



经典教材《计算机操作系统》**最新版**

第8章 文件管理

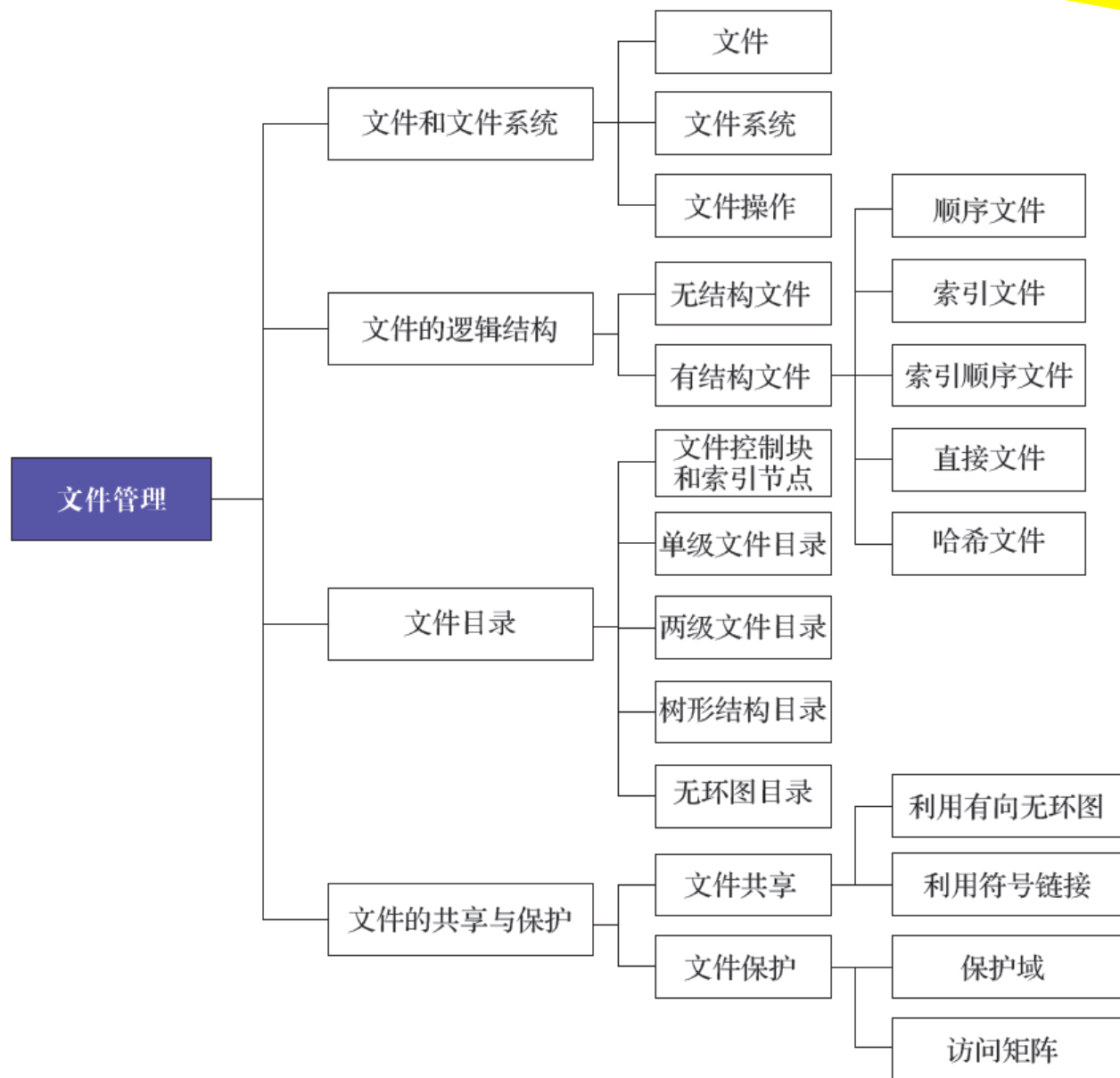
主讲教师：王申





第8章知识导图

第1章	操作系统引论
第2章	进程的描述与控制
第3章	处理机调度与死锁
第4章	进程同步
第5章	存储器管理
第6章	虚拟存储器
第7章	输入/输出系统
第8章	文件管理
第9章	磁盘存储器管理
第10章	多处理机操作系统
第11章	虚拟化和云计算
第12章	保护和安全





操作系统的主要功能

处理机管理	存储器管理	设备管理	文件管理	OS与用户接口
进程控制	内存分配	缓冲管理	文件存储空间管理	用户接口
进程同步	内存保护	设备分配	目录管理	程序接口
进程通信	地址映射	设备处理	文件读写管理	
调度	内存扩充		文件保护	



- 文件存储空间的管理

- 目录管理

- 按名存取

- 文件的读/写管理和保护

- 文件的读/写管理

- 文件保护



内容导航:



8.1 文件和文件系统



8.2 文件的逻辑结构



8.3 文件目录



8.4 文件共享



8.5 文件保护



8.6 Linux文件系统实例

第8章 文件管理



文件和文件系统



数据项、记录和文件



文件名和文件类型



文件系统的层次结构



文件操作



数据项

- 基本数据项：描述一个对象的某种属性，又称字段
- 组合数据项：由若干个基本数据项组成



记录

- 记录是一组相关数据项的集合，用于描述一个对象在某方面的属性。
- 关键字：唯一能标识一个记录的数据项
如：学号、学号+课程号

编号	姓名	性别	工资	
			基本绩效	加班
1000	张三	男	10000	3452
1001	李四	男	10000	2345
1002	王五	男	10000	4567
1003	赵六	男	10000	6789
1004	孙七	女	10000	1234
1005	周八	女	10000	2345

统计.xls

文件：具有文件名的一组相关元素的集合



有结构文件（记录）、无结构文件（字节流）

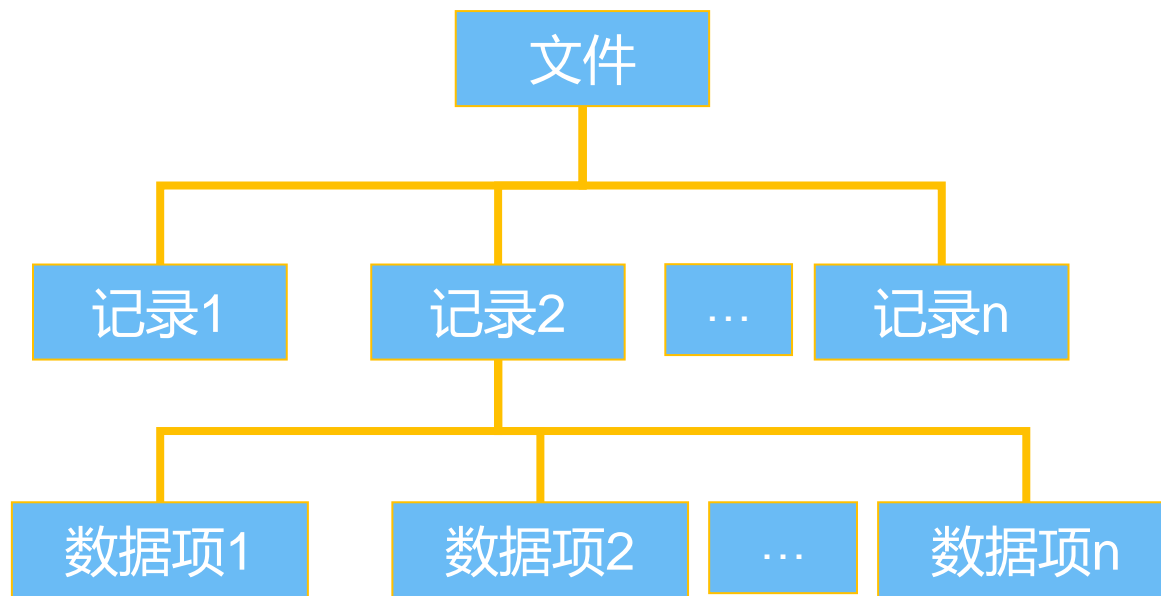


在文件系统中文件是最大的数据单位，描述了一个对象集



文件四个主要属性

- 文件类型
- 文件长度
- 文件的物理位置
- 文件的建立时间





文件名（不同OS对文件名规定不同，NTFS最多支持255个字符）



扩展名：文件名后面的若干个附加字符（.xls）



文件类型

- 按用途分类：系统文件/用户文件/库文件
- 按文件中数据的形式分类：源文件/目标文件/可执行文件
- 按存取控制属性分类：只读文件/读写文件/可执行文件
- 按组织形式和处理方式分类：普通文件/目录文件/特殊文件（I/O设备）



对象及其属性

➤ 对象：文件、目录、磁盘存储空间



对对象操纵和管理的软件集合

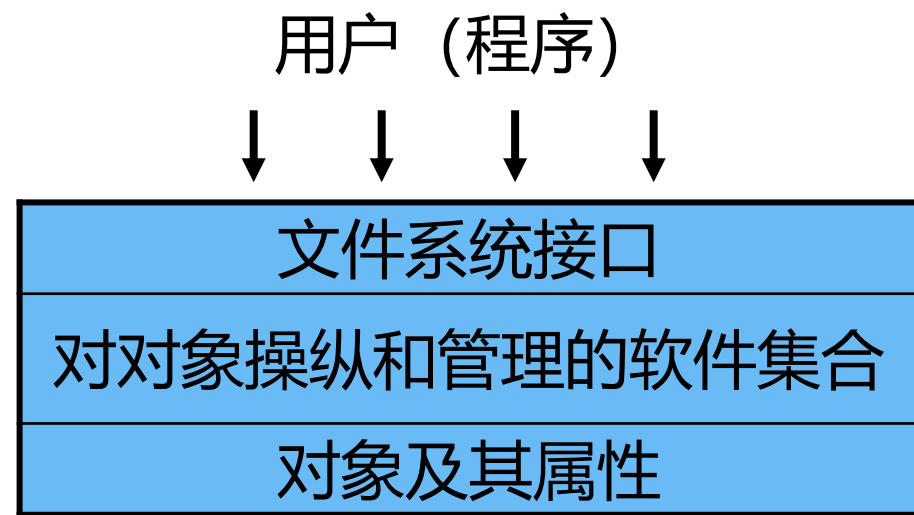
➤ 文件管理系统的核心部分



文件系统的接口

➤ 命令接口

➤ 程序接口



文件系统模型



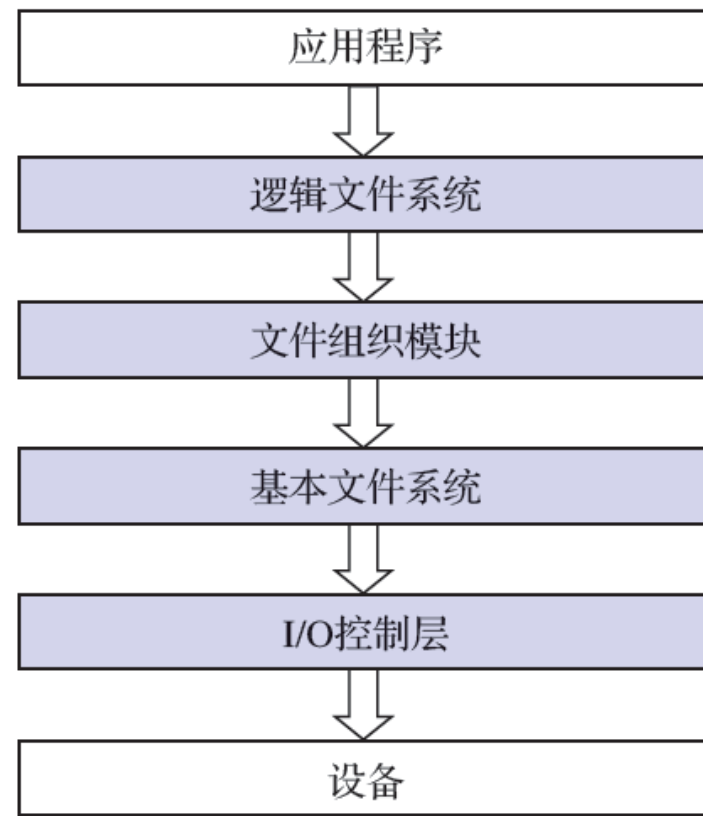
文件系统的功能

- 对文件存储空间的管理
- 对文件目录的管理
- 用于将文件的逻辑地址转换为物理地址的机制
- 对文件读和写的管理
- 对文件的共享和保护



与文件系统有关的软件

- **I/O控制层**：文件系统最底层，设备驱动程序层
- **基本文件系统层**：处理内存和磁盘之间数据块的交换
- **基本I/O管理程序**：完成与磁盘I/O有关的事务
- **逻辑文件系统**：处理与记录和文件相关的操作





