实验 18 动态申请字符类设备号

18.1 本章导读

前一期实验已经介绍了如何静态申请字符类设备号，这个实验给大家介绍如何动态申请设备号。

18.1.1 工具

18.1.1.1 硬件工具

1）iTOP4412 开发板

2）U 盘或者 TF 卡

3）PC 机

4）串口

18.1.1.2 软件工具

1）虚拟机 Vmware

2）Ubuntu12.04.2

3）超级终端（串口助手）

4）源码文件夹“request_ascdev_num”

18.1.2 预备课程

实验 17_静态申请字符类设备号

18.1.3 视频资源

本节配套视频为“视频 18_动态申请字符类设备号”

18.2 学习目标

本章需要学习以下内容：

动态申请字符类设备号

18.3 实验操作

需要的基础知识在“实验 17_静态申请字符类设备号”中都已经介绍过了，包括设备申请函数 alloc_chrdev_region()，dev_t 设备号，以及头文件“include/linux/fs.h”。

这里简单介绍一下动态申请函数。

extern int alloc_chrdev_region(dev_t *, unsigned, unsigned, const char *);

函数有四个参数

参数*dev，存放返回的设备号;

参数 unsigned，一般为 0；

参数 unsigned，次设备号连续编号范围；

参数 const char *，设备名称；

函数调用成功则返回 0,反之返回-1。

将“实验 17_静态申请字符类设备号”中的文件“request_cdev_num.c”改为“request_ascdev_num.c”，然后添加动态申请设备号的代码，如下图所示。

```
static int scdev_init(void)
{
    int ret = 0;
    dev_t num_dev;


    printk(KERN_EMERG "numdev_major is %d!\n",numdev_major);
    printk(KERN_EMERG "numdev_minor is %d!\n",numdev_minor);

    if(numdev_major){
        num_dev = MKDEV(numdev_major,numdev_minor);
        ret = register_chrdev_region(num_dev,DEVICE_MINOR_NUM,DEVICE_NAME);
    }
    else{
        /*动态注册设备号*/
        ret = alloc_chrdev_region(&num_dev,numdev_minor,DEVICE_MINOR_NUM,DEVICE_NAME);
        /*获得主设备号*/
        numdev_major = MAJOR(num_dev);
        printk(KERN_EMERG "adev_region req %d !\n",numdev_major);
    }

    if(ret<0){
        printk(KERN_EMERG "register_chrdev_region req %d is failed!\n",numdev_major);
    }

    printk(KERN_EMERG "scdev_init!\n");
    /*打印信息，KERN_EMERG表示紧急信息*/
    return 0;
}
```

