

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN HỆ ĐIỀU HÀNH

LINUX KERNEL

Mục lục

I. THÀNH VIÊN	3
II. MỨC ĐỘ ĐÁNH GIÁ	3
III. CHI TIẾT	4
1. Phần 1	4
a) Các phần chính.....	4
b) Chạy module.....	5
2. Phần 2	6
a) Các phần chính.....	6
b) Chạy module.....	9
IV. TÀI LIỆU THAM KHẢO.....	11

I. THÀNH VIÊN

MSSV	Họ và tên
1712026	Lê Trần Hữu Đắc
1712214	Phạm Hoàng Nhật Anh

II. MỨC ĐỘ ĐÁNH GIÁ

Yêu cầu	Mức độ hoàn thành
Phần 1: Viết một module tạo một character device để cho phép các tiến trình ở user space có thể open và read các số ngẫu nhiên được sinh ra ở module này.	100%
Phần 2: Chương trình hook vào một system call: <ul style="list-style-type: none">• 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.	100%

III. CHI TIẾT

1. Phần 1

a) Các phần chính

- Driver thực hiện đăng ký số hiệu và tên module.

```
static dev_t dev_num;
```

- Trong hàm tạo:

```
int ret;

ret = alloc_chrdev_region(& dev_num, 0, 1, "RandomNumber");
Major = MAJOR(dev_num);

if(ret < 0) {
    printk(KERN_INFO "RandomNumber: Registration failed\n");
}
```

- Trong hàm hủy:

```
unregister_chrdev_region(dev_num, 1);
```

- Tạo file thiết bị tự động.

```
static struct class * dev_class;
static struct device * dev;
```

- Trong hàm tạo:

```
dev_class = class_create(THIS_MODULE, "Class_RandomNumber");
if (dev_class == NULL) {
    printk(KERN_INFO "RandomNumber: Failed to create a device
class\n");
    unregister_chrdev_region(dev_num, 1);
    return 0;
}
dev_class->devnode = RandomNumber_devnode; //permission
dev = device_create(dev_class, NULL, dev_num, NULL, "RandomNumber");
if (IS_ERR(dev)){
    printk(KERN_INFO "RandomNumber: Failed to create a device\n");
    class_destroy(dev_class);
    unregister_chrdev_region(dev_num, 1);
    return 0;
}
```

- Trong hàm hủy:

```
device_destroy(dev_class, dev_num);
class_destroy(dev_class);
```

- Thao tác với file thiết bị.

```
static struct cdev c_dev;
static struct file_operations fops = {
    .owner = THIS_MODULE,
    .open = randomNumber_open,
    .read = randomNumber_read,
    .release = randomNumber_close,
};
```

- Trong hàm tạo:

```
cdev_init(& c_dev, & fops);
if (cdev_add(&c_dev, dev_num, 1) == -1){
    device_destroy(dev_class, dev_num);
    class_destroy(dev_class);
    unregister_chrdev_region(dev_num, 1);
    return 0;
}
```

- Hàm read (sinh số ngẫu nhiên).

```
static ssize_t randomNumber_read(struct file *filp, char *buffer,
size_t length, loff_t *offset)
{
    printk(KERN_INFO "RandomNumber: read()\n");
    return get_random_int();
}
```

b) Chạy module

- Dùng lệnh make để tạo file (.ko) và sử dụng lệnh insmod để nạp vào nhân hệ thống.

```
nhatanh@nhatanh-virtual-machine:~/Desktop/hdh/cau1$ make
make -C /lib/modules/5.0.0-32-generic/build M=/home/nhatanh/Desktop/hdh/cau1 modules
make[1]: Entering directory '/usr/src/linux-headers-5.0.0-32-generic'
CC [M] /home/nhatanh/Desktop/hdh/cau1/RandomNumber.o
Building modules, stage 2.
MODPOST 1 modules
CC /home/nhatanh/Desktop/hdh/cau1/RandomNumber.mod.o
LD [M] /home/nhatanh/Desktop/hdh/cau1/RandomNumber.ko
make[1]: Leaving directory '/usr/src/linux-headers-5.0.0-32-generic'
nhatanh@nhatanh-virtual-machine:~/Desktop/hdh/cau1$ sudo insmod RandomNumber.ko
nhatanh@nhatanh-virtual-machine:~/Desktop/hdh/cau1$ dmesg
[ 5022.209444] RandomNumber: Successful device registration with MajorID = 240
[ 5022.210623] RandomNumber: Initialize RandomNumber driver successfully
```

