**BỘ MÔN HỆ THỐNG THÔNG TIN – KHOA CÔNG NGHỆ THÔNG TIN**

**ĐẠI HỌC KHOA HỌC TỰ NHIÊN THÀNH PHỐ HỒ CHÍ MINH, ĐẠI HỌC QUỐC GIA TPHCM**

**MÔN HỆ ĐIỀU HÀNH**

**BÁO CÁO PROJECT 1 – LINUX KERNEL MODULE & SYSTEM CALL**

**SINH VIÊN THỰC HIỆN:**

1712841 Phạm Văn Trình

1712850 Trần Trung

1612247 Nguyễn Quang Hưởng

1. **Mô tả tổ chức/ thiết kế đồ án**

### Thông tin đồ án

|  |  |
| --- | --- |
| Tên quy trình | Đồ án Thực Hành Môn Học |
| Công cụ | Hệ Điều Hành Ubuntu |
| Yêu cầu | 1. Hiểu về Linux Kernel module và hệ thống quản lý file và device trong Linux, giao tiếp giữa tiền trình user space và kernel space 2. Chương trình Hook để vào 1 systerm call |
| Mức độ hoàn thành | 100% |

### Phân chia công việc

|  |  |  |  |
| --- | --- | --- | --- |
| **MSSV** | **Họ và tên** | **Công việc** | **Mức độ hoàn thành** |
| 1712841 | Phạm Văn Trình | Hook vào một system call write. | 100% |
| 1612247 | Nguyễn Quang Hưởng | Hook vào một system call open. | 100% |
| 1712850 | Trần Trung | Viết module tạo ra số ngẫu nhiên và tạo character device. | 100% |

**Đồ án**

**Linux Kernel Mod**

* Linux Kernel Module là một đoạn code có khả năng được load vào kernel hoặc unload khỏi kernel khi cần thiết, không cần phải build lại kernel hay khởi động lại máy.
* Kernel module xuất hiện với một ý nghĩa vô cùng to lớn. Nếu không sử dụng kernel module thì ta phải load tất cả driver vào kernel, làm cho dung lượng của kernel tăng cao. Ngoài ra, kernel module rất linh hoạt khi thêm mới hay thay đổi drive trong hệ thống. Ví dụ khi máy tính nhận được thiết bị mới như chuột, bàn phím… thì driver của thiết bị sẽ được tải vào kernel.
* Linux sử dụng một cấu trúc hệ thống file có thứ bậc có đặc tính: thư mục gốc (/) chứa các file, thư mục khác. Trong /dev chứa các file đặc biệt gọi là device file.

**Hook System Call**

* + Hook là một cơ chế mà một ứng dụng có thể chặn các sự kiện, như các thông điệp, thao tác chuột, bàn phím. Hiện nay, có nhiều loại hook khác nhau với các chức năng khác nhau.
  + Bộ nhớ trong của Linux được chia thành 2 vùng phân biệt: user space và kernel space. Trong đó, user space là nơi mà các tiến trình người dùng được thực thi, còn kernel space là nơi kernel thực thi các tác vụ.

**Thiết kế đồ án**

**Random divice:**

- Cấu trúc của chương trình gồm 2 tập tin: 1 kernel module để phát sinh số ngẫu nhiên, 1 chương trình gọi từ user space xuống kernel.

- Ta sử dụng các thư viện trong Linux kernel để có thể sử dụng các hàm, macro để phát triển kernel module.

- Một kernel module có ít nhất 2 macro quan trọng là: module\_init() – giúp xác định hàm nào sẽ được thực thi ngay sau khi lắp vào module, module\_exit() – giúp xác định hàm nào được thực thi ngay trước khi tháo module ra khỏi kernel.

- Thư viện <linux/random.h> cho phép ta dùng hàm get\_random\_bytes để tạo ra số ngẫu nhiên được giới hạn trong phạm vi [0, 1000000]

static int Numbermin = 0;

static int Numbermax = 1000000;

static int randomNumber = 0;

get\_random\_bytes(&randomNumber, sizeof(randomNumber));

randomNumber = min + randomNumber%( Numbermax - Numbermin);

- Để tạo ra character device driver để tạo ra số ngẫu nhiên, ta thực hiện:

* + - 1. Đăng kí số hiệu <major, minor> cho file thiết bị.
      2. Kết nối các thao tác file thiết bị với các hàm tương ứng trong driver.

static int random\_open(struct inode\*, struct file \*);

static int random\_release(struct inode\*, struct file\*);

static ssize\_t random\_read(struct file\*, char\*, size\_t, loff\_t\*);

static struct file\_operations fops={

.open = random\_open,

.release = random\_release,

.read = random\_read,

};

**Chương trình Hook vào 1 systerm call**