

专题18-FLASH驱动程序设计

一、块设备驱动系统架构

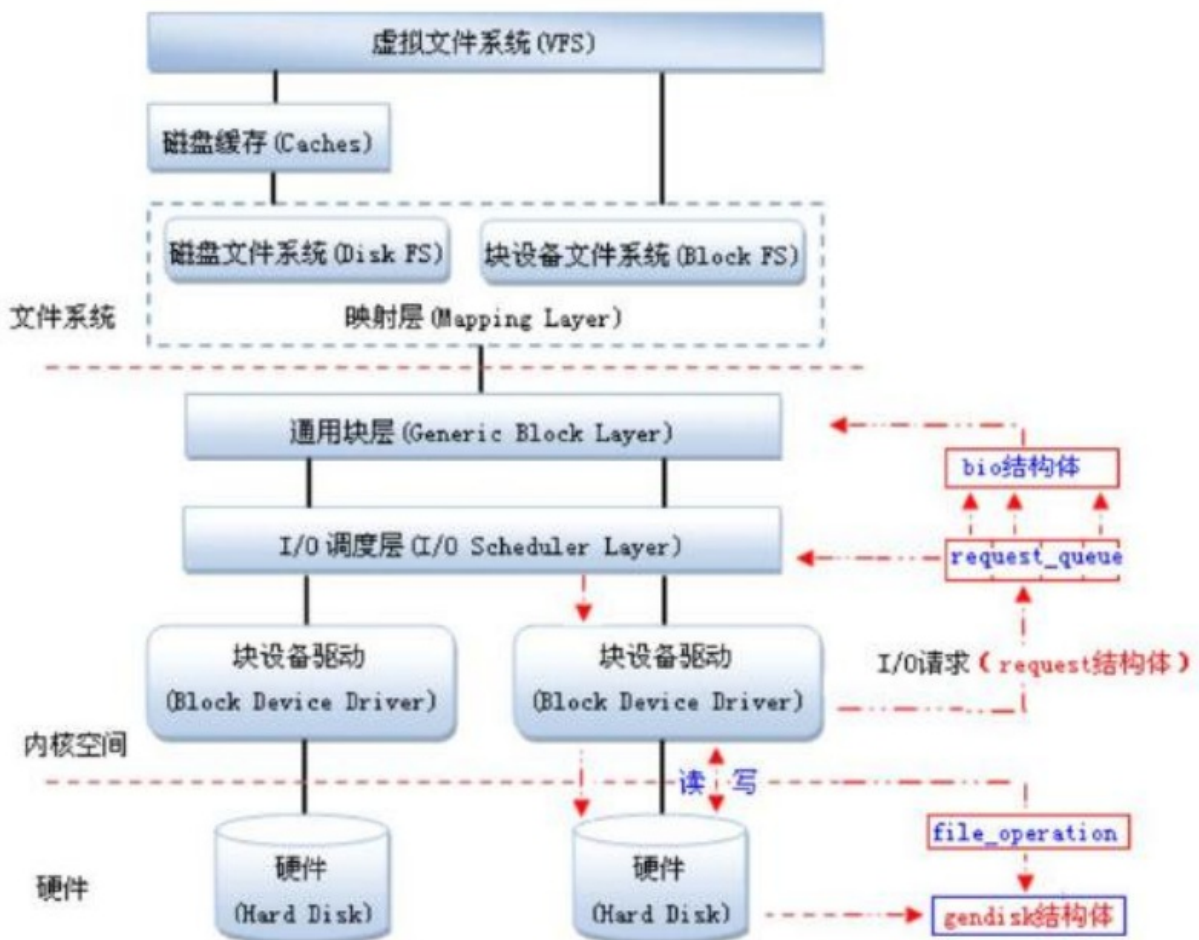
1.1、块设备快速体验

块设备是指只能以块为单位进行访问的设备，块大小一般是512个字节的整数倍。常见的块设备包括硬件，SD卡，光盘等。

体验块设备使用

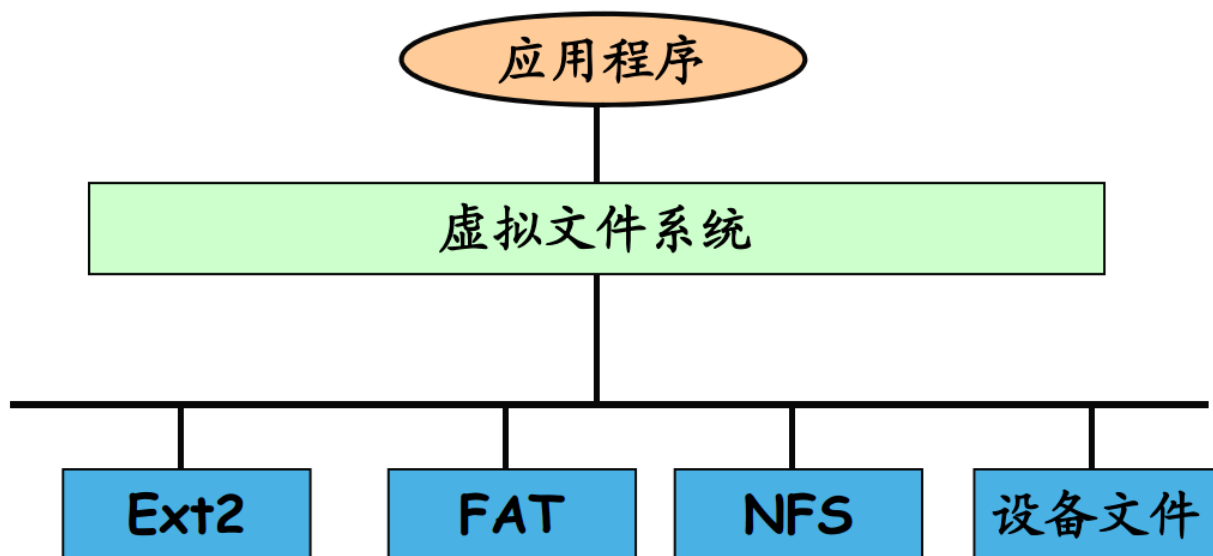
1. insmod simple-blk.ko
2. ls /dev/simp_blkdev0
3. mkfs.ext3 /dev/simp_blk0
4. mkdir -p /mnt/blk
5. mount /dev/simp_blk0 /mnt/blk
6. cp /etc/init.d/* /mnt/blk
7. ls /mnt/blk
