



《操作系统》课第十二次实验报告

学院:	软件学院
姓名:	郁万祥
学号:	2013852
邮箱:	yuwanxiang0114@163.com
时间:	2022.12.2

0. 开篇感言

此次实验，一直会出现一个问题，就是在/dev 目录下虽然已经创建了lkm_example，但是在进行 cat /dev/lkm_example 的时候，会出现问题：cat: /dev/lkm_example: 没有那个设备或地址，问题如图：

```
ywx2013852@ywx2013852-virtual-machine: /dev$ ls
core      loop3      shm        tty28      tty56      ttyS25     vcsa5
cpu       loop4      snapshot  tty29      tty57      ttyS26     vcsa6
cpu_dma_latency loop5      snd        tty3       tty58      ttyS27     vcsu
cuse      loop6      sr0        tty30      tty59      ttyS28     vcsu1
disk      loop7      sr1        tty31      tty6       ttyS29     vcsu2
dma_heap  loop8      stderr     tty32      tty60      ttyS3      vcsu3
dri       loop9      stdin      tty33      tty61      ttyS30     vcsu4
ecryptfs  loop-control stdout     tty34      tty62      ttyS31     vcsu5
fb0       mapper     test       tty35      tty63      ttyS4      vcsu6
fd        mcelog     tty        tty36      tty7       ttyS5      vfio
fd0       mem        tty0       tty37      tty8       ttyS6      vga_arbiter
full      queue      tty1       tty38      tty9       ttyS7      vhci
fuse      net        tty10      tty39      ttyprintk  ttyS8      vhost-net
hidraw0   null       tty11      tty4       ttyS0      ttyS9      vhost-vsock
hpet      nvram      tty12      tty40      ttyS1      udnabuf    vnci
hugepages port        tty13      tty41      ttyS10     uhid       vsock
hwrng     ppp        tty14      tty42      ttyS11     uinput     zero
initctl   psaux      tty15      tty43      ttyS12     urandom    zfs
input     ptmx       tty16      tty44      ttyS13     userio
kmsg      pts        tty17      tty45      ttyS14     vcs
lkm_example random     tty18      tty46      ttyS15     vcs1
log       rfkill     tty19      tty47      ttyS16     vcs2
loop0     rtc        tty2       tty48      ttyS17     vcs3

ywx2013852@ywx2013852-virtual-machine: /dev$
ywx2013852@ywx2013852-virtual-machine: ~/Desktop$ cat /dev/lkm_example
cat: /dev/lkm_example: 没有那个设备或地址
ywx2013852@ywx2013852-virtual-machine: ~/Desktop$
```

