



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

---

**VNUHCM - UNIVERSITY OF SCIENCE**

**Hệ Điều Hành**

**Đồ án 3 – Tìm hiểu và Lập trình Linux Kernel Module**

**BÁO CÁO**

**Giảng viên**

Phạm Tuấn Sơn

Lê Viết Long

## Mục Lục

### Contents

<b>1</b>	<b>THÔNG TIN THÀNH VIÊN .....</b>	<b>3</b>
<b>2</b>	<b>Linux Kernel .....</b>	<b>4</b>
<b>2.1</b>	<b>Giới thiệu về Linux Kernel.....</b>	<b>4</b>
<b>2.2</b>	<b>Giới thiệu về Linux Kernel Module.....</b>	<b>5</b>
<b>2.3</b>	<b>Mã nguồn và các hàm có trong Linux Kernel Module .....</b>	<b>6</b>
<b>3</b>	<b>Hình ảnh minh họa .....</b>	<b>9</b>
	<b>TÀI LIỆU THAM KHẢO .....</b>	<b>9</b>

# 1 THÔNG TIN THÀNH VIÊN

MSSV	Họ tên	Đóng góp
<b>20127577</b>	Phan Nguyễn Phước Nguyên	100%
<b>20127584</b>	Trần Hữu Minh Nhật	100%
<b>20127349</b>	Trần Quốc Thuận	100%

## 2 Linux Kernel

### 2.1 Giới thiệu về Linux Kernel

Năm 1991, dựa trên UNIX Kernel, Linux TõValds đã tạo ra Linux Kernel chạy trên máy tính của ông ấy. Dựa vào chức năng của hệ điều hành, Linux Kernel được chia làm 6 phần.

**Proces management:** có nhiệm vụ quản lý các tiến trình, bao gồm các công việc:

- + Tạo/hủy các tiến trình
- + Lập lịch cho các tiến trình
- + Hỗ trợ các tiến trình giao tiếp với nhau
- + Đồng bộ hoạt động của các tiến trình

**Memory management:** có nhiệm vụ quản lý bộ nhớ, bao gồm các công việc:

- + Cấp phát bộ nhớ, thu hồi bộ nhớ chương trình
- + Đảm bảo chương trình nào cũng có cơ hội đưa vào bộ nhớ
- + Bảo vệ vùng nhớ của mỗi tiến trình.

**Device management:** có nhiệm vụ quản lý thiết bị, bao gồm các công việc:

- + Điều khiển hoạt động của các thiết bị
- + Giám sát trạng thái của các thiết bị
- + Trao đổi dữ liệu với các thiết bị
- + Lập lịch sử dụng các thiết bị

**File system management:** có nhiệm vụ quản lý dữ liệu trên các thiết bị lưu trữ. Quản lý các công việc: thêm, tìm kiếm, sửa, xóa dữ liệu.

**Networking management:** có nhiệm vụ quản lý các gói tin theo mô hình TCP/IP.

**System call Interface:** có nhiệm vụ cung cấp các dịch vụ sử dụng phần cứng cho các tiến trình. Mỗi dịch vụ được gọi là một system call.

Mã nguồn Linux Kernel

Thư mục	Vai trò
/arch	Chứa mã nguồn giúp Linux kernel có thể thực thi được trên nhiều kiến trúc CPU khác nhau.
/block	Chứa mã nguồn triển khai nhiệm vụ lập lịch cho các thiết bị lưu trữ.
/drivers	Chứa mã nguồn triển khai nhiệm vụ điều khiển, giám sát, trao đổi dữ liệu với các thiết bị.
/fs	Chứa mã nguồn triển khai nhiệm vụ quản lý dữ liệu trên các thiết bị lưu trữ.
/ipc	Chứa các mã nguồn triển khai nhiệm vụ giao tiếp giữa các tiến trình.
/kernel	Chứa mã nguồn triển khai nhiệm vụ lập lịch và đồng bộ hoạt động của các tiến trình.
/mm	Chứa mã nguồn triển khai nhiệm vụ quản lý bộ nhớ.
/net	Chứa mã nguồn triển khai nhiệm vụ xử lý các gói tin theo mô hình TCP/IP.

## 2.2 Giới thiệu về Linux Kernel Module

Linux Kernel Module là một file với tên mở rộng là “.ko”. Nó sẽ được lắp vào hoặc tháo ra khỏi Kernel khi cần thiết. Chính vì vậy, nó còn một tên gọi khác là loadable kernel module. Một trong những kiểu loadable phổ biến đó là driver. Việc thiết kế driver theo kiểu loadable module mang lại 3 lợi ích:

- + Giúp giảm kích thước kernel. Do đó, giảm sử lãng phí bộ nhớ và giảm thời gian khởi động hệ thống.

