



www.enjoylinux.cn

字符驱动编程模型 (v2014)

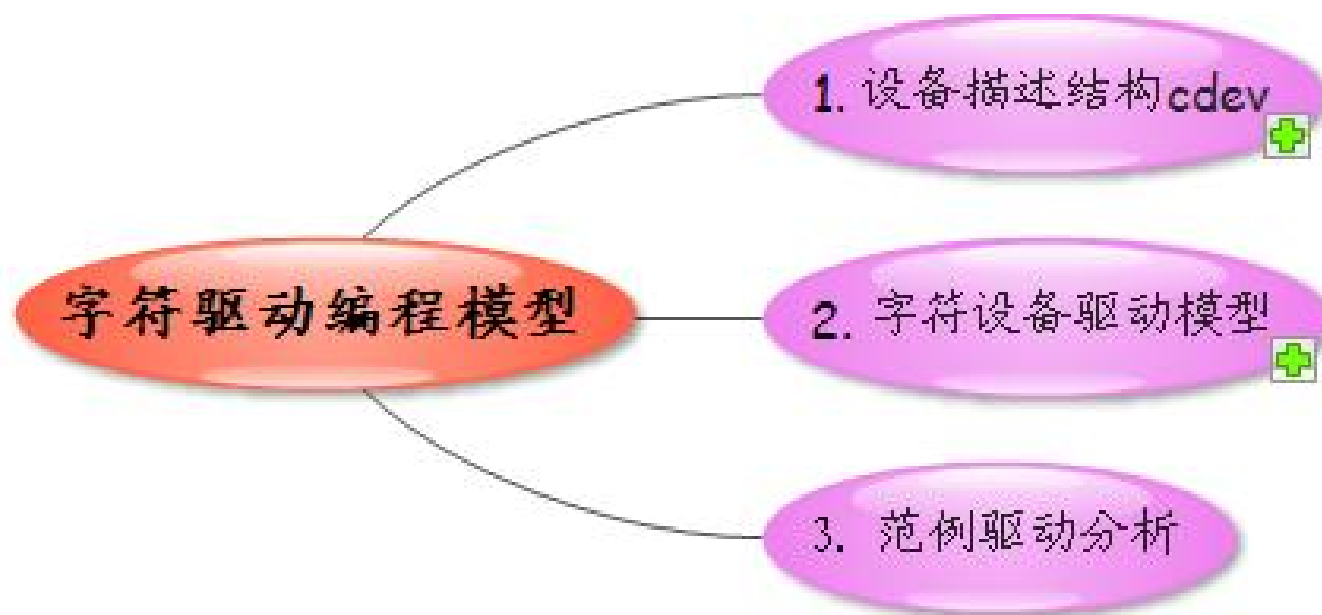


版权声明：本课件及其印刷物、视频的版权归成都国嵌信息技术有限公司所有，并保留所有权力：任何单位或个人未经成都国嵌信息技术有限公司书面授权，不得使用该课件及其印刷物、视频从事商业、教学活动。已经取得书面授权的，应在授权范围内使用，并注明“来源：国嵌”。违反上述声明者，我们将追究其法律责任。