最基本的文件操作：

- 创建文件、删除文件、读文件、写文件。
- 设置文件的读/写位置。



文件的“打开”和“关闭”操作：

- 打开open：系统将文件从外存拷贝到内存打开文件表的一个表目中。
- 关闭close：把文件从打开文件表的表目中删除。









其他文件操作：

- 重命名、改文件拥有者、查询文件状态等。
- 创建/删除目录。



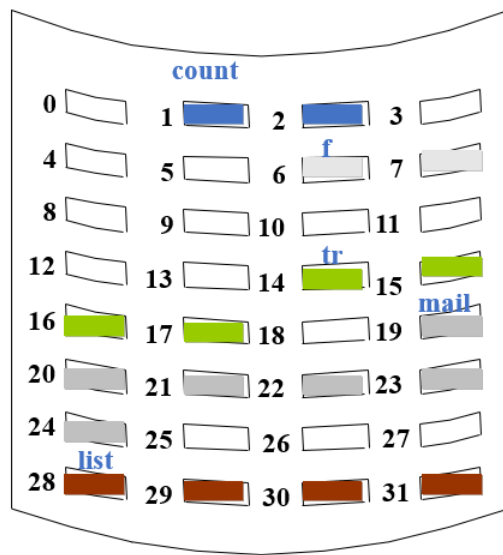
内容导航:

-  8.1 文件和文件系统
-  **8.2 文件的逻辑结构**
-  8.3 文件目录
-  8.4 文件共享
-  8.5 文件保护
-  8.6 Linux文件系统实例

第8章 文件管理

文件目录

文件名	物理地址	
文件名	起始块	块数
count	1	2
tr	14	4
mail	19	6
list	28	4
f	6	2



文件结构

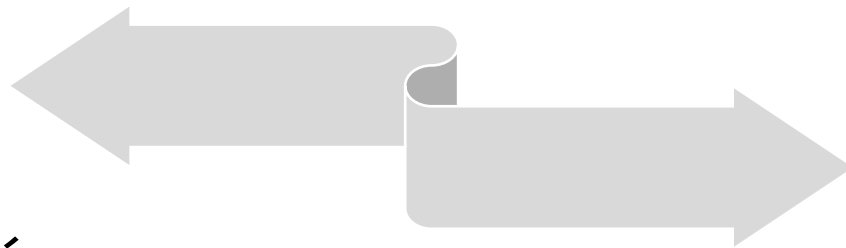
- **文件的逻辑结构**: 从用户观点所观察的文件组织形式
- **文件的物理结构**: 文件的存储结构, 指系统将文件存储在外存上所形成的一种存储组织形式
 - 不仅与存储介质的存储性能有关
 - 也与所采用的外存分配方式有关

文件的逻辑结构和物理结构, 都影响文件的检索速度



按文件是否有结构分类

- 有结构文件（记录式文件）
 - ◆ 定长记录
 - ◆ 变长记录
 - ◆ 例子：数据库
- 无结构文件（流式文件）
 - ◆ 源程序、可执行文件、库函数

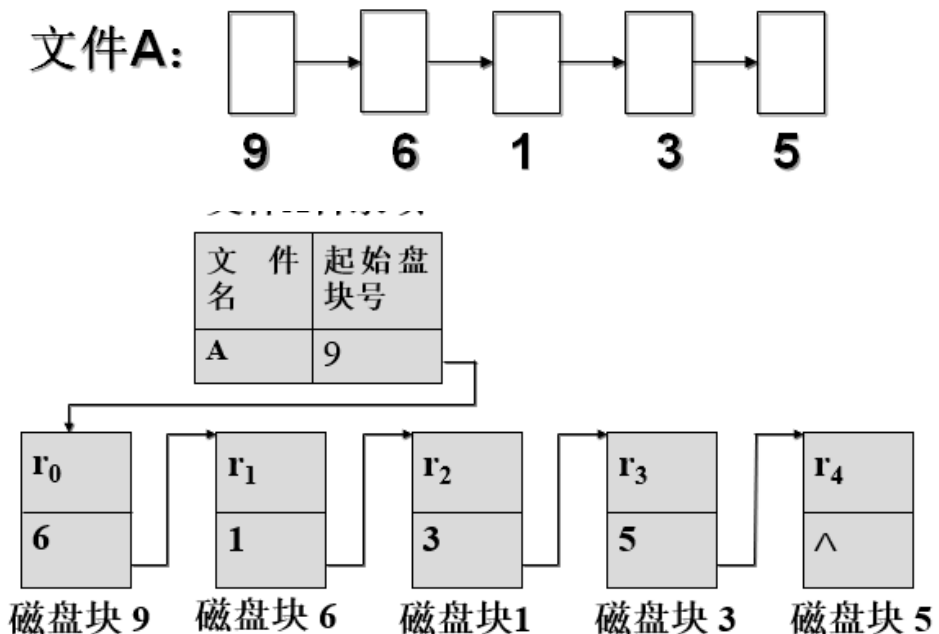


有结构文件按文件的记录方式分类

- 顺序文件
- 索引文件
- 索引顺序文件

串结构：按存入时间的先后排序，
记录间的顺序与关键字无关

- 检索比较费时
- 文件易丢失



顺序结构：指定一个字段为关键字，
记录间的顺序按关键字排序

- 检索时可利用有效的查找算法，折半查找法、插值查找法、跳步查找法等

编号	姓名	性别	工资	
			基本绩效	加班
1000	张三	男	10000	3452
1001	李四	男	10000	2345
1002	王五	男	10000	4567
1003	赵六	男	10000	6789
1004	孙七	女	10000	1234
1005	周八	女	10000	2345



优点：

- 有利于大批记录读写
- 存取效率最高
- 可存储在顺序存储设备（磁带）上

缺点

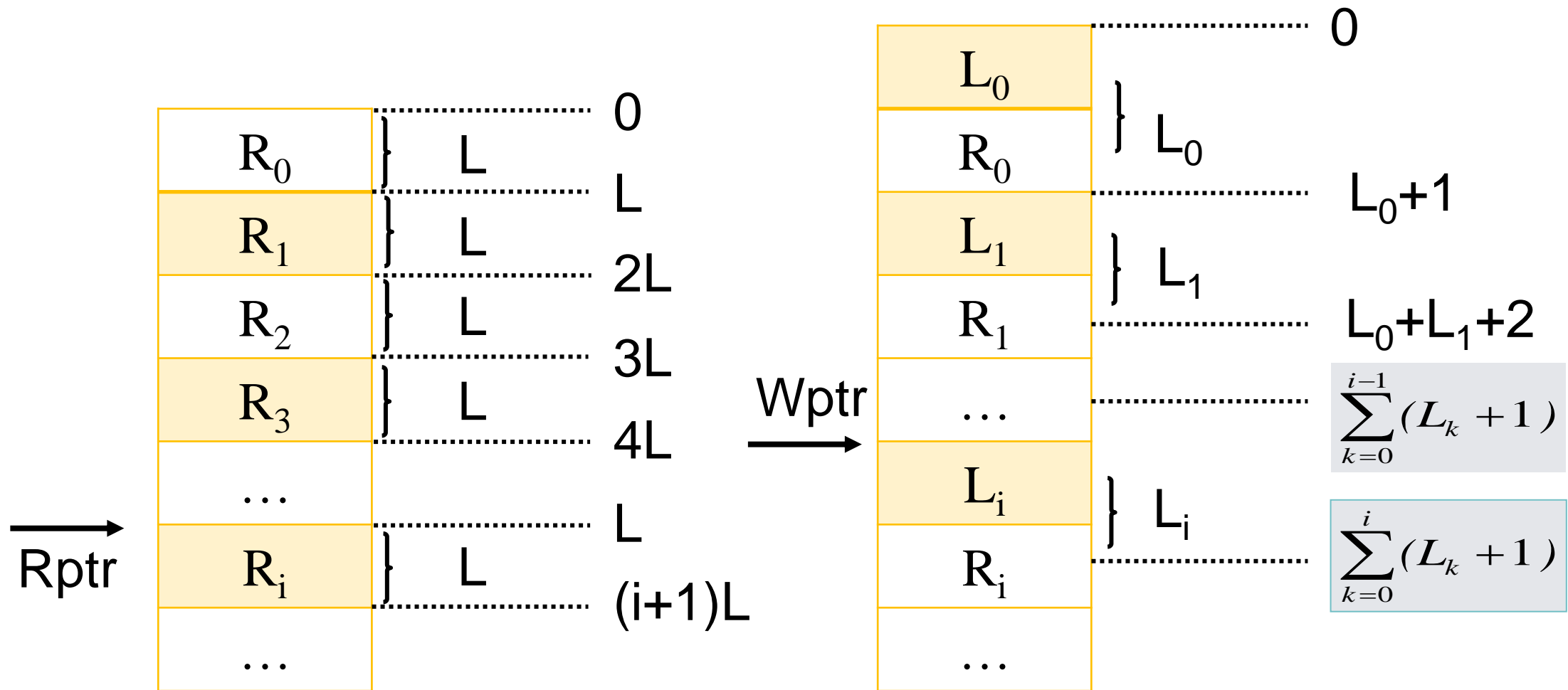
- 查找或修改单个记录：差
- 增加或删除一个记录：难

隐式寻址方式（顺序访问）

- 定长记录: $Rptr=Rptr+L$ / $Wptr=Wptr+L$
- 变长记录: $Rptr=Rptr+L_i$ / $Wptr=Wptr+L_i$

显式寻址方式（直接/随机访问）

- 通过文件中记录的位置
 - 定长记录
 - 变长记录
- 利用关键字



(a) 定长记录文件

(b) 变长记录文件



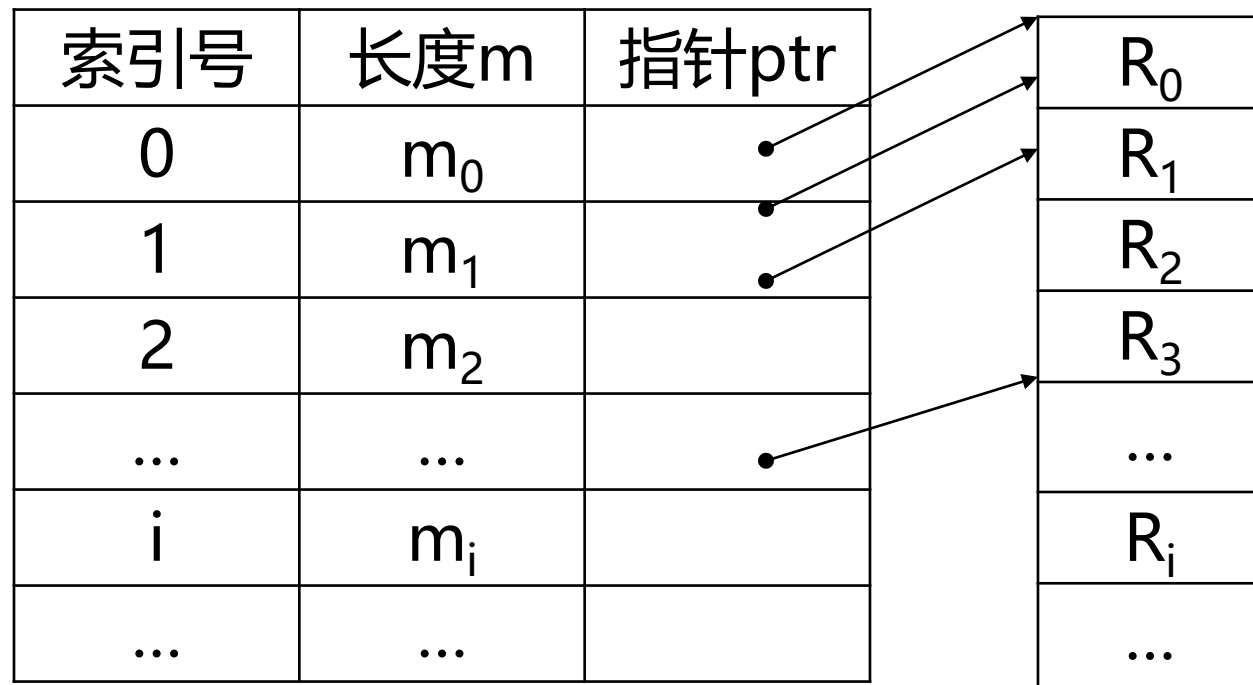
按关键字建立索引

- 为变长记录文件建立一张索引表
- 索引表按关键字排序
- 实现直接存取



具有多个索引表的索引文件

- 为顺序文件建立多个索引表
- 为每一个可能成为检索条件的域配置一张索引表（索引号/书名/姓名/时间）



索引表

逻辑文件

索引文件将需顺序查找的文件改造为可随机查找的文件

- 🏠 有效克服了变长记录文件不便于直接存取的缺点。
- 📁 保留了顺序文件的关键特征：记录按关键字的顺序组织。
- 🔒 新增2个新特征：
 - 引入文件索引表：实现随机访问。
 - 增加溢出文件：记录新增、删除和修改的记录。

索引顺序文件是顺序文件和索引文件相结合的产物

一组记录对应一个索引项

键	逻辑地址	姓名	其他属性
An Qi		An Qi	
Bao Rong		An Kang	
Dong Fan			
Chen Lin	...	Bao Rong	
	

10000条记录：顺序文件平均需要5000次查找

对于索引顺序而言，先将10000条分为100组，平均查找为50次，查到分组后，在有100条记录的分组里查找，平均查找为50次。

故， $50+50 = 100$ 次。

当记录非常多时，查找效率仍然不高，所以继续利用索引的思想，建立两级索引

键	逻辑地址
A	
B	
C	
	...

