap$ ls -l /var/spool/cron
total 4
drwx-wx--T 2 root crontab 4096 Aug 13 00:25 crontabs
```

- Ngoài ra có thể thêm các quyền đặc biệt này bằng cách sử dụng giá trị số. Đối với SUID là 4, SGID là 2 và sticky bit là 1. Thêm vào phía trước permissions-number của các quyền cơ bản.
- Thêm cờ SUID và các quyền cơ bản cho file thực thi test

```
nhidb@ubuntu:~/Desktop/thuctap$ ls -l test
---------- 1 nhidb nhidb 0 Aug 22 11:42 test
nhidb@ubuntu:~/Desktop/thuctap$ chmod +4755 test
nhidb@ubuntu:~/Desktop/thuctap$ ls -l test
-rwsr-xr-x 1 nhidb nhidb 0 Aug 22 11:42 test
```

d. Khái niệm real user id (ruid), effective user id (euid), saved user id (saved_uid)

1. Real user id (ruid)

- Với một tiến trình thì real userid là userid khởi động tiến trình đấy. Nó được dùng để xác định xem các file mà tiến trình đó đã truy cập vào.

2. Effective user id (euid)

- Thường thì *EUID* sẽ trùng với *RUID* nhưng trong một số trường hợp sẽ có sự khác biệt. Đôi khi bạn cần sử dụng tạm thời danh tính của một user khác (thường sẽ là root, nhưng cũng có thể là bất kì người dùng nào).
- Khi chạy lệnh passwd thì nó sẽ chạy dưới quyền root nên euid của nó là 0 (uid của root)

3. Saved user id (saved_uid)

- Được sử dụng khi một process đang chạy dưới quyền root và nó cần thực hiện một số tác vụ không yêu cầu quyền root. Khi đó EUID sẽ được lưu vào bên trong SUID và tác vụ đó sẽ được chuyển thành tác vụ bình thường. Khi hoàn thành EUID sẽ được lấy từ SUID và chuyển lại tài khoản root.

4. Xem euid, ruid, suid của một tiến trình

   ps -eo pid, ruid, euid, suid | grep <PID>

5. Hàm setuid(uid_t uid) và seteuid(uid_t uid):
   - **Hàm setuid(uid_t uid):** Đặt lại các giá trị ruid, euid và suid cho tiến trình về giá trị uid. Nếu giá trị uid giống với ruid hoặc suid thì hàm luôn thành công. Nếu uid không trùng với ruid thì hàm chỉ thành công nếu tiến trình có quyền phù hợp. Khi đó cả ba giá trị ruid, euid và suid sẽ được đặt lại thành uid.
   - **Hàm seteuid(uid_t uid):** Đặt lại euid thành giá trị của uid nếu uid giống với ruid hoặc suid. Nếu không trùng thì hàm sẽ thực hiện nếu có quyền thích hợp

e. Lệnh chattr (Change attribute)

- Câu lệnh cho phép thay đổi thuộc tính của file giúp bảo vệ file khỏi bị xóa hoặc ghi đè nội dung dù là user root
- Cú pháp lệnh: **chattr [operator] [flags] [filename]**
  - Tham số [-+=] dùng để

Ký hiệu	-	+	=
Ý nghĩa	Gỡ bỏ thuộc tính	Thêm thuộc tính cho file	Giữ nguyên thuộc tính

  - Các flag (hay thuộc tính) của file:

- i: Flag này khiến file không thể rename, không thể tạo symlink, không thể thực thi, không thể write. Chỉ có user root mới set và unset được thuộc tính này.
- a: Flag này khiến file không thể rename, không thể tạo symlink, không thể thực thi, chỉ có thể nối thêm nội dung vào file. Chỉ có user root mới set và unset được thuộc tính này.

```
root@ubuntu:~/thuctap# cat test.txt
Hello!!
root@ubuntu:~/thuctap# chattr +i test.txt
root@ubuntu:~/thuctap# lsattr
----i--------------- ./test.txt
root@ubuntu:~/thuctap# rm test.txt
rm: cannot remove 'test.txt': Operation not permitted
root@ubuntu:~/thuctap# mv test.txt test2.txt
mv: cannot move 'test.txt' to 'test2.txt': Operation not permitted
root@ubuntu:~/thuctap#
```

f. Access Control List

- ACLs là cách khác để xác định quyên trên tệp và thư mục. Chúng cho phép gán quyền cho một người dùng hoặc một nhóm bất kỳ, thậm chí không tương ứng với owner hoặc owning group. Một tệp hoặc thư mục có thể có nhiều ACL
- Kiểm tra trạng thái khởi đầu của ACL: **getfacl name**
- Sử dụng setfacl để thêm hoặc sửa đổi một hoặc nhiều quy tắc trong ACL của tệp cú pháp là: **setfacl -M [rules] [files], setfacl -m [rules] [files]**
  - -m (–modify) và -M (–modify-file)
  - Để loại bỏ các mục ACL: -x (–remove) và -X (–remove-fileao
  - Các rules ở dạng:
    - u:name:permissions: Đặt ACL truy cập cho user (tên người dùng hoặc UID)
    - g:name:permissions: Đặt ACL truy cập cho group (tên nhóm hoặc GID)
    - m:permissions: Đặt mask quyền có hiệu lực. Đây là sự kết hợp của tất cả các quyền của nhóm sở hữu và tất cả các mục nhập của người dùng và nhóm.
    - o:permissions: Đặt ACL truy cập cho others

