

迅为电子Linux驱动教程 —字符驱动

北京迅为电子有限公司





本节目标

- 完成字符驱动
 - 完善file_operations中的函数
- 编写简单的应用



完成字符驱动

- file_operations中的函数比较多，选取用的比较多的函数简单介绍，后面的驱动教程中调用了对应的函数，再详细介绍
- int (*open) (struct inode *, struct file *)
 - 打开函数
- int (*release) (struct inode *, struct file *)
 - 释放close函数
- long (*unlocked_ioctl) (struct file *, unsigned int, unsigned long)
 - io控制函数



完成字符驱动

- `ssize_t (*read) (struct file *, char __user *, size_t, loff_t *)`
 - 读函数
- `ssize_t (*write) (struct file *, const char __user *, size_t, loff_t *)`
 - 写函数
- `loff_t (*llseek) (struct file *, loff_t, int)`
 - 定位函数
- 如果需要不同的设备节点有不同的功能，只需要在注册设备的时候添加不同的file_operations结构体即可



编写编译运行测试

- 将驱动视频教程20中的“create_cnode.c”改为“char_driver.c”
- 修改编译文件Makefile
- 将驱动视频教程09中“invoke_hello.c”改为“invoke_char_driver.c”，编译应用命令如下
 - arm-none-linux-gnueabi-gcc -o invoke_char_driver invoke_char_driver.c -static



谢谢！