8. umount /mnt/blk
9. ls /mnt/blk

1.2、块设备驱动系统架构



1.2.1、系统架构-VFS

VFS是对各种具体文件系统的一种封装，为用户程序访问文件提供统一的接口。



1.2.2、系统架构-Cache

当用户发起文件访问请求的时候，首先会到 Disk Cache中寻找文件是否被缓存了，如果在cache中，则直接从cache中读取。如果数据不在缓存中，就必须要到具体的文件系统中读取数据了。

1.2.3、Mapping Layer

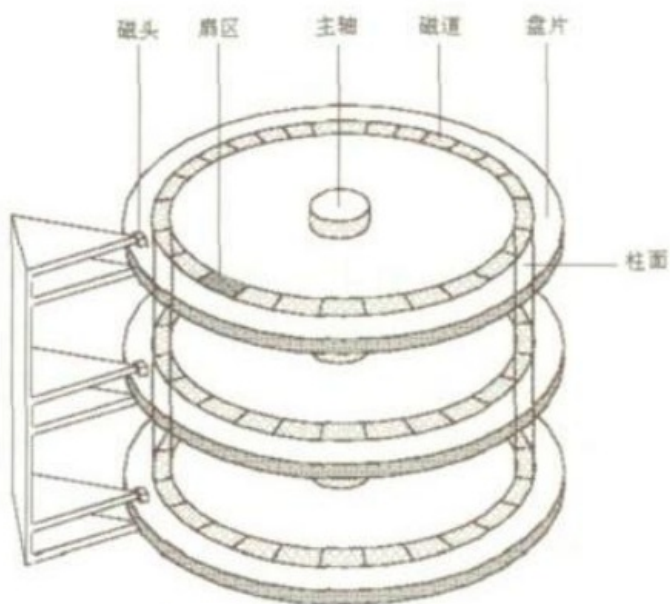
1. 首先确定文件系统的block size,然后计算 所请求的数据包含多少个block。
2. 调用具体文件系统的函数来访问文件的 inode 结构，确定所请求的数据在磁盘上的地址。

