

网址：www.icourses.cn，主页搜索“苏曙光”即可进入MOOC课堂

第八章 设备管理

-  8.1设备管理概念
-  8.2 Spooling系统
-  8.3.1 Linux模块机制
-  8.3.2 Linux驱动程序
-  8.3.3 Windows驱动程序

华中科技大学.苏曙光老师.《操作系统原理》MOOC课程组版权所有

网址：www.icourses.cn，主页搜索“苏曙光”即可进入MOOC课堂

《操作系统原理》

8.3.2

Linux驱动程序（LDD）

教师：苏曙光

华中科技大学软件学院

华中科技大学.苏曙光老师.《操作系统原理》MOOC课程组版权所有

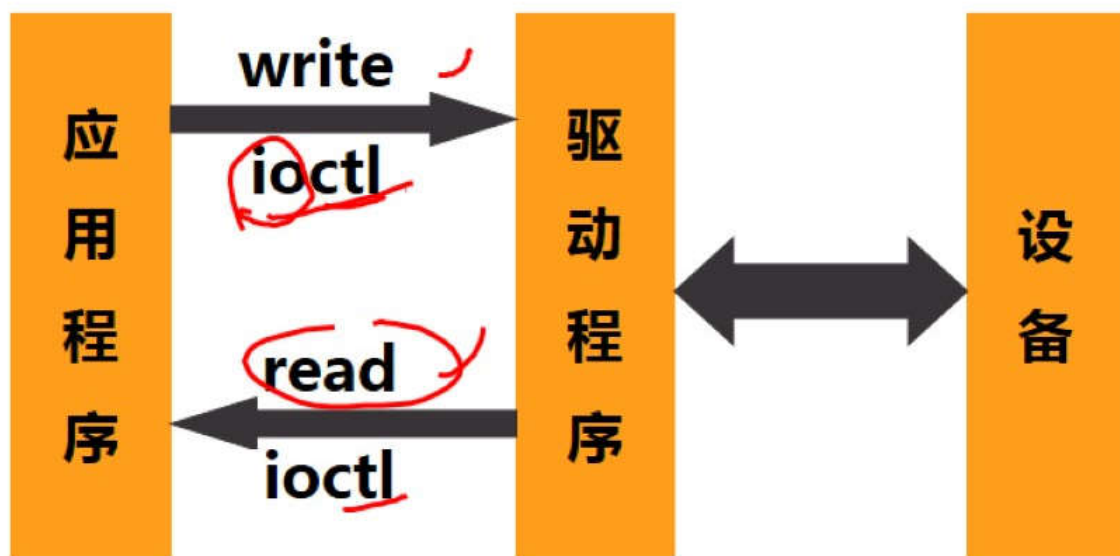


网址：www.icourses.cn，主页搜索“苏曙光”即可进入MOOC课堂

- LDD程序结构
- LDD程序加载方式
- LDD应用程序测试
- 例子：字符设备驱动程序

网址：www.icourses.cn，主页搜索“苏曙光”即可进入MOOC课堂

应用程序通过驱动程序间接访问设备



网址：www.icourses.cn，主页搜索“苏曙光”即可进入MOOC课堂

用户态与内核态

- Linux的两种运转模式。
 - ☒ 内核态
 - ☐ 用户态
- 驱动程序工作在内核态。

网址：www.icourses.cn，主页搜索“苏曙光”即可进入MOOC课堂

Linux设备的分类

■ 字符设备

- 以字节为单位逐个进行I/O操作
- 如串口设备

■ 块设备

- 块设备的存取是通过buffer、cache来进行
- 可以进行随机访问
- 例如IDE硬盘设备
- 支持可安装文件系统

■ 网络设备

- 通过BSD套接口访问 (SOCKET)

