

用户级文件系统

裁剪UNIX V6++源代码，管理二级文件系统。

简介： 识别、研读 UNIX V6++系统中 **文件系统**和**缓存管理模块**。按需摘取其中的代码和数据结构，用以管理二级文件系统。

目标： 学习UNIX V6++如何分解文件系统功能，依托开源软件进行二次开发。

任务

- 把一个大文件当一张磁盘用。比如，myDisk.img。
- 把随意的一个纯文本文件，你的课设报告和你的头像存进这个文件系统，分别放在/home/texts，/home/reports和/home/photos文件夹



分配空间，存放
文件数据 和 inodes

myDisk.img

myDisk.img文件布局

- 磁盘

- UNIX文件卷

- **UNIX V6++ Volume**

- Linux ext2 Volume
 - Linux ext3、ext4……

- Windows文件卷

- FAT16
 - FAT32
 - NTFS

myDisk.img是一张磁盘，存放着一个文件系统。这张磁盘存放的是什么文件系统，我们就称这张磁盘是什么卷。比如，

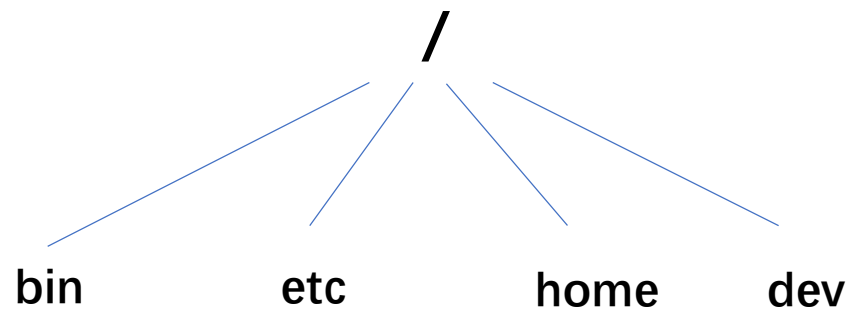
1、存放的是Unix V6++文件系统，磁盘就叫做UNIX V6++文件卷；

超级块	inode区	数据区
-----	--------	-----

2、存放的是FAT32文件系统，磁盘就叫做 FAT32 文件卷；