课程索引



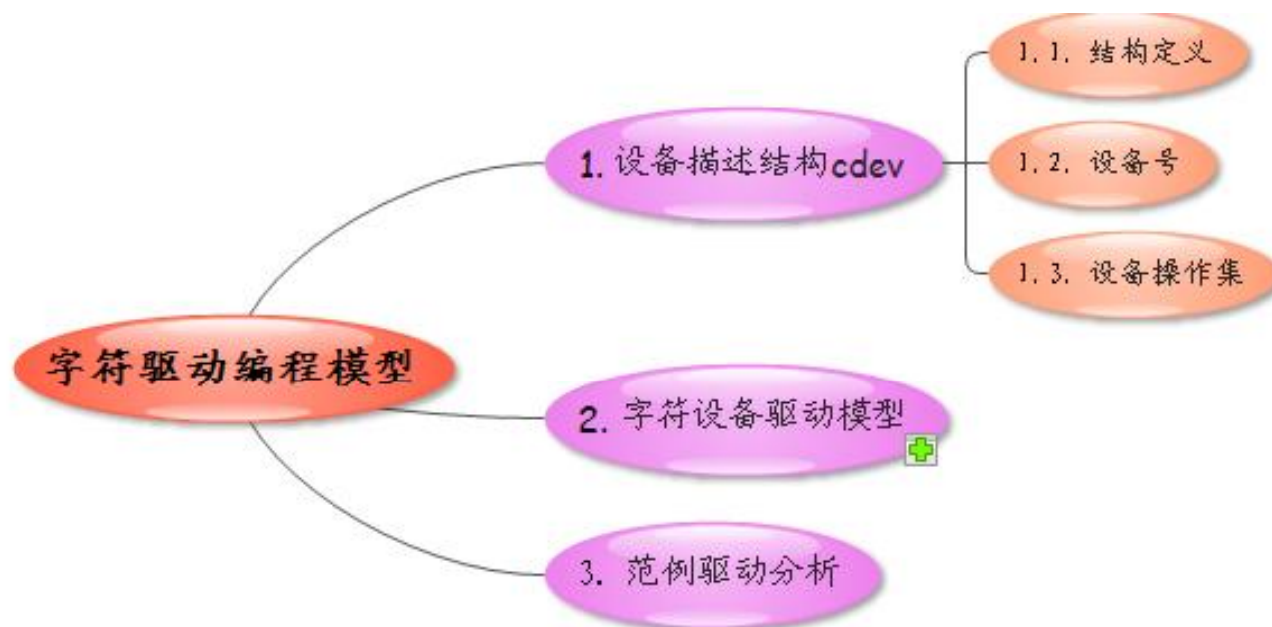
www.enjoylinux.cn



嵌入式Linux技术咨询QQ号: 550491596
嵌入式Linux学习交流QQ群: 65212116



1. 设备描述结构



嵌入式Linux技术咨询QQ号: 550491596
嵌入式Linux学习交流QQ群: 65212116



1. 驱动模型



www.enjoylinux.cn

在Linux系统中,设备的类型非常繁多,如:字符设备,块设备,网络接口设备,USB设备,PCI设备,平台设备,混杂设备.....,而设备类型不同,也意味着其对应的驱动程序模型不同,这样就导致了我们需要去掌握众多的驱动程序模型。那么能不能从这些众多的驱动模型中提炼出一些具有共性的规则,则是我们能不能学好Linux驱动的关键。

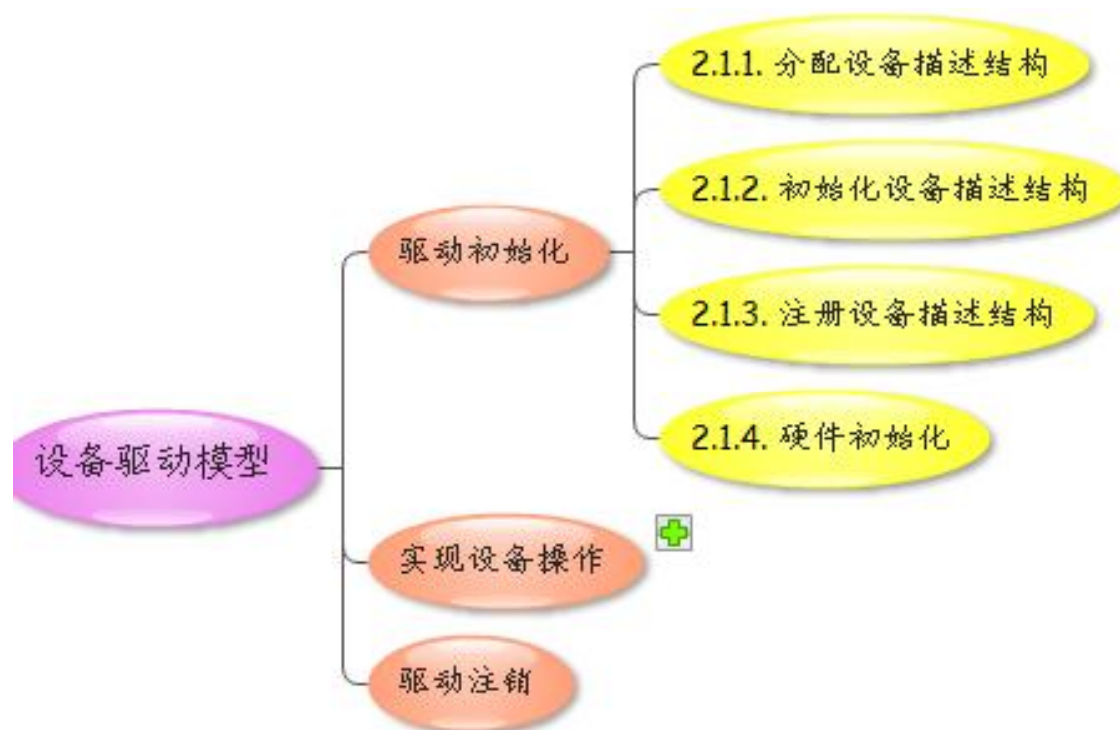
嵌入式Linux技术咨询QQ号: 550491596
嵌入式Linux学习交流QQ群: 65212116



