

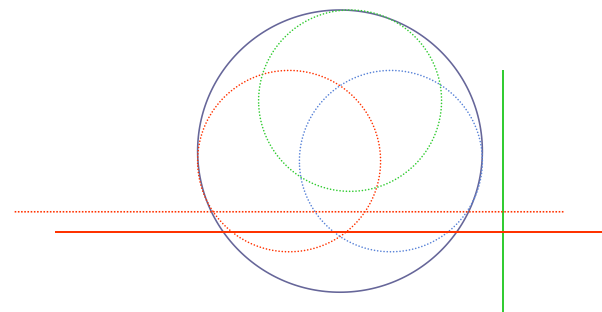


张涛

Review

文件系统的概述

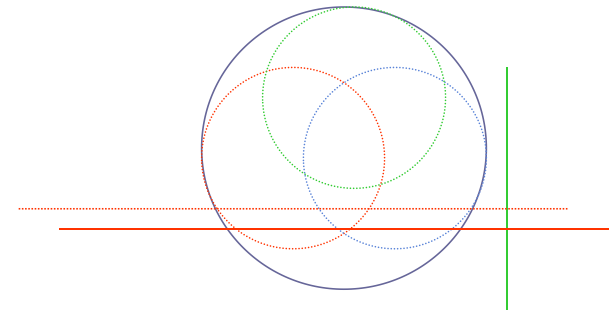
文件的结构和存取法



Today we focus on...

文件的目录

文件系统的使用



5.3 文件的目录

- 文件控制块FCB
- 简单的文件目录
- 二级目录
- 多级目录
- 路径和工作目录
- 索引结点

文件目录组织的目标

- **基本原则**：能方便而迅速地对目录进行检索，从而准确地找到所需文件。
- 对目录管理的要求如下：
 - 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。

5.3.1 文件控制块FCB

- **文件控制块**是操作系统为管理文件而设置的数据结构，存放了为管理文件所需的所有有关信息
- 文件控制块与文件一一对应，是文件存在的标志
- 文件控制块的内容：
 - 基本信息类：① 文件名；② 文件物理位置；③ 文件逻辑结构；④ 文件的物理结构
 - 存取控制信息类
 - 使用信息类

文件 名	扩展 名	属 性	备 用	时 间	日 期	第 一 块 号	盘 块 数
---------	---------	--------	--------	--------	--------	------------------	-------------