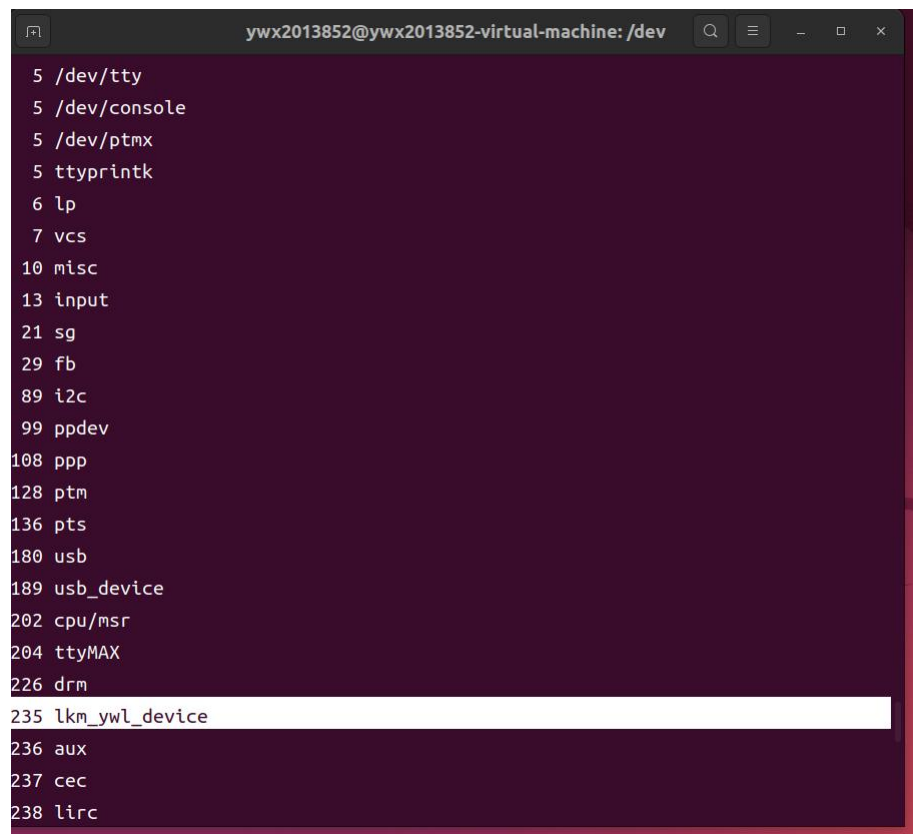


经过查询和通过同学之间的讨论，问题的出现的原因主要可以从两个方面入手：

1、设备号可能不同，可能是 237，也可能是 238，(我的甚至是 235!!)

这里介绍一个查看主设备号的方法：

先按照正常步骤添加设备，如果能够正常的 cat，就没有问题，但是如果不能的话，不要着急 `sudo rm lab12ywx` 来删除设备，可以先使用命令 `cat /proc/devices` 查看一下添加设备的主设备号



```
ywx2013852@ywx2013852-virtual-machine: /dev
5 /dev/tty
5 /dev/console
5 /dev/ptmx
5 ttyprintk
6 lp
7 vcs
10 misc
13 input
21 sg
29 fb
89 i2c
99 ppdev
108 ppp
128 ptm
136 pts
180 usb
189 usb_device
202 cpu/msr
204 ttyMAX
226 drm
235 lkm_ywl_device
236 aux
237 cec
238 lirc
```

就像这里，我的是 235。

2、如果你已经在/dev 中添加设备了，就需要重新启动，这样添加的设备才可以被删除，才可以更换信息，重新添加设备。



1. 实验题目

Linux kernel Module Development (2)

2. 实验目标

- 1、使用 C 语言，编写程序，实现一个 linux 内核模块，可以读取和写一个指定的文件。
- 2、使用 Makefile 文件，进行编译命令的编辑。

3. 原理方法

1、linux 内核模块：模块是在内核空间运行的程序，实际上是一种目标对象文件，没有链接，不能独立运行，但是其代码可以在运行时链接到系统中作为内核的一部分运行或从内核中取下，从而可以动态扩充内核的功能。这种目标代码通常由一组函数和数据结构组成，用来实现一种文件系统，一个驱动程序，或其它内核上层的功能。模块机制的完整叫法应该是动态可加载内核模块，一般就简称为模块。

2、sysfs 是一个基于内存的文件系统，它的作用是将内核信息以文件的方式提供给用户程序使用。该文件系统的目录层次结构严格按照内核的数据结构组织。除了二进制文件外，sysfs 文件内容均以 ASCII 格式保存，且一个文件只保存一个数据，另外，一个文件不可大于一个内存页（通常为 4096 字节）。项目要求将 char 数组保存到属性文件。



