Linux字符设备驱动模型

## 整体框架



## 设备描述结构

struct cdev{

struct kobject kobj;

struct module \*ower;

const struct file\_operations \*ops;//设备操作集

struct list\_head list;

dev\_t dev;//设备号

unsigned int count;//设备数

};

## 设备号—操作

Linux内核中使用dev\_t类型来定义设备号，dev\_t这种类型实质为32位的unsigned int ，其中高12位位主设备号，低20位为次设备号。

问1：如果知道主设备号、此设备号，怎么组合成dev\_t类型。

答：dev\_t dev = MKDEV(主设备号，次设备号)

问2：如何从dev\_t中分解除主设备号

答：主设备号 = MAJOR(dev\_t dev)

问3：如何从dev\_t中分解除次设备号

答：次设备号= MINOR(dev\_t dev)

## 设备号—分配

静态分配 register\_chrdev\_region函数

int register\_chrdev\_region(dev\_t from, unsigned count, const char \*name)

功能：申请使用从from开始的count个设备号（主设备号不变，次设备号增加）

缺点：如果申请使用的设备号已经被内核中的其他驱动使用了，则申请失败。

动态分配

int alloc\_chrdev\_region(dev\_t \*dev, unsigned baseminor, unsigned count,const char \*name)

功能:请求内核动态分配count个设备号，且次设备号从baseminor开始。

优点：简单，易于驱动推广

缺点：无法在安装驱动前创建设备文件（因为安装前还没有分配到主设备号）

解决方法：安装驱动后，从/proc/devices中查询设备号

## 设备号 –注销

void unregister\_chrdev\_region(dev\_t from,unsigned count)

## 操作函数集

struct file\_opreations 是一个函数指针的集合，定义能在设备上进行的操作。结构中的函数指针指向驱动中的函数，这些函数实现了一个针对设备的操作，对于不支持的草最则设置函数指针为NULL。例：

struct file\_operations mem\_fops = {

.owner = THIS\_MODULE,

.llseek = mem\_seek,

.read = mem\_read,

.write = mem\_write,

.ioctl = mem\_ioctl,

.open = mem\_open,

.release = mem\_release,

};

## 描述结构—分配

静态分配

struct cdev mdev;

动态分配

struct cdev \*cdev\_alloc(void)

## 描述结构 –初始化

void cdev\_init(struct cdev \*cdev, const struct file\_operations \*fops)

## 描述结构 –添加

int cdev\_add(struct cdev \*p, dev\_t dev, unsigned count)

## 设备操作

有很多，比如open close read write lseek ioctl mmap 等等

设备操作 –open int (\*open)(struct inode \*, struct file \*)

打开设备，响应open系统调用，完成如下工作

1. 标明次设备号
2. 启动设备

设备操作 –release void (\*release)(struct inode \*, struct file \*)

有时也称为close，完成关闭设备操作。

设备操作 –read ssize\_t (\*read) (struct file \*, char \_\_user \*, size\_t, loff\_t \*)

read设备方法通常完成两件事：

1. 从设备中读取数据（属于硬件访问操作）
2. 将读取的数据返回给应用程序

设备操作 –write ssize\_t (\*write) (struct file \*, const char \_\_user \*, size\_t, loff\_t \*)

write设备方法通常完成两件事

1. 从应用程序提供的地址读取数据
2. 将数据写入设备（属于硬件访问类操作）

## 驱动注销

int cdev\_del(struct cdev \*p)