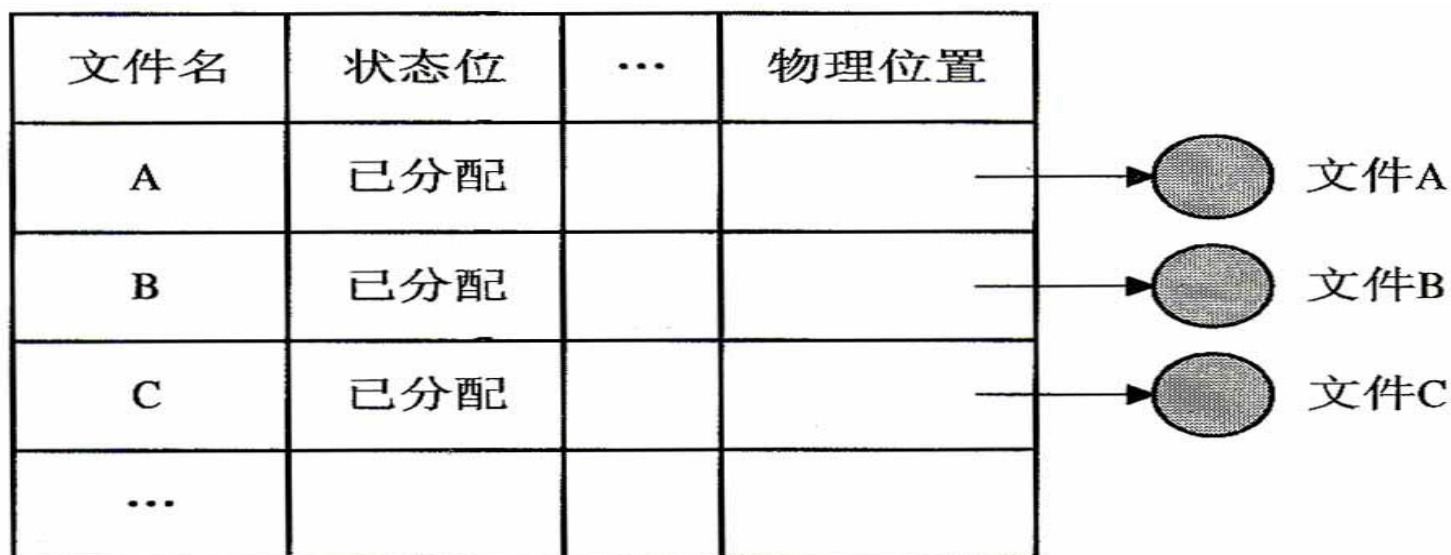
MS-DOS的文件控制快

文件目录与目录文件

- **文件目录**：把所有的FCB组织在一起，就构成了文件目录，即文件控制块的有序集合
- **目录项**：构成文件目录的项目（一个FCB就是目录表中的一个目录项）
- **目录文件**：为了实现对文件目录的管理，通常将文件目录以文件的形式保存在外存，这个文件就叫目录文件

