ĐẠI HỌC KHOA HỌC TỰ NHIÊN, ĐHQG-HCM
KHOA CÔNG NGHỆ THÔNG TIN

HỆ ĐIỀU HÀNH
Báo cáo Đồ án 1

DRIVER ẨN FILE TRÊN LINUX

Nhóm thực hiện

1. Phạm Hùng Việt	1612809
2. Hồng Thanh Hoài	1612855
3. Huỳnh Hoàng Huy	1612861

Giáo viên lý thuyết
ThS. Nguyễn Văn Giang

Giáo viên hướng dẫn
ThS. Lê Quốc Hòa

Tháng 11 năm 2018

Lời cảm ơn

Trong quá trình thực hiện đồ án này, nhóm chúng em đã nhận được rất nhiều sự giúp đỡ cũng như hỗ trợ từ các thầy cô Trường Đại học Khoa học Tự nhiên, ĐHQG-HCM và các bạn bè trong trường. Nhóm chúng em xin bày tỏ lòng cảm ơn chân thành đến mọi người vì đã hướng dẫn, chỉ bảo rất tận tình.

Đặc biệt, nhóm chúng em xin bày tỏ lòng biết ơn sâu sắc đến các thầy cô khoa Công nghệ thông tin, cụ thể là thầy Nguyễn Văn Giang và thầy Lê Quốc Hòa đã giảng dạy cũng như hướng dẫn rất kĩ lưỡng để chúng em có thể hoàn thành tốt đồ án này.

Một lần nữa, chúng em xin bày tỏ lòng biết ơn sâu sắc đến với các thầy cô và bạn bè.

Tháng 11 năm 2018,
Đại học Khoa học Tự nhiên, ĐHQG-HCM.

Mục lục

1 Giới thiệu nhóm và phân công công việc

1.1 Giới thiệu nhóm

Nhóm gồm 3 thành viên.

STT	Họ và tên	MSSV	Email
1	Phạm Hùng Việt	1612809	1612809@student.hcmus.edu.vn
2	Hồng Thanh Hoài	1612855	hthoai1006@gmail.com
3	Huỳnh Hoàng Huy	1612861	huynhhoanghuy11111998@gmail.com

1.2 Phân công công việc

STT	Họ và tên	Nội dung công việc
1	Phạm Hùng Việt	Hàm `hidefile_init`, `hidefile_exit`. Định nghĩa lại các struct `file_operations`.
2	Hồng Thanh Hoài	Phần giao tiếp phía người dùng (`user_space`). Viết báo cáo.
3	Huỳnh Hoàng Huy	Hàm `allocate_memmory`, `reallocate_memmory`. Hàm `hook_functions`, `backup_functions`.

2 Giới thiệu đồ án

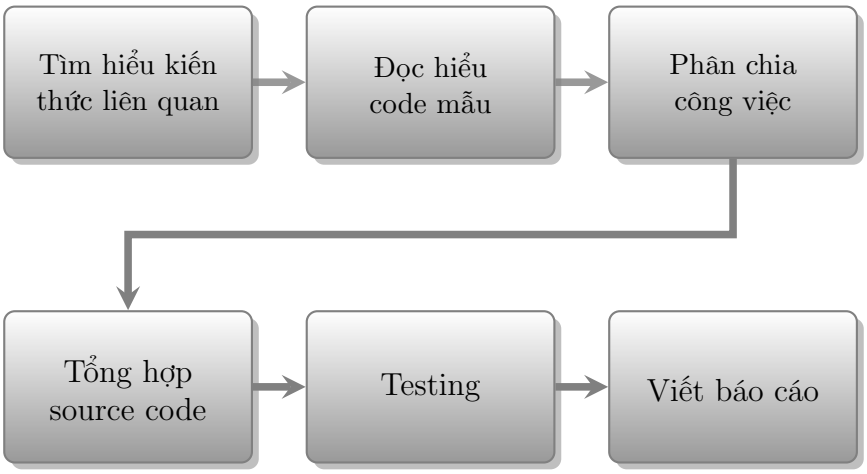
2.1 Mục tiêu

- Hiểu được hệ thống quản lý tập tin trên Linux.
- Lập trình driver để ẩn file trên Linux.