- + Không phải biên dịch lại kernel khi thêm mới driver hoặc khi thay đổi driver.

- + Không cần phải khởi động lại hệ thống khi thêm mới driver.

Phần lớn các driver đều là các loadable kernel module, nhưng không phải là tất cả. Vẫn có một số driver được tích hợp luôn vào trong kernel, đặc biệt là các bus driver. Chúng được gọi là built-in-driver. Các device thường sẽ là các loadable kernel module.

Ngược lại, không phải là loadable kernel module nào cũng là driver, ví dụ kvm.ko là loadable kernel module nhưng không phải là driver. Trên thực tế, loadable kernel module được chia làm ba loại chính: device driver, system call và device system.

## 2.3 Mã nguồn và các hàm trong linux kernel module

### Make file

```
obj-m = mymodule.o
all:
|   make -C /lib/modules/$(shell uname -r)/build/ M=$(PWD) modules
clean:
|   make -C /lib/modules/$(shell uname -r)/build M=$(PWD) clean
```

### Kbuild

```
EXTRA_CFLAGS = -Wall
obj-m = mymodule.o
```

## Các thư viện Linux Kernel

```
#include <linux/module.h>      /* Needed by all modules */
#include <linux/kernel.h>      /* Needed for KERN_INFO */
#include <linux/init.h>        /* Needed for the macros */

#include <linux/version.h>

#include <linux/types.h>
#include <linux/kdev_t.h>
#include <linux/fs.h>
#include <linux/device.h>
#include <linux/cdev.h>
#include <linux/uaccess.h>
#include <linux/random.h>
```

## Module information

```
MODULE_LICENSE("GPL");
MODULE_AUTHOR("goporo");
MODULE_DESCRIPTION("Module dung de tao so ngau nhien");
MODULE_SUPPORTED_DEVICE("Character device");
```

## Hàm khởi tạo character device driver

- + **alloc\_chrdev\_region** để cấp phát động
- + **class\_create** để tạo lớp device file
- + **cdev\_init** để tạo các cdev thích hợp
- + **cdev\_add** để đăng kí cdev
- + **device\_create** dùng để tạo device file

## Hàm gỡ character device driver

- + **device\_destroy** dùng để hủy các device file
- + **class\_destroy** dùng để hủy lớp device
- + **unregister\_chrdev\_region** dùng để hủy các device major đã đăng kí

### Hàm thao tác với character device file

+ Tạo cấu trúc **file\_operations** dùng để thao tác với device file

```
static const struct file_operations pugs_fops =  
{  
    .owner      = THIS_MODULE,  
    .open       = myOpen,  
    .release    = myRelease,  
    .read       = myRead  
};
```



# 3 Hình ảnh minh họa



```
mingnhat@mingnhat-VirtualBox: ~/Downloads/project3
mingnhat@mingnhat-VirtualBox:~/Downloads$ cd project3
mingnhat@mingnhat-VirtualBox:~/Downloads/project3$ make
make -C /lib/modules/3.13.0-170-generic/build/ M=/home/mingnhat/Downloads/project3 modules
make[1]: Entering directory `/usr/src/linux-headers-3.13.0-170-generic'
CC [M] /home/mingnhat/Downloads/project3/mymodule.o
Building modules, stage 2.
MODPOST 1 modules
CC /home/mingnhat/Downloads/project3/mymodule.mod.o
LD [M] /home/mingnhat/Downloads/project3/mymodule.ko
make[1]: Leaving directory `/usr/src/linux-headers-3.13.0-170-generic'
mingnhat@mingnhat-VirtualBox:~/Downloads/project3$ sudo insmod mymodule.ko
[sudo] password for mingnhat:
insmod: ERROR: could not insert module mymodule.ko: File exists
mingnhat@mingnhat-VirtualBox:~/Downloads/project3$ sudo cat /dev/chardev0
19324
mingnhat@mingnhat-VirtualBox:~/Downloads/project3$ sudo cat /dev/chardev0
-11343
mingnhat@mingnhat-VirtualBox:~/Downloads/project3$ sudo cat /dev/chardev0
-2990
mingnhat@mingnhat-VirtualBox:~/Downloads/project3$ sudo cat /dev/chardev0
-16890
mingnhat@mingnhat-VirtualBox:~/Downloads/project3$
```

## TÀI LIỆU THAM KHẢO

- File trong tài liệu hướng dẫn thực hành của thầy Phạm Tuấn Sơn

([OS Do an 3 Linux Kernel Module.mp4 - Google Drive](#))

Project tham khảo