3、存放的是Linux ext2文件系统，这就是Linux系统管理的标准机械硬盘；

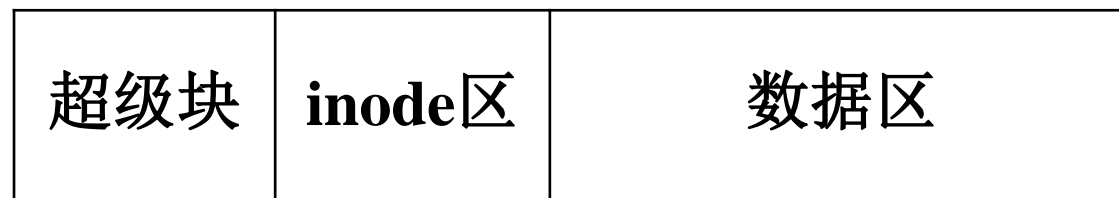
1、文件树和 磁盘格式化



任务： 写format程序

格式化磁盘

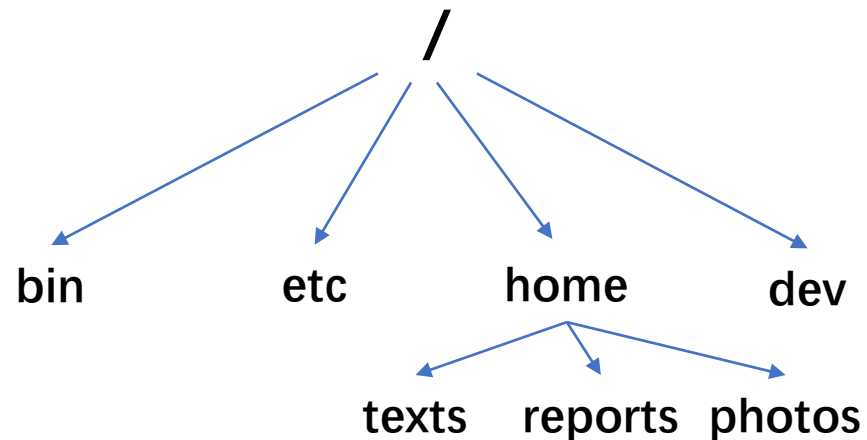
这是我们要交的第1个程序



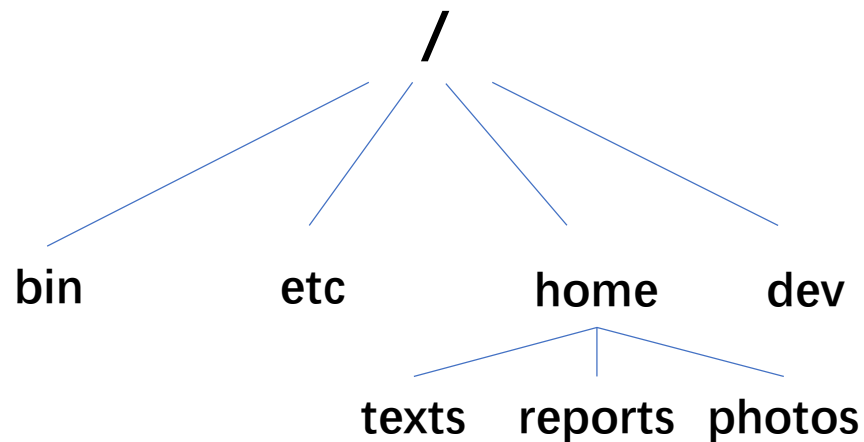
2、文件系统的使用

- 第2个程序 secondFS
- 实现系统调用
 - creat
 - open
 - read
 - write
 - lseek (seek)
 - close
 - mkdir
 - rm命令
 - 你自己玩的其它
- 用这些系统调用完成PPT2中的任务

超级块	inode区	数据区
-----	--------	-----



格式化后
创建3个子目录



例：

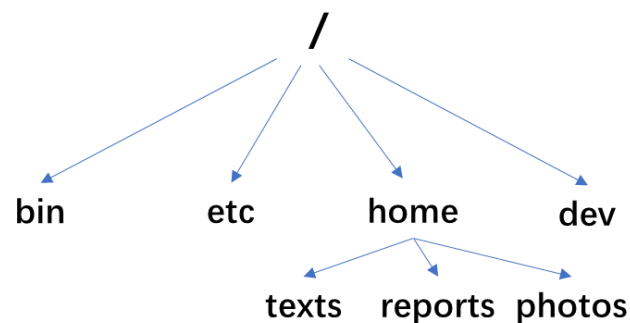
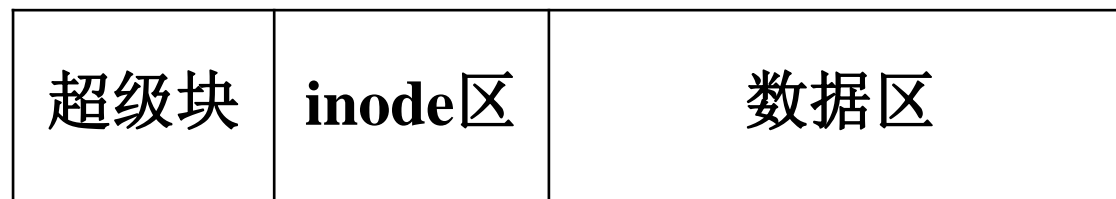
`mkdir /home/texts`

- 1、打开/home目录文件
- 2、为 text目录分配一个新的 inode、一个新的磁盘数据块
- 3、将text目录文件的目录项 写入 /home目录文件
- 4、写text目录文件