100条记录

键	逻辑地址
An	
Angle	
Bao	
	...

10000条记录

姓名	其他属性
An Qi	
An Kang	
...	
Angle Dong	
...	
Bao Rong	
...	...

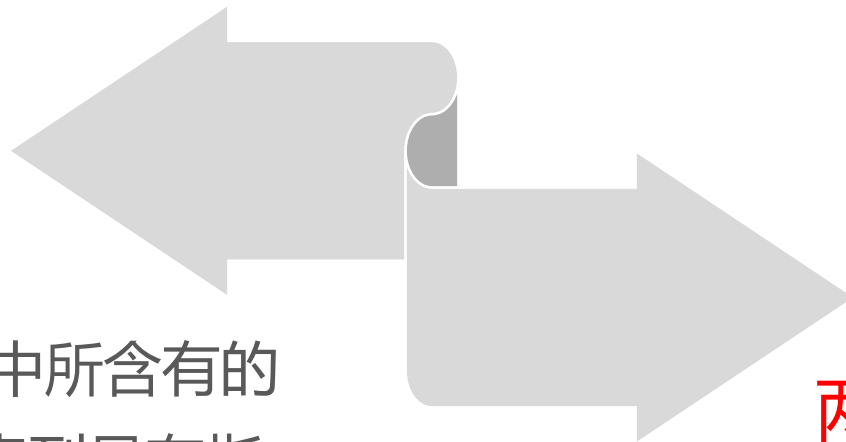
注意：N个记录，K级索引，最优分组每组 $N^{1/(k+1)}$

故， $50+50+50 = 150$ 次



一级索引文件

- 如果在一个顺序文件中所含有的记录数为 N ，则为检索到具有指定关键字的记录，平均须查找 $N/2$ 个记录；
- 对于索引顺序文件，则为能检索到具有指定关键字的记录，平均只要查找 \sqrt{N} 个记录数，因而其检索效率比顺序文件约提高 $\sqrt{N}/2$ 倍



两级索引文件

- 为了进一步提高检索效率，可以为顺序文件建立多级索引，即为索引文件再建立一张索引表，从而形成两级索引表
- 两级索引表 $(3/2)(N)^{(1/3)}$



直接文件：

- 关键字本身决定了记录的物理地址。
- 键值转换：关键字到记录物理地址的转换。





哈希文件：

- 利用Hash函数将记录键值转换为记录的地址。
- 应用最广泛的一种直接文件



内容导航:

-  8.1 文件和文件系统
-  8.2 文件的逻辑结构
-  **8.3 文件目录**
-  8.4 文件共享
-  8.5 文件保护
-  8.6 Linux文件系统实例

第8章 文件管理

目录实现了对文件的有效管理

目录是一种数据结构，
用于标识文件及其物理地址

对目录管理的要求

- 实现“按名存取”
- 提高对目录的检索速度
- 文件共享
- 允许文件重名



文件控制块 (FCB)：管理和控制文件的数据结构，与文件——对应



文件目录：文件控制块的集合，即一个FCB就是一个文件目录项

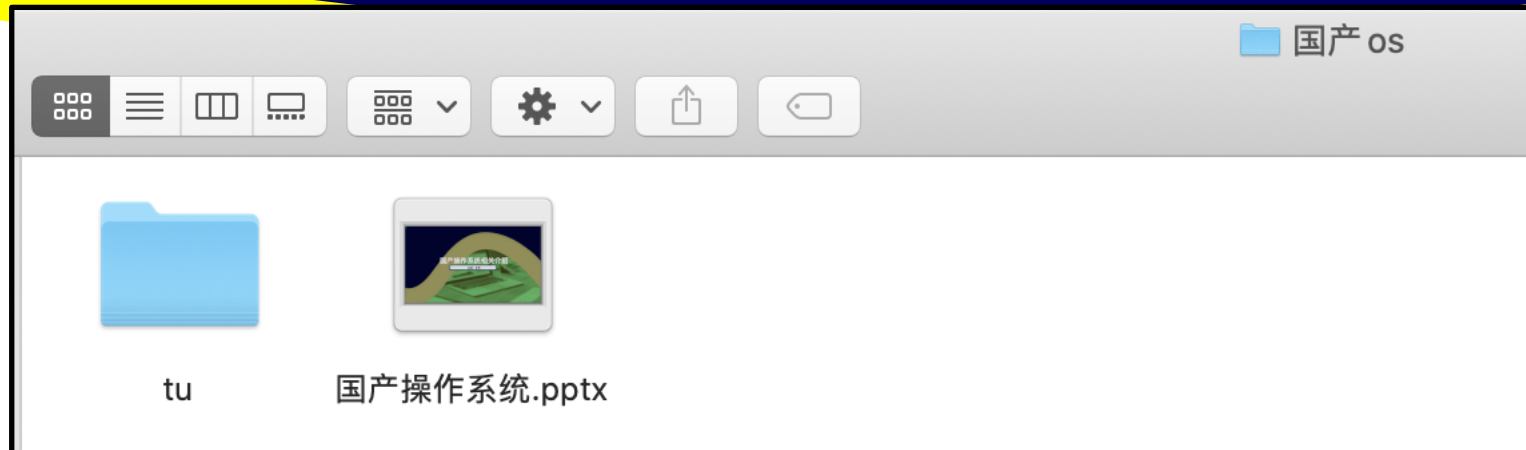


目录文件：文件目录也被看做是一个文件



文件控制块包含信息

- 基本信息类：文件名、文件物理位置、文件逻辑结构、文件物理结构
- 存取控制信息类：文件主的存取权限、核准用户的存取权限以及一般用户的存取权限
- 使用信息类：文件的建立日期和时间、文件上一次修改的日期和时间及当前使用信息



不是特别严谨的例子

```
→ /Users/s[redacted]/Documents/北邮/网络空间安全学院/教学/操作系统 - [redacted]/PPT/国产 os: ll
total 4216
drwxr-xr-x@ 223 s[redacted] staff  7.0K Aug 29  2020 tu
-rw-r--r--@   1 s[redacted] staff  2.1M Apr 25 11:26 国产操作系统.pptx
```

- 一条记录就一个文件控制块（FCB）；FCB实现了文件名和文件之间的映射，用户可以“按名存取”
- 多条记录的集合就是文件目录
- 文件目录存储起来后就是目录文件



目录结构的组织，是设计好文件系统的重要环节

- 关系到文件系统的存取速度
- 也关系到文件的共享性（不同的文件名访问同一个文件）和安全性

目录结构和文件都驻留在磁盘上。

OS 所有用户的文件都在同一目录中(早期操作系统)

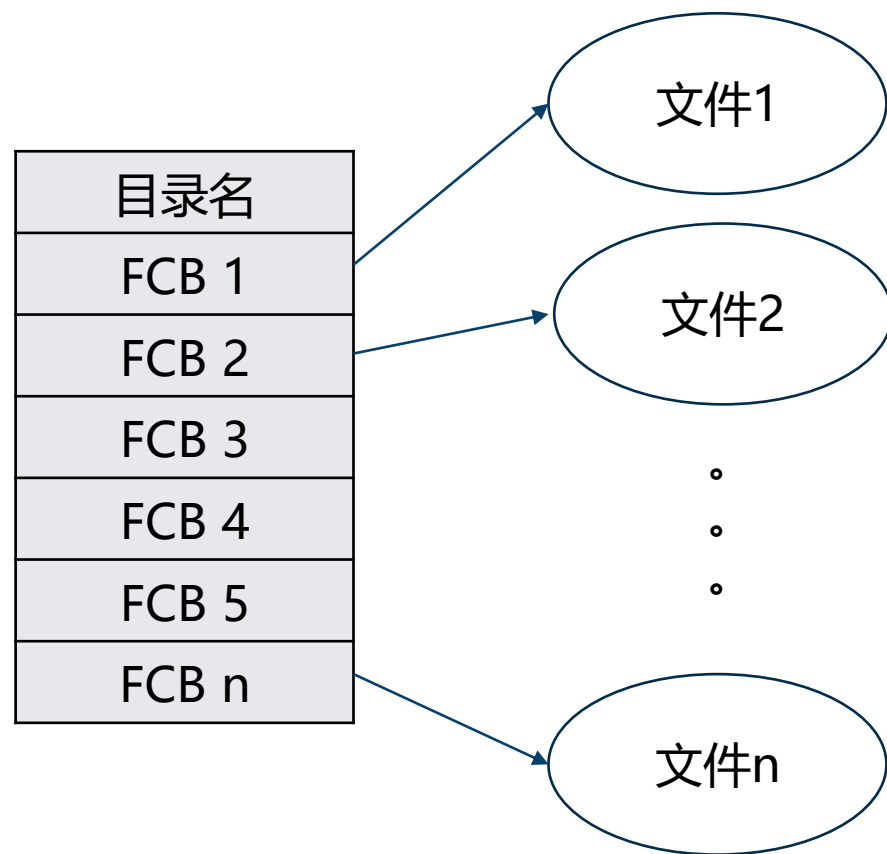
OS 最简单的文件目录

OS 目录项:

- 文件名、扩展名、文件长度、文件类型、物理地址、文件说明以及其他文件属性

OS 缺点:

- 查找速度慢
- 不允许重名
- 不便于实现文件共享

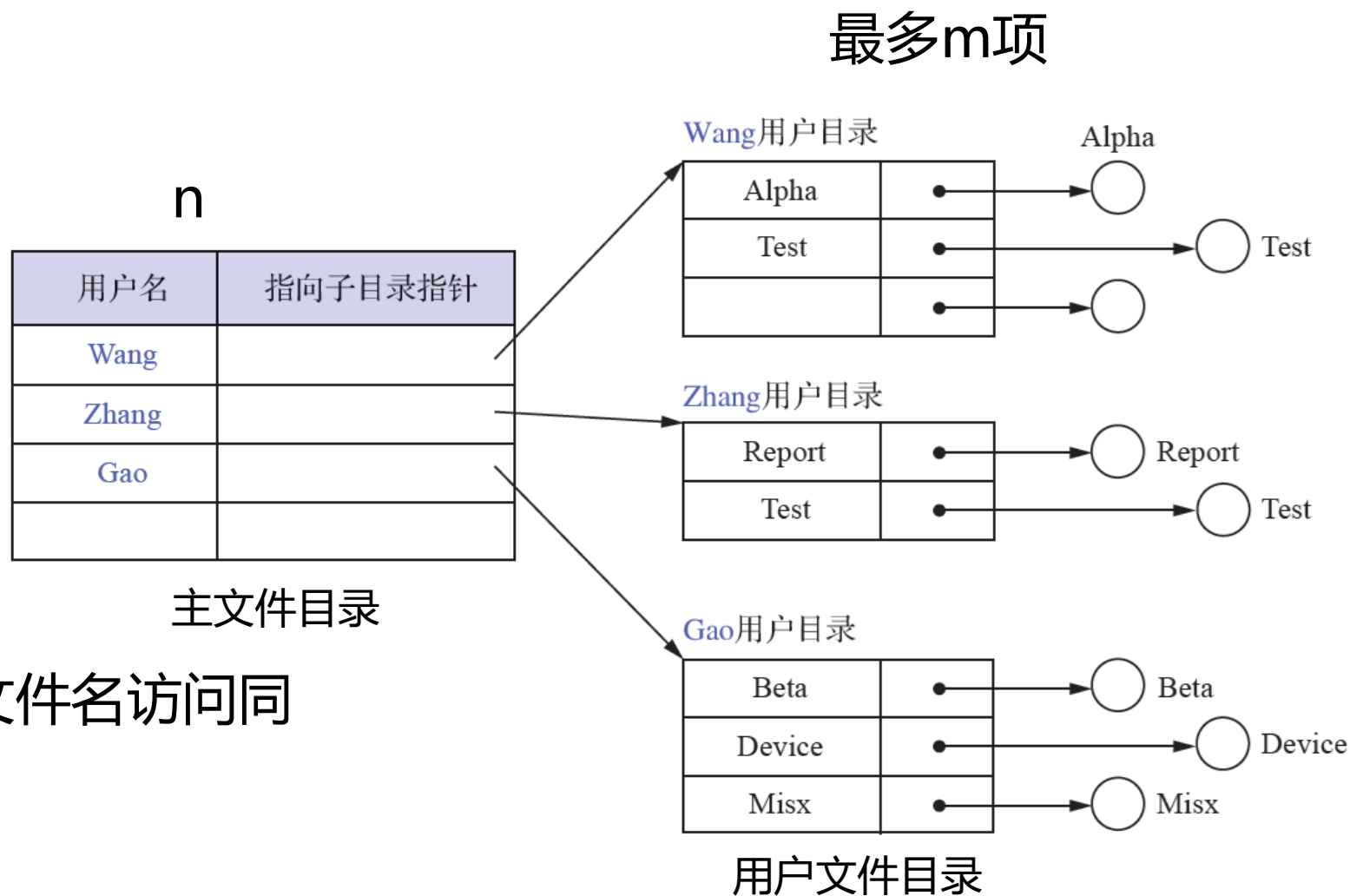


整个系统一张表

每个用户有自己的目录结构

特点:

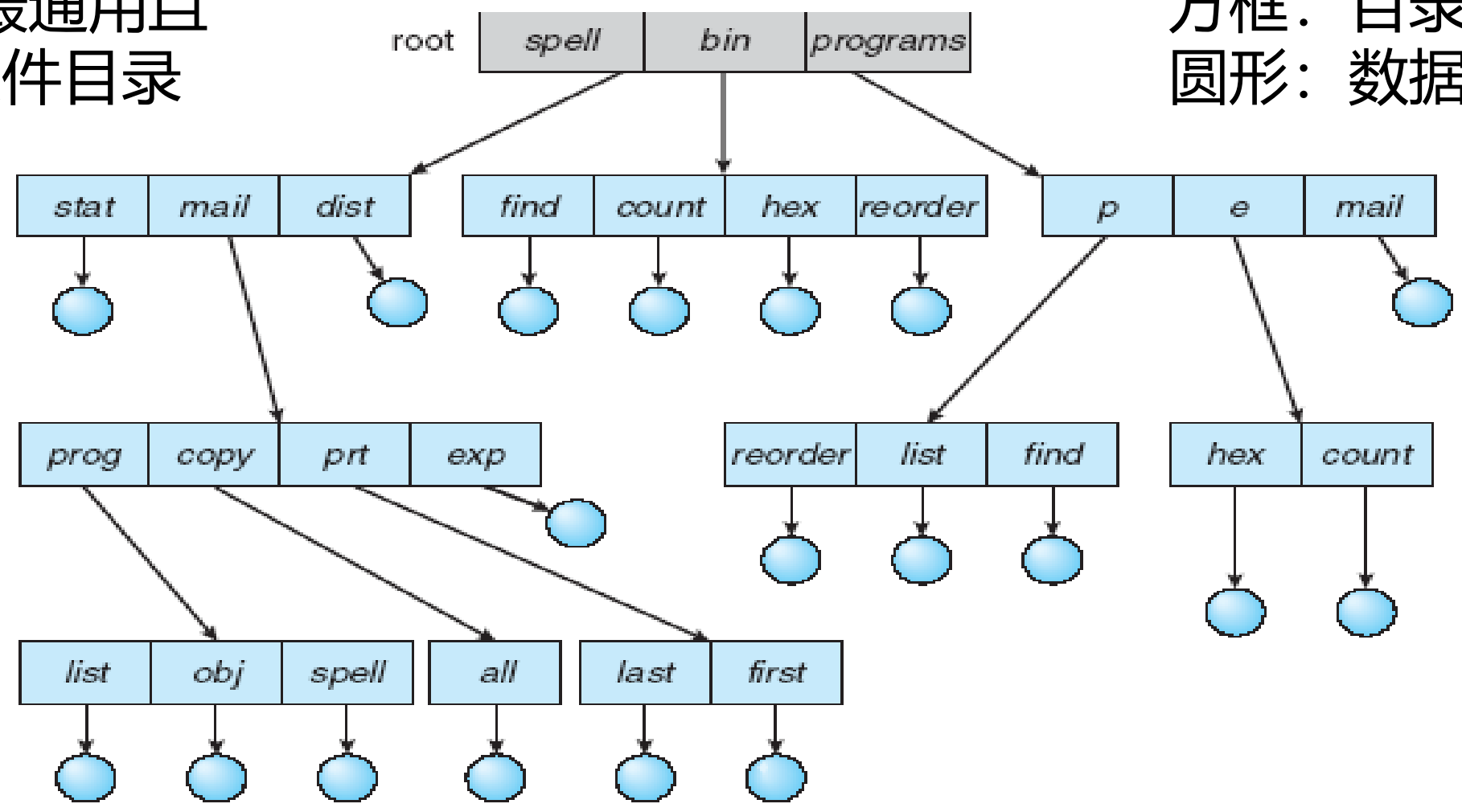
- 提高了检索目录的速度 $(nxm)/(n+m)$
- 不同用户可以有相同的文件名
- 不同用户可以使用不同文件名访问同一个共享文件
- 不便于共享文件



用户不能对自己的文件进行分类!!!

现代OS最通用且实用的文件目录

方框：目录文件
圆形：数据文件



路径名 – 从根目录到任何数据文件的通路

- 绝对路径名：从根目录开始的路径名
- 相对路径名：从当前目录开始的路径名（由于对每一层目录而言，需要将其调入内存后才可以对下一层目录进行读取；但是外存与内存之间的访问效率很低，**所以，提出了相对路径，即从当前目录进行下级目录的查询，减少了I/O读写次数，提高了文件访问速率。**）



