

Character device driver

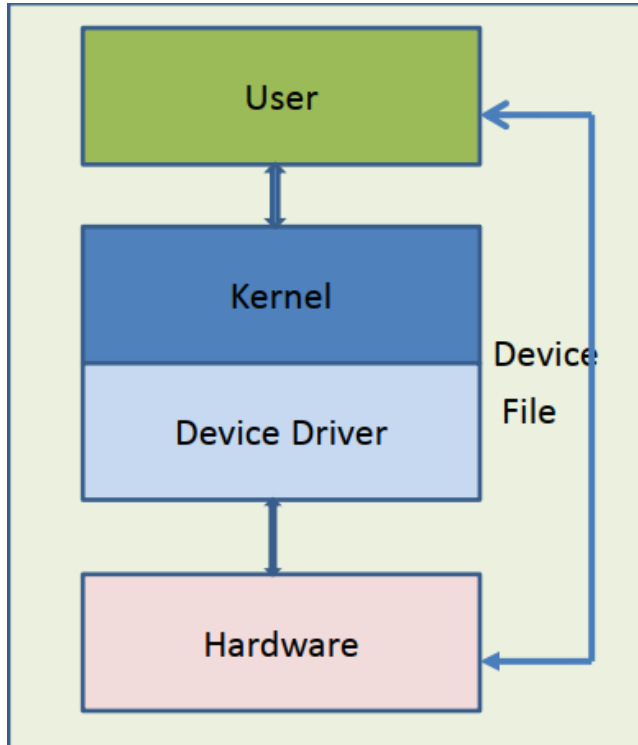
Giảng Viên: Lưu An Phú



Agenda

- Objective
- Implement file operation
- Driver basic functions
- Practice

Driver main functions



- Service user space
- Call service from kernel space
- Working directly with hardware

- Cây phân cấp device
 - Character device
 - Block device
 - Network device

- All device file place in /dev folder
- Major number
- Minor number
- Create a device file
 - By udev
 - By mknod command
 - By device driver

Device operation

- open
- close
- read
- write
- ioctl

Implement file operation in driver

- `static int dev_open(struct inode *inodep, struct file *filep)`
- `static int dev_close(struct inode *inodep, struct file *filep)`
- `static ssize_t dev_read(struct file*filep, char __user *buf, size_t len, loff_t *offset)`
- `static ssize_t dev_write(struct file*filep, const char __user *buf, size_t len, loff_t *offset)`

- `unsigned long copy_from_user(void *to, const void __user *from, unsigned long n);`
- `unsigned long copy_to_user(void __user *to, const void *from, unsigned long n);`

Register file operation with kernel

```
15 static struct file_operations fops = {
16     .open = dev_open,
17     .release = dev_close,
18     .read = dev_read,
19     .write = dev_write,
20
21     static int __init exam_init(void)
22 {
23     int ret;
24
25     ret = alloc_chrdev_region(&dev, 0, 1, "example");
26     if (ret)
27     {
28         printk("can not register major no\n");
29         return ret;
30     }
31     printk(KERN_INFO "register successfully major now is: %d\n", MAJOR(dev));
32     my_cdev = cdev_alloc();
33     my_cdev->owner = THIS_MODULE;
34     my_cdev->ops = &fops;
35     my_cdev->dev = dev;
36     cdev_add(my_cdev, dev, 0);
37
38     return 0;
39 }
```

Basic steps of character driver

- Define file operation
- Define other interface
- Initialize private resource
- Create device file
- Register file operation and other interface

Create device file

- `void cdev_init (struct cdev * cdev, const struct file_operations * fops);`
- `int cdev_add (struct cdev * p, dev_t dev, unsigned count);`
- `struct class *class_create(struct module *owner, const char *name);`
- `struct device *device_create(struct class *class, struct device *parent, dev_t devt, const char *fmt,);`

Destroy device file

- `void cdev_del(struct cdev *);`
- `void device_destroy(struct class *class, dev_t devt);`
- `void class_destroy (struct class * cls);`
- `void unregister_chrdev_region(dev_t from, unsigned count);`

Compare create and destroy a device

Create device file

- `cdev_init`
- `cdev_add`
- `class_create`
- `device_create`

Destroy device file

- `unregister_chrdev_region`
- `cdev_del`
- `class_destroy`
- `device_destroy`

Example code



- Viết 1 driver tạo ra device file có tên `/dev/sample_led`. Mỗi khi user ghi 1 vào device file thì led sáng, user ghi 0 vào thì led tắt. Nếu led đang sáng, đọc file sẽ trả về 1, led tắt đọc file sẽ trả về 0.

- Driver code
 - static long dev_ioctl(struct file *filep, unsigned int cmd, unsigned long arg)
- User space code
 - #include <sys/ioctl.h>
 - int ioctl(int fd, unsigned long request, ...);

Example code



- Viết 1 cặp ioctl tương ứng giữa driver và application. User có thể truyền command xuống để thực hiện nhấp nháy đèn led.
- Cắm led vào bảng nhựa thành hình số 8, trong hàm init của driver thực hiện configure gpio, sau đó tạo ra device file. Khi ghi 1 số bất kỳ có 1 chữ số thì hiển thị lại trên led của bảng nhựa.

- Viết 1 cặp driver và application. Khi hệ thống boot lên sẽ chạy application và insmod driver. Application sẽ tiến hành mount device file vào folder /home/root/char_dev. Folder trên nếu không tồn tại thì sẽ được application tạo ra. Mỗi khi user mở folder đấy lên sẽ nhìn thấy 3 file image1.jpg, image2.jpg, image3.jpg. Nếu user bình thường thì mở 3 file đều trả về dữ liệu đã mã hóa. Nếu user là “phula” thì mở ra sẽ nhìn được nội dung file ảnh 1 cách bình thường.
 - Gợi ý: Ở đâu đó trong máy tính sẽ lưu trữ 3 file ảnh ở dạng mã hóa, nếu user bình thường thì ở hàm đọc file sẽ đọc từ file ảnh mã hóa rồi trả về cho read buffer, nếu user là phula thì sẽ tiến hành giải mã rồi mới trả về cho read buffer.

Thank you