任务1

ddl: 4月1日

- 参考Unix V6++文件系统的静态数据结构，设计二级文件系统的metadata。
- 以伪代码的形式给出磁盘格式化和以下系统调用的实现过程
 - creat
 - open
 - read
 - write
 - lseek (seek)
 - close
 - mkdir
 - rm命令
 - 你自己玩的其它

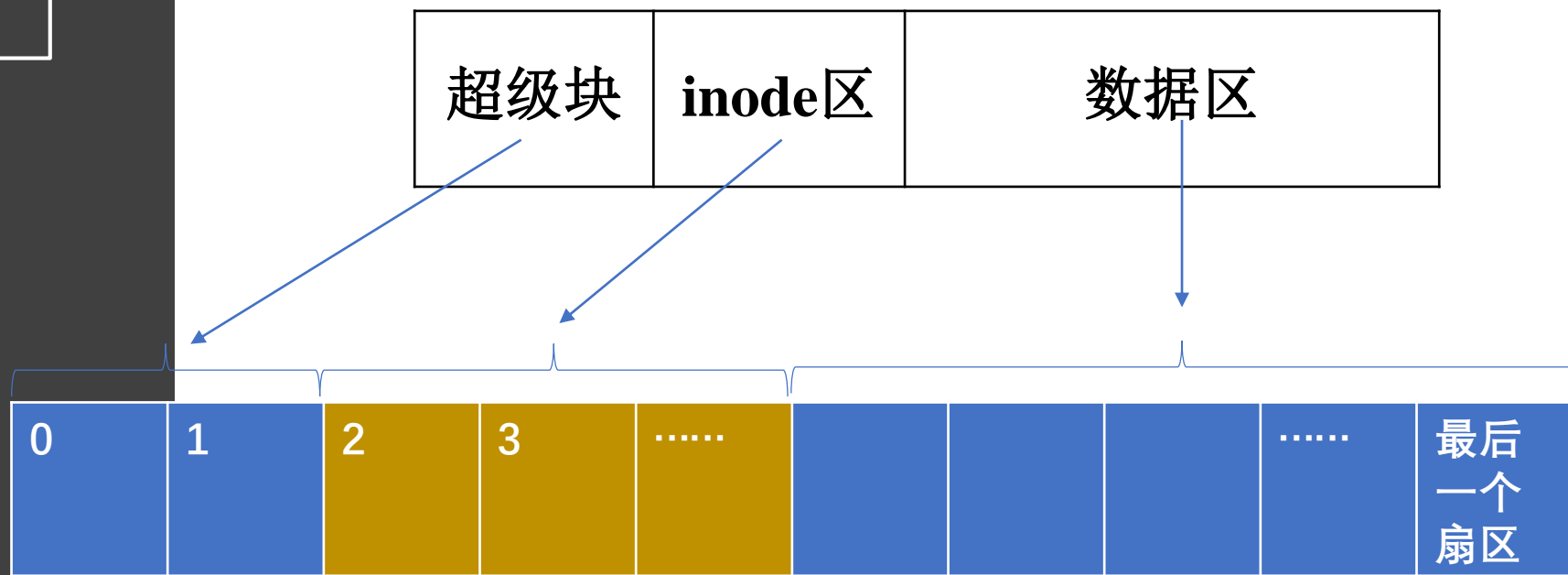


任务：写format程序
格式化磁盘
这是我们要交的第1个程序

- 第2个程序 secondFS
- 实现系统调用
 - creat
 - open
 - read
 - write
 - lseek (seek)
 - close
 - mkdir
 - rm命令
 - 你自己玩的其它
- 用这些系统调用完成PPT2中的任务