4. 具体步骤

1、使用 C 语言创建一个内核 module，实现读取和写一个指定的文件的功能。



```
#define EXAMPLE_MSG "Hello World YangWanli 2013774!\n" //设备初始信息
#define MSG_BUFFER_LEN 128 //设备文件自定义存储大小

static int device_open(struct inode *, struct file *);
static int device_release(struct inode *, struct file *);
static ssize_t device_read(struct file *, char *, size_t, loff_t *);
static ssize_t device_write(struct file *, const char *, size_t, loff_t *);

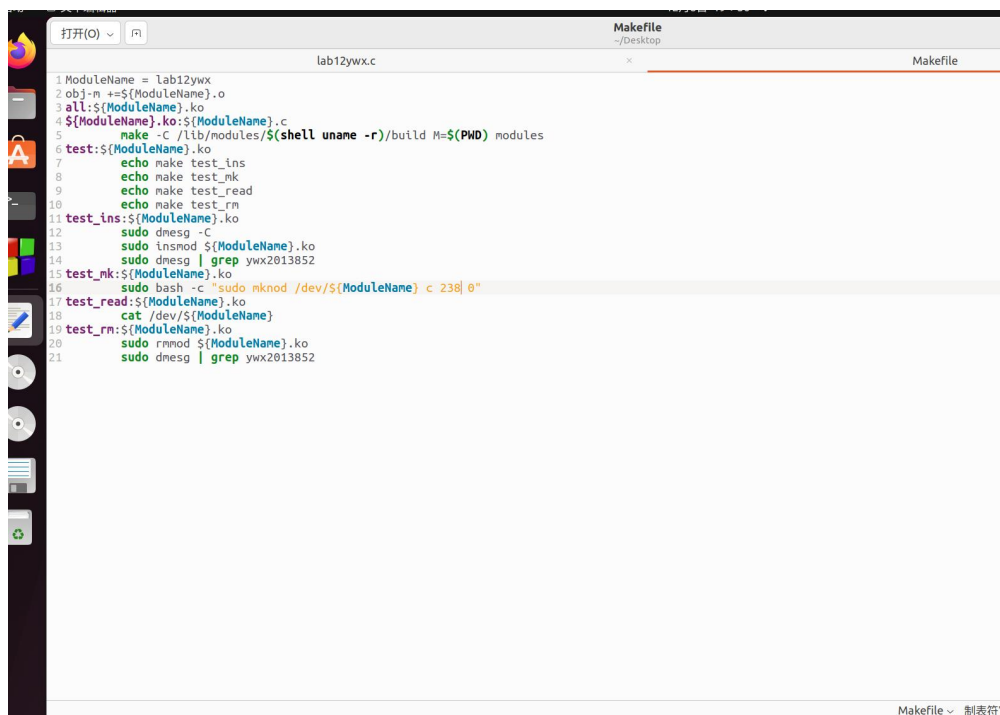
static int major_num; //设备号
static int device_open_count = 0; //设备打开次数
static char msg_buffer[MSG_BUFFER_LEN]; //设备文件的缓冲区 (存储区)
static char *msg_ptr; //读文件的指针
static int public_offset; //标识写文件的offset

//声明各个功能
/* This structure points to all of the device functions */
static struct file_operations file_ops = {
    .read = device_read,
    .write = device_write,
    .open = device_open,
    .release = device_release
};

static ssize_t device_read(struct file *flp, char *buffer, size_t len, loff_t *offset)
{
    int bytes_read = 0;
    printk(KERN_INFO "[2013774ywl] Read from /dev/%s c %d 0\n", DEVICE_NAME, major_num);
    if (*msg_ptr == 0) {
        msg_ptr = msg_buffer;
    }
    while (len && *msg_ptr) {
        //逐个字符读取
        put_user(*(msg_ptr++), buffer++);
        len--;
        bytes_read++;
    }
    printk("ywx2013852 %d bytes read\n", bytes_read);
    return bytes_read;
}

static ssize_t device_write(struct file *flp, const char *buffer, size_t len, loff_t *offset)
{
    int value;
    int flag;
    flag = O_APPEND & flp->f_flags;
    if (O_APPEND == flp->f_flags)
```

2、为 Module 编写一个 Makefile。



```
1 ModuleName = lab12ywx
2 obj-m += $(ModuleName).o
3 all:$(ModuleName).ko
4 $(ModuleName).ko:$(ModuleName).c
5     make -C /lib/modules/$(shell uname -r)/build M=$(PWD) modules
6 test:$(ModuleName).ko
7     echo make test_ins
8     echo make test_mk
9     echo make test_read
10    echo make test_rm
11 test_ins:$(ModuleName).ko
12    sudo dmesg -c
13    sudo insmod $(ModuleName).ko
14    sudo dmesg | grep ywx2013852
15 test_mk:$(ModuleName).ko
16    sudo bash -c "sudo mknod /dev/$(ModuleName) c 238 0"
17 test_read:$(ModuleName).ko
18    cat /dev/$(ModuleName)
19 test_rm:$(ModuleName).ko
20    sudo rmmod $(ModuleName).ko
21    sudo dmesg | grep ywx2013852
```



