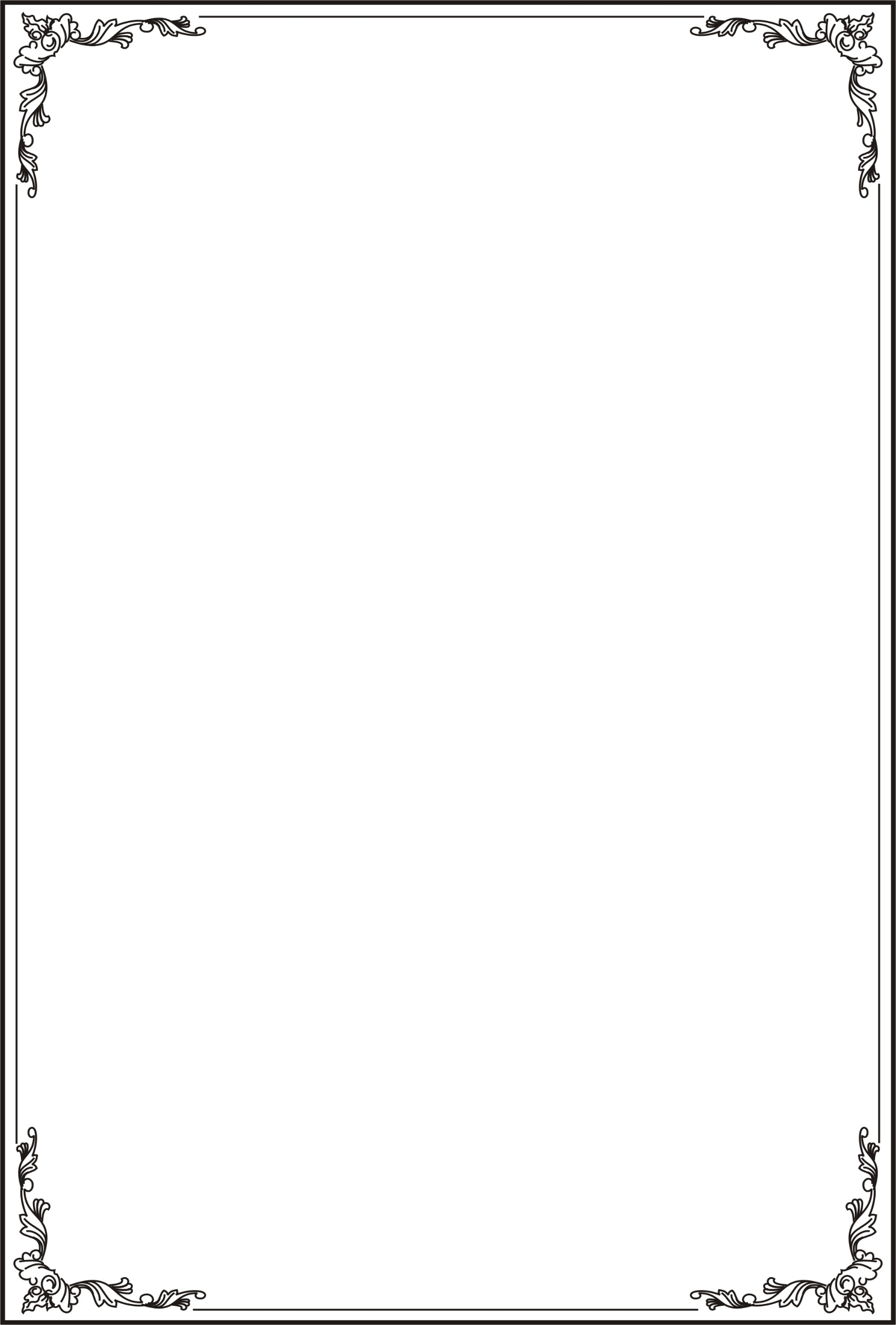
**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**ĐẠI HỌC KHOA HỌC TỰ NHIÊN**