2.2 Thời gian và công cụ thực hiện

- Thời gian thực hiện: từ ngày 14/10/2018 đến ngày 04/11/2018.
  - 14/10/2018 – 21/10/2018: Tìm hiểu kiến thức liên quan, cụ thể là hệ thống quản lý tập tin trên Linux.
  - 22/10/2018 – 27/10/2018: Đọc hiểu code từ những nguồn giáo viên gợi ý và nhóm tự tìm hiểu.
  - 28/10/2018 – 03/11/2018: Tổng hợp source code, testing.
  - 03/11/2018 – 04/11/2018: Viết báo cáo.
- Công cụ làm việc nhóm: Facebook.
- Công cụ tạo máy ảo: VMware Workstation 8.0.
- Hệ điều hành dùng để kiểm thử: CentOS 6.0.

2.3 Các bước thực hiện



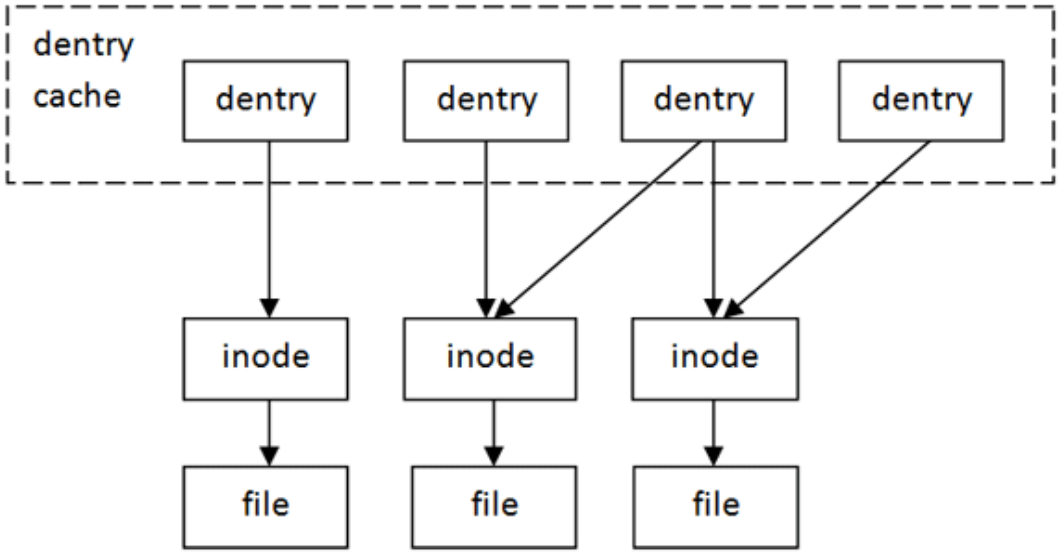
3 Nội dung đồ án

3.1 Tổng quan

Linux sử dụng cấu trúc hệ thống file có thứ bậc, giống một cấu trúc cây từ trên xuống dưới, với root (/) tại cơ sở của hệ thống file và tất cả các thư mục khác trải ra từ đó.

Virtual File System (VFS) là một tầng phần mềm trong kernel cung cấp khả năng tương tác với hệ thống tập tin ở mức độ người dùng. VFS thực thi những *system calls* như `open`, `stat`, `chmod`. Khi ta truyền tham số `filePath` vào, VFS sẽ dùng để tìm thông qua *directory entry cache (dentry cache)*. Điều này sẽ cung cấp cho ta một cơ chế rất nhanh để chuyển đường dẫn tệp thành một *dentry* cụ thể, phục vụ cho những tác vụ tiếp theo.

Một dentry độc lập thường có một con trỏ trỏ đến `inode` - một cấu trúc dữ liệu chứa các metadata của mỗi file, thư mục trong các hệ thống file Linux.



Hình 1: Mỗi cấu trúc inode sẽ có một số inode riêng.

Driver ẩn file sẽ thực hiện việc *hook* inode và file operations của một file cụ thể để thay đổi con trỏ trên cấu trúc inode và file operations thành hàm do ta định nghĩa lại.

