**PROG02: LẬP TRÌNH ỨNG DỤNG LINUX**

Mục lục

[**A.** **Lý thuyết** 1](#_Toc157787148)

[**1.** **Nắm vững khái niệm user, group; owner của file, folder; các cờ phân quyền đối với file, folder** 1](#_Toc157787149)

[**2.** **Nắm vững khái niệm real user id và effective user id của một process** 2](#_Toc157787150)

[**3.** **Nắm vững khái niệm cờ setuid và setgid của một file binary** 3](#_Toc157787151)

[**B.** **Thực hành** 4](#_Toc157787152)

[**1.** **Tìm kiếm và hiển thị thông tin user theo id** 4](#_Toc157787153)

[**2.** **Chương trình cho phép user đổi mật khẩu mypasswd** 5](#_Toc157787154)

[**C.** **Tài liệu tham khảo** 7](#_Toc157787155)

1. **Lý thuyết**
2. **Nắm vững khái niệm user, group; owner của file, folder; các cờ phân quyền đối với file, folder**

* User (người dùng) là người sử dụng hệ thống hoặc các tài nguyên trên đó. Mỗi người dùng thường có một tài khoản duy nhất được định danh bằng tên người dùng (username) và một số nguyên duy nhất được gọi là User ID (UID).
* Group (nhóm) là một tập hợp các người dùng được tổ chức lại với nhau để chia sẻ quyền lợi hoặc quản lý dễ dàng hơn. Mỗi nhóm được định danh bằng tên nhóm (group name) và một số nguyên duy nhất gọi là Group ID (GID). Các người dùng trong cùng một nhóm thường có thể chia sẻ quyền lợi hoặc tài nguyên.
* Owner(**Chủ sở hữu**) là người dùng hoặc nhóm sở hữu file hoặc folder trên hệ thống. owner có quyền kiểm soát truy cập và sửa đổi file/folder.
* Các cờ phân quyền xác định quyền hạn và quyền truy cập của user, group và others (người dùng khác) đối với một tài nguyên cụ thể. Có ba loại cờ phân quyền chính
  + Quyền sỡ hữu (user) : xác định quyền truy cập của chủ sở hữu file hoặc folder.
  + **Quyền nhóm (Group):** xác định quyền truy cập của các thành viên trong nhóm sở hữu file hoặc folder.
  + Quyền khác (Other): xác định quyền truy cập của tất cả người dùng khác không thuộc nhóm sở hữu file hoặc folder.

Mỗi cờ phân quyền bao gồm 3 quyền:

* + Đọc (read): Cho phép người dùng xem nội dung file, folder
  + Ghi (write): cho phép người dùng chỉnh sửa nội dung file, folders
  + Thực thi (execute): Cho phép người dùng thực thi file hoặc truy cập folder.

Trong Unix/Linux, cờ phân quyền thường được hiển thị dưới dạng 10 ký tự. Trong đó:

* + Kí tự đầu tiên: biểu thị kiểu file, bao gồm các giá trị
    - - : file thông thường
    - d: thư mục
    - c: thiết bị
    - l: liên kết
  + 9 ký tự tiếp theo, mỗi 3 ký tự lần lượt là cờ phân quyền cho user, group, other, bao gồm các giá trị
    - r: quyền đọc
    - w: quyền ghi
    - x: quyền thực thi
    - - : không có quyền

Ví dụ, ‘**drwxr-xr--'**, biểu thị

* + d: biểu thị là thư mục (directory)
  + rwx: quyền của chủ sỡ hữu (read, write, execute)
  + r-x: quyền của nhóm (read, execute)
  + r--: quyền của người dùng khác (read)

1. **Nắm vững khái niệm real user id và effective user id của một process**

* Real user id:
  + Đại diện cho người dùng thực tế đã khởi chạy quá trình.
  + Xác định quyền truy cập của quy trình vào các tệp và thư mục do người dùng cụ thể đó sở hữu.
  + Không thay đổi trong suốt thời gian tồn tại của quy trình và không thể được sửa đổi trực tiếp bởi chính quy trình đó.
  + Phản ánh quyền sở hữu.
* Effective user id:
  + Xác định các đặc quyền của người dùng mà quy trình hiện đang nắm giữ để truy cập tài nguyên.
  + Xác định khả năng truy cập các tệp và thư mục của quy trình dựa trên ID này, bất kể Real Uid.
  + Có thể được thay đổi tạm thời bởi quy trình bằng cách sử dụng hàm setuid() cho các hoạt động cụ thể.
  + Phản ánh các đặc quyền truy cập hiện tại.

