

操作系统

Operating Systems

L31 目录与文件系统

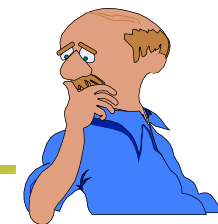
File System

lizhijun_os@hit.edu.cn

综合楼411室

授课教师：李治军

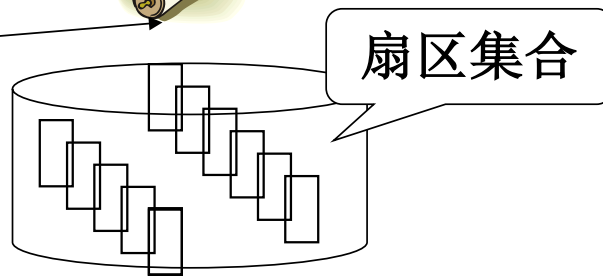
文件，抽象一个磁盘块集合



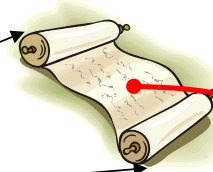
■ 用户眼里文件的样子



■ 磁盘上的文件的样子

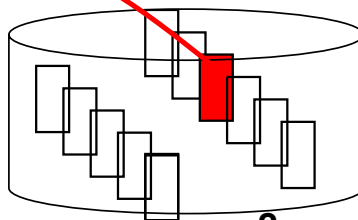


■ 磁盘文件：建立了字符流到盘块集合的映射关系



**test.c中的202-112
字符对应盘块789**

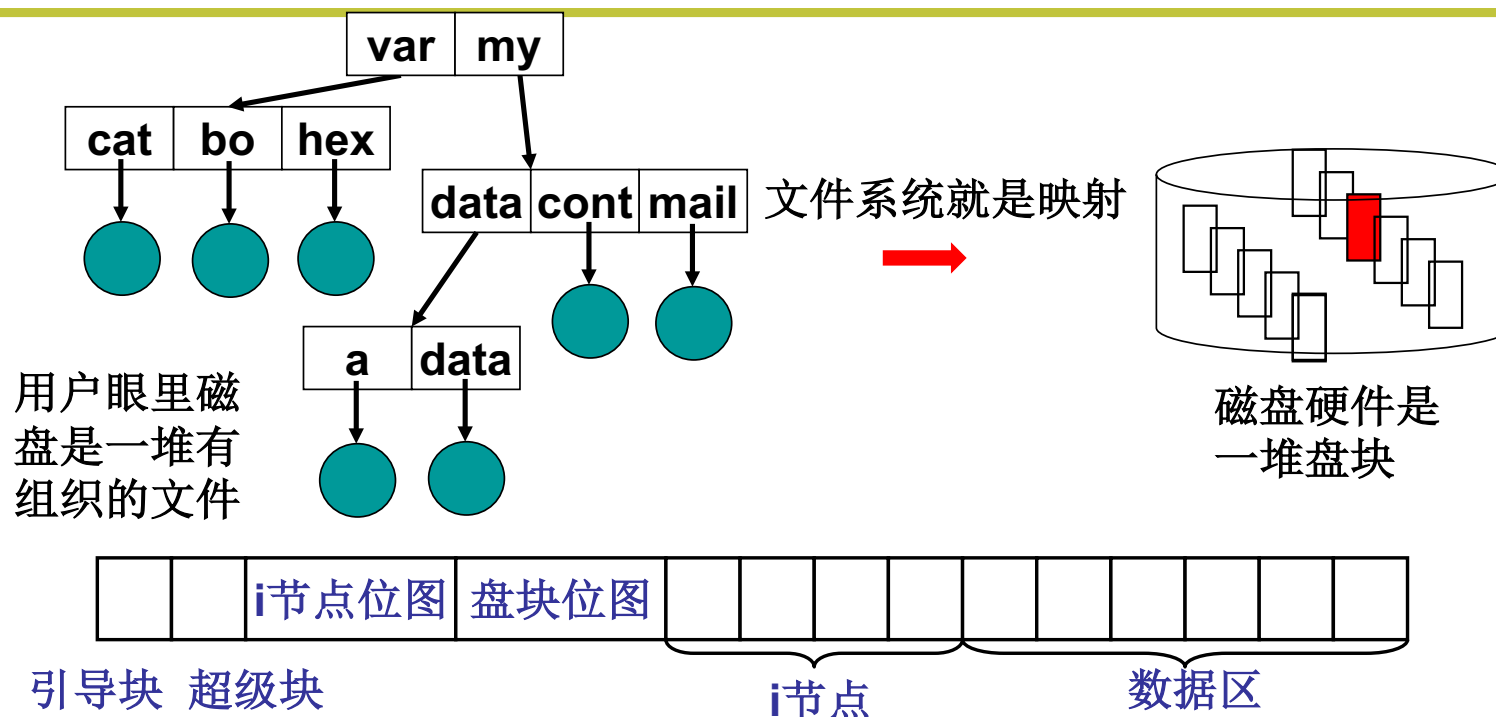
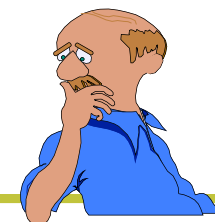
**将202-212字符
删去**