主目录 – 根目录



当前目录 - 工作目录

- `cd /spell/mail/prog`
- `type list`

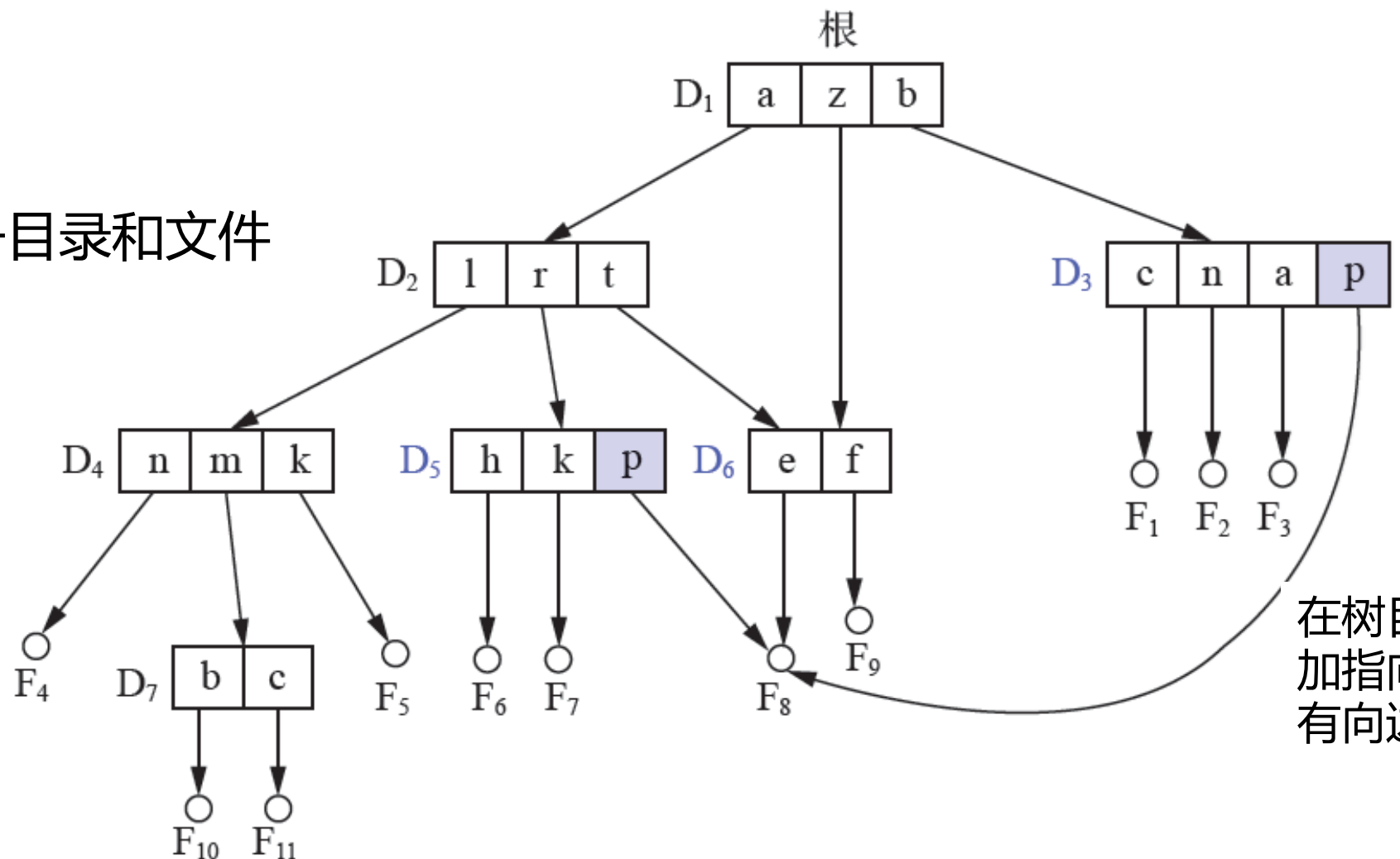
明显提高对目录
的检索速度



更加有效地进行
文件管理和保护

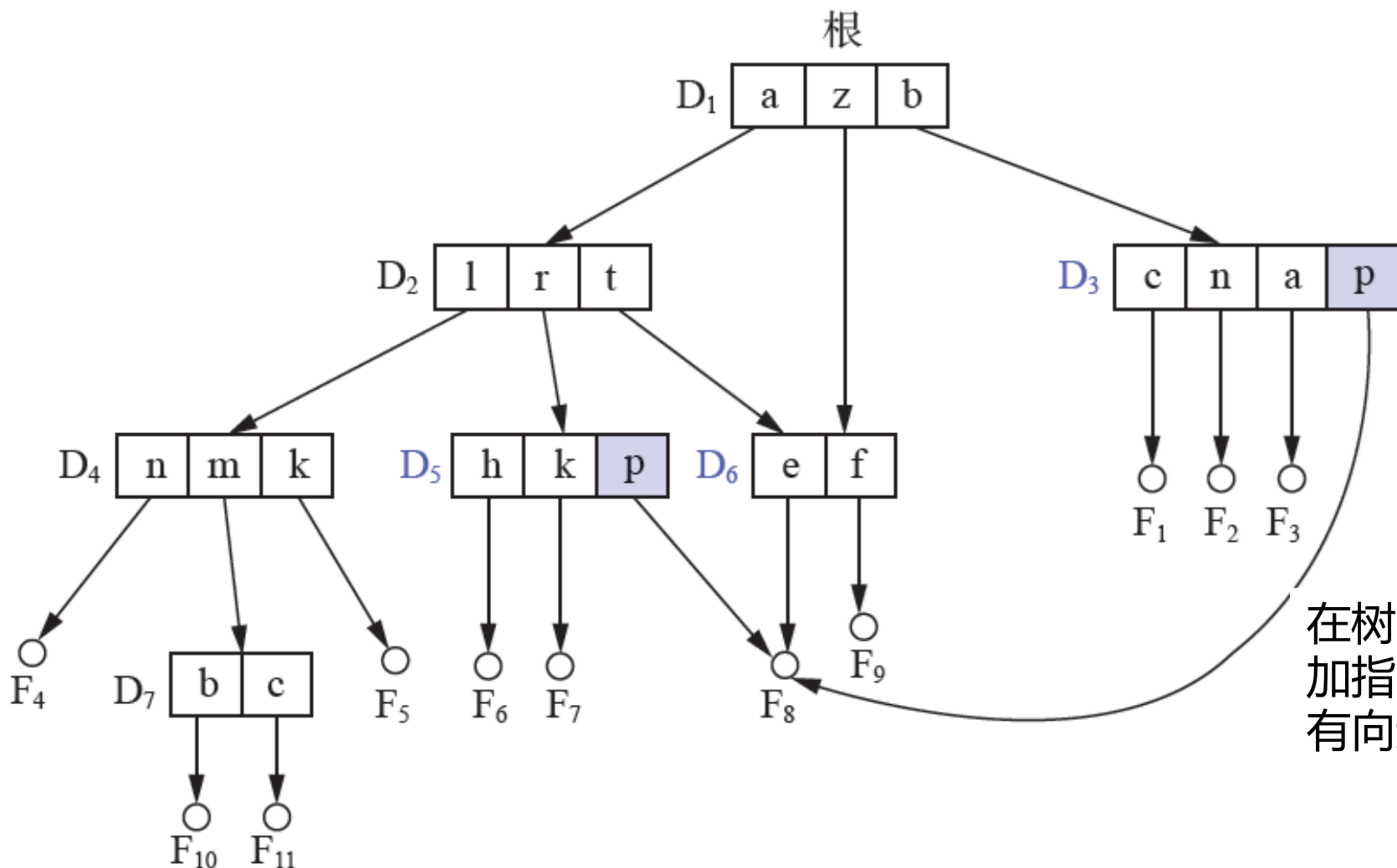
树形目录结构可以：对文件进行分类，更有效地进行文件的管理和保护，且层次结构清晰。**但是，树形结构不便于实现文件的共享。**为此，提出了“无环图目录结构”。

有共享的子目录和文件

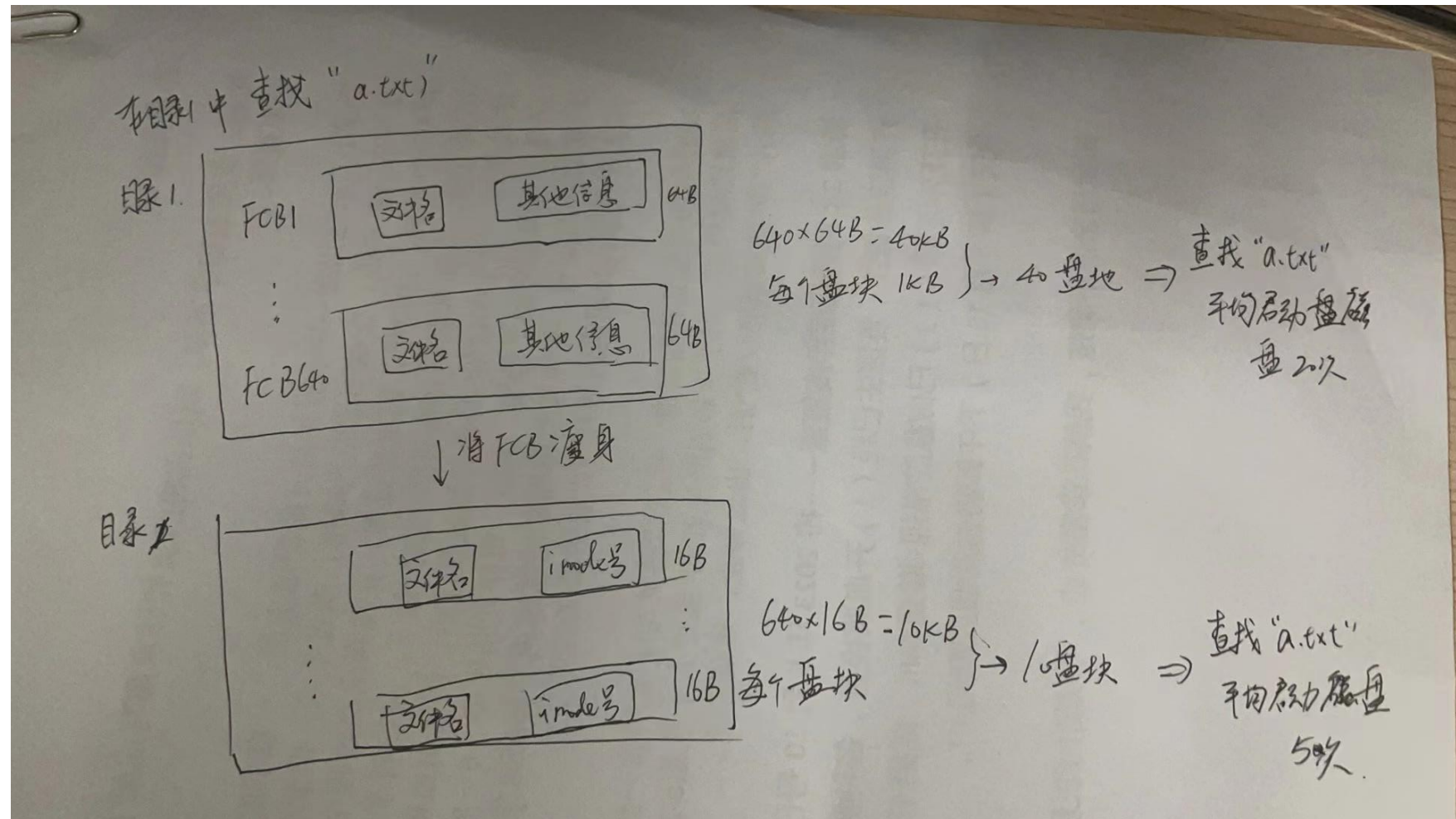


在树目录基础上，增加指向同一个节点的有向边有向无环图。

- 1、无环图目录：可以用不同文件名指向同一个文件。
- 2、由于一个文件有多个指向，所以，**当删除文件时不能直接将文件本身删除**。为此，设置了一个共享计数器，某个用户删除文件时，共享计数器减一，删除该用户的FCB，而不会对文件本身操作。



注意点：FCB是和文件一一对应，但是不是文件本身，只是保存了对文件信息的描述



- OS 设目录文件所占盘块数为 N ，则查找一个目录项平均需要调入盘块 $(N+1)/2$ 次
- OS 假如一个FCB为64B，盘块大小为1KB，则每个盘块中只能存放16个FCB；若一个文件目录中共有640个FCB ($640 \times 64\text{B} = 40\text{KB}$)，需占用40个盘块 ($40\text{KB} / 1\text{KB} = 40$)，故平均查找一个文件需启动磁盘20次
- OS 引入索引节点后，一个目录仅占16B，其中14B是文件名，2B为i节点指针。1KB的盘块中可容纳64个目录项；若一个文件目录中共有640个FCB ($640 \times 16\text{B} = 10\text{KB}$)，需占用10个盘块 ($10\text{KB} / 1\text{KB} = 10$)，故平均查找一个文件需启动磁盘5次
- OS 可使平均启动磁盘次数减少到原来的1/4



索引结点的引入：减少磁盘访问次数

- 把文件名与文件描述信息分开，文件描述信息用索引结点 (iNode) 保存，简称为i节点 (UNIX)



磁盘索引结点：存放在磁盘上的索引结点

- ①文件拥有者标识符，②文件类型，③文件存取权限，④文件物理地址，⑤文件长度，⑥文件连接计数，⑦文件存取时间



内存索引结点：存放在内存上的索引结点

- 磁盘索引结点+ ①索引节点编号，②状态，③访问计数，④文件所属文件系统的逻辑设备号，⑤链接指针

```
→ /Users/chenming/Documents/北邮/网络空间安全学院/教学/操作系统-王中/PPT/国产os: ls -li  
94099605 tu 94078280 国产操作系统.pptx
```

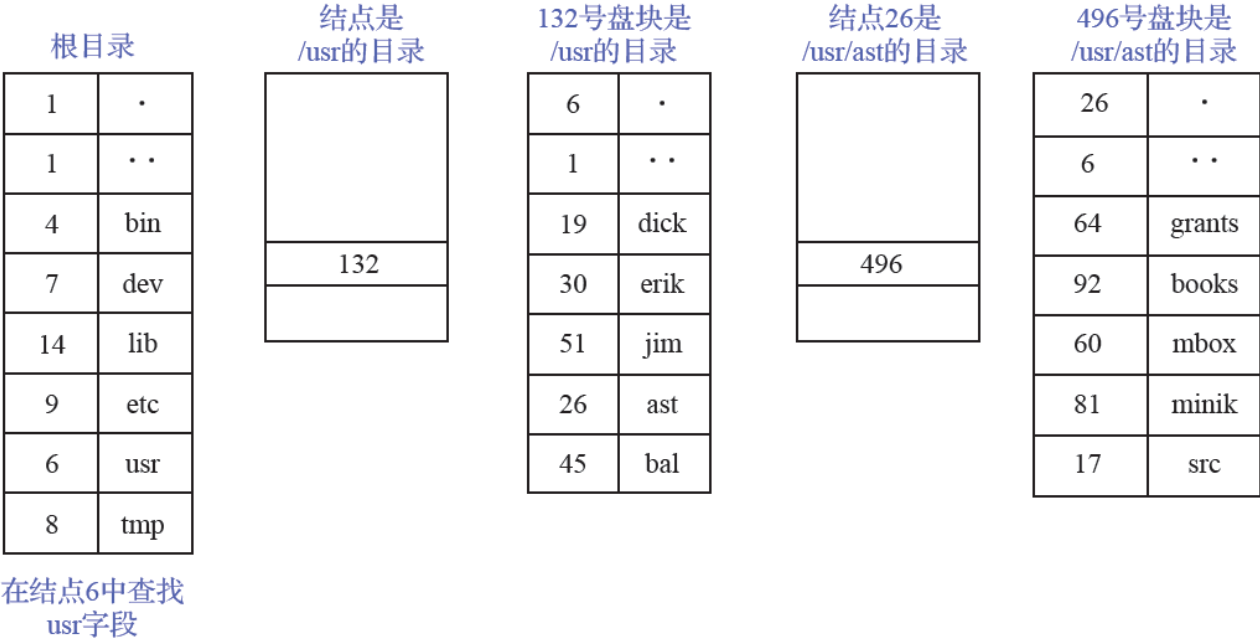
目录操作





线性检索法（顺序检索法）：

- 查找/usr/ast/mbox的过程。









Hash方法：

- 系统利用用户提供的文件名，将之变换为文件目录的索引值，再利用该索引值到目录中去查找。
- 模式匹配（通配符* ? ）时，无法使用Hash方法。
- 处理冲突使用有效规则。



内容导航:

