

第25课 字符设备驱动程序概念纠正之另一种写法

版权所有：深圳百问网科技有限公司

电话：0755-86200561 淘宝地址：100ask.taobao.com

地址：广东省深圳市龙岗区布吉中海怡翠山庄13-2-2B

作者：jason Q：125707942 群84174029 28664149

前言

先回忆一下旧字符设备驱动的框架（忘了的直接去面壁）：

- ① 确定主设备号 `Major`;
- ② 构造 `file_operations`;
- ③ `register_chrdev`;
- ④ 创建类和设备。

这个框架最大的弊端出现在 `register_chrdev` 这个函数。它的定义在 `linux-2.6.22.6/fs/Char_dev.c`，定位到274行你会发现 `register_chrdev` 的定义中有这么一句：

版权所有：深圳百问网科技有限公司

电话：0755-86200561 淘宝地址：100ask.taobao.com

地址：广东省深圳市龙岗区布吉中海怡翠山庄13-2-2B

作者：jason Q：125707942 群84174029 28664149

```
cd = __register_chrdev_region(major, 0, 256, name);
```

问题就出在256上，这样做的结果就是向内核连续注册了256个次设备号，也就把major这个主设备号下的所有次设备号都占用了，而其实我们99%的情况下都用不到那么多的次设备号。这是巨大的浪费，这样做违背了内核的精神。最节约的情况就是用多少注册多少。

于是就有了以下2个函数：

- ① `register_chrdev_region(dev_t from, unsigned count, const char *name)`
- ② `alloc_chrdev_region(dev_t *dev, unsigned baseminor, unsigned count, const char *name)`

版权所有：深圳百问网科技有限公司

电话：0755-86200561 淘宝地址：100ask.taobao.com

地址：广东省深圳市龙岗区布吉中海怡翠山庄13-2-2B

作者：jason Q：125707942 群84174029 28664149

register_chrdev_region函数的第一个参数dev_t from是一个unsigned型的数，高12位表示主设备号，低20位表示次设备号。因此这个参数中包含了起始的次设备号。第二个参数unsigned count表示需要注册的连续的次设备号的个数。最后一个参数表示设备或驱动的名字。

在解释**alloc_chrdev_region**函数的参数之前先解释一下这2个函数的区别和用法。**register_chrdev_region**这个函数在调用之前需要确定from这个参数的值，就是说设备号是已知的。而**alloc_chrdev_region**这个函数调用的时候不需要知道设备的设备号是多少，内核会动态分配一个给他。在分配好了之后会把设备号写到设备号变量的地址，所以我们看到在调用**alloc_chrdev_region**这个函数的时候它的第一

版权所有：深圳百问网科技有限公司

电话：0755-86200561 淘宝地址：100ask.taobao.com

地址：广东省深圳市龙岗区布吉中海怡翠山庄13-2-2B

作者：jason Q：125707942 群84174029 28664149

个参数是设备号变量的地址。`alloc_chrdev_region`它的第二个参数是起始次设备号，第三个参数是要注册的次设备号个数，最后一个设备或驱动的名字。

这两个函数最终调用的都是`__register_chrdev_region`，同时需要跟`cdev_init(struct cdev *cdev, const struct file_operations *fops)`和`cdev_add(struct cdev *p, dev_t dev, unsigned count)`来配合使用。这2个函数的功能在以前的字符设备驱动程序中是在`register_chrdev`这一个函数中完成的，在这里需要我们自己来调用。

这样一来，`register_chrdev`这个函数就被完美的替代了。我们就可以通过控制注册的起始次设备号以及多少个次设备号，实现了用多少注册多少的目的。

版权所有：深圳百问网科技有限公司

电话：0755-86200561 淘宝地址：100ask.taobao.com

地址：广东省深圳市龙岗区布吉中海怡翠山庄13-2-2B

作者：jason Q：125707942 群84174029 28664149

框架

1

确定主设备号

2

构造file_operations

3

告诉内核

4

创建类和设备

版权所有：深圳百问网科技有限公司

电话：0755-86200561 淘宝地址：100ask.taobao.com

地址：广东省深圳市龙岗区布吉中海怡翠山庄13-2-2B

作者：jason Q：125707942 群84174029 28664149



确定主设备号

用法: `static int major;`

我们一直说第一步是确定主设备号，但其实这一步仅仅是定义了一个全局变量而已。

版权所有：深圳百问网科技有限公司

电话：0755-86200561 淘宝地址：100ask.taobao.com

地址：广东省深圳市龙岗区布吉中海怡翠山庄13-2-2B

作者：jason Q: 125707942 群84174029 28664149



构造file_operations

用法:

```
static struct file_operations hello_fops = {  
    .owner = THIS_MODULE,  
    .open = hello_open,  
};
```

这一步需要自己构造open函数，注意返回值即可。

版权所有：深圳百问网科技有限公司

电话：0755-86200561 淘宝地址：100ask.taobao.com

地址：广东省深圳市龙岗区布吉中海怡翠山庄13-2-2B

作者：jason Q：125707942 群84174029 28664149



告诉内核

用法： 主设备号已知，

```
devid = MKDEV(major, 0);  
register_chrdev_region(devid, HELLO_CNT, "hello");
```

主设备号未知，

```
alloc_chrdev_region(&devid, 0, HELLO_CNT, "hello");  
major = MAJOR(devid);
```

```
cdev_init(&hello_cdev, &hello_fops);  
cdev_add(&hello_cdev, devid, HELLO_CNT);
```

版权所有：深圳百问网科技有限公司

电话：0755-86200561 淘宝地址：100ask.taobao.com

地址：广东省深圳市龙岗区布吉中海怡翠山庄13-2-2B

作者：jason Q: 125707942 群84174029 28664149



创建类和设备

用法: `cls = class_create(THIS_MODULE, "hello");`
`class_device_create(cls, NULL, MKDEV(major, 0), NULL,`
`"hello0");`

需要说明的是，之前在这个主设备号下注册了多少个次设备号这里就可以创建多少个设备。

版权所有：深圳百问网科技有限公司

电话：0755-86200561 淘宝地址：100ask.taobao.com

地址：广东省深圳市龙岗区布吉中海怡翠山庄13-2-2B

作者：jason Q：125707942 群84174029 28664149

其实新旧**2**种字符设备驱动的框架本质上区别不大，因为它们实现的功能其实是一样的，区别就在于新的框架更加节省资源。**2**种框架一对比你更深入的理解它们的实现，由于字符设备驱动框架逻辑上比较简单，所以这里也没有深入。

版权所有：深圳百问网科技有限公司

电话：0755-86200561 淘宝地址：100ask.taobao.com

地址：广东省深圳市龙岗区布吉中海怡翠山庄13-2-2B

作者：jason Q：125707942 群84174029 28664149