- 2 -



文件系统，抽象整个磁盘(第四层抽象)



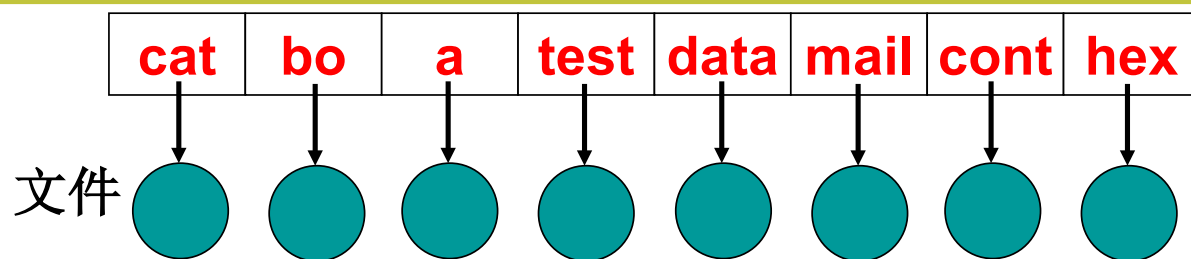
在其他计算机上：应用结构+存储的数据可以得到那

棵文件树，找到文件、读写文件...

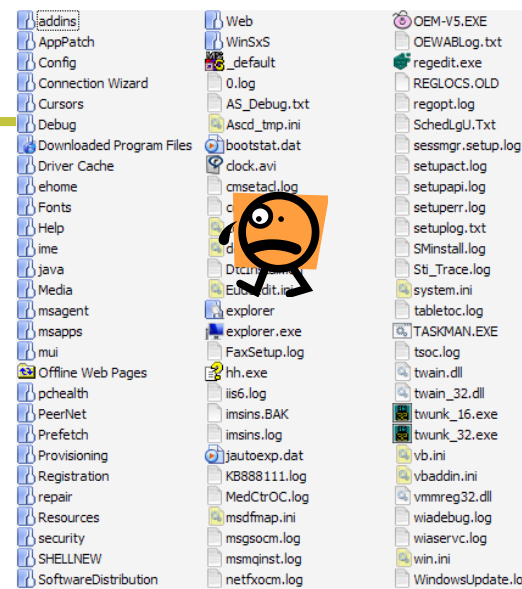
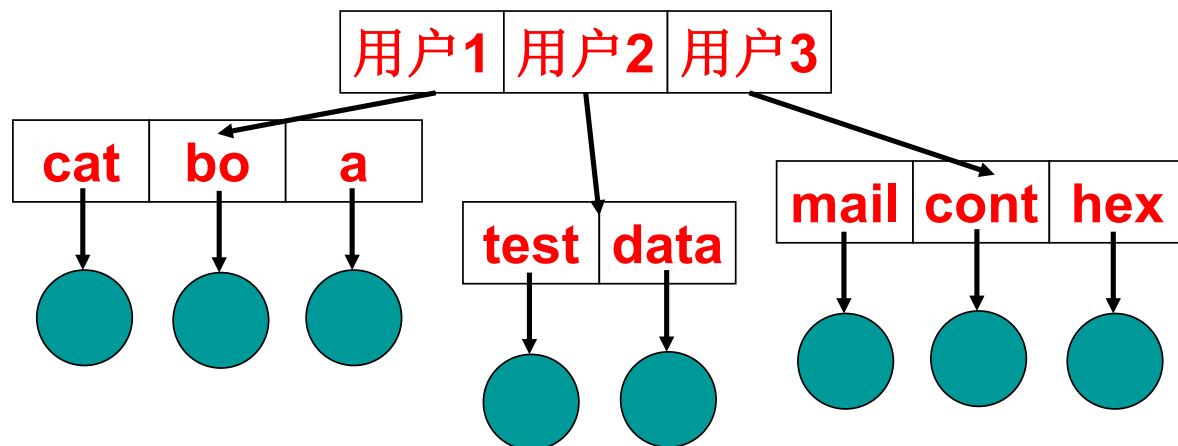
这才是系统，这
就是文件系统