1. 驱动模型



www.enjoylinux.cn



嵌入式Linux技术咨询QQ号: 550491596
嵌入式Linux学习交流QQ群: 65212116



1. 设备描述结构



www.enjoylinux.cn

在任何一种驱动模型中，设备都会用内核中的一种结构来描述。我们的字符设备在内核中使用 **struct cdev** 来描述。

```
struct cdev {  
    struct kobject kobj;  
    struct module *owner;  
    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  
};
```

嵌入式Linux技术咨询QQ号: 550491596
嵌入式Linux学习交流QQ群: 65212116





www.enjoylinux.cn

1.1 设备号



查看/dev目录下
设备号

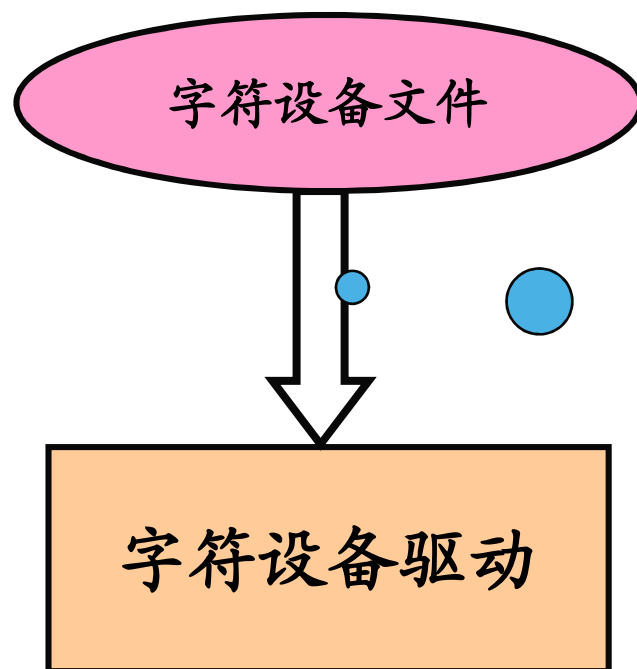
嵌入式Linux技术咨询QQ号: 550491596
嵌入式Linux学习交流QQ群: 65212116



1.1 主设备号



www.enjoylinux.cn



字符设备文件与字符驱动程序如何建立起对应关系??