1.2.4、Generic Block Layer

Linux内核把块设备看作是由若干个扇区组成的数据空间。上层的读写请求在通用块层 被构造成一个或多个bio结构。

1.2.4、I/O Scheduler Layer

I/O调度层负责采用某种算法(如：电梯调度 算法)将I/O操作进行排序。

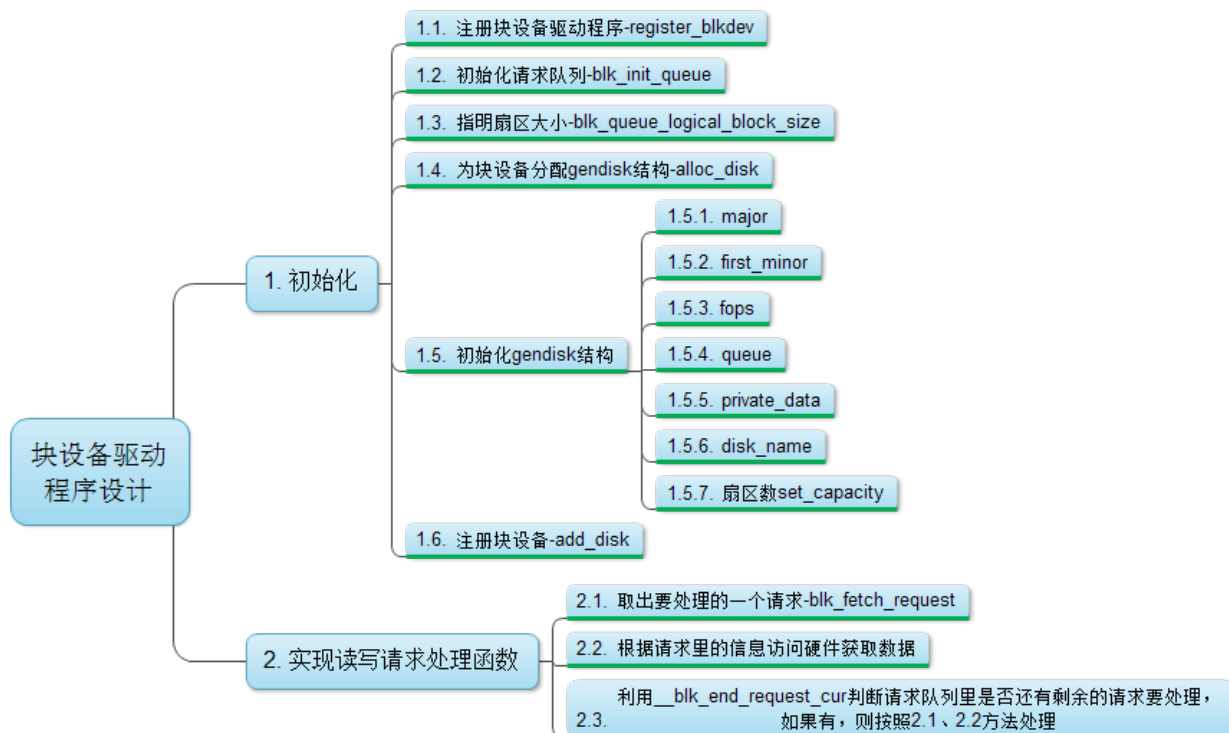


电梯调度算法的基本原则：如果电梯现在朝上运动，如果当前楼层的上方和下方都有请求，则先响应所有上方的请求，然后才向下响应下方的请求；如果电梯向下运动，则刚好相反。

1.2.6、块设备驱动

在块系统架构的最底层，由块设备驱动根据 排序好的请求，对硬件进行数据访问。

二、块设备驱动系统实例分析