故事先从多个文件开始...

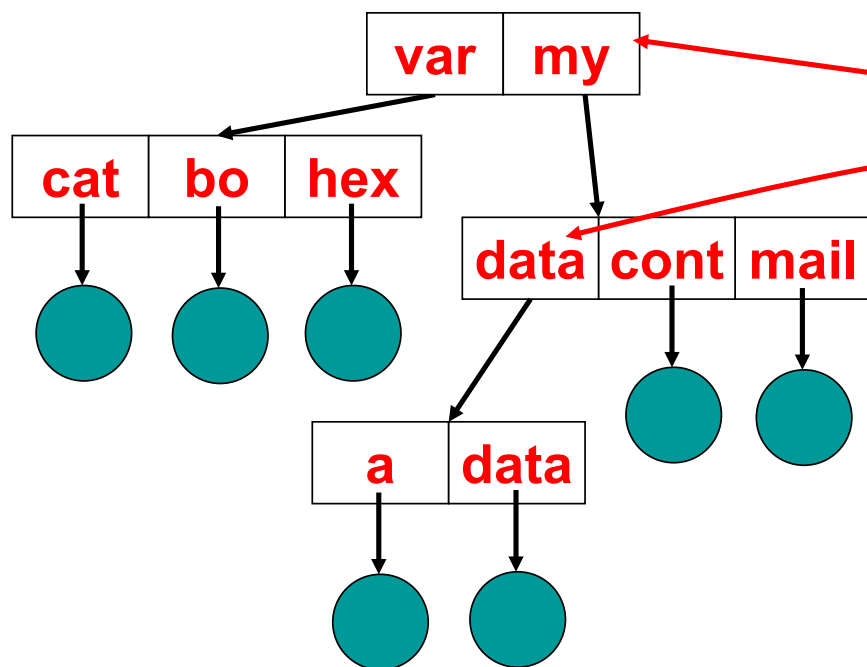


- 所有文件放在一层(一个大集合)
- 怎么办? 集合划分、分治



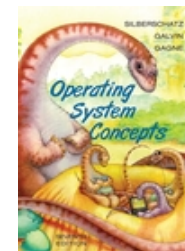
引入目录树

- 将划分后的集合再进行划分: k 次划分后, 每个集合中的文件数为 $O(\log_k N)$



■ 这一树状结构扩展性好、表示清晰, 最常用

■ 引入了一个新的东西: 目录, 表示一个文件集合



实现目录成为了关键问题...