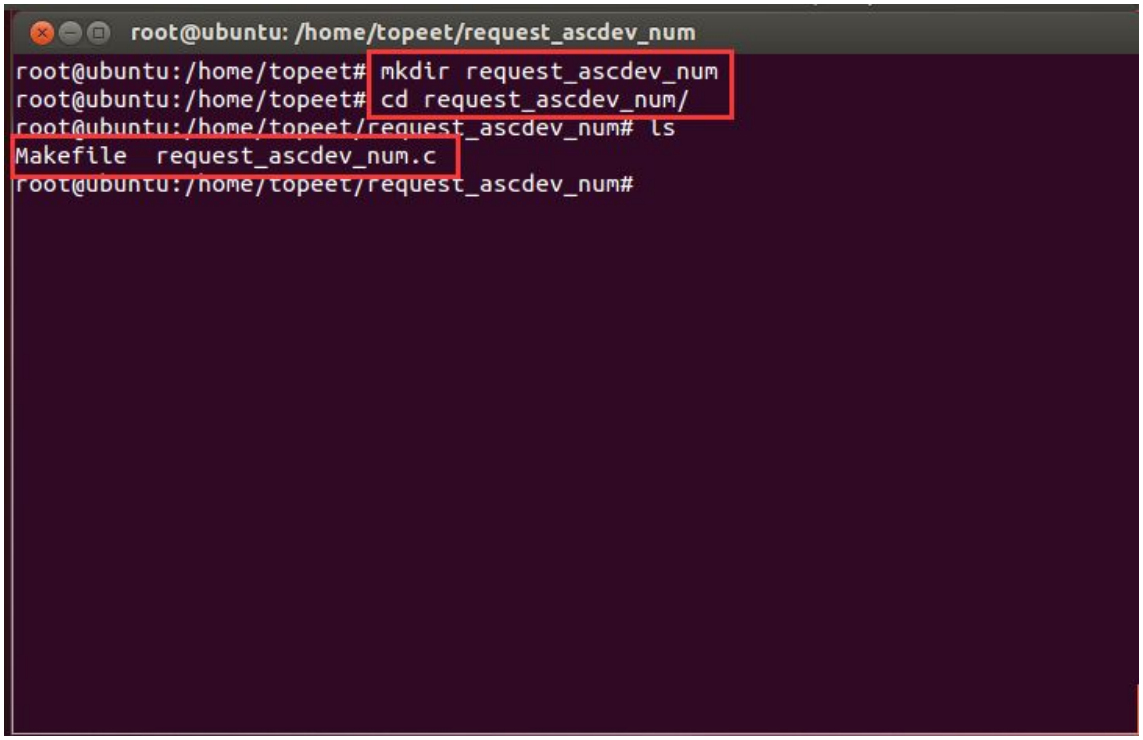
将设备节点宏定义由“sscdev”改为“ascdev”，如下图所示。

```
#define DEVICE_NAME "ascdev"
#define DEVICE_MINOR_NUM 2
#define DEV_MAJOR 0
#define DEV_MINOR 0

MODULE_LICENSE("Dual BSD/GPL");
/*声明是开源的，没有内核版本限制*/
MODULE_AUTHOR("iTOPEET_dz");
/*声明作者*/

int numdev_major = DEV_MAJOR;
int numdev_minor = DEV_MINOR;

/*输入主设备号*/
```