3、Module 的插入与卸载。

我们可以进入/dev 下使用命令 ls 查看此时的设备列表里面是否含有刚刚添加的

lab12ywx

```
ywx2013852@ywx2013852-virtual-machine: /dev$ ls
core          loop3      shm        tty29  tty57    ttyS26  vcsa6
cpu           loop4      snapshot  tty3   tty58    ttyS27  vcsu
cpu_dma_latency loop5      snd        tty30  tty59    ttyS28  vcsu1
cuse          loop6      sr0        tty31  tty6     ttyS29  vcsu2
disk          loop7      sr1        tty32  tty60    ttyS3   vcsu3
dma_heap      loop8      stderr     tty33  tty61    ttyS30  vcsu4
dri           loop9      stdin      tty34  tty62    ttyS31  vcsu5
ecryptfs      loop-control stdout     tty35  tty63    ttyS4   vcsu6
fb0           mapper     tty        tty36  tty7     ttyS5   vfio
fd            mcelog     tty0       tty37  tty8     ttyS6   vga_arbiter
fd0           mem        tty1       tty38  tty9     ttyS7   vhci
full          queue      tty10      tty39  ttyprintk ttyS8   vhost-net
fuse          net        tty11      tty4   ttyS0    ttyS9   vhost-vsock
hidraw0       null       tty12      tty40  ttyS1    udmabuf vmci
hpet          nvram      tty13      tty41  ttyS10   uhid    vsock
hugepages     port       tty14      tty42  ttyS11   uinput  zero
hwrng         ppp        tty15      tty43  ttyS12   urandom zfs
initctl       psaux     tty16      tty44  ttyS13   userio
input         ptmx       tty17      tty45  ttyS14   vcs
kmsg          pts        tty18      tty46  ttyS15   vcs1
lab12ywx      random     tty19      tty47  ttyS16   vcs2
log           rfkill     tty2       tty48  ttyS17   vcs3
loop0         rtc        tty20      tty49  ttyS18   vcs4
ywx2013852@ywx2013852-virtual-machine: /dev$
```

4、功能的测试

Make:

```
ywx2013852@ywx2013852-virtual-machine: ~/Desktop$ make
make -C /lib/modules/5.15.0-56-generic/build M=/home/ywx2013852/Desktop modules
make[1]: 进入目录"/usr/src/linux-headers-5.15.0-56-generic"
CC [M] /home/ywx2013852/Desktop/lab12ywx.o
MODPOST /home/ywx2013852/Desktop/Module.symvers
CC [M] /home/ywx2013852/Desktop/lab12ywx.mod.o
LD [M] /home/ywx2013852/Desktop/lab12ywx.ko
BTF [M] /home/ywx2013852/Desktop/lab12ywx.ko
Skipping BTF generation for /home/ywx2013852/Desktop/lab12ywx.ko due to unavailability of vmlinux
make[1]: 离开目录"/usr/src/linux-headers-5.15.0-56-generic"
```

Make test:



```
ywx2013852@ywx2013852-virtual-machine:~/Desktop$ make test
echo make test_ins
make test_ins
echo make test_mk
make test_mk
echo make test_read
make test_read
echo make test_rm
make test_rm
```

Make test_ins:

```
ywx2013852@ywx2013852-virtual-machine:~/Desktop$ make test_ins
sudo dmesg -C
sudo insmod lab12ywx.ko
sudo dmesg | grep ywx2013852
[ 1557.301939] ywx2013852 lab12ywx module loaded with device major number 235
```

Make test_mk:

```
ywx2013852@ywx2013852-virtual-machine:~/Desktop$ make test_mk
sudo bash -c "sudo mknod /dev/lab12ywx c 235 0"
ywx2013852@ywx2013852-virtual-machine:~/Desktop$
```