**KHOA CÔNG NGHỆ THÔNG TIN**

🙠🙟🖳🙝🙢



**BÁO CÁO ĐỒ ÁN 2**

**LINUX KERNEL MODULE**

**Bộ môn: Hệ điều hành**

GVHD: Nguyễn Thị Thanh Huyền

Nhóm thực hiện gồm các thành viên:

1. Danh Phi Long - 1712083
2. Nguyễn Tấn Phát - 1712121
3. Lê Hận Thương Hoài - 1712452

**Mục lục**

[**1.** **Thông tin thành viên nhóm:** 3](#_Toc24053239)

[**2.** **Bảng phân công công việc:** 3](#_Toc24053240)

[**3.** **Nội dung thực hiện đồ án:** 3](#_Toc24053241)

[**3.1** **Ý tưởng thực hiện đồ án** 3](#_Toc24053242)

[**3.2** **Chức năng của chương trình** 4](#_Toc24053243)

[**3.3** **Đánh giá mức độ hoàn thành đồ án** 6](#_Toc24053244)

[**3.4** **Một số lưu ý** 6](#_Toc24053245)

[**4.** **Tài liệu tham khảo** 6](#_Toc24053246)

1. **Thông tin thành viên nhóm:**

|  |  |  |  |
| --- | --- | --- | --- |
| STT | MSSV | Họ tên | Ghi chú |
| 1 | 1712083 | Danh Phi Long | Nhóm trưởng |
| 2 | 1712121 | Nguyễn Tấn Phát |  |
| 3 | 1712452 | Lê Hận Thương Hoài |  |

1. **Bảng phân công công việc:**

|  |  |  |  |
| --- | --- | --- | --- |
| STT | MSSV | Họ tên | Công việc |
| 1 | 1712083 | Danh Phi Long | Module + Hook |
| 2 | 1712121 | Nguyễn Tấn Phát | Hook + Readme |
| 3 | 1712452 | Lê Hận Thương Hoài | Module + Report |

1. **Nội dung thực hiện đồ án:**
   1. **Ý tưởng thực hiện đồ án**

Phần 1: Mục tiêu hiểu về Linux kernel module và hệ thống quản lý file và device trong linux, giao tiếp giữa tiến trình ở user space và kernel space.

* Viết một module dùng để tạo ra số ngẫu nhiên.
* Module này sẽ tạo một character device để cho phép các tiền trình ở user space có thể open và read các số ngẫu nhiên

Phần 2: Chương trình hook vào một system call:

* syscall open => ghi vào dmesg tên tiến trình mở file và tên file được mở
* syscall write => ghi vào dmesg tên tiến trình, tên file bị ghi và số byte được ghi
  1. **Chức năng của chương trình**

**Thông tin phần mềm**

* Được viết bằng **C.**
* Source có chứa **Comments**.
* Phần module.

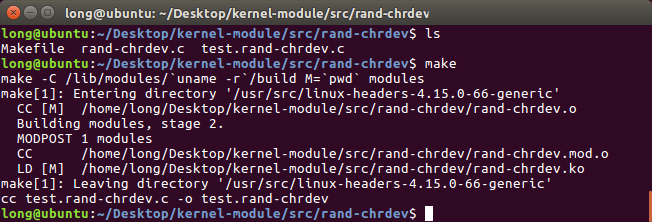
|  |  |
| --- | --- |
| **Các hàm sử dụng** | **Chức năng** |
| static int dev\_open(struct inode \*, struct file \*); | Gọi khi 1 file sử dụng device |
| static ssize\_t dev\_read(struct file \*, char \_\_user \*, size\_t, loff\_t \*); | Truyền dữ liệu cho file khi file thực hiện read |
| static int dev\_release(struct inode \*, struct file \*); | Gọi khi close file, không sử dụng device nữa |
| static \_\_init int chrdev\_init(void) | Khởi tạo module khi insmod |
| static \_\_exit void chrdev\_exit(void) | Thoát module khi rmmod |

* Phần hook.