答案：主设备号

嵌入式Linux技术咨询QQ号: 550491596

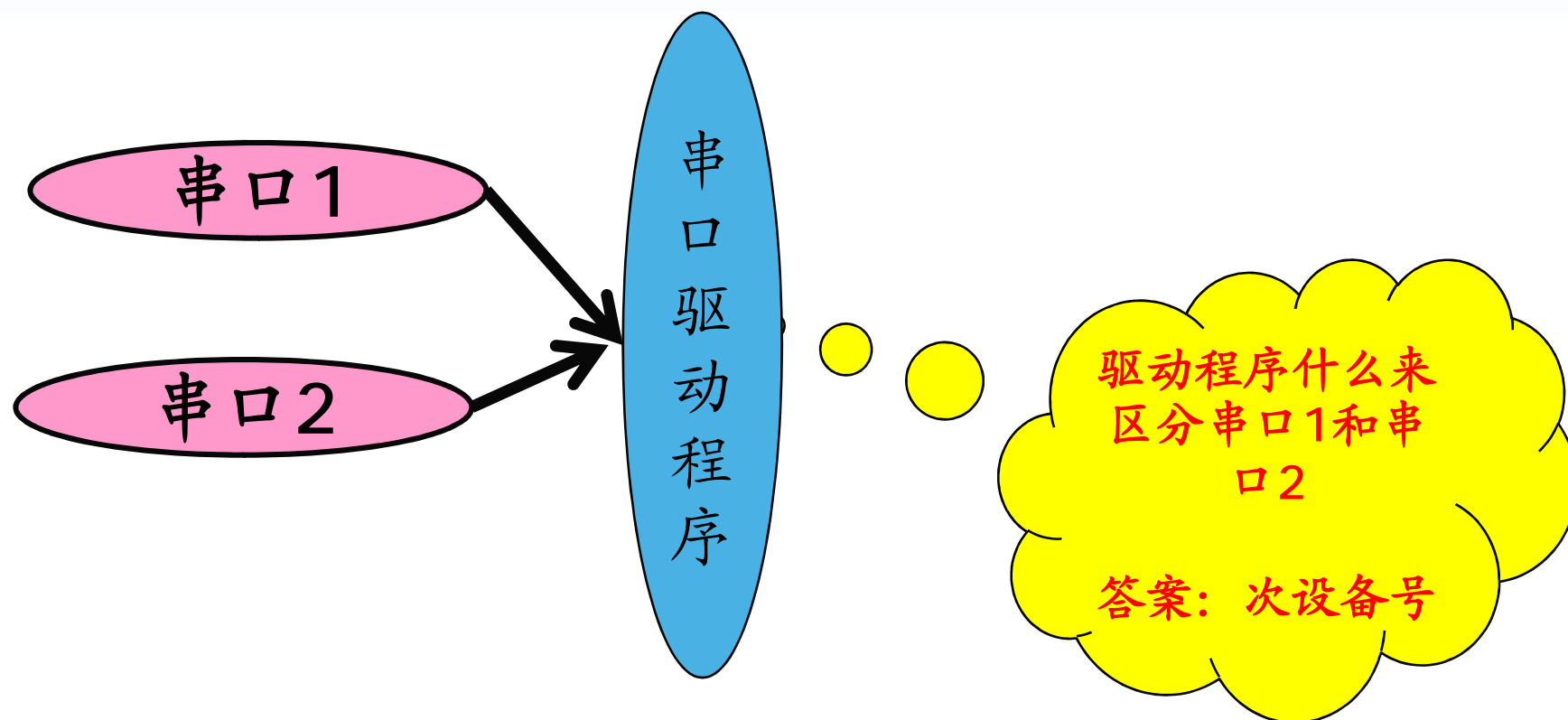
嵌入式Linux学习交流QQ群: 65212116





www.enjoylinux.cn

1.1 次设备号



嵌入式Linux技术咨询QQ号: 550491596

嵌入式Linux学习交流QQ群: 65212116



1.1 设备号-操作



www.enjoylinux.cn

Linux内核中使用`dev_t`类型来定义设备号，`dev_t`这种类型其实质为32位的`unsigned int`，其中高12位为主设备号，低20位为次设备号。

问1: 如果知道主设备号，次设备号，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`中分解出主设备号?

答: 主设备号 = `MAJOR(dev_t dev)`

问3: 如何从`dev_t`中分解出次设备号?

答: 次设备号 = `MINOR(dev_t dev)`

嵌入式Linux技术咨询QQ号: 550491596

嵌入式Linux学习交流QQ群: 65212116



1.1 设备号-分配

如何为设备分配一个主设备号？

✓ 静态申请

开发者自己选择一个数字作为主设备号，然后通过函数 `register_chrdev_region` 向内核申请使用。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

使用 `alloc_chrdev_region` 由内核分配一个可用的主设备号。优点：因为内核知道哪些号已经被使用了，所以不会导致分配到已经被使用的号。

嵌入式Linux技术咨询QQ号：550491596
嵌入式Linux学习交流QQ群：65212116





www.enjoylinux.cn

1.1 设备号-注销

不论使用何种方法分配设备号，都应该在驱动退出时，使用unregister_chrdev_region函数释放这些设备号。



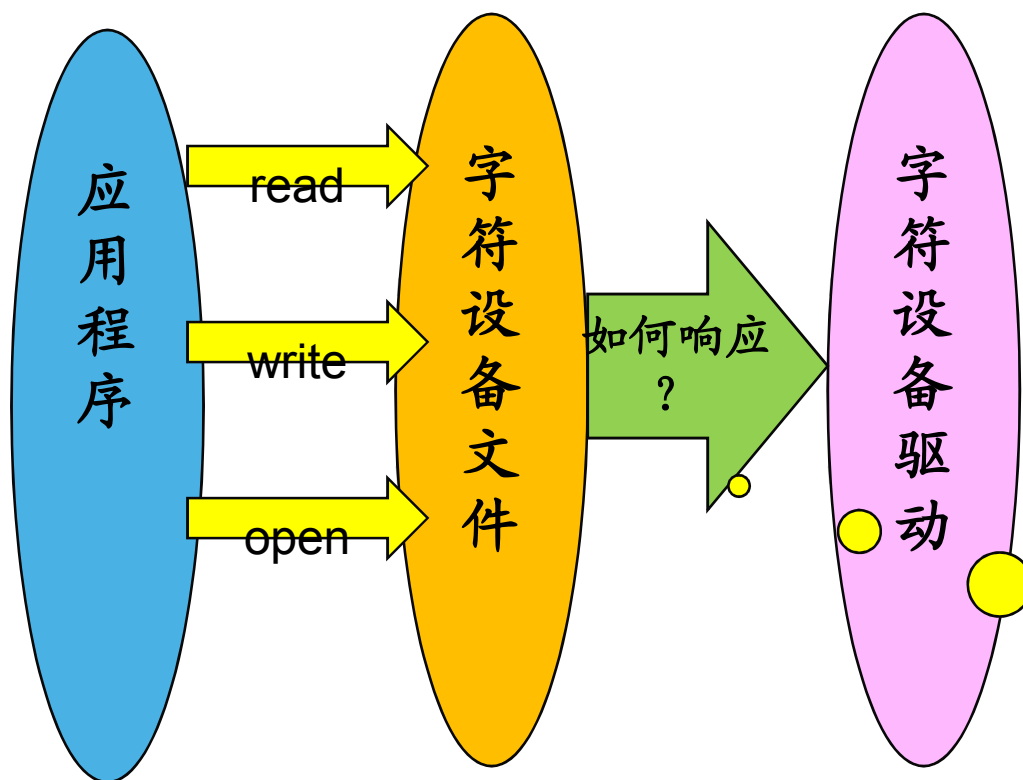
嵌入式Linux技术咨询QQ号: 550491596
嵌入式Linux学习交流QQ群: 65212116