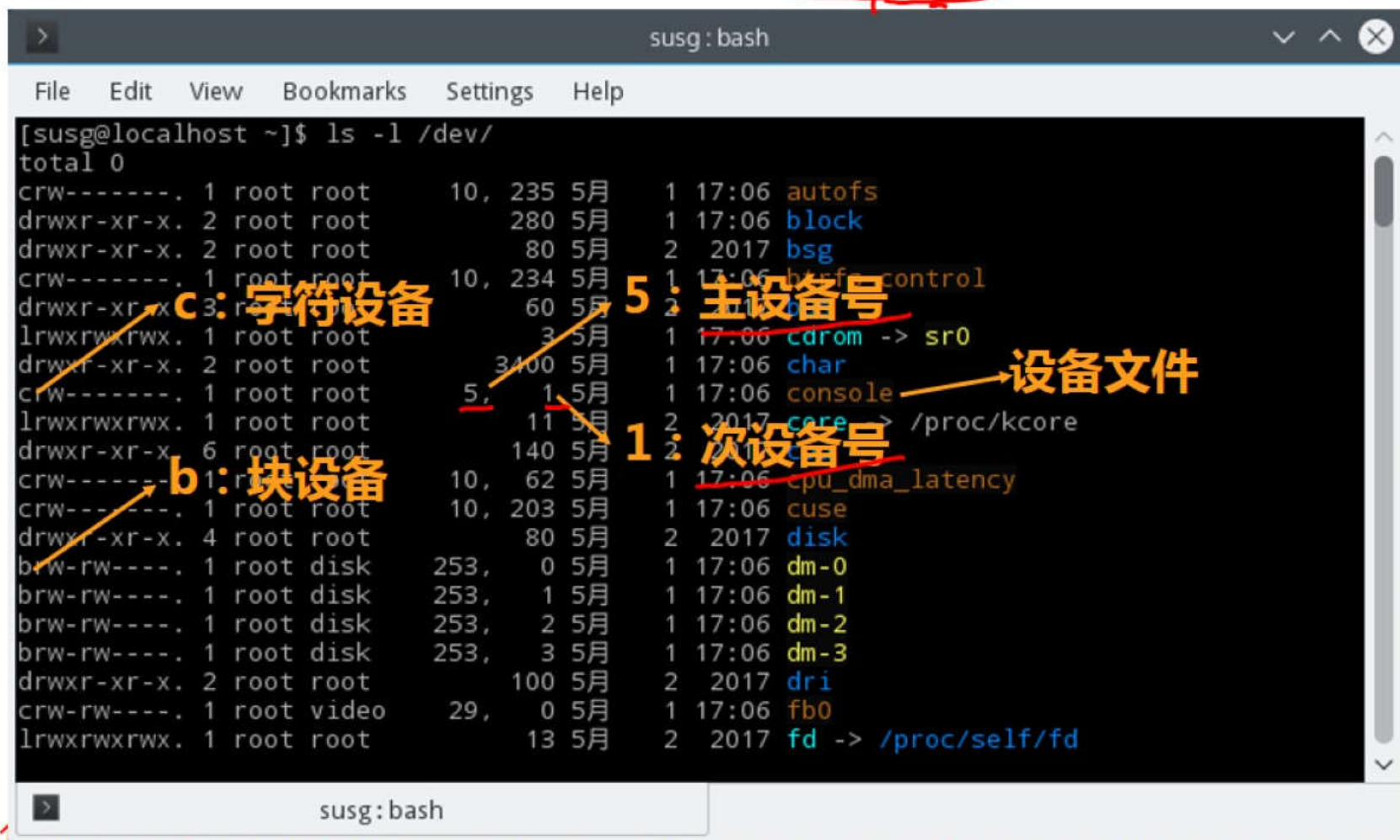
网址：www.icourses.cn，主页搜索“苏曙光”即可进入MOOC课堂

■ 设备文件

- 硬件设备作为文件看待：设备文件
- 用文件接口来完成设备的操作
 - 打开、关闭、读写和I/O控制等操作
- 字符设备和块设备通过设备文件访问。
 - Linux文件系统中可以找到设备文件

网址：www.icourses.cn，主页搜索“苏曙光”即可进入MOOC课堂

列出设备文件 `ls -l /dev`



The terminal window displays the output of the command `ls -l /dev/`. The output lists various device files with their permissions, owner, group, size, and name. Annotations are present on the image:

- c: 字符设备** (Character Device) points to the first column of the output.
- b: 块设备** (Block Device) points to the second column of the output.
- 5: 主设备号** (Major Device Number) points to the third column of the output.
- 1: 次设备号** (Minor Device Number) points to the fourth column of the output.
- 设备文件** (Device File) points to the fifth column of the output.

```
[susg@localhost ~]$ ls -l /dev/
total 0
crw----- 1 root root      10, 235 5月  1 17:06 autofs
drwxr-xr-x 2 root root      280 5月  1 17:06 block
drwxr-xr-x 2 root root       80 5月  2 2017 bsg
crw----- 1 root root     10, 234 5月  1 17:06 btcontrol
drwxr-xr-x 3 root root       60 5月  2 2017 btrf
lrwxrwxrwx 1 root root       3 5月  1 17:06 cdrom -> sr0
drwxr-xr-x 2 root root    3400 5月  1 17:06 char
crw----- 1 root root       5,  1 5月  1 17:06 console
lrwxrwxrwx 1 root root       11 5月  2 2017 core -> /proc/kcore
drwxr-xr-x 6 root root     140 5月  2 2017 cpio
crw----- 1 root root     10,  62 5月  1 17:06 cpu_dma_latency
crw----- 1 root root     10, 203 5月  1 17:06 cuse
drwxr-xr-x 4 root root       80 5月  2 2017 disk
brw-rw---- 1 root disk    253,  0 5月  1 17:06 dm-0
brw-rw---- 1 root disk    253,  1 5月  1 17:06 dm-1
brw-rw---- 1 root disk    253,  2 5月  1 17:06 dm-2
brw-rw---- 1 root disk    253,  3 5月  1 17:06 dm-3
drwxr-xr-x 2 root root     100 5月  2 2017 dri
crw-rw---- 1 root video    29,  0 5月  1 17:06 fb0
lrwxrwxrwx 1 root root      13 5月  2 2017 fd -> /proc/self/fd
```


网址：www.icourses.cn，主页搜索“苏曙光”即可进入MOOC课堂

主设备号和次设备号

■ 主设备号

- 标识该设备种类，标识驱动程序
- 主设备号的范围：1-255
- Linux内核支持动态分配主设备号

■ 次设备号

- 标识同一设备驱动程序的不同硬件设备
- 次设备号只在驱动程序内部使用，系统内核直接把次设备号传递给驱动程序，由驱动程序去管理。

网址：www.icourses.cn，主页搜索“苏曙光”即可进入MOOC课堂



功能完整的LDD结构

- 设备的打开
- 设备的释放
- 设备的读操作
- 设备的写操作
- 设备的控制操作
- 设备的中断和轮询处理
- 驱动程序的注册
- 驱动程序的注销

功能完整的
Linux设备驱动
程序结构

华中科技大学.苏曙光老师.《操作系统原理》MOOC课程组版权所有

网址：www.icourses.cn，主页搜索“苏曙光”即可进入MOOC课堂

简单字符驱动程序的例子：实现了5个函数

```
static int my_open(struct inode * inode, struct file * filp)
```

```
{ 设备打开时的操作 ... }
```

```
static int my_release(struct inode * inode, struct file * filp)
```

```
{ 设备关闭时的操作 ... }
```

```
static int my_write(struct file *file, const char * buffer, size  
loff_t * ppos)
```

```
{ 设备写入时的操作 ... }
```

```
static int my_init(void)
```

```
{设备的注册：初始化硬件，注册设备，创建设备节点... }
```

```
static void my_exit(void)
```

```
{设备的注销：删除设备节点，注销设备... }
```

华中科技大学.苏曙光老师.《操作系统原理》MOOC课程组版权所有

网址：www.icourses.cn，主页搜索“苏曙光”即可进入MOOC课堂

打开和关闭操作

my_open和my_release在设备打开和关闭时调用

```
static int my_open(struct inode * inode, struct file * filp){  
    MOD_INC_USE_COUNT;  
    return 0;  
}
```

```
static int my_release(struct inode * inode, struct file * filp)  
    MOD_DEC_USE_COUNT;  
    return 0;  
}
```

■ MOD_INC_USE_COUNT

■ MOD_DEC_USE_COUNT

华中科技大学.苏曙光老师.《操作系统原理》MOOC课程组版权所有

网址：www.icourses.cn，主页搜索“苏曙光”即可进入MOOC课堂

写入操作

```
static int my_write(struct file *file, const char *buffer, size_t count,
loff_t * ppos){
    char led_status = 0;
    copy_from_user(&led_status, buffer, sizeof(led_status));
    if(led_status == 0x01){
        AT91F_PIOB_SetOutput(LED);
    }else{
        AT91F_PIOB_ClearOutput(LED);
    }
    return 0;
}
```