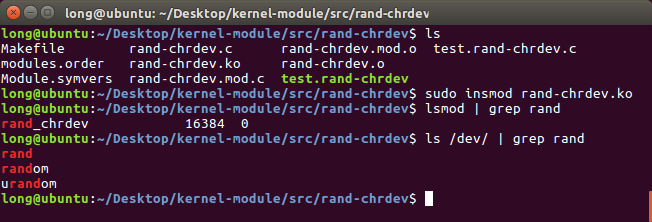
|  |  |
| --- | --- |
| **Các hàm sử dụng** | **Chức năng** |
| asmlinkage unsigned long (original\_open) (const char \_\_user, int, int); | Giữ chức năng gốc của sys\_open |
| asmlinkage unsigned long (original\_write) (unsigned int, const char \_\_user, size\_t); | Giữ chức năng gốc của sys\_write |
| asmlinkage unsigned long our\_sys\_open(const char\* \_\_user file\_name, int flags, int mode); | Hook, bắt sự kiện sys\_open được sử dụng |
| asmlinkage unsigned long our\_sys\_write(unsigned int fd, const char \_\_user \*buffer, size\_t count); | Hook, bắt sự kiện sys\_write được sử dụng |
| int make\_rw(unsigned long address); | Disable write protection |
| int make\_ro(unsigned long address); | Enable write protection |
| static \_\_init int hook\_init(void); | Khởi tạo module khi insmod |
| static void \_\_exit hook\_cleanup(void); | Thoát module khi rmmod |

**Demo**

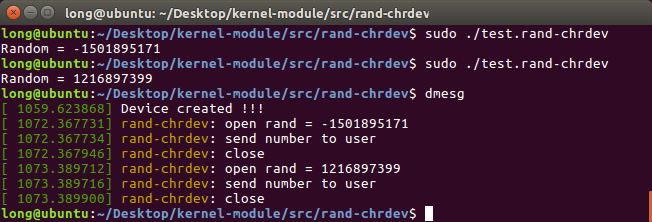
* Built and make file:

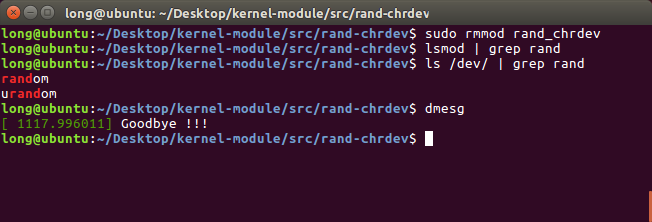
****

* Nạp module:

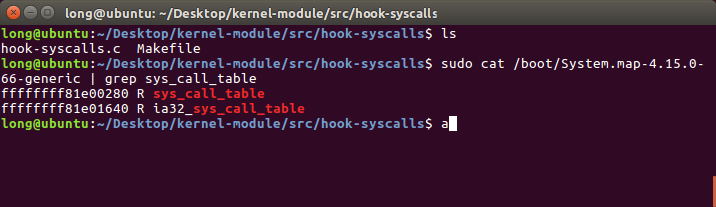
****

* Character device:

****

****

* Hook- system call:

****

* 1. **Đánh giá mức độ hoàn thành đồ án**

|  |  |  |
| --- | --- | --- |
| **STT** | **Chức năng** | **Mức độ hoàn thành** |
| 1 | Viết module dùng để tạo ra số ngẫu nhiên | 100% |
| 2 | Tạo một character device để cho phép các tiền trình ở user space có thể open và read các số ngẫu nhiên | 100% |
| 3 | Hook vào một system call | 100% |

* 1. **Một số lưu ý**

1. **Tài liệu tham khảo**

*https://uwnthesis.wordpress.com/2016/12/26/basics-of-making-a-rootkit-from-syscall-to-hook*