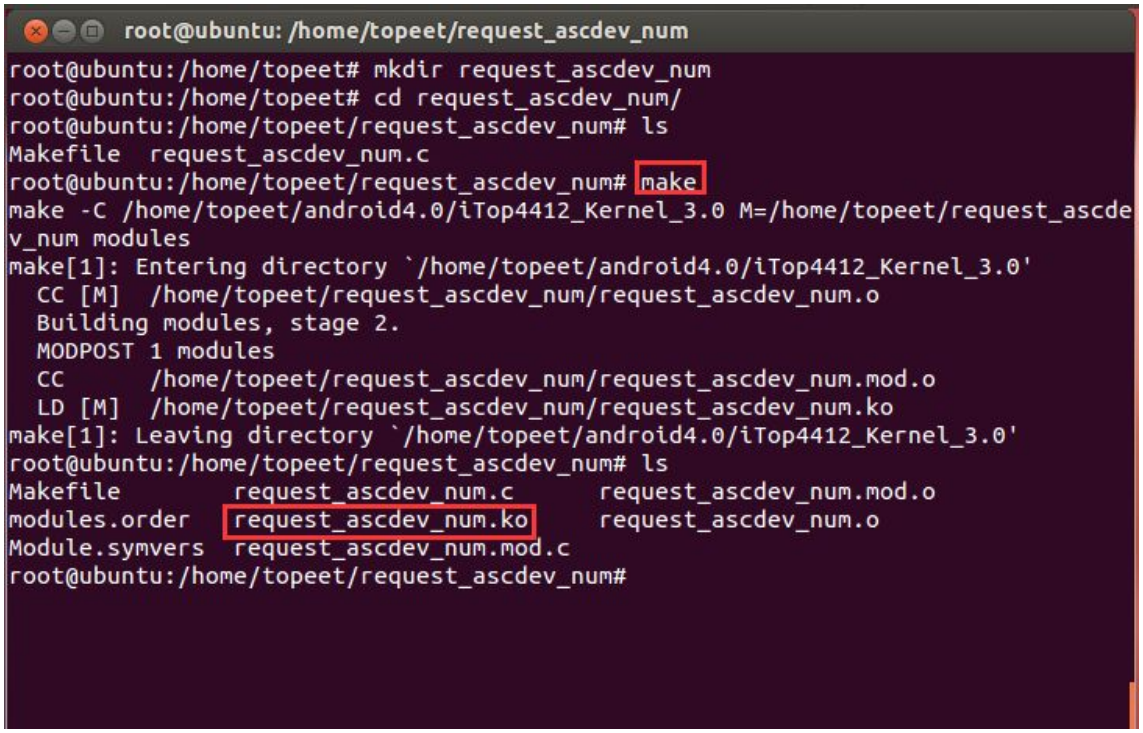
然后修改一下 Makefile 文件，如下图所示，将“request_cdev_num”改为“request_ascdev_num”。

```
#!/bin/bash
#通知编译器我们要编译模块的哪些源码
#这里是编译itop4412_hello.c这个文件编译成中间文件mini_linux_module.o
obj-m += request_ascdev_num.o

#源码目录变量，这里用户需要根据实际情况选择路径
#作者是将Linux的源码拷贝到目录/home/topeet/android4.0下并解压的
KDIR := /home/topeet/android4.0/iTop4412_Kernel_3.0

#当前目录变量
PWD ?= $(shell pwd)

#make命名默认寻找第一个目标
#make -C就是指调用执行的路径
#$(KDIR)Linux源码目录，作者这里指的是/home/topeet/android4.0/iTop4412_Kernel_3.0
#$(PWD)当前目录变量
#modules要执行的操作
all:
    make -C $(KDIR) M=$(PWD) modules

#make clean执行的操作是删除后缀为o的文件
clean:
    rm -rf *.mod.c *.o *.order *.ko *.mod.o *.symvers
```

在 Ubuntu 系统下，使用命令“mkdir request_ascdev_num”新建文件夹“request_ascdev_num”，将写好的文件 request_ascdev_num.c、编译脚本拷贝到 module_param 文件夹下，如下图所示。



使用 Makefile 命令编译驱动命令“Make”编译应用，如下图所示，生成驱动模块“request_ascdev_num.ko”。



将上图中的文件 request_ascdev_num.ko 拷贝到 U 盘。

启动开发板，将 U 盘插入开发板，使用命令“mount /dev/sda1 /mnt/udisk/”加载 U 盘，然后使用命令“insmod /mnt/udisk/request_ascdev_num.ko”加载模块 request_ascdev_num.ko，如下图所示。

```
[root@iTOP-4412]# mount /dev/sda1 /mnt/udisk/
[root@iTOP-4412]# insmod /mnt/udisk/request_ascdev_num.ko
[   71.273612] numdev_major is 0!
[   71.275311] numdev_minor is 0!
[   71.278379] adev_region req 249 !
[   71.289533] scdev_init!
```

使用命令“cat /proc/devices”查看已经被注册的主设备号，如下图所示。

```
[   71.278379] adev_region req 249 !
[   71.289533] scdev init!
[root@iTOP-4412]# cat /proc/devices
Character devices:
  1 mem
  4 ttyS
  5 /dev/tty
  5 /dev/console
  5 /dev/ptmx
 10 misc
 13 input
 21 sg
 29 fb
 81 video4linux
 89 i2c
108 ppp
116 alsa
128 ptm
136 pts
153 rc522_test
166 ttyACM
180 usb
188 ttyUSB
189 usb_device
204 ttySAC
216 rfcomm
249 ascdev
250 roccat
251 BaseRemoteCtl
252 media
253 ttyGS
254 rtc
```

如上图所示，可以看到主设备号 249 已经动态被申请到。

这个模块也是可以通过模块参数来静态配置，和上一期实验中的用法一样，这里就不重复介绍了。