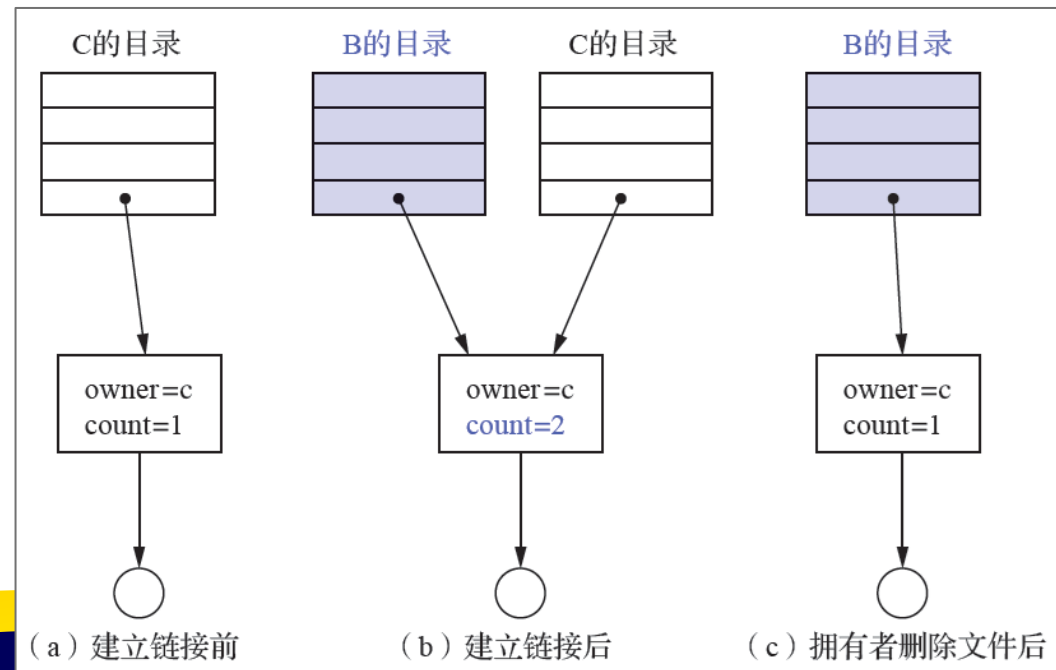
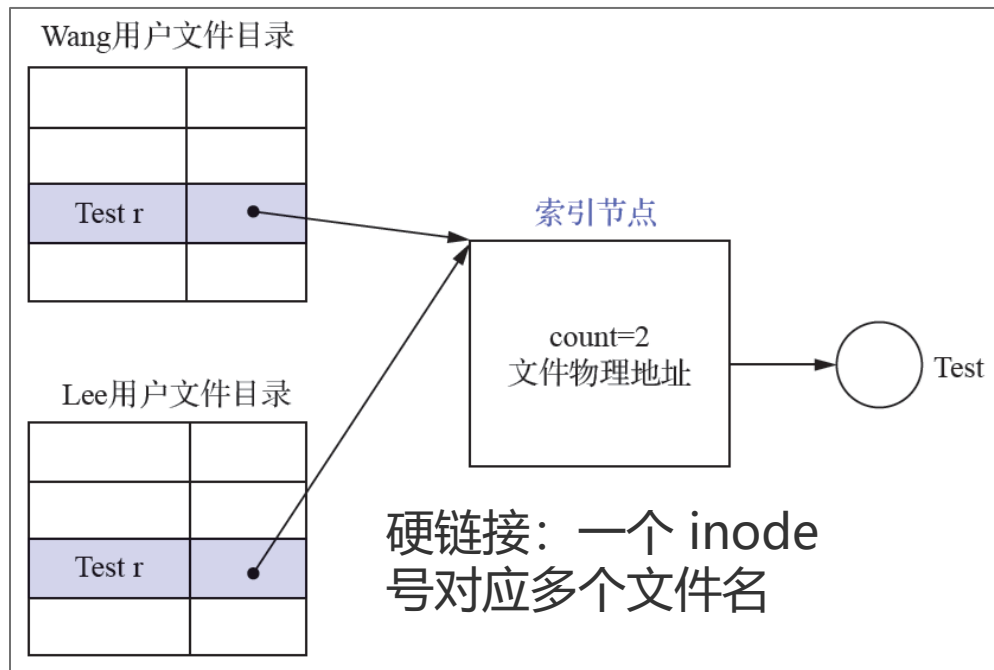
-  8.1 文件和文件系统
-  8.2 文件的逻辑结构
-  8.3 文件目录
-  **8.4 文件共享**
-  8.5 文件保护
-  8.6 Linux文件系统实例

第8章 文件管理

文件共享：多个用户（进程）共享同一份文件（注意：inode 号是文件的唯一标识而非文件名。文件名仅是为了方便人们的记忆和使用）

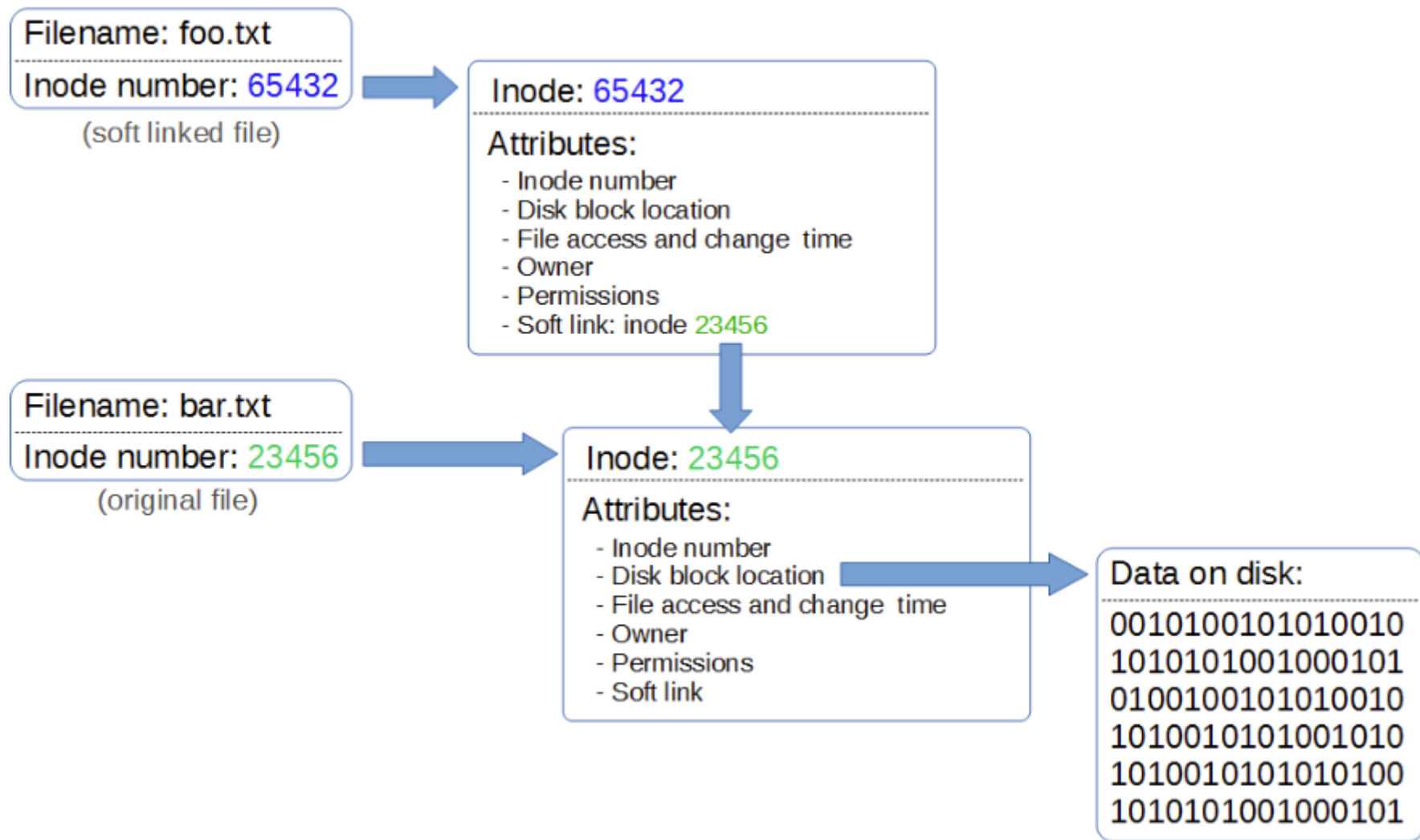
基于有向无环图实现共享

- 有向无环图：树形目录中，允许一个文件有多个父目录
- 利用索引结点实现共享，设置链接计数count





文件共享—利用符号链接实现共享（软链接）





基本思想：允许一个文件有多个父目录，但其中仅有一个作为主父目录，其它的都是通过符号链接方式与之相链接。

- Linux系统中的创建符号链接命令：ln -s 源文件 目标文件
- 新文件内容中只包含被链接文件的路径名。



优点：

- 不会发生在文件拥有者删除一个共享文件后留下一个悬空指针的情况。

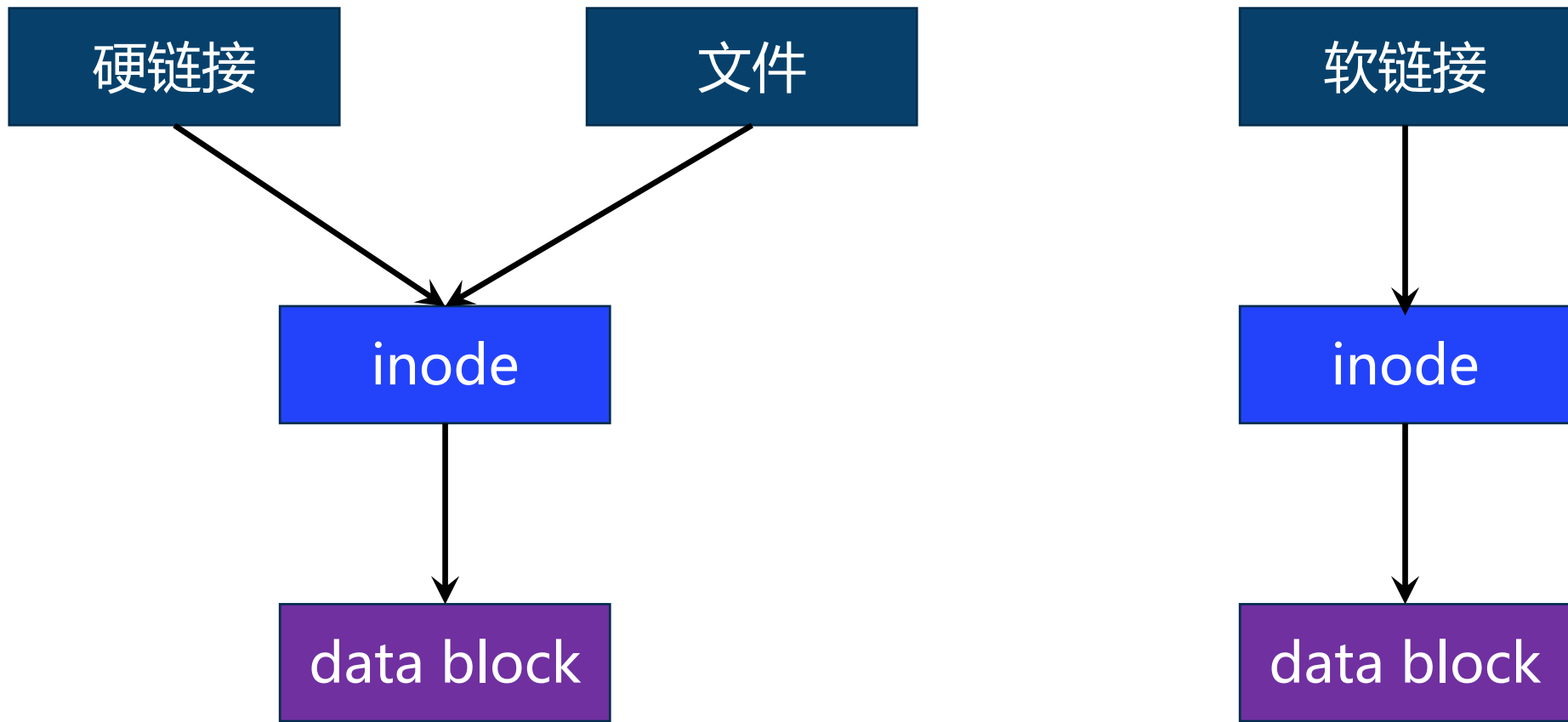


缺点：

- 每次访问共享文件时都可能要多次地读盘，这使每次访问文件的开销甚大，且增加了启动磁盘的频率。



软链接 vs 硬链接








软链接 vs 硬链接

软链接就是一个普通文件，只是数据块内容特殊。软链接有着自己的 inode 号以及用户数据块。因此软链接的创建与使用没有类似硬链接的诸多限制：

- 软链接有自己的文件属性及权限等；
- 可对不存在的文件或目录创建软链接；
- 软链接可交叉文件系统；
- 软链接可对文件或目录创建；
- 创建软链接时，链接计数不会增加；
- 删除软链接并不影响被指向的文件，但若被指向的原文件被删除，则相关软连接被称为死链接（若被指向路径文件被重新创建，死链接可恢复为正常的软链接）



内容导航:

-  8.1 文件和文件系统
-  8.2 文件的逻辑结构
-  8.3 文件目录
-  8.4 文件共享
-  **8.5 文件保护**
-  8.6 Linux文件系统实例