■ 首先需要回答：目录怎么用？

- 用 **“/my/data/a”** 定位文件a

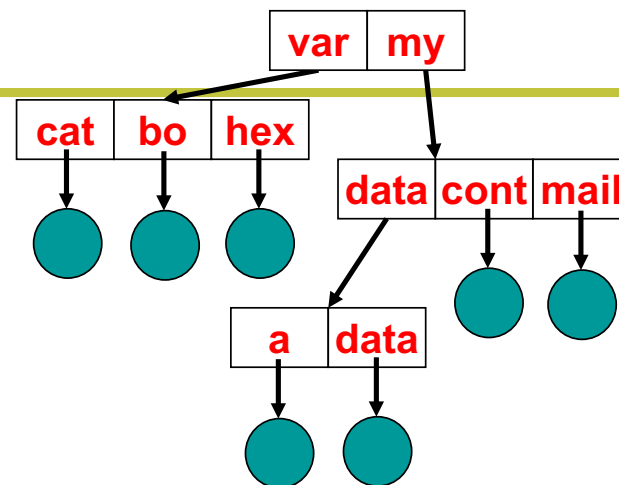
问题：更准确的说，是要干什么？

根据**/my/data/a**，得到文件a的FCB

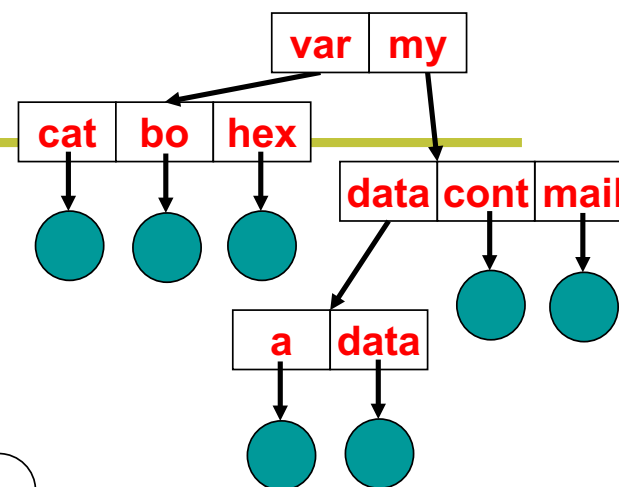
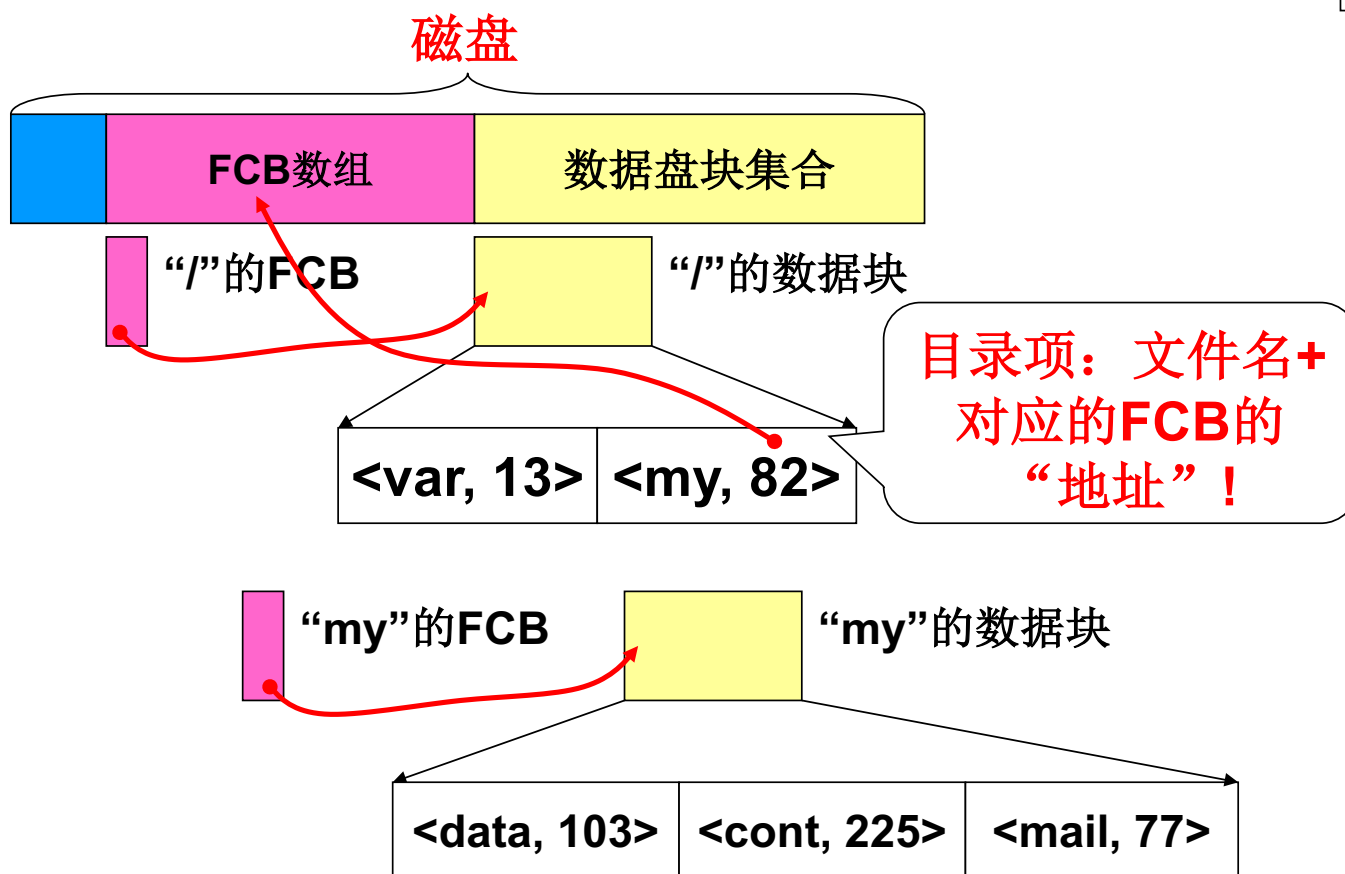
问题：那么目录中应该存什么？