Trong đồ án này, nhóm sẽ triển khai driver như một kernel module. Module là một đối tượng file được thiết kế đặc biệt. Khi làm việc với các module, Linux sẽ liên kết chúng với kernel và tải chúng vào vùng địa chỉ.

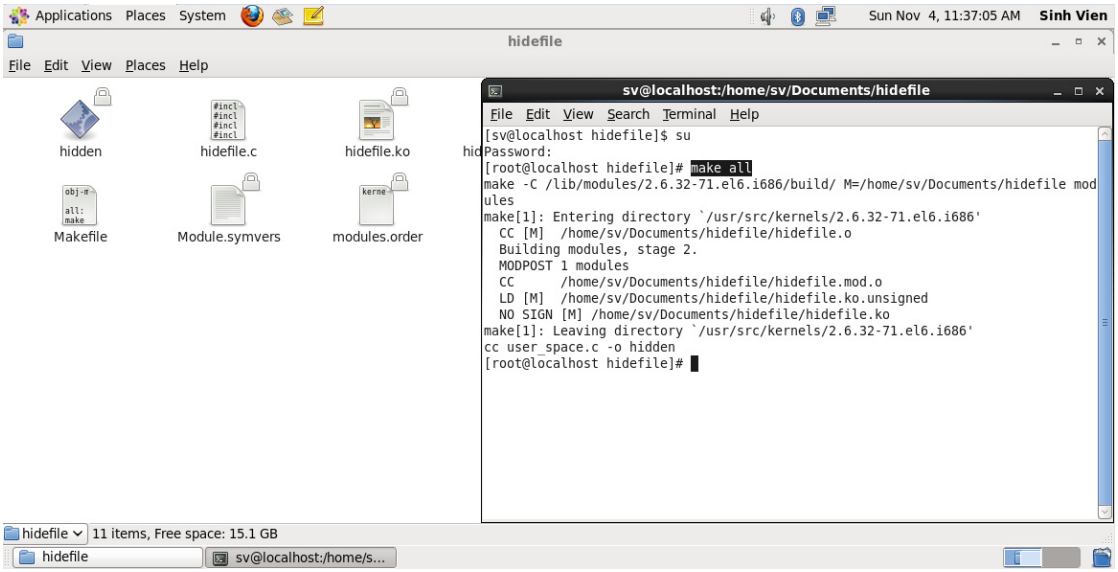
3.2 Các hàm chính

- `hidefile_init`
  - Tham số đầu vào: Không có.

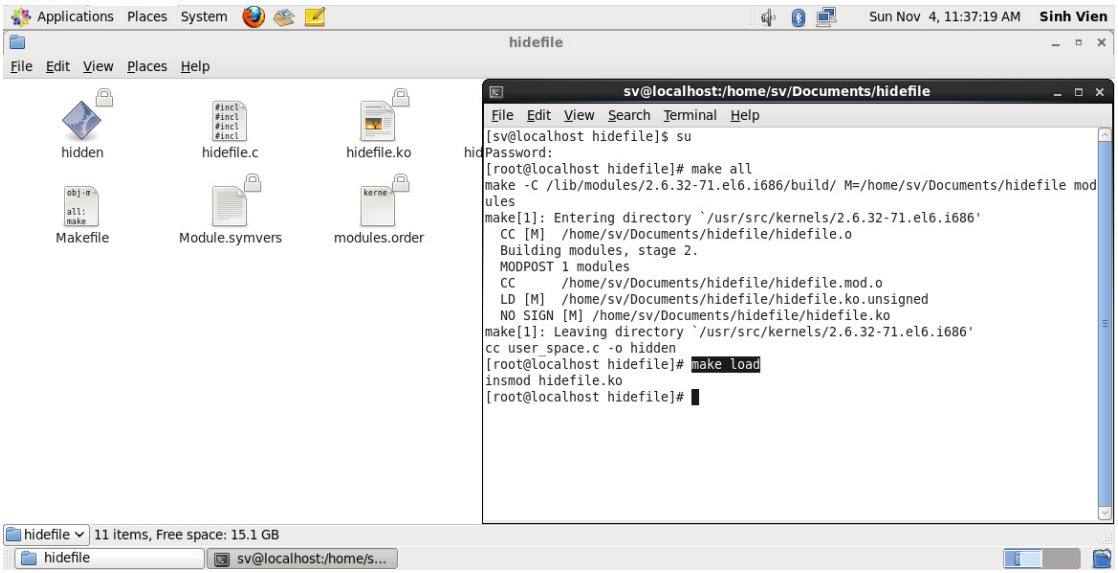
  - Công dụng: Khởi tạo module và đăng ký một character device.