- Tạo file Test.exe và tương tác với driver.

```
nhatanh@nhatanh-virtual-machine:~/Desktop/hdh/cau1$ gcc Test.c -o Test
nhatanh@nhatanh-virtual-machine:~/Desktop/hdh/cau1$ ./Test /dev/RandomNumber
Random number: 1866636278
```

- Dùng rmmod để gỡ bỏ driver.

```
nhatanh@nhatanh-virtual-machine:~/Desktop/hdh/cau1$ sudo rmmod RandomNumber
nhatanh@nhatanh-virtual-machine:~/Desktop/hdh/cau1$ dmesg
[ 6490.662153] RandomNumber: Successful device registration with MajorID = 240
[ 6490.664789] RandomNumber: Initialize RandomNumber driver successfully
[ 6502.509826] RandomNumber: open()
[ 6502.509829] RandomNumber: read()
[ 6502.509946] RandomNumber: close()
[ 6511.283952] RandomNumber: The device has been disconnected from the system
```

2. Phần 2

a) Các phần chính

- Xác định tham số của syscall cần hook.

– Syscall open:

```
asm linkage int (*original_call) (const char __user*, int, mode_t);
```

– Syscall write:

```
asm linkage size_t (*original_call) (unsigned int, const char *,
size_t);
```

- Viết hàm thực thi syscall với đúng tham số của syscall gốc:

– Syscall open:

```
asm linkage int our_sys_open(const char __user* filename, int flags,
mode_t mode)
{
    printk(KERN_INFO "\n");

    printk(KERN_INFO "%s opened %s\n", current->comm, filename);
    return original_call(filename, flags, mode);
}
```

– Syscall write:

```

asmlinkage size_t our_sys_write(unsigned int fd, const char *buf,
size_t nbytes)
{
    size_t wrotebytes = original_call(fd, buf, nbytes);

    char fileName[256];
    int fileDesc = fd;
    struct files_struct *files = current->files;
    char *tmp;
    char *pathname;
    struct file *file;
    struct path *path;

    spin_lock(&files->file_lock);
    file = fcheck_files(files, fileDesc);
    if (!file) {
        spin_unlock(&files->file_lock);
        return -ENOENT;
    }
    path = &file->f_path;
    path_get(path);
    spin_unlock(&files->file_lock);
    tmp = (char *)__get_free_page(GFP_KERNEL);
    if (!tmp) {
        path_put(path);
        return -ENOMEM;
    }
    pathname = d_path(path, tmp, PAGE_SIZE);
    path_put(path);
    if (IS_ERR(pathname)) {
        free_page((unsigned long)tmp);
        return PTR_ERR(pathname);
    }
    strcpy(fileName, pathname);
    free_page((unsigned long)tmp);

    printk(KERN_INFO "\n");

    printk(KERN_INFO "%s opened %s wrote %zu byte(s)\n", current-
>comm, fileName, wrotebytes);

    return wrotebytes;
}

```

- Viết hàm init và exit cho custom syscall:
- Syscall open:

```
static int __init entry_point(void)
{
    printk(KERN_INFO "Hookopen loaded successfully..\n");
    // sys_call_table address in System.map
    system_call_table_addr =
(void*)kallsyms_lookup_name("sys_call_table");
    original_call = system_call_table_addr[__NR_open];

    make_rw((unsigned long)system_call_table_addr);
    system_call_table_addr[__NR_open] = our_sys_open;
    return 0;
}
```

```
static void __exit exit_point(void)
{
    printk(KERN_INFO "Unloaded hookopen successfully\n");
    // Restore the original call
    system_call_table_addr[__NR_open] = original_call;
    make_ro((unsigned long)system_call_table_addr);
}
```