第8章 文件管理

影响文件安全性的主要因素

- 人为因素（有意无意的破坏）
- 系统因素（磁盘）
- 自然因素（硬盘数据随着时间会丢失）

采取的措施

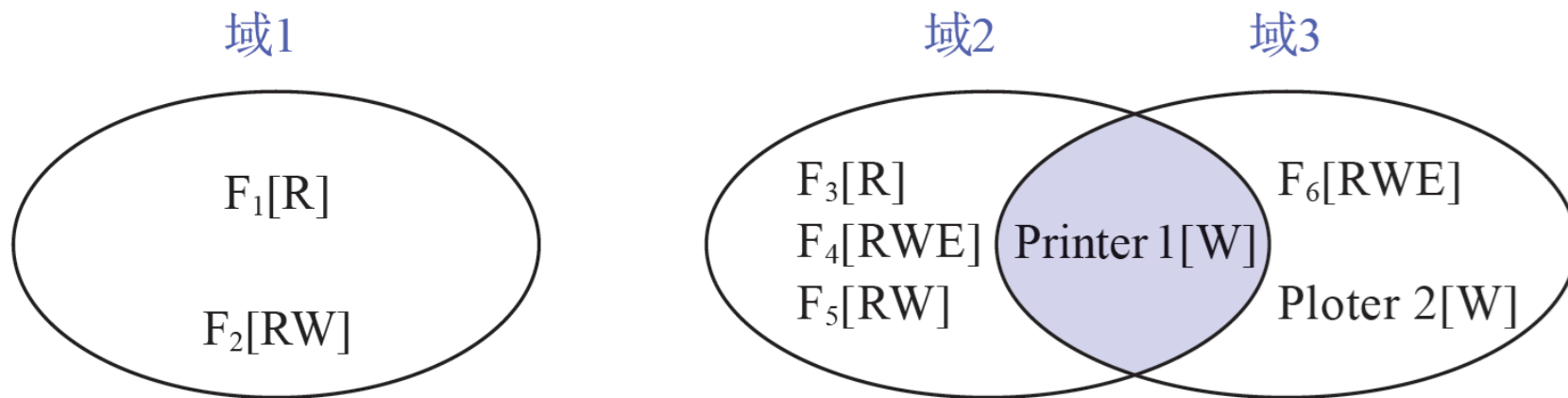
- 存取控制机制
- 系统容错技术
- 建立后备系统

访问权

- 一个进程能对某对象执行操作的权利，称为访问权
- 表示：(对象名，权集)，如(F_1 , {R/W})

保护域

- 进程对一组对象访问权的集合，称为保护域
- 如进程和域一一对应，则称为静态联系方式，该域称为静态域
- 如进程和域是一对多关系，则称为动态联系方式



域2和域3的进程都可使用打印机

- 用矩阵来描述系统的访问控制
- 行代表域；列代表对象；每一项为访问权

域 \ 对象	F ₁	F ₂	F ₃	F ₄	F ₅	F ₆	打印机1	绘图仪2
域D ₁	R	R, W						
域D ₂			R	R, W, E	R, W		W	
域D ₃						R, W, E	W	W

- 能够将进程从一个保护域切换为另一个域
- 如，由于域 D_1 和 D_2 所对应的项目中有一个S（即切换权），则允许在域 D_1 中的进程切换到域 D_2 中。

域 \ 对象	F ₁	F ₂	F ₃	F ₄	F ₅	F ₆	打印机1	绘图仪2	域D ₁	域D ₂	域D ₃
域D ₁	R	R, W								S	
域D ₂			R	R, W, E	R, W		W				S
域D ₃						R, W, E	W	W			

■ 复制权

- 利用复制权将某个域所拥有的访问权 $\text{access}(i, j)$ 扩展到同一列的其他域中

域 \ 对象	F ₁	F ₂	F ₃
域D ₁	E		W'
域D ₂	E	R'	E
域D ₃	E		

(a) 增加复制权前的访问矩阵

域 \ 对象	F ₁	F ₂	F ₃
域D ₁	E		W'
域D ₂	E	R'	E
域D ₃	E	R	W

(b) 增加复制权后的访问矩阵

■ 所有权

- 利用所有权 (owner right) 能增加某种访问权, 或者能删除某种访问权
- 图中的 “O”表示所有权

域 \ 对象	F ₁	F ₂	F ₃
域D ₁	O, E		W
域D ₂		R', O	R', O, W
域D ₃	E		

(a) 增加所有权前的访问矩阵

域 \ 对象	F ₁	F ₂	F ₃
域D ₁	O, E		
域D ₂		O, R', W'	R', O, W
域D ₃		W	W

(b) 增加所有权后的访问矩阵

图 (b) 所示为: 在域D₁中运行的进程删除了在域D₃中运行的进程对文件F₁的执行权; 在域D₂中运行的进程增加了在域D₃中运行的进程对文件F₂和F₃的写访问权。

■ 控制权

- 用于改变矩阵内同一行（域）中的各项访问权，亦即，用于改变在某个域中运行的进程对不同对象的访问权

域 \ 对象	F ₁	F ₂	F ₃	F ₄	F ₅	F ₆	打印机1	绘图仪2	域D ₁	域D ₂	域D ₃
域D ₁	R	R, W									
域D ₂			R	R, W, E	R, W		W				Control
域D ₃						R, E	W	W			

若在access (D₂, D₃) 中包括了控制权，则一个在域D₂中运行的进程能够改变对域D₃的各项访问权



访问矩阵的实现



访问矩阵：本身是稀疏矩阵



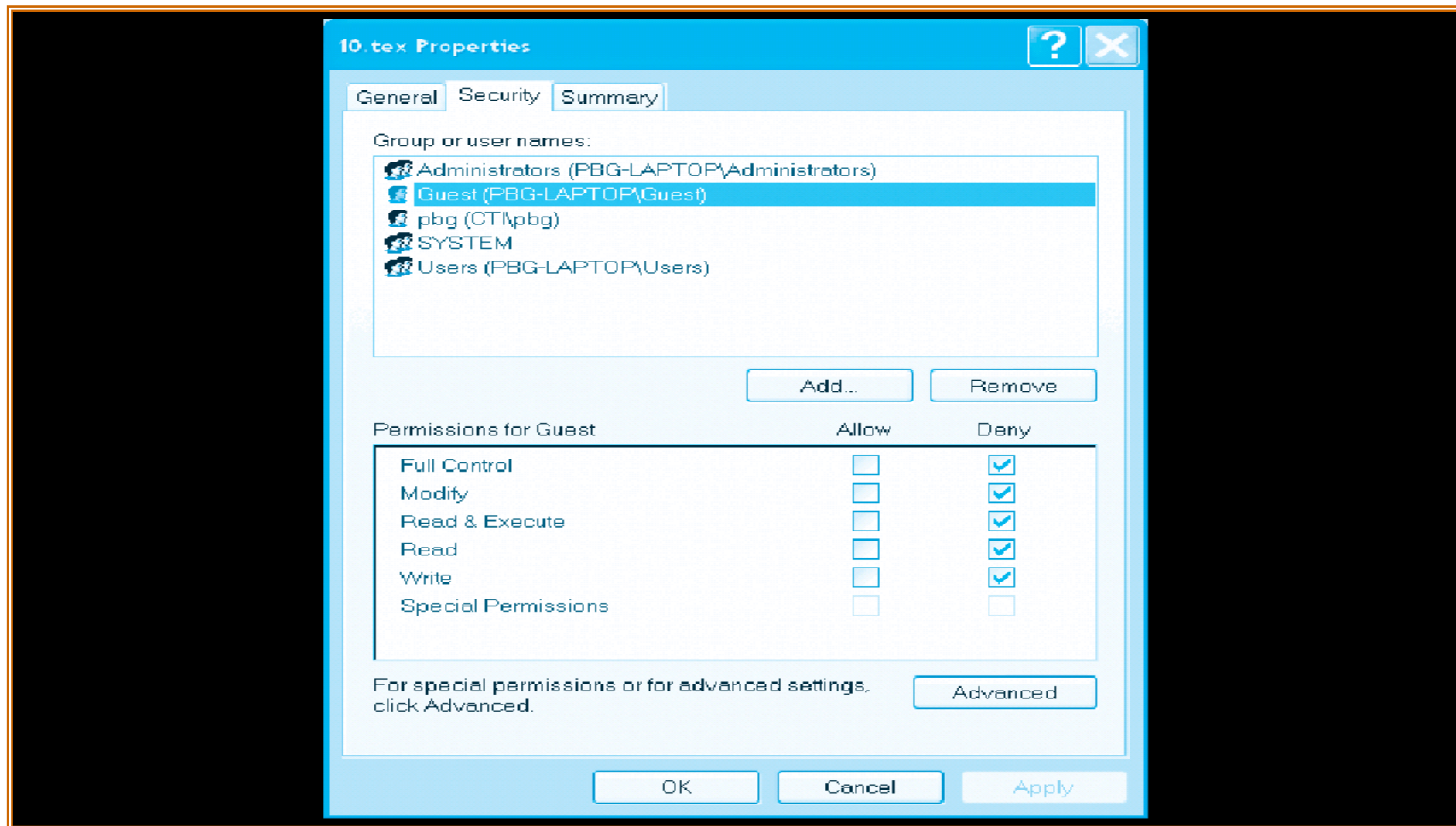
访问控制表ACL

- 对访问矩阵按列(对象)划分，为每一列建立一张访问控制表ACL，删除空项，由(域，权集)组成
- 当对象是文件时，ACL存放在文件的文件控制表中



访问权限表

- 对访问矩阵按行(域)划分，为每一行建立一张访问权限表
- 当对象是文件时，访问权限表用来描述一个用户对文件的一组操作



-rw-rw-r--	1	pbg	staff	31200	Sep 3 08:30	intro.ps
drwx-----	5	pbg	staff	512	Jul 8 09:33	private/
drwxrwxr-x	2	pbg	staff	512	Jul 8 09:35	doc/
drwxrwx---	2	pbg	student	512	Aug 3 14:13	student-proj/
-rw-r--r--	1	pbg	staff	9423	Feb 24 2003	program.c
-rwxr-xr-x	1	pbg	staff	20471	Feb 24 2003	program
drwx--x--x	4	pbg	faculty	512	Jul 31 10:31	lib/
drwx-----	3	pbg	staff	1024	Aug 29 06:52	mail/
drwxrwxrwx	3	pbg	staff	512	Jul 8 09:35	test/



内容导航:

-  8.1 文件和文件系统
-  8.2 文件的逻辑结构
-  8.3 文件目录
-  8.4 文件共享
-  8.5 文件保护
-  **8.6 Linux文件系统实例**