- `hidefile_exit`
  - Tham số đầu vào: Không có.
  - Công dụng: Hủy module – khôi phục lại các con trỏ đã thay đổi của inode và file operations của file vừa ẩn, sau đó hủy đăng ký character device.
- Hàm `filldir()` mới cho thư mục cha của file
  - Mô tả: Để ẩn một file từ hệ thống, ta sẽ *hook* hàm `filldir()` cho thư mục cha. Hàm này được gọi từ hàm `readdir()` (hiển thị file trong thư mục). Để thay đổi lời gọi hàm `filldir()`, ta sẽ định nghĩa lại hàm `readdir()` và gọi hàm `filldir()` từ hàm vừa định nghĩa lại.
  - Các hàm thay đổi:
    ```
    int parent_readdir (struct file * file, void * dirent,
    filldir_t filldir)
    static int new_filldir (void *buf, const char *name,
    int namelen, loff_t offset,u64 ux64, unsigned ino)
    ```
- `allocate_memmory`
  - Tham số đầu vào: Không có.
  - Công dụng: Cấp phát bộ nhớ để lưu các con trỏ trên inode và file operations của file bị thay đổi.
- `reallocate_memmory`
  - Tham số đầu vào: Không có.
  - Công dụng: Lưu lại các con trỏ trên inode và file operations của file bị thay đổi.
- `hook_functions`
  - Tham số đầu vào: `file_path` (đường dẫn file).
  - Công dụng: *Hook* inode, file operations và số inode của một file cụ thể.
- `backup_functions`
  - Tham số đầu vào: Không có.
  - Công dụng: Khôi phục lại con trỏ trỏ đến inode gốc và file operations của một file cụ thể.
- Hàm `main` (trong *user_ space*): Nhận chuỗi đường dẫn từ người dùng và gửi xuống kernel.

