kernel hacker 修炼之道——李万鹏

男儿立志出乡关， 学不成名死不还。 埋骨何须桑梓地， 人生无处不青山。 ——西乡隆盛诗

kernel hacker修炼之道之驱动-流水灯

分类： linux驱动编程 2011-04-02 21:45 2015人阅读 评论(4) 收藏 举报

浅析linux驱动之流水灯

作者：李万鹏

module.h包含可装载模块需要的大量符号和函数定义，包含init.h的目的是指定初始化和清除代码，struct file是在<linux/fs.h>中定义的。
注册字符设备使用
int register_chrdev(unsigned int major, const char *name, struct file_operations *fops);
移除字符设备使用
int unregister_chrdev(unsigned int major, const char *name);
init函数是在insmod时被调用的，exit函数是在rmmod时被调用的。
使用命令 cat /proc/devices 可以查看设备的主设备号。
如果内核没有为我们生成设备文件，我们需要手动生成：
mknod /dev/led_driver c 234 0
__init标记表示该函数只在初始化期间使用，在模块被装载后，模块装载器就会将初始化函数扔掉，这样可将函数占用的内存释放出来。
__exit修饰的函数只在模块卸载或系统关闭时被调用，如果模块被直接内嵌到内核中，或者内核的配置不允许卸载模块，则被标记为__exit
的模块将被简单丢弃。

驱动程序：

```
 #include <linux/module.h>
#include <linux/fs.h>
#include <linux/init.h>
#include <mach/regs-gpio.h>
#include <mach/hardware.h>

#define LED_MAJOR 234
#define DEVICE_NAME "led_driver"

static unsigned long led_table[]={
S3C2410_GPF3,
S3C2410_GPF4,
S3C2410_GPF5,
S3C2410_GPF6,
};

static unsigned long led_cfg_table[]={
S3C2410_GPF3_OUTP,
S3C2410_GPF4_OUTP,
S3C2410_GPF5_OUTP,
S3C2410_GPF6_OUTP,
};
```

```
int led_ioctl(struct inode *inode, struct file *file, unsigned int cmd , unsigned long arg){
switch(cmd){
case 0:
s3c2410_gpio_setpin(led_table[0],0);
s3c2410_gpio_setpin(led_table[1],1);
s3c2410_gpio_setpin(led_table[2],1);
s3c2410_gpio_setpin(led_table[3],1);
break;
case 1:
s3c2410_gpio_setpin(led_table[0],1);
s3c2410_gpio_setpin(led_table[1],0);
s3c2410_gpio_setpin(led_table[2],1);
s3c2410_gpio_setpin(led_table[3],1);
break;
case 2:
s3c2410_gpio_setpin(led_table[0],1);
s3c2410_gpio_setpin(led_table[1],1);
s3c2410_gpio_setpin(led_table[2],0);
s3c2410_gpio_setpin(led_table[3],1);
break;
case 3:
s3c2410_gpio_setpin(led_table[0],1);
s3c2410_gpio_setpin(led_table[1],1);
s3c2410_gpio_setpin(led_table[2],1);
s3c2410_gpio_setpin(led_table[3],0);
break;
default:
return -EINVAL;
break;
}
return 0;
}

struct file_operations led_ops = {
.owner = THIS_MODULE,
.ioctl = led_ioctl,
};

static int __init init_led(void)
{
int i, ret;
ret = register_chrdev(LED_MAJOR,DEVICE_NAME,&led_ops);
if(ret < 0){
printk(DEVICE_NAME, "can't register major number\n");
}
for(i = 0; i < 4; i++){
s3c2410_gpio_cfgpin(led_table[i],led_cfg_table[i]);
s3c2410_gpio_setpin(led_table[i],1);
}
printk(DEVICE_NAME "initialized\n");
return 0;
}

static void __exit exit_led(void)
{
unregister_chrdev(LED_MAJOR,DEVICE_NAME);
}

module_init(init_led);
module_exit(exit_led);
MODULE_LICENSE("GPL");
MODULE_AUTHOR("liwanpeng");
```

测试程序：

```
 #include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <fcntl.h>
#include <unistd.h>

int main(){
int fd, ret;
char *i;
fd = open("/dev/led_driver",0);
if(fd < 0){
printf("open device error");
}
else{
while(1){
ioctl(fd,0);
printf("led0 is on!\n");
sleep(1);
ioctl(fd,1);
printf("led1 is on!\n");
sleep(1);
ioctl(fd,2);
printf("led2 is on!\n");
sleep(1);
ioctl(fd,3);
printf("led3 is on!\n");
sleep(1);
}
close(fd);
}
return 0;
}
```

效果：此时LED灯也轮流闪烁。

```
[root@LWP usb]# ./led_test
led0 is on!
led1 is on!
led2 is on!
led3 is on!
led0 is on!
led1 is on!
led2 is on!
led3 is on!
led0 is on!
led1 is on!
led2 is on!
led3 is on!
led0 is on!
led1 is on!
led2 is on!
```