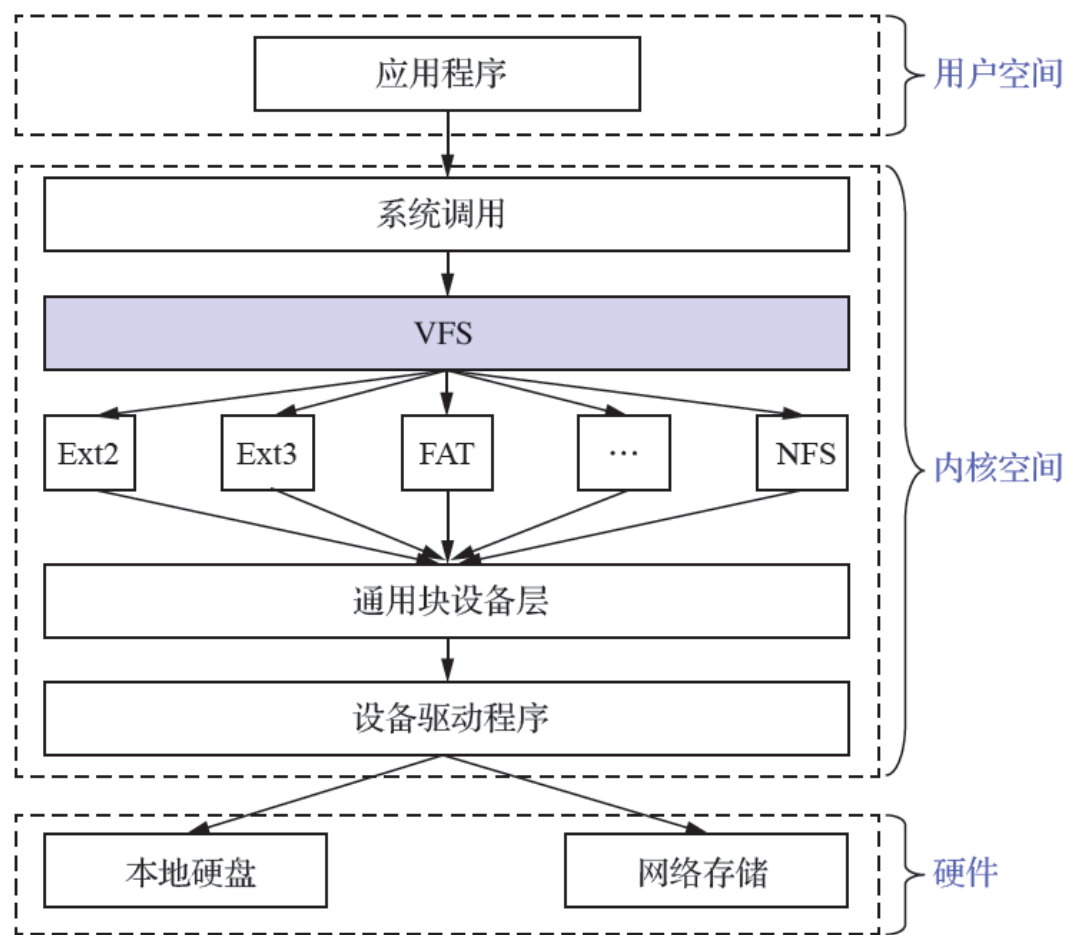
第8章 文件管理



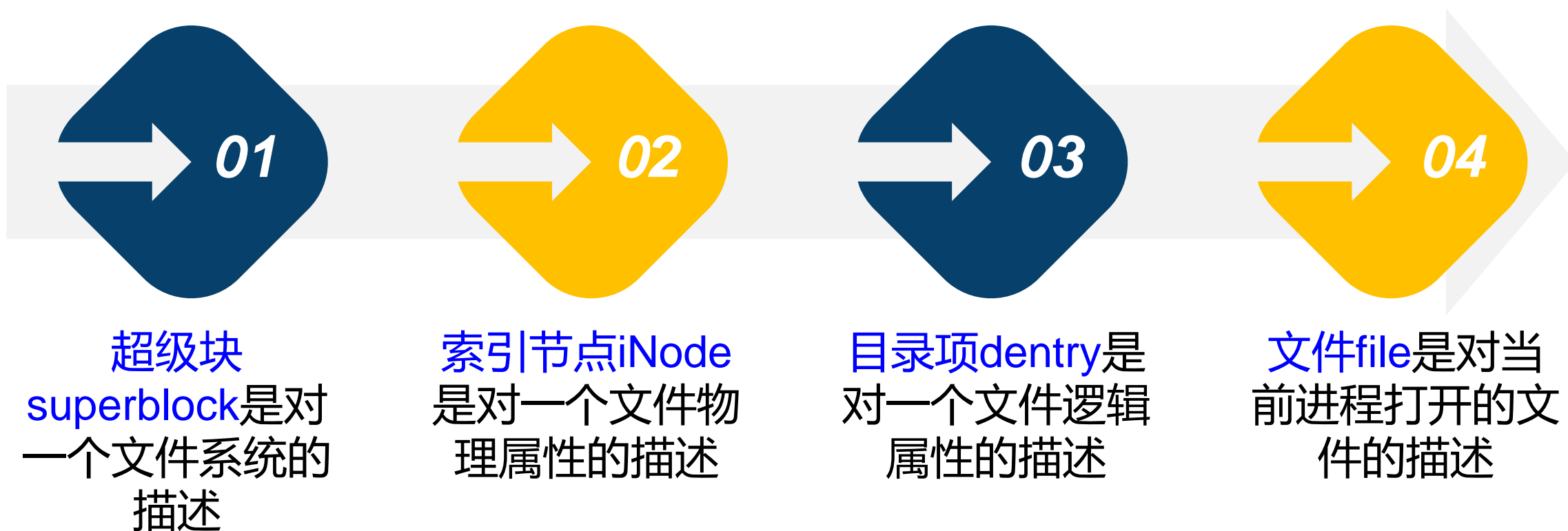
Linux的**虚拟文件系统VFS**隐藏了各种硬件的具体细节，包括本地存储设备和远程网络存储设备等，并且把文件系统的相关操作和不同文件系统的具体细节分离开，为所有的设备提供了统一的接口。



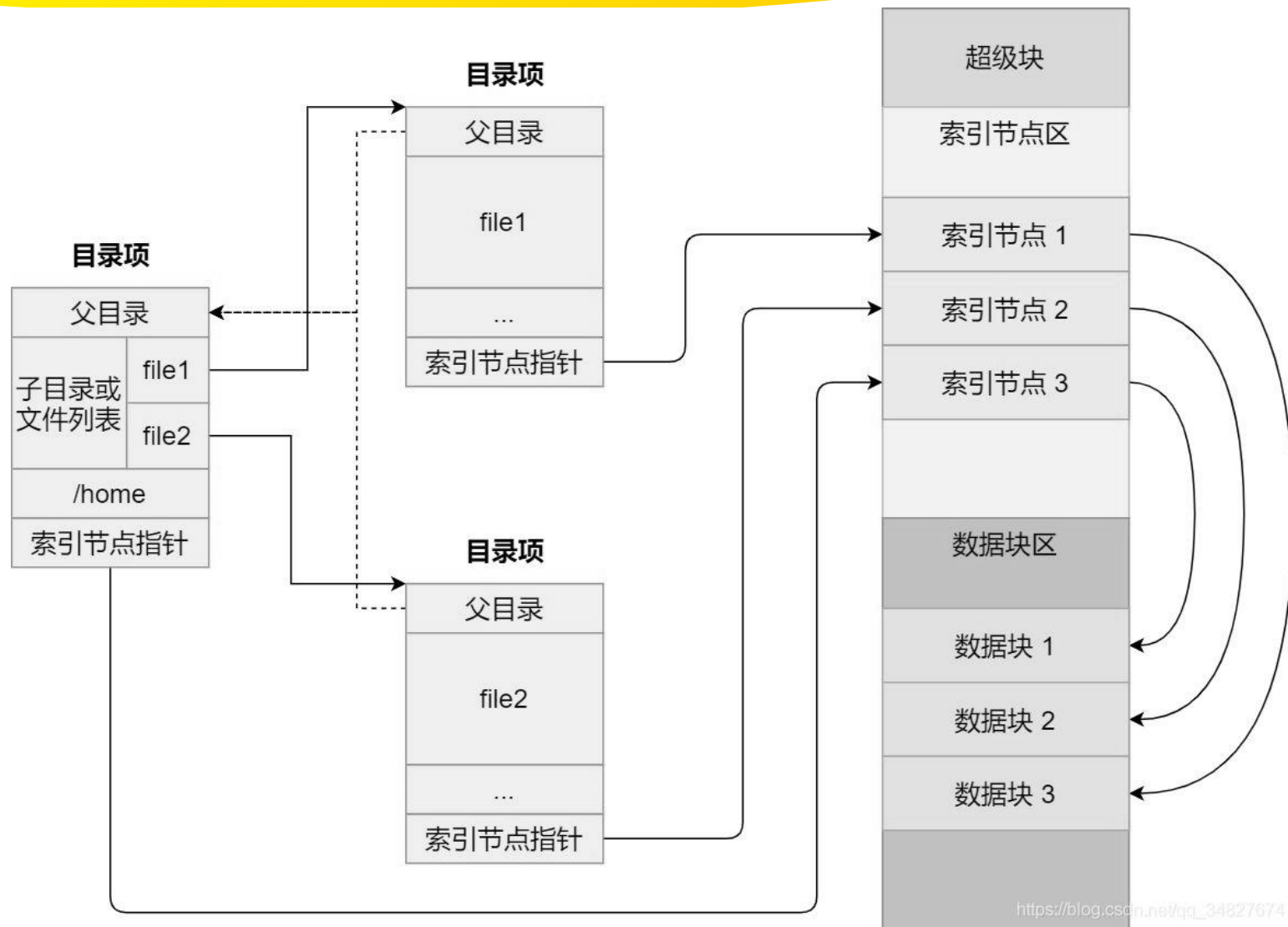
VFS使得Linux系统可以支持多达数十种不同的文件系统。



■ VFS支持的四个主要文件系统对象

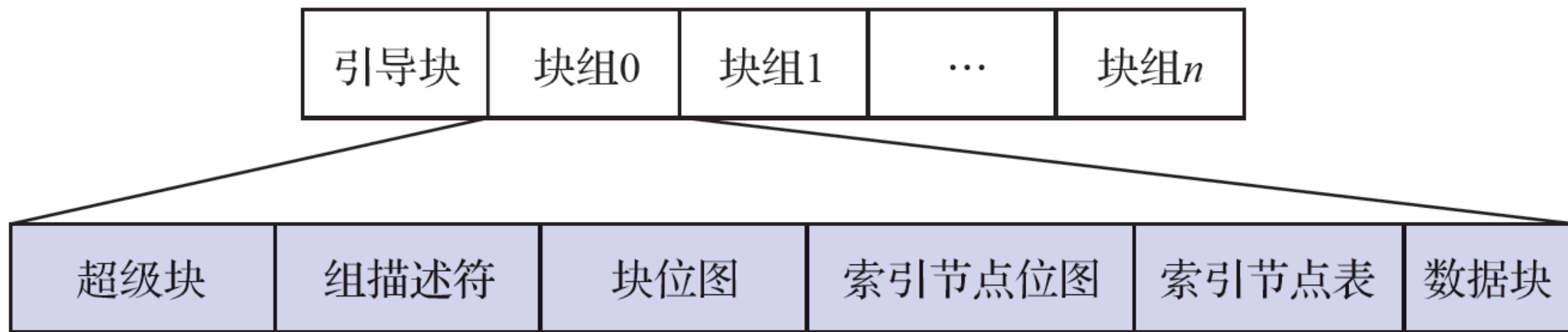


- OS 超级块: 记录整个文件系统的信息。
- OS 组描述符: 描述每个组块的开始与结束的block号码。
- OS 块位图: 数据块位示图, 用一个二进制位来表示数据块是否空闲。
- OS 索引节点位图: 类似数据块位示图, 用一个二进制位来表示iNode是否被使用。
- OS 索引节点表: 每个iNode的固定大小为128B, 每个文件仅占用一个iNode, 文件系统能够创建的文件数量与iNode的数量有关。
- OS 数据块: 每个数据块中只能放一个文件的数据, ext2支持的数据块的大小有1KB、2KB、4KB三种。





实例2: Linux ext2文件系统





学而时习之（第8章总结）

第1章	操作系统引论
第2章	进程的描述与控制
第3章	处理机调度与死锁
第4章	进程同步
第5章	存储器管理
第6章	虚拟存储器
第7章	输入/输出系统
第8章	文件管理
第9章	磁盘存储器管理
第10章	多处理机操作系统
第11章	虚拟化和云计算
第12章	保护和安全



本章学习结束

简答题

1	2	3	4	5	6	7	8
9	10	11	12				

计算题

13	14	15	16	17	
----	----	----	----	----	--

综合应用题

18	19	20	
----	----	----	--

标黄色为本次作业

简答题

1	2	3	4	5	6	7	8
9	10	11	12				

计算题

13	14	15	16	17	
----	----	----	----	----	--

综合应用题

18	19	20	
----	----	----	--

标黄色为本次作业

多维度发力构建良好生态环境

大力推动国产操作系统落地生根

2018年以来，在“中兴事件”“华为事件”等的影响下，国人尤其觉得倘若核心技术受制于人，我国就无法在相应的科技领域中占据主导地位，而这必将会严重影响国家安全、社会稳定以及经济发展等。例如，采用Windows 10等知识产权非我国所有的操作系统，即有可能给我国信息安全带来潜在风险。

值得庆幸的是，近年来，在“核高基”（核心电子器件+高端通用芯片+基础软件产品）等国家科技重大专项的支持与引导下，国产操作系统领域研发人员不断增强自主创新能力，同时促进自主研发的操作系统充分参与市场竞争，使得国产操作系统市场占有率得以大幅提升。那么，国产操作系统究竟能否成功呢？

多维度发力构建良好生态环境

大力推动国产操作系统落地生根

判定一个操作系统能否成功，主要得看它的生态建设情况。

在上述国际形势下，中国信息技术（information technology, IT）产业从基础硬件到系统软件，再到行业应用软件，在不同层面均迎来了国产替代潮。如今，国产操作系统的应用领域正在逐步拓展，金财、金农、金企等“十二金”工程以及通信、能源、交通等一系列关键基础设施，都已开始进行部分产品国产化替代。

坚信，随着国家战略向国产操作系统倾斜，行业市场规模必将越来越大，成熟的开源软件（如Linux等）所构成的生态圈，也必将会孕育出蓬勃发展的国产操作系统生态环境。但是与此同时，我们也要清晰地认识到研发国产操作系统依旧任重而道远，我们必须促使更多企业投入国产操作系统研发中来，同时不断培养IT产业高级人才，以实现国产操作系统生态发展的良性循环。