Make test_test:

```
ywx2013852@ywx2013852-virtual-machine: ~/Desktop
52!Hello World YuWanxiang 2013852!Hello World YuWanxiang 2013852!Hello World YuW
anxiang 2013852!Hello World YuWanxiang 2013852!Hello World YuWanxiang 2013852!He
llo World YuWanxiang 2013852!Hello World YuWanxiang 2013852!Hello World YuWanxia
ng 2013852!Hello World YuWanxiang 2013852!Hello World YuWanxiang 2013852!Hello W
orld YuWanxiang 2013852!Hello World YuWanxiang 2013852!Hello World YuWanxiang 20
13852!Hello World YuWanxiang 2013852!Hello World YuWanxiang 2013852!Hello World
YuWanxiang 2013852!Hello World YuWanxiang 2013852!Hello World YuWanxiang 201385
2!Hello World YuWanxiang 2013852!Hello World YuWanxiang 2013852!Hello World YuW
anxiang 2013852!Hello World YuWanxiang 2013852!Hello World YuWanxiang 2013852!H
ello World YuWanxiang 2013852!Hello World YuWanxiang 2013852!Hello World YuWanxi
ang 2013852!Hello World YuWanxiang 2013852!Hello World YuWanxiang 2013852!Hello
World YuWanxiang 2013852!Hello World YuWanxiang 2013852!Hello World YuWanxiang
2013852!Hello World YuWanxiang 2013852!Hello World YuWanxiang 2013852!Hello Wor
ld YuWanxiang 2013852!Hello World YuWanxiang 2013852!Hello World YuWanxiang 2013
852!Hello World YuWanxiang 2013852!Hello World YuWanxiang 2013852!Hello World Yu
Wanxiang 2013852!Hello World YuWanxiang 2013852!Hello World YuWanxiang 2013852!H
ello World YuWanxiang 2013852!Hello World YuWanxiang 2013852!Hello World YuWanxi
ang 2013852!Hello World YuWanxiang 2013852!Hello World YuWanxiang 2013852!Hello
World YuWanxiang 2013852!Hello World YuWanxiang 2013852!Hello World YuWanxiang 2
013852!Hello World YuWanxiang 2013852!Hello World YuWanxiang 2013852!Hello World
YuWanxiang 2013852!Hello World YuWanxiang 2013852!Hello World YuWanxiang 201385
2!Hello World YuWanxiang 2013852!Hello World YuWanxiang 2013852!Hello World YuW
anxiang 2013852!Hello World YuWanxiang 2013852!Hello World YuWanxiang 2013852!H
ello World YuWanxiang 2013852!^Z
[1]+ 已停止                  make test_read
ywx2013852@ywx2013852-virtual-machine:~/Desktop$
```



Make test_rm:

```
ywx2013852@ywx2013852-virtual-machine:~/Desktop$ make test_rm
sudo rmmod lab12ywx.ko
[sudo] ywx2013852 的密码:
sudo dmesg | grep ywx2013852
[ 1540.359123] ywx2013852 Close /dev/lkm_ywl_device c 235 0
```

5. 总结心得

之前的实验过程中，总是抱怨编译内核的时间过于长，每次都在 waiting，其实，问题总比困难多，出现了问题，自然就会产生相应的解决方式，无论是操作系统，还是其他的技术，都是在这个过程中不断发展进步的。

6. 参考资料

源码:

lab12ywx:

```
#include <linux/init.h>
#include <linux/module.h>
#include <linux/kernel.h>
#include <linux/fs.h>
#include <linux/uaccess.h>

MODULE_LICENSE("GPL");
MODULE_AUTHOR("YuWanxiang");
MODULE_DESCRIPTION("A simple example of linux module.");
MODULE_VERSION("0.01");

#define DEVICE_NAME "lkm_ywl_device" //设备名称
#define EXAMPLE_MSG "Hello World YuWanxiang 2013852!\n" //设备初始信息
#define MSG_BUFFER_LEN 128 //设备文件自定义存储大小

static int device_open(struct inode *, struct file *);
static int device_release(struct inode *, struct file *);
```



```
static ssize_t device_read(struct file *, char *, size_t, loff_t *);
static ssize_t device_write(struct file *, const char *, size_t, loff_t *);

static int major_num; //设备号
static int device_open_count = 0; //设备打开次数
static char msg_buffer[MSG_BUFFER_LEN]; //设备文件的缓冲区（存储区）
static char *msg_ptr; //读文件的指针
static int public_offset; //标识写文件的 offset

//声明各个功能
/* This structure points to all of the device functions */
static struct file_operations file_ops = {
    .read = device_read,
    .write = device_write,
    .open = device_open,
    .release = device_release
};

static ssize_t device_read(struct file *flip, char *buffer, size_t len, loff_t *offset)
{
    int bytes_read = 0;
    printk(KERN_INFO "[2013774ywl] Read from /dev/%s c %d 0\n", DEVICE_NAME, major_num);
    if (*msg_ptr == 0) {
        msg_ptr = msg_buffer;
    }
    while (len && *msg_ptr) {
        //逐个字符读取
        put_user(*(msg_ptr++), buffer++);
        len--;
        bytes_read++;
    }
    printk("ywx2013852 %d bytes read\n", bytes_read);
    return bytes_read;
}

static ssize_t device_write(struct file *flip, const char *buffer, size_t len, loff_t *offset)
{
    int value;
    int flag;
    flag = O_APPEND & flip->f_flags;
    if(O_APPEND == flag){
```




```
        printk("ywx2013852 append\n");
        value = copy_from_user(msg_buffer + public_offset, buffer, len);
        public_offset += len;
    }
    else if(flag == 0){
        printk("ywx2013852 overlay\n");
        memset(msg_buffer, '\0', MSG_BUFFER_LEN);
        value = copy_from_user(msg_buffer, buffer, len);
        public_offset = len;
    }
    printk(KERN_INFO "ywx2013852 length of message:%d\n", (int)len);
    return len;
}

static int device_open(struct inode *inode, struct file *file) {
    printk(KERN_INFO "ywx2013852 Open /dev/%s c %d 0\n", DEVICE_NAME, major_num);

    if (device_open_count) {
        return -EBUSY;
    }
    device_open_count++;
    try_module_get(THIS_MODULE);
    return 0;
}

static int device_release(struct inode *inode, struct file *file)
{
    printk(KERN_INFO "ywx2013852 Close /dev/%s c %d 0\n", DEVICE_NAME, major_num);
    device_open_count--;
    module_put(THIS_MODULE);
    return 0;
}

static int __init lkm_example_init(void)
{
    strncpy(msg_buffer, EXAMPLE_MSG, 31);
    public_offset = 31;

    msg_ptr = msg_buffer;
    major_num = register_chrdev(0, DEVICE_NAME, &file_ops);
    if (major_num < 0) {
        printk(KERN_ALERT "ywx2013852 Could not register device: %d\n", major_num);
```



```
        return major_num;
    } else {
        printk(KERN_INFO "ywx2013852 lab12ywx module loaded with device major number %d\n",
major_num);
        return 0;
    }
}

static void __exit lkm_example_exit(void)
{
    unregister_chrdev(major_num, DEVICE_NAME);
    printk(KERN_INFO "Goodbye World YuWanxiang lab12 !\n");
}

/* Register module functions */
module_init(lkm_example_init);
module_exit(lkm_example_exit);
```

Makefile:

```
ModuleName = lab12ywx
obj-m += ${ModuleName}.o
all:${ModuleName}.ko
${ModuleName}.ko:${ModuleName}.c
    make -C /lib/modules/$(shell uname -r)/build M=$(PWD) modules
test:${ModuleName}.ko
    echo make test_ins
    echo make test_mk
    echo make test_test
    echo make test_rm
test_ins:${ModuleName}.ko
    sudo dmesg -C
    sudo insmod ${ModuleName}.ko
    sudo dmesg | grep ywx2013852
test_mk:${ModuleName}.ko
    sudo bash -c "sudo mknod /dev/${ModuleName} c 238 0"
test_test:${ModuleName}.ko
    cat /dev/${ModuleName}
test_rm:${ModuleName}.ko
    sudo rmmod ${ModuleName}.ko
    sudo dmesg | grep ywx2013852
```