- Syscall write:

```
static int __init entry_point(void)
{
    printk(KERN_INFO "Hookwrite loaded successfully\n");
    // sys_call_table address in System.map
    system_call_table_addr =
(void*)kallsyms_lookup_name("sys_call_table");
    original_call = system_call_table_addr[__NR_write];

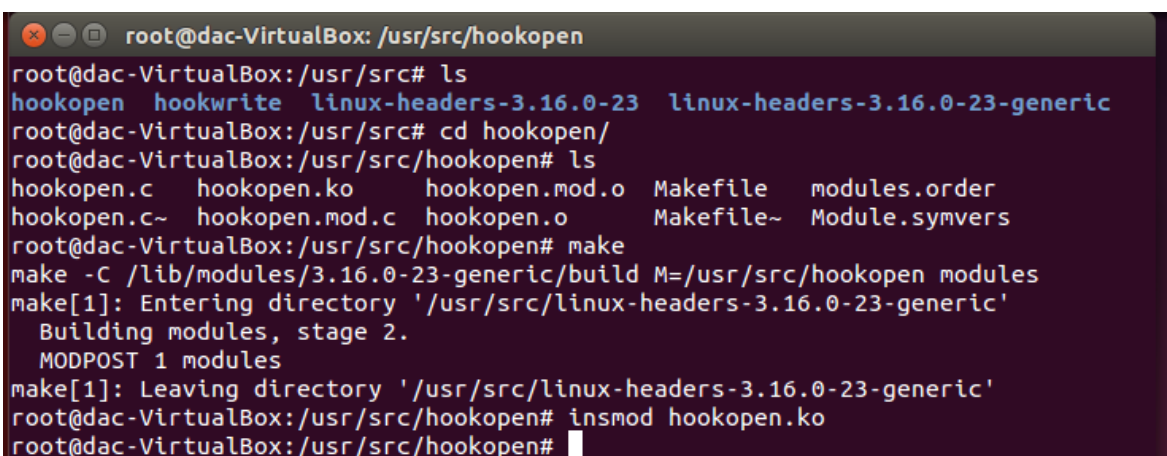
    make_rw((unsigned long)system_call_table_addr);
    system_call_table_addr[__NR_write] = our_sys_write;
    return 0;
}
```



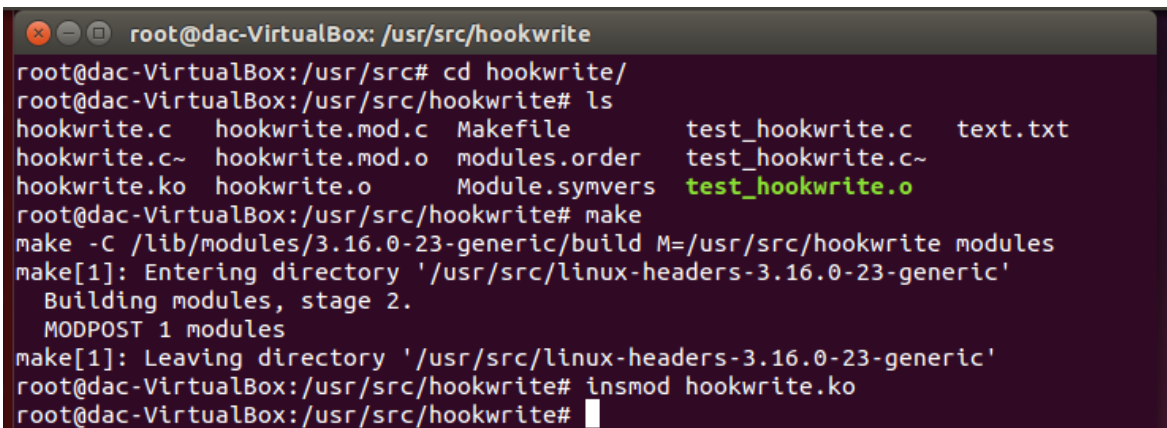
```
static void __exit exit_point(void)
{
    printk(KERN_INFO "Unloaded hookwrite successfully\n");
    // Restore the original call
    system_call_table_addr[__NR_write] = original_call;
    make_ro((unsigned long)system_call_table_addr);
}
```

b) Chạy module

- Phiên bản header linux sử dụng sẽ là linux-3.16.0-23-generic.
- Dùng lệnh make để tạo file (.ko) và sử dụng lệnh insmod để nạp vào nhân hệ thống.