www.enjoylinux.cn

1.2 操作函数集



分析
file_operations
定义

嵌入式Linux技术咨询QQ号: 550491596
嵌入式Linux学习交流QQ群: 65212116



1.2 操作函数集



www.enjoylinux.cn

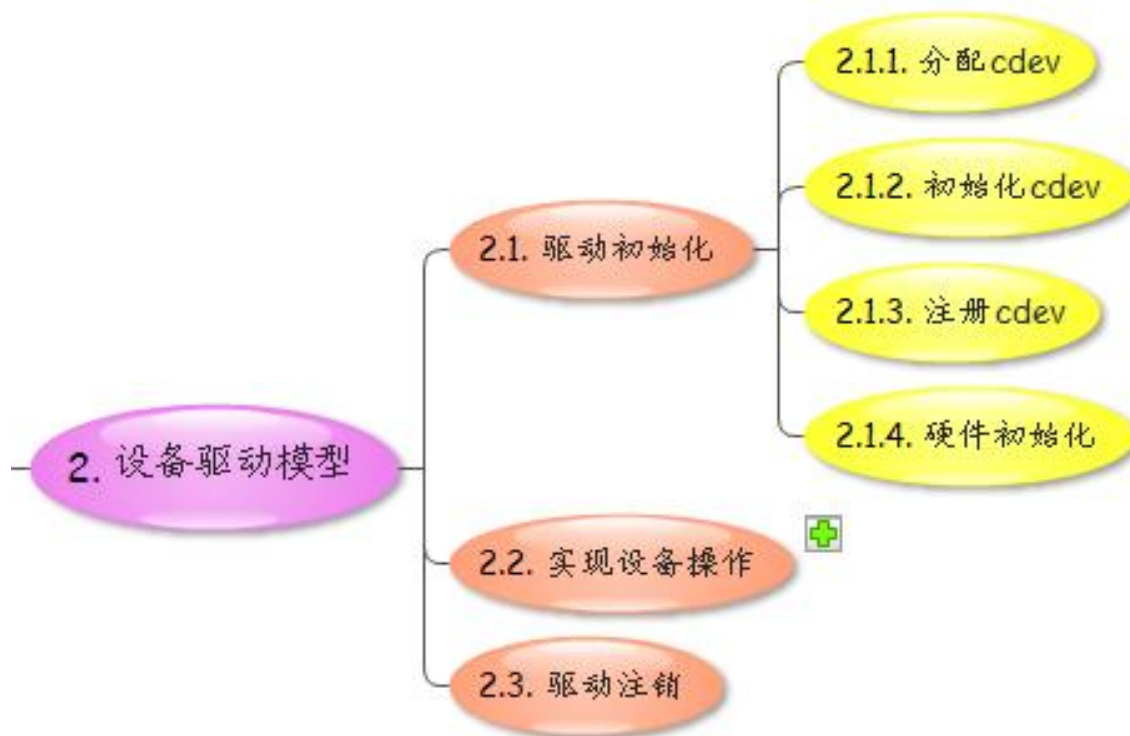
Struct `file_operations`是一个函数指针的集合，定义能在设备上进行的操作。结构中的函数指针指向驱动中的函数，这些函数实现一个针对设备的操作，对于不支持的操作则设置函数指针为 `NULL`。例如：

```
struct file_operations dev_fops = {  
    .llseek = NULL,  
    .read = dev_read,  
    .write = dev_write,  
    .ioctl = dev_ioctl,  
    .open = dev_open,  
    .release = dev_release,  
};
```

