深入浅出Linux设备驱动之并发控制（3） PDF转换可能丢失图片或格式，建议阅读原文 https://www.100test.com/kao_ti2020/144/2021_2022__E6_B7_B1_E5_85_A5_E6_B5_85_E5_c103_144577.htm 接下来，我们给globalvar的驱动程序增加open()和release()函数，并在其中借助自旋锁来保护对全局变量int globalvar_count（记录打开设备的进程数）的访问来实现设备只能被一个进程打开（必须确保globalvar_count最多只能为1）： #include #include #include #include #include MODULE_LICENSE("GPL").#define MAJOR_NUM 254static ssize_t globalvar_read(struct file *, char *, size_t, loff_t*).static ssize_t globalvar_write(struct file *, const char *, size_t, loff_t*).static int globalvar_open(struct inode *inode, struct file *filp).static int globalvar_release(struct inode *inode, struct file *filp).struct file_operations globalvar_fops ={ read: globalvar_read, write: globalvar_write, open: globalvar_open, release:globalvar_release,}.static int global_var = 0.static int globalvar_count = 0.static struct semaphore sem.static spinlock_t spin = SPIN_LOCK_UNLOCKED.static int __init globalvar_init(void){ int ret. ret = register_chrdev(MAJOR_NUM, "globalvar", amp.sem). } return ret.}static void __exit globalvar_exit(void){ int ret. ret = unregister_chrdev(MAJOR_NUM, "globalvar"). if (ret) { printk("globalvar unregister failure"). } else { printk("globalvar unregister success"). }}static int globalvar_open(struct inode *inode, struct file *filp){ //获得自选锁 spin_lock(amp.spin). return -EBUSY. } globalvar_count . //释放自选锁 spin_unlock(amp.sem))

{ return - ERESTARTSYS. } if (copy_to_user(buf, amp.sem). return - EFAULT. } up(amp.sem)) { return - ERESTARTSYS. } if (copy_from_user(amp.sem). return - EFAULT. } up(&amp.sem). return sizeof(int).}module_init(globalvar_init).module_exit(globalvar_exit).为了上述驱动程序的效果，我们启动两个进程分别打开/dev/globalvar。在两个终端中调用./globalvartest.o测试程序，当一个进程打开/dev/globalvar后，另外一个进程将打开失败，输出"device open failure"，如下图： 输出结果 