myDisk.img文件 数据块和布局

- 数据块等长，512字节



- 二级文件系统的磁盘驱动

myDisk.img文件

任务2

元数据 (metadata)
是二进制数据结构

把metadata写入myDisk.img

完成第1个程序 Format

```
class SuperBlock
{
    ...;
    /* Members */
public:
    int     s_isize;        /* 外存Inode区占用的盘块数 */
    int     s_fsize;        /* 盘块总数 */
    int     s_nfree;        /* 直接管理的空闲盘块数量 */
    int     s_free[100];    /* 直接管理的空闲盘块索引表 */

    int     s_ninode;       /* 直接管理的空闲外存Inode数量 */
    int     s_inode[100];   /* 直接管理的空闲外存Inode索引表 */

    int     s_flock;        /* 封锁空闲盘块索引表标志 */
    int     s_ilock;        /* 封锁空闲Inode表标志 */

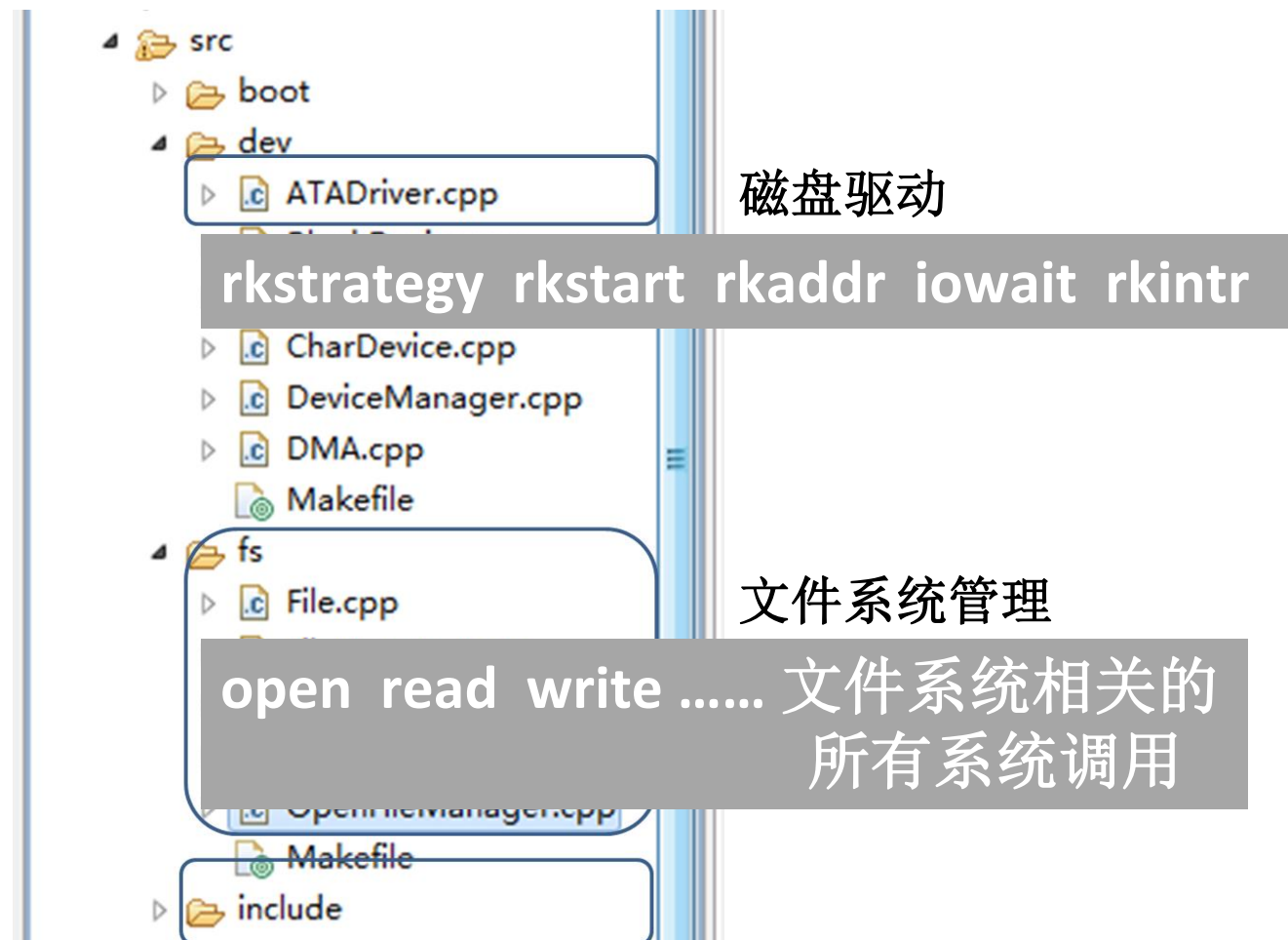
    int     s_fmod;         /* 内存中super block副本被修改标志 */
    int     s_ronly;        /* 本文件系统只能读出 */
    int     s_time;         /* 最近一次更新时间 */
    int     padding[47];    /* 填充使SuperBlock块大小为1024字节 */
};
```