```
nhidb@ubuntu:~/Desktop/thuctap$ ls -dl Test
drwxrwxr-x 2 nhidb nhidb 4096 Aug 23 14:47 Test
nhidb@ubuntu:~/Desktop/thuctap$ getfacl Test
# file: Test
# owner: nhidb
# group: nhidb
user::rwx
group::rwx
other::r-x

nhidb@ubuntu:~/Desktop/thuctap$ setfacl -m user:userB:rwx,group:nhidb:rwx Test
nhidb@ubuntu:~/Desktop/thuctap$ getfacl Test
# file: Test
# owner: nhidb
# group: nhidb
user::rwx
user:userB:rwx
group::rwx
group:nhidb:rwx
mask::rwx
other::r-x

nhidb@ubuntu:~/Desktop/thuctap$ sudo chmod g-w Test
nhidb@ubuntu:~/Desktop/thuctap$ getfacl Test
# file: Test
# owner: nhidb
# group: nhidb
user::rwx
user:userB:rwx                  #effective:r-x
group::rwx                      #effective:r-x
group:nhidb:rwx                 #effective:r-x
mask::r-x
other::r-x
```

- Khi tệp có ACL, 'ls –l' hiển thị dấu cộng (+) sau các quyền

```
drwxr-xr-x+ 2 nhidb nhidb  4096 Aug 23 14:47 Test
```

- **Default ACLs**: Default ACL ảnh hưởng đến các thư mục con cũng như là các files. Nói cách khác, các thư mục con và tệp sẽ kế thừa default ACL của thư mục cha

```
nhidb@ubuntu:~/Desktop/thuctap$ setfacl -d -m group:nhidb:r-x Test
nhidb@ubuntu:~/Desktop/thuctap$ getfacl Test
# file: Test
# owner: nhidb
# group: nhidb
user::rwx
user:userB:rwx                  #effective:r-x
group::rwx                      #effective:r-x
group:nhidb:rwx                 #effective:r-x
mask::r-x
other::r-x
default:user::rwx
default:group::rwx
default:group:nhidb:r-x
default:mask::rwx
default:other::r-x
```

II. Thực hành

a. Set password cho 1 user bất kỳ bằng quyền người dùng thường.

Mô tả: có 3 user: A, B, C. Chương trình tên là: mypasswd. Khi đăng nhập bằng user A. Chạy chương trình ./mypassword. Chương trình sẽ hỏi tên user và mật khẩu mới. Nhập tên user B (hoặc C) và mật khẩu mới 123456a@ thì mật khẩu user B (hoặc C) sẽ được đổi sang 123456a@

```cpp
#include <bits/stdc++.h>
#include <pwd.h>
#include <grp.h>
#include <string>
#include <unistd.h>

using namespace std;

int main() {
    char user[100]="";
    struct passwd *entry;
    cout << "Enter user: ";
    cin >> user;
    entry = getpwnam(user);
    int changeid = (int) entry->pw_uid;
    setuid(changeid);
    cout << "User id: " << changeid << endl;
    system("passwd");
}
```

```
nhidb@ubuntu:~$ ps -ef | grep userB
userB       6539    6538  0 14:18 pts/0    00:00:00 ./mypassword
root        7407    7397  0 14:33 pts/0    00:00:00 su userB
userB       7408    7407  0 14:33 pts/0    00:00:00 sh
userB       8540    7414  0 15:29 pts/0    00:00:00 ./mypasswd
userB       8541    8540  0 15:29 pts/0    00:00:00 sh -c passwd
nhidb       8544    8442  0 15:29 pts/1    00:00:00 grep --color=auto userB
nhidb@ubuntu:~$
^C
nhidb@ubuntu:~/Desktop/thuctap$ ./mypasswd
Enter user: userB
User id: 1002
Changing password for userB.
Current password:
```

b. *Có 2 user thường*, chạy tiến trình bằng quyền user 1 nhưng thực hiện lệnh id thì in ra thông tin user 2

Mô tả: có 2 user A, B. Chương trình tên là: myid. Khi đăng nhập bằng user A. Sau đó chạy chương trình ./myid thì sẽ in ra user B. Kiểm tra tiến trình myid bằng lệnh ps thì thấy user chạy là user B