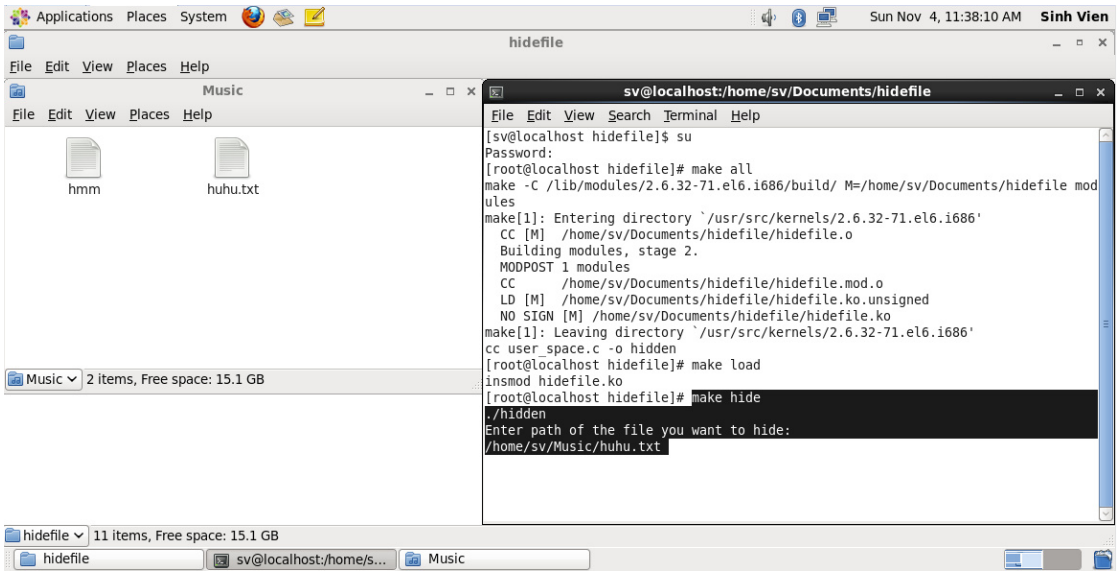
4 Kiểm thử



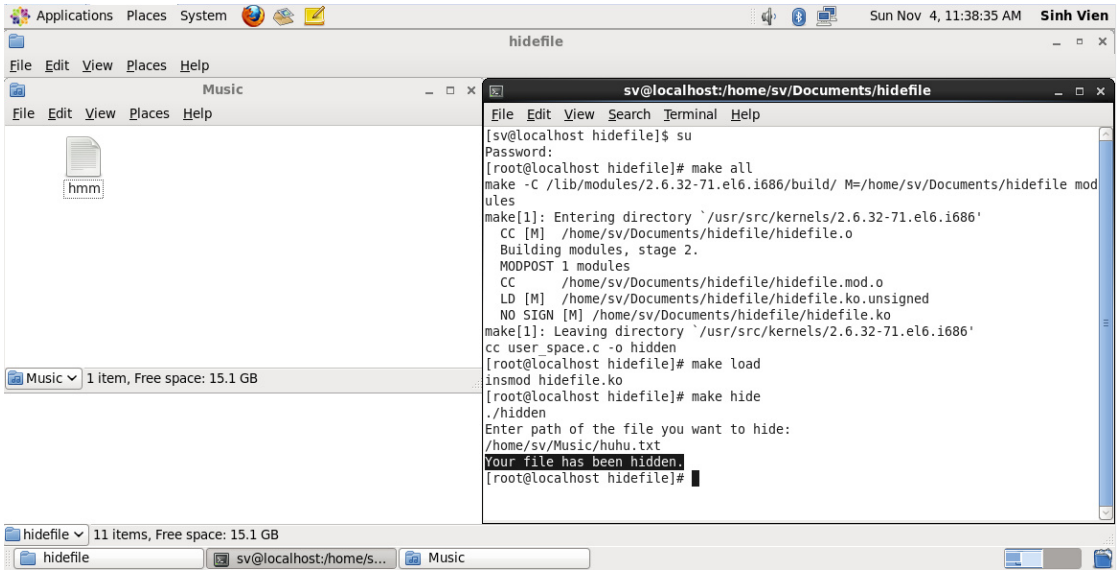
Hình 2: *make all.*



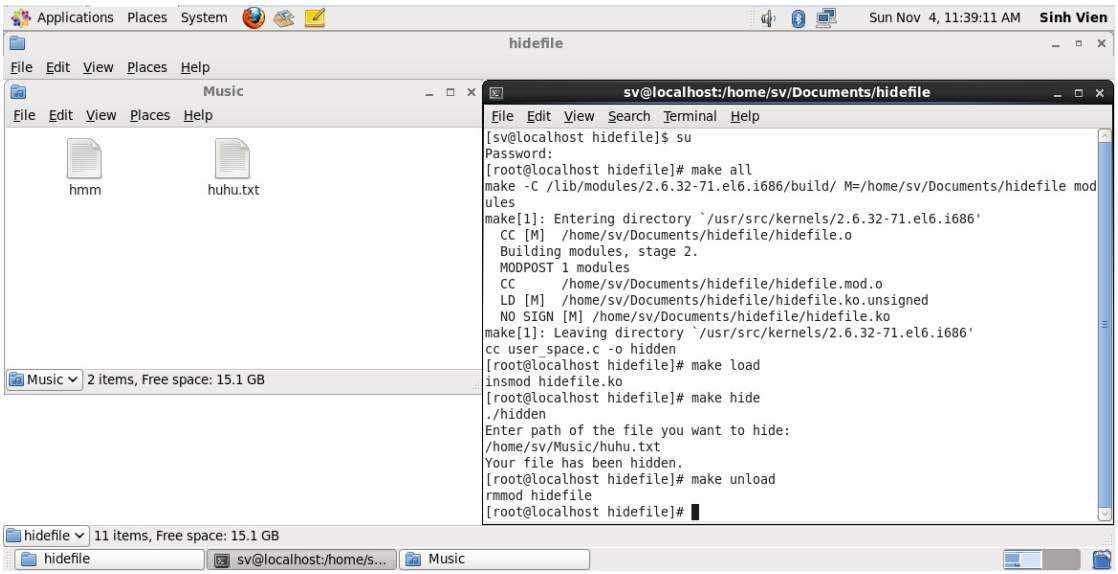
Hình 3: *make load.*



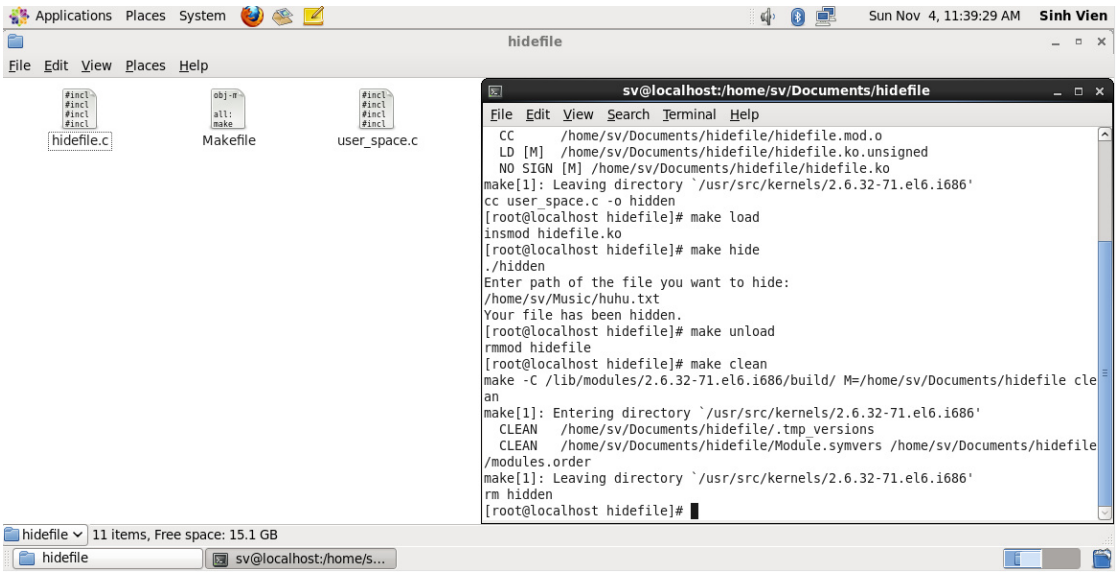
Hình 4: *make hide*, sau đó nhập đường dẫn file cần ẩn vào.



Hình 5: File *huhu.txt* đã được ẩn.



Hình 6: *make unload*, file hiện trở lại.



Hình 7: *make clean*.

5 Đánh giá và tổng kết quá trình

5.1 Mức độ hoàn thành của đồ án

STT	Nội dung	Hoàn thành
1	Tìm hiểu kỹ kiến thức liên quan.	100%
2	Phân chia công việc rõ ràng.	100%
3	Trình bày code sạch sẽ, comment đầy đủ.	95%
4	Có thể ẩn và khôi phục file bị ẩn.	95%
5	Kết quả đúng với yêu cầu đồ án.	95%
Mức độ hoàn thành tổng thể của đồ án:		**97%**

5.2 Những vấn đề chưa thực hiện được

- Một số hàm nhóm còn chưa thông suốt.
- Chưa thể kiểm tra tính hợp lệ của đường dẫn file người dùng nhập vào.

Tài liệu

[1] Apriorit, *Linux Driver Tutorial: How to Write a Simple Linux Device Driver*.

[2] CodeProject, *Driver to hide files in Linux OS*.

[3] derekmolloy, *Writing a Linux Kernel Module — Part 2: A Character Device*.

[4] nixCraft, *Understanding UNIX / Linux filesystem Inodes*.

[5] Georgia Tech on Udacity, *Advanced OS*.

[6] Ian! D. Allen, *CST8207-18W – GNU/Linux Operating Systems*.