23-驱动理论

1、模块化编程
2、驱动理论
3、函数接口
4、中断编程
5、内核定时器
6、内核延时执行机制
7、内核同步机制

一、模块化编程

驱动调试阶段，以单个模块加载内核，将.c文件利用模板编译为.ko文件

insmod xxx.ko //加载

rmmod xxx.ko //卸载

制作文件系统：制作很多的命令 /bin 出错？ /etc 下面的配置文件

单模块编程、多模块编程

传参： module_param、module_param_array

二、驱动理论

应用层通过调用驱动层注册的接口，完成对硬件的操作。

应用层：open、close、read、write、ioctl、lseek、poll等

驱动层：file_operations指定的函数

应用层的open --- 驱动层的open

驱动层：注册/dev节点 字符设备c、网络设备s、块设备b

杂项设备注册、经典方式注册、linux2.6版本

杂项设备：misc_register struct miscdevice描述一个字符设备(指定次设备号、节点名字、关联的file_operations)

经典方式注册：register_chrdev(int major,char *pathname,struct file_operations *f_ops);

linux2.6版本：申请设备号、添加设备到内核

注册设备节点：class_create、device_create

编写驱动：
框架：GPIO 映射地址：ioremap、iounmap
硬件配置：分析数据手册、参考裸机程序、复杂驱动：内核提供的接口
填充file_operations？ led灯:ioctl、write key：read 串口：read、write
三、函数接口
file_operations:open、release、read、write、ioctl、llseek、poll、fasync
open:硬件开启的操作
release：硬件的关闭的操作
read：copy_to_user 应用层读操作 驱动层返回给应用层数据
write：copy_from_user 应用层写操作 驱动层读取应用层数据
ioctl：参数传递 参数个数、放在被系统占用 cmd不能为2
lseek：参数的传递 偏移量、参考位置 (考虑文件大小)
poll：轮询机制(一定时间内去轮询等待队列上的事件，如果有时间解除阻塞返回真)
fasync：异步通知(驱动有事件发生，内核可以向应用层发送指定的信号)
四、中断编程
将IO配置中断，中断发生执行指定的中断服务函数。
request_irq：注册中断
应用层无法感知中断发生以及读取相应数据？
注册设备、添加等待队列。
五、内核定时器
1、定义struct timer_list结构体指定定时器函数、超时时间、传递的参数
2、初始化结构体
3、添加定时器到内核
4、修改定时器的计时时间

利用定时器：延时、消抖　中断服务函数：修改推后定时器 定时器：唤醒等待队列

六、内核延时执行机制(相当于内核的线程操作)

1、小任务 (创建、调度、终止、删除、传参)

2、工作队列 (创建、调度、终止、删除、传参)

七、内核的同步机制

信号量、锁

信号量：类似睡眠锁，可以用在不同的函数中

锁：忙等待锁，在一个函数当中保护一段临界区的代码

复习

预习平台设备总线