

深入浅出Linux字符设备驱动程序解析 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/213/2021_2022__E6_B7_B1_E5_85_A5_E6_B5_85_E5_c103_213938.htm Linux下的设备驱动程序被组织为一组完成不同任务的函数的集合，通过这些函数使得linux的设备操作犹如文件一般。在应用程序看来，硬件设备只是一个设备文件，应用程序可以象操作普通文件一样对硬件设备进行操作，如open()、close()、read()、write()等。Linux主要将设备分为二类：字符设备和块设备。字符设备是指设备发送和接收数据以字符的形式进行；而块设备则以整个数据缓冲区的形式进行。字符设备的驱动相对比较简单。下面我们来假设一个非常简单的虚拟字符设备：这个设备中只有一个4个字节的全局变量int global_var，而这个设备的名字叫做"globalvar"。对"globalvar"设备的读写等操作即是对其中全局变量global_var的操作。驱动程序是内核的一部分，因此我们需要给其添加模块初始化函数，该函数用来完成对所控设备的初始化工作，并调用register_chrdev() 函数注册字符设备：

```
static int __init globalvar_init(void){ if
(register_chrdev(MAJOR_NUM, " globalvar ", amp.global_var,
sizeof(int)). ... }write() 函数 当设备特殊文件进行write () 系统调用时，将调用驱动程序的write () 函数： ssize_t (*write) (struct
file *, const char *, size_t, loff_t *). 向设备发送数据。如果没有这个函数，write 系统调用会向调用程序返回一个-EINVAL。如果返回值非负，则表示成功写入的字节数。 globalvar_write函数中内核空间与用户空间的内存交互需要借助第2节所介绍的函数：
```

```
static ssize_t globalvar_write(struct file *filp, const char *buf,
```

```
size_t len, loff_t *off){...copy_from_user(&global_var, buf,
sizeof(int))....}ioctl() 函数 该函数是特殊的控制函数，可以通过它向设备传递控制信息或从设备取得状态信息，函数原型为：int (*ioctl) (struct inode *,struct file *,unsigned int ,unsigned long). unsigned int参数为设备驱动程序要执行的命令的代码，由用户自定义，unsigned long参数为相应的命令提供参数，类型可以是整型、指针等。如果设备不提供ioctl 入口点，则对于任何内核未预先定义的请求，ioctl 系统调用将返回错误（-ENOTTY, "No such ioctl for device, 该设备无此ioctl 命令"）。如果该设备方法返回一个非负值，那么该值会被返回给调用程序以表示调用成功。100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com
```