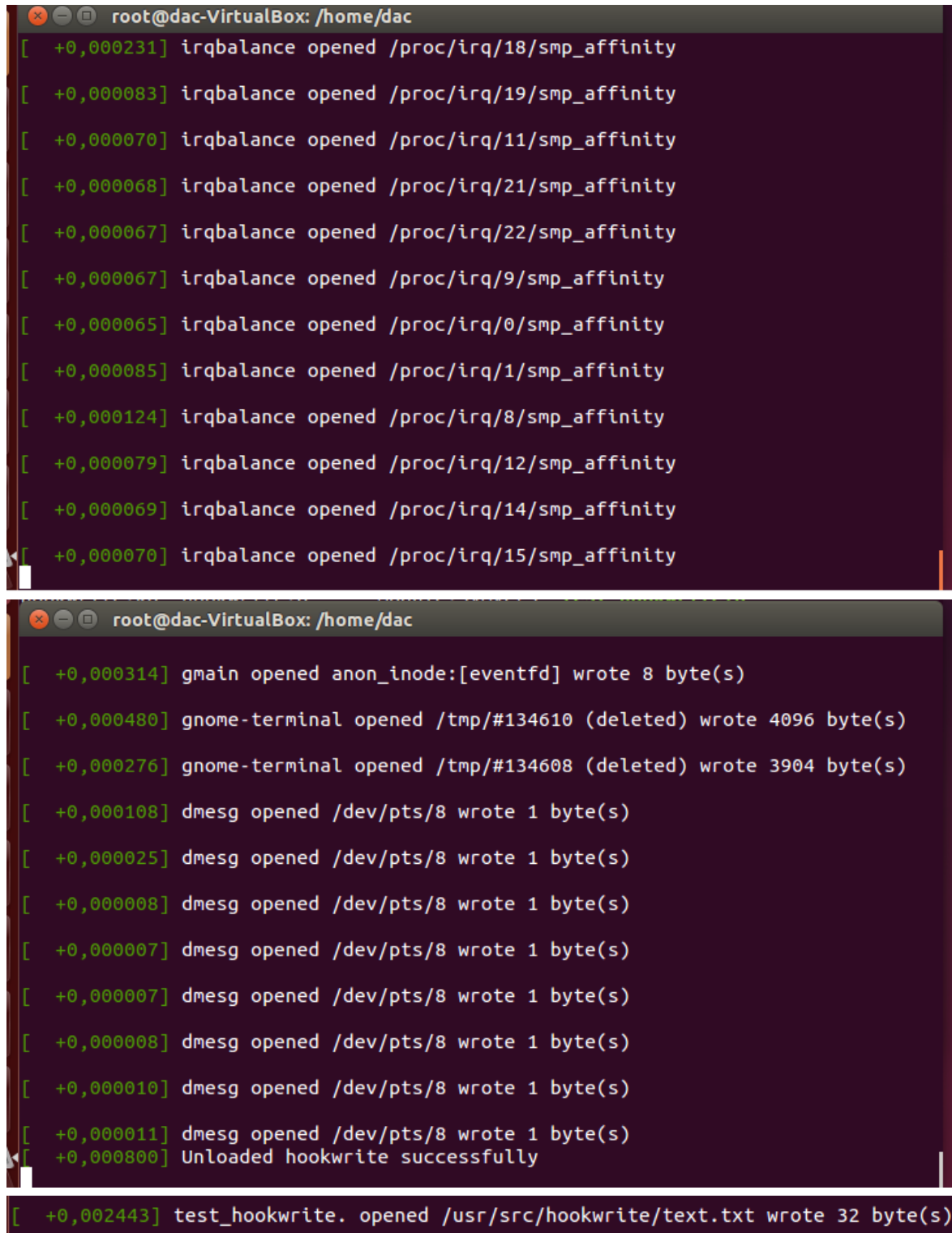


```
root@dac-VirtualBox: /usr/src/hookopen
root@dac-VirtualBox:/usr/src# ls
hookopen  hookwrite  linux-headers-3.16.0-23  linux-headers-3.16.0-23-generic
root@dac-VirtualBox:/usr/src# cd hookopen/
root@dac-VirtualBox:/usr/src/hookopen# ls
hookopen.c  hookopen.ko  hookopen.mod.o  Makefile  modules.order
hookopen.c~ hookopen.mod.c hookopen.o      Makefile~  Module.symvers
root@dac-VirtualBox:/usr/src/hookopen# make
make -C /lib/modules/3.16.0-23-generic/build M=/usr/src/hookopen modules
make[1]: Entering directory '/usr/src/linux-headers-3.16.0-23-generic'
Building modules, stage 2.
MODPOST 1 modules
make[1]: Leaving directory '/usr/src/linux-headers-3.16.0-23-generic'
root@dac-VirtualBox:/usr/src/hookopen# insmod hookopen.ko
root@dac-VirtualBox:/usr/src/hookopen#
```



```
root@dac-VirtualBox: /usr/src/hookwrite
root@dac-VirtualBox:/usr/src# cd hookwrite/
root@dac-VirtualBox:/usr/src/hookwrite# ls
hookwrite.c  hookwrite.mod.c  Makefile  test_hookwrite.c  text.txt
hookwrite.c~ hookwrite.mod.o  modules.order  test_hookwrite.c~
hookwrite.ko hookwrite.o      Module.symvers  test_hookwrite.o
root@dac-VirtualBox:/usr/src/hookwrite# make
make -C /lib/modules/3.16.0-23-generic/build M=/usr/src/hookwrite modules
make[1]: Entering directory '/usr/src/linux-headers-3.16.0-23-generic'
Building modules, stage 2.
MODPOST 1 modules
make[1]: Leaving directory '/usr/src/linux-headers-3.16.0-23-generic'
root@dac-VirtualBox:/usr/src/hookwrite# insmod hookwrite.ko
root@dac-VirtualBox:/usr/src/hookwrite#
```