三、简单块设备驱动设计

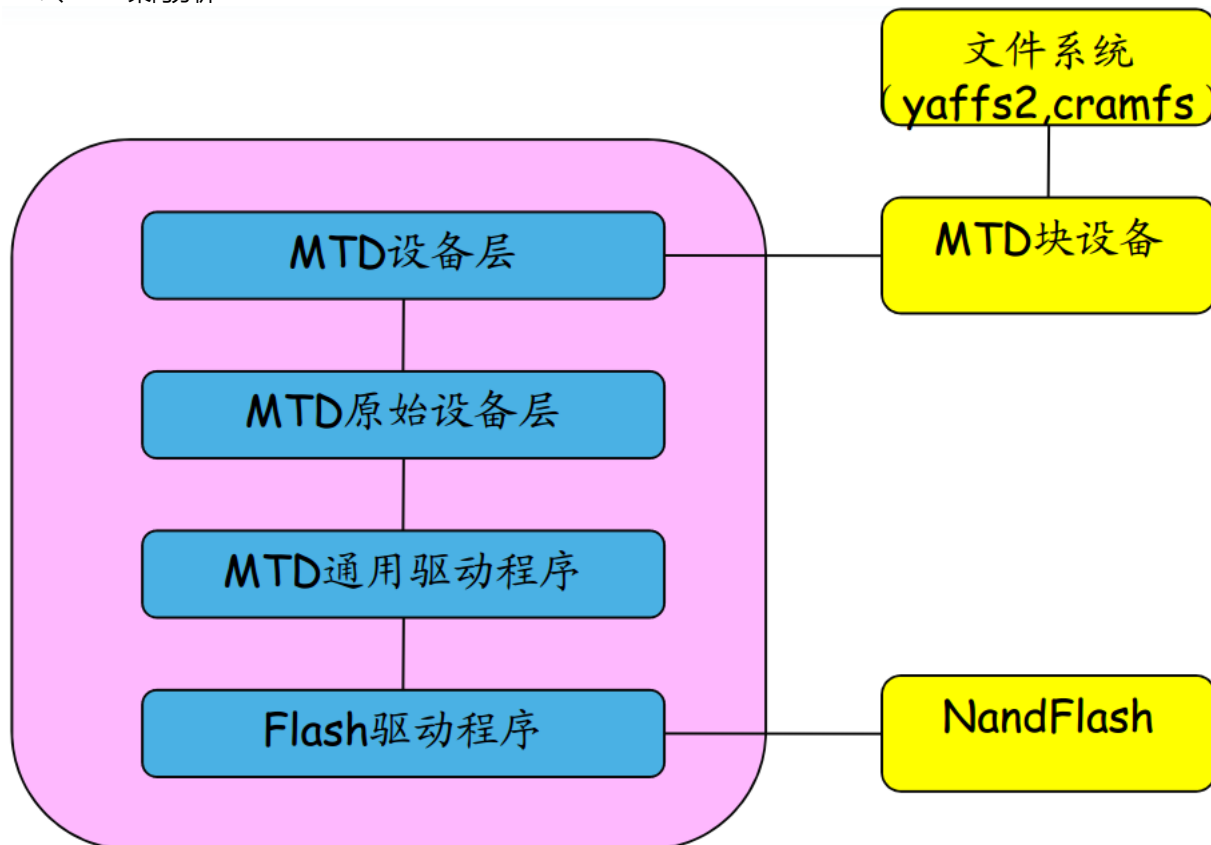
根据课程视频的代码会导致linux系统死机。

四、MTD系统架构

4.1、MTD设备体验

FLASH在嵌入式系统中是必不可少的，它是 bootloader、linux内核和文件系统的最佳载体。在Linux内核中引入了MTD子系统为NOR FLASH和NAND FLASH设备提供统一的接口，从而使得FLASH驱动的设计大为简化。

4.2、MTD架构分析



五、yaffs2文件系统使用

5.1、MTD分区设置

5.2、yaffs2文件系统制作

5.3、U-boot参数设置

5.4、下载烧写与启动

六、nandflash驱动程序设计