嵌入式Linux技术咨询QQ号: 550491596
嵌入式Linux学习交流QQ群: 65212116



2.1 字符设备初始化



嵌入式Linux技术咨询QQ号: 550491596
嵌入式Linux学习交流QQ群: 65212116



2.1 描述结构-分配



www.enjoylinux.cn

cdev变量的定义可以采用静态和动态两种办法

- 静态分配

```
struct cdev mdev;
```

- 动态分配

```
struct cdev *pdev = cdev_alloc();
```

嵌入式Linux技术咨询QQ号: 550491596

嵌入式Linux学习交流QQ群: 65212116



2.1 描述结构-初始化



struct cdev的初始化使用**cdev_init**函数来完成。

```
cdev_init(struct cdev *cdev, const struct file_operations *fops)
```

参数:

cdev: 待初始化的cdev结构

fops: 设备对应的操作函数集

嵌入式Linux技术咨询QQ号: 550491596

嵌入式Linux学习交流QQ群: 65212116



2.1 描述结构-注册



字符设备的注册使用 **cdev_add** 函数来完成。

```
cdev_add(struct cdev *p, dev_t dev, unsigned count)
```

参数:

p: 待添加到内核的字符设备结构

dev: 设备号

count: 该类设备的设备个数

嵌入式Linux技术咨询QQ号: 550491596

嵌入式Linux学习交流QQ群: 65212116

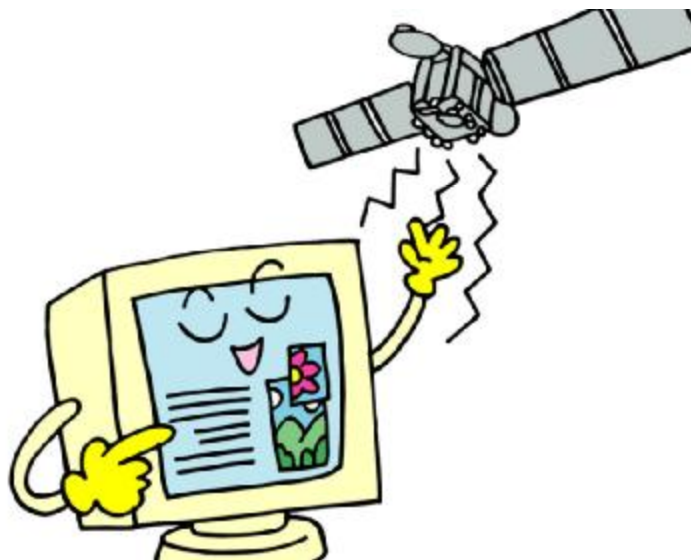


2.1 硬件初始化



www.enjoylinux.cn

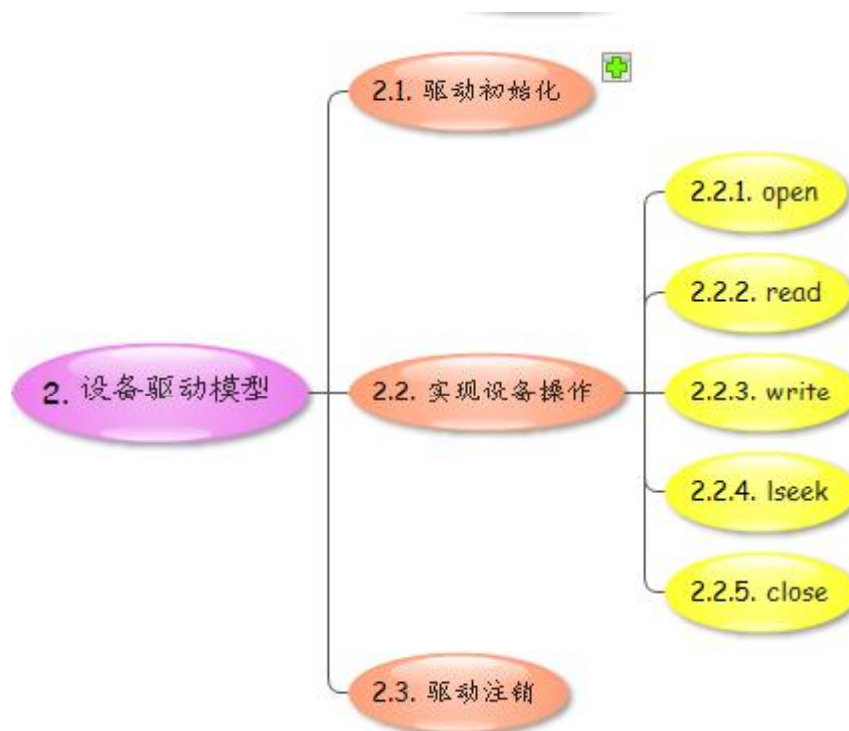
根据相应硬件的芯片手册，完成初始化。



嵌入式Linux技术咨询QQ号: 550491596
嵌入式Linux学习交流QQ群: 65212116



2.2 实现设备操作



嵌入式Linux技术咨询QQ号: 550491596
嵌入式Linux学习交流QQ群: 65212116



2.2 手把手带你来分析



分析
file_operations

嵌入式Linux技术咨询QQ号: 550491596
嵌入式Linux学习交流QQ群: 65212116



2.2 设备操作原型



✓ `int (*open) (struct inode *, struct file *)`

打开设备，响应open系统

✓ `int (*release) (struct inode *, struct file *);`

关闭设备，响应close系统调用

✓ `loff_t (*llseek) (struct file *, loff_t, int)`

重定位读写指针，响应lseek系统调用

嵌入式Linux技术咨询QQ号: 550491596

嵌入式Linux学习交流QQ群: 65212116



2.2 设备操作原型



www.enjoylinux.cn

✓ `ssize_t (*read) (struct file *, char __user *, size_t, loff_t *)`

从设备读取数据，响应read系统调用

✓ `ssize_t (*write) (struct file *, const char __user *, size_t, loff_t *)`

向设备写入数据，响应write系统调用

嵌入式Linux技术咨询QQ号: 550491596