华中科技大学.苏曙光老师.《操作系统原理》MOOC课程组版权所有

网址：www.icourses.cn，主页搜索“苏曙光”即可进入MOOC课堂

文件操作结构体的定义

```
struct file_operations{
    struct module *owner;
    loff_t (*llseek) (struct file *, loff_t, int);
    ssize_t(*read) (struct file *, char *, size_t, loff_t*);
    ssize_t(*write) (struct file *, const char *, size_t, loff_t*);
    int(*readdir) (struct file *, void *, filldir_t);
    unsigned int(*poll) (struct file *, struct poll_table_struct *);
    int(*ioctl) (struct inode*, struct file *, unsigned int, unsigned long);
    int(*mmap) (struct file *, struct vm_area_struct *);
    int(*open) (struct inode*, struct file *);
    int(*flush) (struct file *);
    int(*release) (struct inode*, struct file *);
    int(*fsync) (struct file *, struct dentry*, intdatasync);
    int(*fasync) (int, struct file *, int);
    int(*lock) (struct file *, int, struct file_lock*);
    ssize_t(*readv) (struct file *, const struct iovec*, unsigned long, loff_t*);
    ssize_t(*writev) (struct file *, const struct iovec*, unsigned long, loff_t*);
    ssize_t(*sendpage) (struct file *, struct page *, int, size_t, loff_t*, int);
};
```

华中科技大学.苏曙光老师.《操作系统原理》MOOC课程组版权所有

网址：www.icourses.cn，主页搜索“苏曙光”即可进入MOOC课堂

文件操作结构体初始化

```
static struct file_operations my_fops = {  
    open: my_open,  
    write: my_write,  
    release: my_release  
};
```

```
fd = open("/dev/my_led", O_RDWR);  
  
write(fd, &led_on, 1);  
  
close(fd);
```

华中科技大学.苏曙光老师.《操作系统原理》MOOC课程组版权所有

网址：www.icourses.cn，主页搜索“苏曙光”即可进入MOOC课堂

设备注册（初始化）



```
static int __init my_init(void)
{
    //硬件初始化
    AT91F_PIOB_Enable(LED);
    //字符设备注册
    Led_Major = register_chrdev(0, DEVICE_NAME, &my_fops);
    // 创建设备文件
    Devfs_Led_Dir = devfs_mk_dir(NULL, "my_led", NULL);
    Devfs_Led_Raw = devfs_register(Devfs_Led_Dir, "0",
                                   DEVFS_FL_DEFAULT, Led_Major, 1,
                                   S_IFCHR|S_IRUSR|S_IWUSR, &my_fops, NULL);
}
```


网址：www.icourses.cn，主页搜索“苏曙光”即可进入MOOC课堂

设备注销（反初始化）

```
static void __exit my_exit(void)
{
    //删除设备文件
    devfs_unregister(Devfs_Led_Raw);
    devfs_unregister(Devfs_Led_Dir);
    //注销设备
    unregister_chrdev(Led_Major, DEVICE_NAME);
}
```

网址：www.icourses.cn，主页搜索“苏曙光”即可进入MOOC课堂

设备注册（初始化）和设备注销（反初始化）的登记

//向Linux系统记录设备初始化的函数名称

module_init(my_init);

//向Linux系统记录设备退出的函数名称

module_exit(my_exit);

网址：www.icourses.cn，主页搜索“苏曙光”即可进入MOOC课堂

驱动程序编译

Makefile文件

```
OBJ=my_led.o
SOURCE=my_led.c
CC=gcc
COMP=-Wall -O2 -DMODULE -D_KERNEL -I /home/linux/linux-2.4.19-rmk7/include -c
$(OBJ):$(SOURCE)
    $(CC) $(COMP) $(SOURCE)
clean:
    rm $(OBJ)
```

运行make,生成名为my_led.o的驱动程序

华中科技大学.苏曙光老师.《操作系统原理》MOOC课程组版权所有

网址：www.icourses.cn，主页搜索“苏曙光”即可进入MOOC课堂

驱动程序加载/查看/移除

□ 驱动程序模块插入内核

```
#insmod my_led.o
```

□ 查看是否载入？载入成功会显示设备名my_led

```
#cat /proc/devices
```

□ 从内核移除设备

```
#rmmod my_led
```

网址：www.icourses.cn，主页搜索“苏曙光”即可进入MOOC课堂

驱动测试应用程序

```
int main(void)
{
    int fd;
    char led_on = 1;
    fd = open("/dev/my_led", O_RDWR); //打开led设备
    write(fd, &led_on, 1); //LED开
    close(fd); //关闭设备文件
    return 0;
}
```