

LẬP TRÌNH LINUX MODULE KERNEL



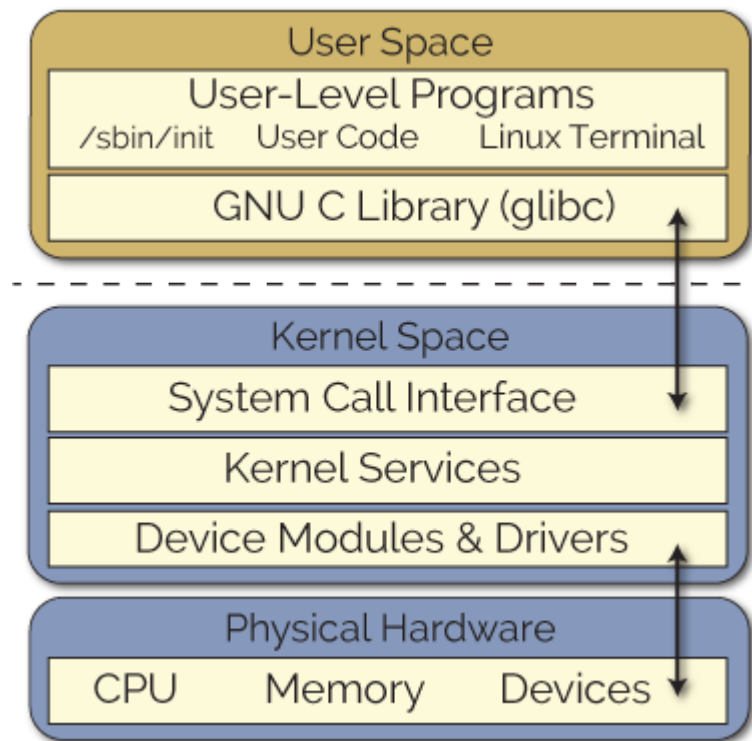
Trương Văn Hoàng – MSSV: 1612857

Đại học Khoa học Tự nhiên

Khoa Công nghệ thông tin

Môn: Hệ điều hành

I. Kernel Module là gì?



Kernel module đơn giản chỉ là file để chúng ta thêm code vào nhằm mục đích mở rộng chức năng của Kernel.

II. Các bước thêm module vào Kernel

1. Tạo module

- Cài đặt Linux Headers:

```

~$ sudo apt-get update
~$ apt-cache search linux-headers-$(uname -r)
~amd64 - Header files for Linux 3.16.0-4-amd64
~$ sudo apt-get install linux-headers-3.16.0-4-amd64
~$ cd /usr/src/linux-headers-3.16.0-4-amd64/

```

- Ta sẽ thực hiện viết một module đơn giản bằng ngôn ngữ C, ở đây chúng ta lưu là `hello.c`

```

1  #include<linux/module.h>
2  #include<linux/init.h>
3  #include<linux/kernel.h>
4
5  static int Hello_init(void){
6      printk(KERN_ALERT "Hello, How are you ?\n");
7      return 0;
8  }
9
10 static void Hello_exit(void){
11     printk(KERN_ALERT "Goodbye, see you again\n");
12 }
13 module_init(Hello_init);
14 module_exit(Hello_exit);

```

2. Biên dịch module

- Muốn biên dịch Module ta cần có file Makefile như sau:

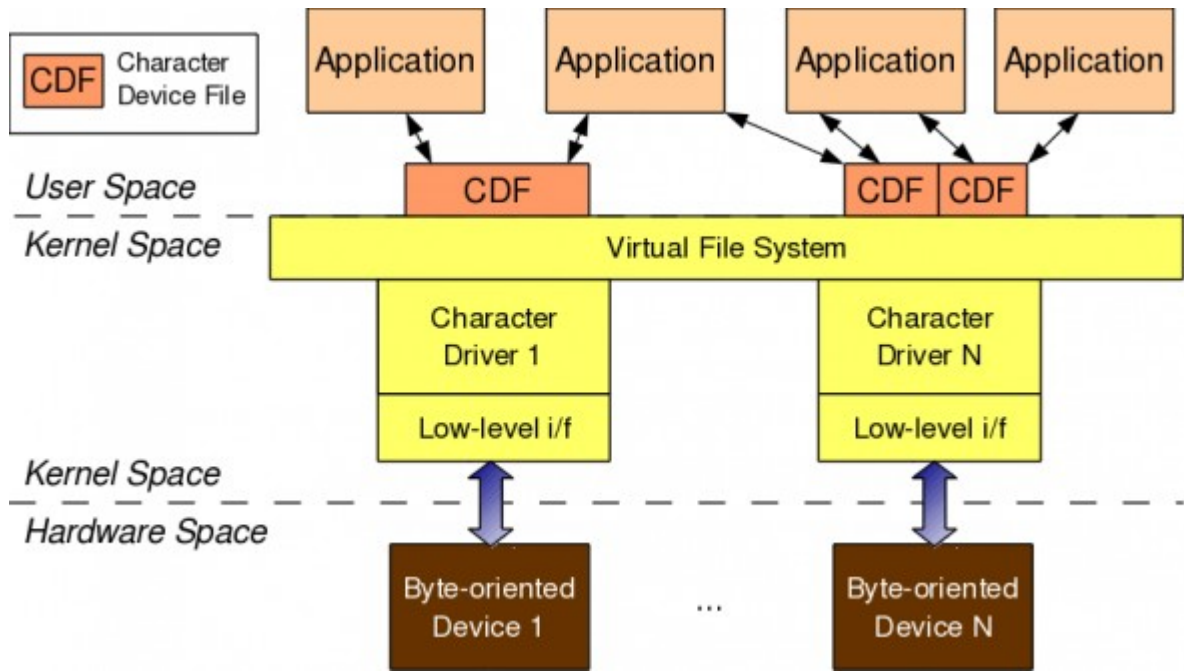
```

1  obj-m+=hello.o
2
3  all:
4      make -C /lib/modules/$(shell uname -r)/build/ M=$(PWD) modules
5  clean:
6      make -C /lib/modules/$(shell uname -r)/build/ M=$(PWD) clean

```

- Chúng ta mở Terminal và vào thư mục chứa file hello.c và Makefile để thực hiện lệnh make, khi đó chúng ta sẽ thu được file hello.ko
- Insert module hello.ko vào Kernel thông qua lệnh \$ sudo su insmod hello.ko
- Như vậy là chúng ta đã insert được module vào Kernel, để gỡ bỏ module đi ta sẽ thực hiện lệnh \$ sudo rmmod hello.ko

III. Character Device



- Bất kỳ một ứng dụng nào ở tầng người dùng (user space) muốn thao tác với một thiết bị kiểu character device trong tầng phần cứng (hardware space) sẽ sử dụng character device driver tương ứng trong tầng nhân (kernel space). Việc sử dụng các character driver được thực hiện thông qua các file thiết bị (device files) tương ứng, được liên kết với driver thông qua hệ thống file ảo (virtual file system – VFS).

- Điều này có nghĩa là các ứng dụng có thể thực hiện các thao tác file thông thường trên các file thiết bị. Các thao tác file này sẽ được VFS diễn giải ra các hàm tương ứng trong driver liên kết với nó. Các hàm này sau đó sẽ thực hiện các truy cập ở mức thấp đến các thiết bị thật sự để đạt được kết quả mong muốn. Tuy nhiên, cần lưu ý rằng mặc dù các ứng dụng truy cập đến các file thiết bị sử dụng các thao tác file thông thường như đối với file dữ liệu (mở, đọc, ghi, đóng, ...), nhưng hiệu quả là khác so với các thao tác thông thường trên file dữ liệu. Ví dụ việc đọc dữ liệu từ thiết bị sẽ không là dữ liệu đã được ghi ra và ngược lại ... Việc kết nối từ ứng dụng đến thiết bị được thực hiện hoàn chỉnh thông qua 4 thực thể chính liên quan gồm:

1. Application (ứng dụng)

2. Character device file (File thiết bị)
3. Character device driver (Driver thiết bị)
4. Character device (Thiết bị)

- Số hiệu file thiết bị (Major và minor number):

Việc kết nối giữa ứng dụng và file thiết bị được thực hiện thông qua tên file thiết bị.