嵌入式Linux学习交流QQ群: 65212116



2.2 Struct file



在Linux系统中，每一个打开的文件，在内核中都会关联一个struct file，它由内核在打开文件时创建，在文件关闭后释放。

✓ 重要成员：

```
loff_t f_pos /*文件读写指针*/
```

```
struct file_operations *f_op /*该文件所对应的操作*/
```

嵌入式Linux技术咨询QQ号：550491596
嵌入式Linux学习交流QQ群：65212116



2.2 Struct inode



www.enjoylinux.cn

每一个存在于文件系统里面的文件都会关联一个 **inode** 结构，该结构主要用来记录文件物理上的信息。因此，它和代表打开文件的file结构是不同的。一个文件没有被打开时不会关联file结构，但是却会关联一个 **inode** 结构。

✓重要成员：

`dev_t` `i_rdev`: 设备号

嵌入式Linux技术咨询QQ号: 550491596

嵌入式Linux学习交流QQ群: 65212116



3.2 设备操作-open



www.enjoylinux.cn

open设备方法是驱动程序用来为以后的操作完成初始化准备工作的。在大部分驱动程序中，open完成如下工作：

- ✓ 标明次设备号
- ✓ 启动设备

嵌入式Linux技术咨询QQ号：550491596
嵌入式Linux学习交流QQ群：65212116



3.2 设备操作-release



www.enjoylinux.cn

release方法的作用正好与open相反。
这个设备方法有时也称为close，它应该：

✓ 关闭设备。

嵌入式Linux技术咨询QQ号：550491596
嵌入式Linux学习交流QQ群：65212116



2.2 设备操作-read

read设备方法通常完成2件事情:

- ✓ 从设备中读取数据(属于硬件访问类操作)
- ✓ 将读取到的数据返回给应用程序

```
ssize_t (*read) (struct file *filp, char __user *buff, size_t count, loff_t *offp)
```

参数分析:

filp: 与字符设备文件关联的file结构指针, 由内核创建。

buff : 从设备读取到的数据, 需要保存到的位置。由read系统调用提供该参数。

count: 请求传输的数据量, 由read系统调用提供该参数。

offp: 文件的读写位置, 由内核从file结构中取出后, 传递进来。

嵌入式Linux技术咨询QQ号: 550491596
嵌入式Linux学习交流QQ群: 65212116



2.2 设备操作-read



buff参数是来源于用户空间的指针，**这类指针都不能被内核代码直接引用**，必须使用专门的函数

- ✓ int **copy_from_user**(void *to, const void __user *from, int n)
- ✓ int **copy_to_user**(void __user *to, const void *from, int n)

嵌入式Linux技术咨询QQ号: 550491596
嵌入式Linux学习交流QQ群: 65212116

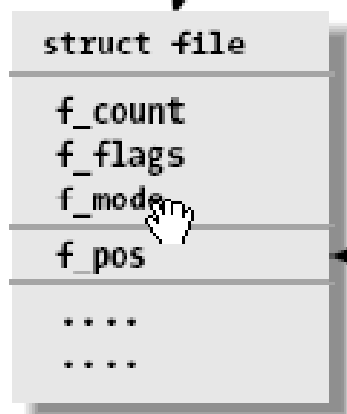


2.2 设备操作-read

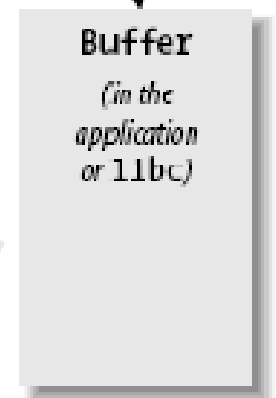


www.enjoylinux.cn

```
ssize_t dev_read(struct file *file, char *buf, size_t count, loff_t *ppos);
```