- 存放目录下的所有文件的**FCB**吗？如果是，解析**my**要干什么？

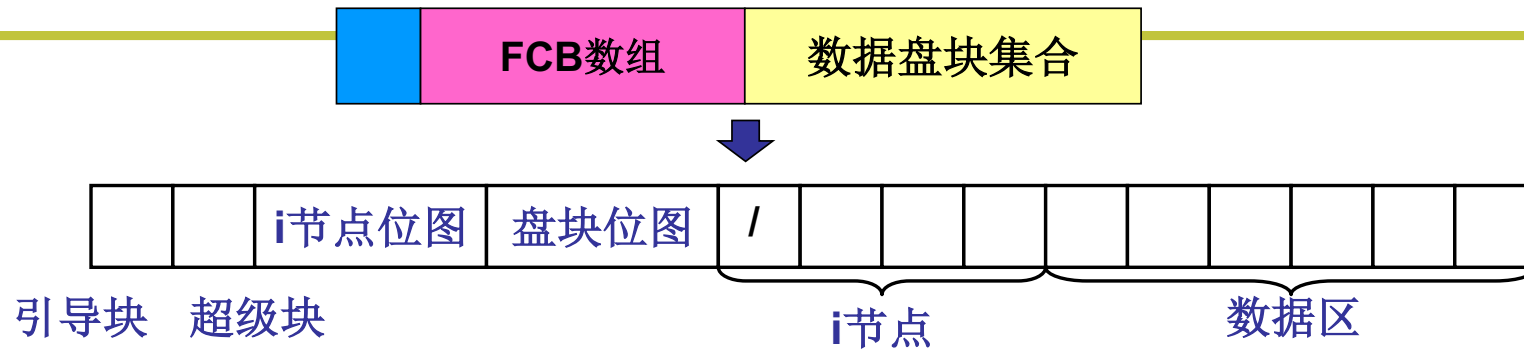
问题：有什么办法(目录存什么)让系统效率更高？



树状目录的完整实现



要使整个系统能自举，还需存一些信息

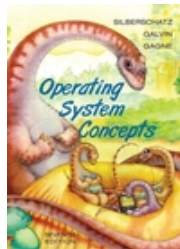


- **inode**位图: 哪些**inode**空闲，哪些被占用

- **盘块位图**: 哪些盘块是空闲的，硬盘大小不同这个位图的大小也不同

空闲位图(位向量)... 表示磁盘块2、3、4、5、
0011110011101 8、9、10、12空闲

- **超级块**: 记录两个位图有多大等信息



“完成全部映射下”的磁盘使用



用户

读test.c 202-212个字节

open(/xx/test.c)

目录解析找到/, 读入/内容找到xx, 再找到test.c的inode

read(fd)

根据找到的FCB和file中的202-212字节找盘块789

写入电梯队列

add_request(789)

磁盘中断

从队列中取出789, 算出cyl, head, sector

写磁盘控制器

outp(cyl, head, sector)