任务 3

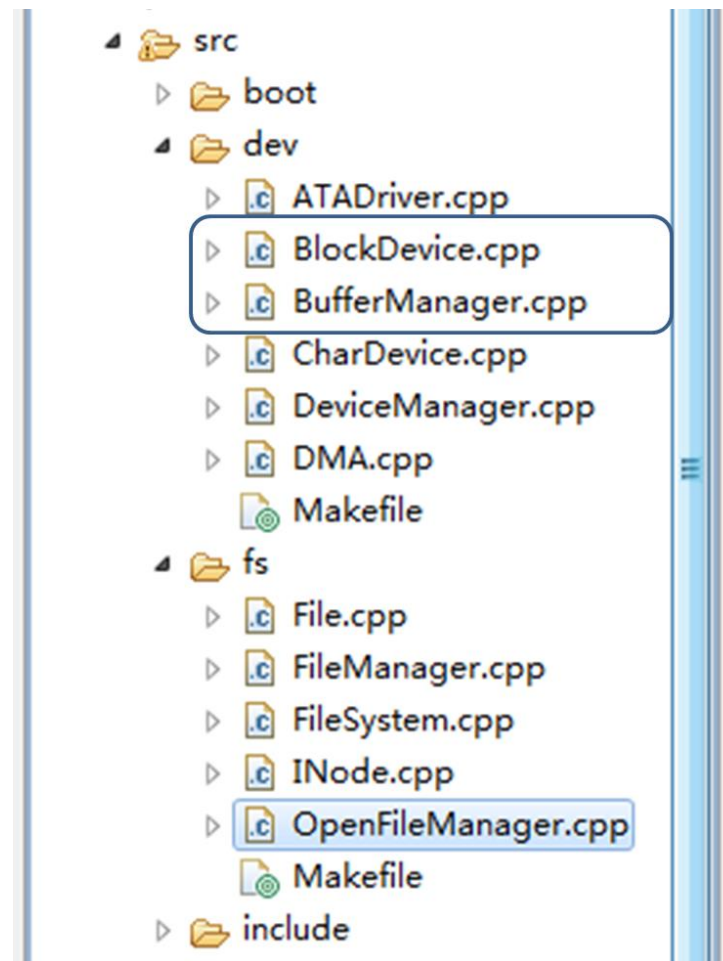
精读 UNIX V6++ 文件系统源代码

按任务要求裁剪源代码，
用在secondFS程序中，
完成课设任务



任务 4 磁盘高速缓存 (加分项)

实现磁盘高速缓存
提升系统性能



任务 5 并发 (加分项)