`copy_to_user()`



Kernel Space
(nonswappable)

User Space
(swappable)

嵌入式Linux技术咨询QQ号: 550491596
嵌入式Linux学习交流QQ群: 65212116



2.2 设备操作-write

write设备方法通常完成2件事情:

- ✓ 从应用程序提供的地址中取出数据
- ✓ 将数据写入设备(属于硬件访问类操作)

`ssize_t (*write) (struct file *, const char __user *, size_t, loff_t *)`

其参数类似于read

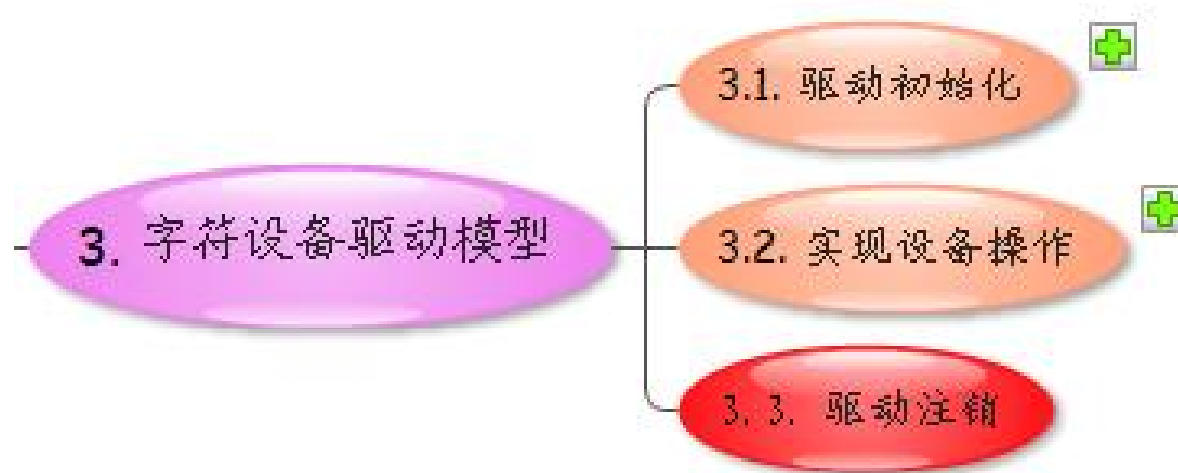
嵌入式Linux技术咨询QQ号: 550491596
嵌入式Linux学习交流QQ群: 65212116



2.3 驱动注销



www.enjoylinux.cn



嵌入式Linux技术咨询QQ号: 550491596
嵌入式Linux学习交流QQ群: 65212116

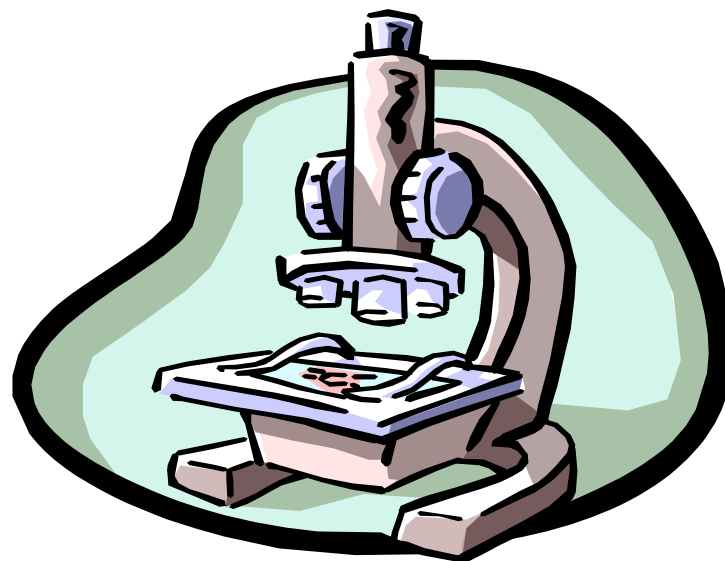




www.enjoylinux.cn

2.3 驱动注销

当我们从内核中卸载驱动程序的时候，需要使用 **cdev_del** 函数来完成字符设备的注销。



嵌入式Linux技术咨询QQ号: 550491596
嵌入式Linux学习交流QQ群: 65212116



3. 手把手带你读代码



www.enjoylinux.cn



分析 字符设备驱动范例

嵌入式Linux技术咨询QQ号: 550491596
嵌入式Linux学习交流QQ群: 65212116