1. **Nắm vững khái niệm cờ setuid và setgid của một file binary**

* **setuid** và **setgid** là hai cờ phân quyền đặc biệt có thể gán cho các tệp thực thi nhị phân. Có ảnh hưởng đến cách OS chạy chương trình và cách xử lý các quyền của người dùng và nhóm trong việc thực thi chương trình. thường được sử dụng để cho phép người dùng trên hệ thống máy tính chạy các chương trình có đặc quyền nâng cao tạm thời để thực hiện một tác vụ cụ thể.
* **setuid:** Khi cờ setuid được đặt trên một tệp thực thi, chương trình sẽ chạy với đặc quyền của người dùng sở hữu tệp thay vì đặc quyền của người dùng đang thực thi chương trình. Thường được sử dụng để cho phép người dùng thông thường thực hiện các tác vụ yêu cầu đặc quyền nâng cao, chẳng hạn như thay đổi mật khẩu hoặc truy cập một số tài nguyên hệ thống nhất định. Lệnh đặt cờ setuid:

**chmod u+s executable\_file**

* **setgid:** Khi cờ setgid được đặt trên tệp thực thi, chương trình sẽ chạy với quyền nhóm của nhóm sở hữu tệp, thay vì quyền nhóm của người dùng đang thực thi chương trình. Thường được sử dụng để cho phép người dùng chạy chương trình với sự cho phép của một nhóm cụ thể, bất kể tư cách thành viên trong nhóm. Lệnh đặt cờ setgid:

**chmod g+s executable\_file**

1. **Thực hành**

* Sử dụng ngôn ngữ lập trình C/C++ trên môi trường Ubuntu, xây dựng 2 chương trình (lưu ý không gọi các chương trình có sẵn như id, passwd, cần xử lý các thông tin trong file /etc/passwd, /etc/shadow):

1. **Tìm kiếm và hiển thị thông tin user theo id**

*Mô tả: Chương trình tên là myid. Khi chạy ./myid, chương trình hỏi nhập username. Người dùng nhập username, chương trình sẽ tìm user có username tương ứng và hiển thị thông tin id, username, thư mục home, các group của user. Nếu không có user tương ứng thì báo không tìm thấy.*

**Ý tưởng:** dùng thư viện <pwd.h> và <grp.h> để trích xuất thông tin trong file /etc/paswd và /etc/group.

**Giải thích chương trình:**

* Sử dụng hàm getpwnam() để lấy bản ghi của username trong file /etc/passwd. Hàm sẽ trả về một struct passwd bao gồm các thuộc tính như ảnh sau, sau đó tiến hành in ra 3 thông số pw\_uid, pw\_name, và pw\_dir theo yêu cầu:

A screenshot of a computer screen

Description automatically generated

* Sử dụng hàm getgrouplist() để truy xuất tất cả group id của username lưu vào mảng groups[], tiến hành duyệt tất cả các id trong mảng groups và dùng hàm getgrgid() để lấy thông tin của group theo group id, và in ra thông số gr\_name của group. Hàm getgrgid() trả về một struct group bao gồm các thuộc tính như hình sau:

A group of black text

Description automatically generated

Biên dịch và chạy chương trình:

A screenshot of a computer screen

Description automatically generated

1. **Chương trình cho phép user đổi mật khẩu mypasswd**

*Mô tả: Khi một user A chạy chương trình ./mypassword. Chương trình sẽ yêu cầu nhập mật khẩu cũ, nếu mật khẩu cũ đúng thì sẽ hỏi mật khẩu mới. Sau đó chương trình sẽ đổi mật khẩu user sang mật khẩu mới.*

**Ý tưởng:** trích xuất thông tin ở file /etc/shadow để đối chiếu mật khẩu cũ của người dùng và cập nhật bản ghi của người dùng bao gồm mật khẩu mới của người dùng và ngày sửa đổi.

**Giải thích chương trình:**

* Đầu tiên sử dụng hàm getuid() để lấy id của người dùng chạy chương trình. Sau đó dùng hàm getpwuid() để lấy thông tin user trong trong file /etc/passwd bao gồm tên user. Sử dụng hàm getspnam() để lấy thông tin về bản ghi của user trong file /etc/shadow dựa trên tên user, hàm trả về một struct spwd bao gồm các thuộc tính như hình sau:

A computer screen shot of a computer program

Description automatically generated

* Tiếp theo người dùng nhập mật khẩu cũ bằng hàm getpass() thực hiện mã hóa mật khẩu này bằng hàm crypt() và so sánh với thuộc tính sp\_pwdp được trả về ở bước trước. Nếu hai giá trị giống nhau thì tiến hành nhập mật khẩu mới, và thực hiện cập nhật mật khẩu bằng hàm change\_shadowfile\_entry()
* Hàm change\_shadowfile\_entry():
  + Tạo ra một file tạm
  + Tiến hành duyệt qua tất cả các bản ghi trong file /etc/shadow, nếu là người dùng cần thay đổi mật khẩu thi ghi giá trị bản ghi đã sửa mật khẩu và ngày thay đổi vào file tạm, ngược lại thì ghi giá trị bản ghi cũ
  + Cuối cùng chép giá trị của file tạm vào file /etc/shadow để hoàn thành sửa đổi mật khẩu
* **Biên dịch chương trình:** Do cần quyền root để chỉnh sửa file /etc/shadow nên ta cần biên dịch chương trình với quyền root và đặt cờ setuid cho file thực thi.

A screen shot of a computer

Description automatically generated

**Link video demo của 2 chương trình:** [duclt22\_prog02.mp4 - Google Drive](https://drive.google.com/file/d/1jL149W5o_S4vrMJG6nPHYjQNc3EfO4hE/view)

**Link mã nguồn 2 chương trình:** [VCS\_Training/PROG\_02 at master · ltduc147/VCS\_Training (github.com)](https://github.com/ltduc147/VCS_Training/tree/master/PROG_02)

1. **Tài liệu tham khảo**

* <https://en.wikipedia.org/wiki/Setuid>
* Sách The Linux Programming Interface (chương 8, chương 9)