在第2个程序secondFS中
允许多个用户同时访问
我们的二级文件系统

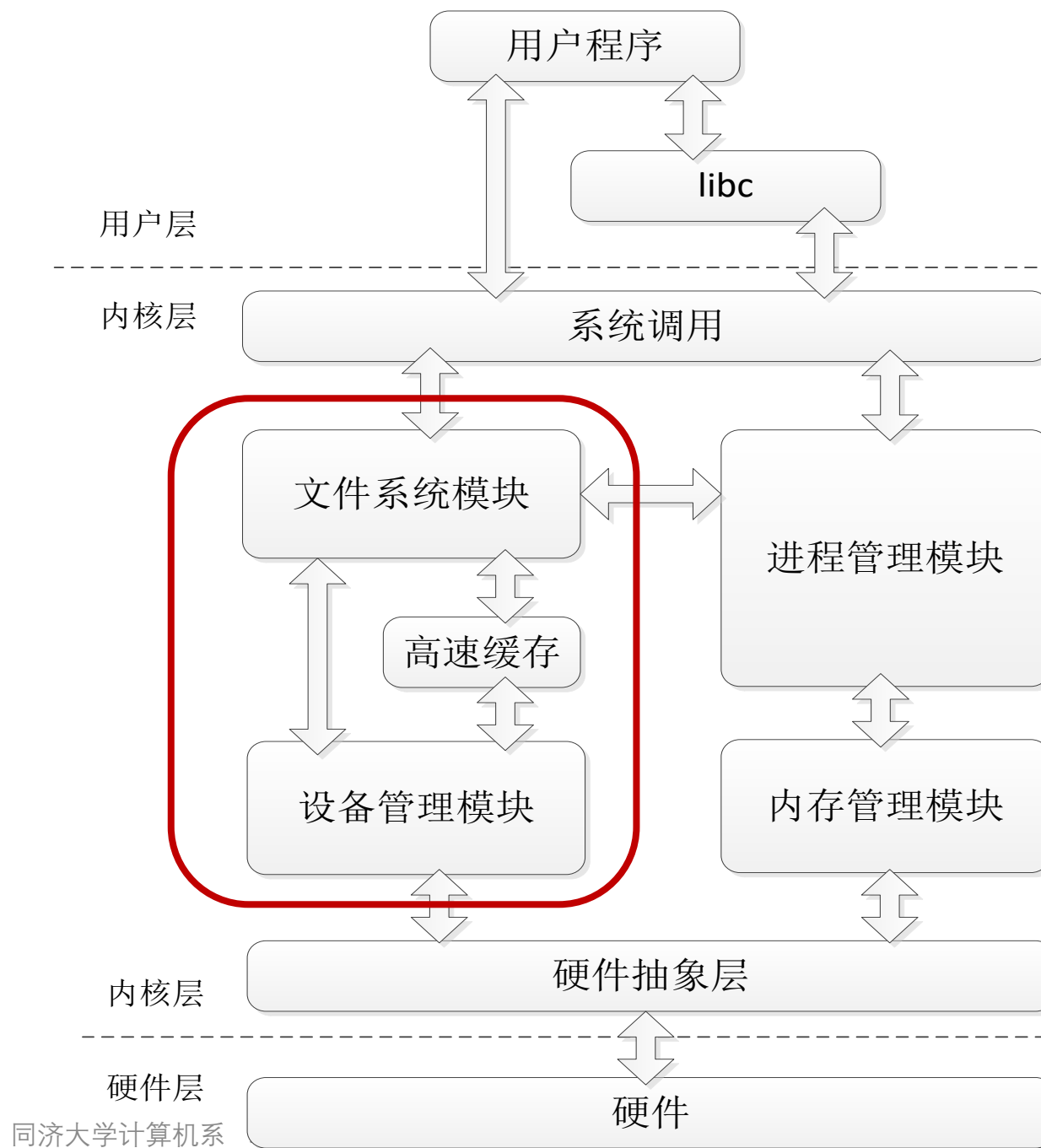
提示：

- 1、需要区分不同用户提交的系统调用。这可能需要为每个用户提供一个界面。
- 2、需要实现内存打开文件结构，合理解决多用户同时访问同一个文件的问题。

PS：磁盘高速缓存层和内存打开文件结构是共享数据结构，为所有用户共享。

附录A

UNIX系统架构



任务6 设备驱动

- 二级文件系统secondFS初始化的时候，将超级块读入内存中的数据结构SuperBlock。
- 二级文件系统secondFS运行时，以数据块为单位，数据块的块号（blkno）为索引，将磁盘上的数据（文件数据和metadata）读入磁盘高速缓存池。



myDisk.img文件

IO请求队列需要嘛？
设备队列需要嘛？
自由缓存队列？同学们自己设计。

附录B UNIX文件系统架构