Bài tập : Viết Module tạo một character device để cho phép các tiến trình ở userspace có thể open và read các số ngẫu nhiên

1. File Module

Source code: https://github.com/VanHoangTruong/Linux-Module-Kernel/blob/master/my_module.c

- Trong file my_module.c chúng ta thiết lập để tạo device my_module ở các câu lệnh:

```

47     charClass = class_create(THIS_MODULE,"1612857");
48     if(IS_ERR(charClass)){
49         unregister_chrdev(Major,"my_module");
50         printk(KERN_ALERT "Failed to Register device class\n");
51         return PTR_ERR(charClass);
52     }
53     printk(KERN_ALERT "device class Register correctly\n");
54     charDevice = device_create(charClass, NULL, MKDEV(Major,0),NULL,"my_module");
55     if(IS_ERR(charDevice)){
56         class_destroy(charClass);
57         unregister_chrdev(Major,"my_module");
58         printk(KERN_ALERT "Failed to create device\n");
59         return PTR_ERR(charDevice);
60     }
61     printk(KERN_ALERT "device class created correctly\n");
    
```

- Chúng ta có hàm randomNumber như sau:

LINUX MODULE KERNEL

6

```
23 static ssize_t randomNumber(struct file *filp, char *buffer, size_t length, loff_t *offset)
24 {
25     int randomNumber;
26     get_random_bytes(&randomNumber, sizeof(randomNumber));
27     int ret = copy_to_user(buffer, &randomNumber, 4);
28     if (ret == 0){
29         printk(KERN_ALERT "Sent %ld character to the user\n ", sizeof(randomNumber));
30         return sizeof(randomNumber);
31     } else {
32         printk(KERN_ALERT "Failed to send!\n");
33         return -EFAULT;
34     }
35 }
```

2. File Makefile

```
1  obj-m+=my_module.o
2
3  all:
4      make -C /lib/modules/$(shell uname -r)/build/ M=$(PWD) modules
5      $(CC) random_n.c -o randomNumber
6  clean:
7      make -C /lib/modules/$(shell uname -r)/build/ M=$(PWD) clean
8      rm random_n
```

3. File chương trình để thực hiện Module my_module

Source file:

[https://github.com/VanHoangTruong/Linux-Module-Kernel/blob/master/
random_n.c](https://github.com/VanHoangTruong/Linux-Module-Kernel/blob/master/random_n.c)

- Sau khi thực hiện viết xong 3 file trên ta mở Terminal tại thư mục chứa 3 file trên và thực hiện lệnh make, sau đó insmod rồi công việc còn lại chỉ là test kết quả thu được.

```

root@vanhoang-Vostro-14-3468:/home/vanhoang/Documents/sss# make
make -C /lib/modules/4.15.0-34-generic/build/ M=/home/vanhoang/Documents/sss modules
make[1]: Entering directory '/usr/src/linux-headers-4.15.0-34-generic'
  CC [M] /home/vanhoang/Documents/sss/my_module.o
/home/vanhoang/Documents/sss/my_module.c: In function 'randomNumber':
/home/vanhoang/Documents/sss/my_module.c:27:2: warning: ISO C90 forbids mixed de
clarations and code [-Wdeclaration-after-statement]
  int ret = copy_to_user(buffer,&randomNumber,4);
  ^~~~
Building modules, stage 2.
MODPOST 1 modules
  CC      /home/vanhoang/Documents/sss/my_module.mod.o
  LD [M]  /home/vanhoang/Documents/sss/my_module.ko
make[1]: Leaving directory '/usr/src/linux-headers-4.15.0-34-generic'
cc random_n.c -o randomNumber
root@vanhoang-Vostro-14-3468:/home/vanhoang/Documents/sss# insmod my_module.ko
root@vanhoang-Vostro-14-3468:/home/vanhoang/Documents/sss# ./randomNumber
The random Number is: 855198483
root@vanhoang-Vostro-14-3468:/home/vanhoang/Documents/sss#

```

Trong hình trên ta đã tạo ra được số ngẫu nhiên là **855198483**