- Kết quả chạy



The image displays two screenshots of a terminal window running in a VirtualBox environment. The terminal title bar indicates the user is 'root' at 'dac-VirtualBox' in the directory '/home/dac'. The first screenshot shows a series of log messages from the 'irqbalance' service, indicating it has opened various files in the '/proc/irq' directory to manage CPU affinity for different interrupt requests (IRQs). The second screenshot shows log messages from 'gmain', 'gnome-terminal', and 'dmesg', along with a successful 'hookwrite' operation. The third line of the second screenshot shows 'test_hookwrite.' opening a file in '/usr/src/hookwrite/text.txt'.

```
root@dac-VirtualBox: /home/dac
[ +0,000231] irqbalance opened /proc/irq/18/smp_affinity
[ +0,000083] irqbalance opened /proc/irq/19/smp_affinity
[ +0,000070] irqbalance opened /proc/irq/11/smp_affinity
[ +0,000068] irqbalance opened /proc/irq/21/smp_affinity
[ +0,000067] irqbalance opened /proc/irq/22/smp_affinity
[ +0,000067] irqbalance opened /proc/irq/9/smp_affinity
[ +0,000065] irqbalance opened /proc/irq/0/smp_affinity
[ +0,000085] irqbalance opened /proc/irq/1/smp_affinity
[ +0,000124] irqbalance opened /proc/irq/8/smp_affinity
[ +0,000079] irqbalance opened /proc/irq/12/smp_affinity
[ +0,000069] irqbalance opened /proc/irq/14/smp_affinity
[ +0,000070] irqbalance opened /proc/irq/15/smp_affinity

root@dac-VirtualBox: /home/dac
[ +0,000314] gmain opened anon_inode:[eventfd] wrote 8 byte(s)
[ +0,000480] gnome-terminal opened /tmp/#134610 (deleted) wrote 4096 byte(s)
[ +0,000276] gnome-terminal opened /tmp/#134608 (deleted) wrote 3904 byte(s)
[ +0,000108] dmesg opened /dev/pts/8 wrote 1 byte(s)
[ +0,000025] dmesg opened /dev/pts/8 wrote 1 byte(s)
[ +0,000008] dmesg opened /dev/pts/8 wrote 1 byte(s)
[ +0,000007] dmesg opened /dev/pts/8 wrote 1 byte(s)
[ +0,000007] dmesg opened /dev/pts/8 wrote 1 byte(s)
[ +0,000008] dmesg opened /dev/pts/8 wrote 1 byte(s)
[ +0,000010] dmesg opened /dev/pts/8 wrote 1 byte(s)
[ +0,000011] dmesg opened /dev/pts/8 wrote 1 byte(s)
[ +0,000800] Unloaded hookwrite successfully

[ +0,002443] test_hookwrite. opened /usr/src/hookwrite/text.txt wrote 32 byte(s)
```

IV. TÀI LIỆU THAM KHẢO

Các file pdf hướng dẫn

<https://sites.google.com/site/embedded247/ddcourse/kernelmoduleprogramming>

<https://vimentor.com/vi/lesson/device-file>