5.3.2 简单的文件目录

- **一级目录：**整个目录组织是一个线性结构，系统中的所有文件都建立在一张目录表中。它主要用于单用户操作系统。



简单目录结构的特点

■ 优点：

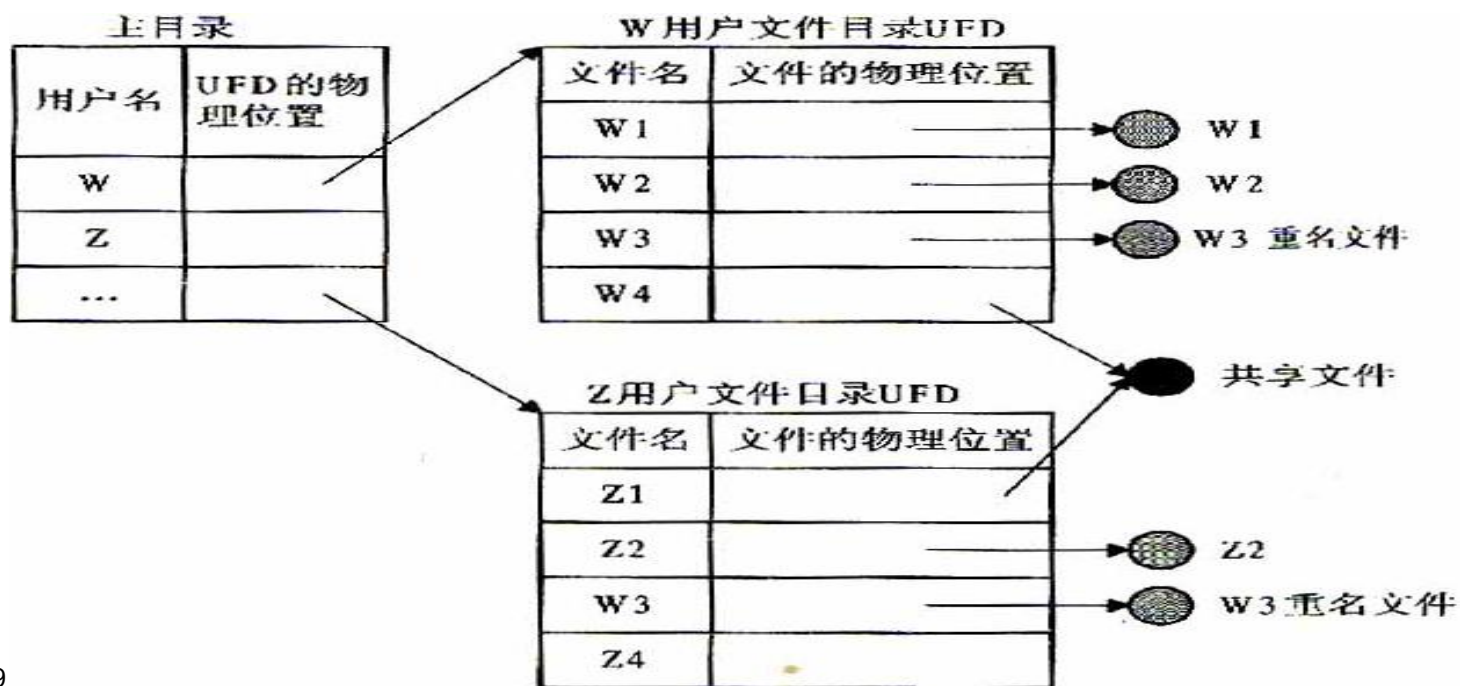
- 结构简单、清晰，便于维护和查找。
- 可实现按名存取。

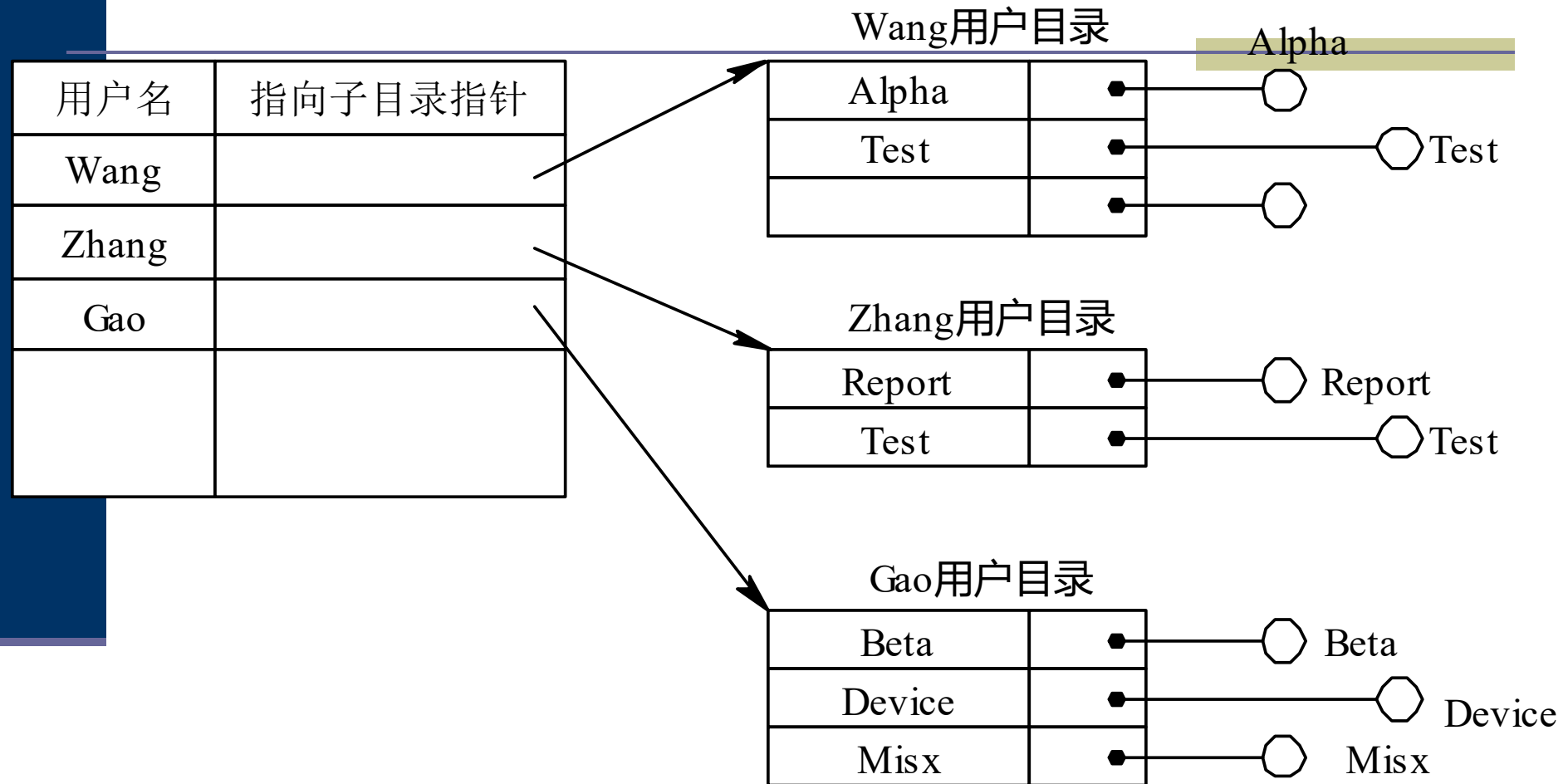
■ 缺点：

- 查找速度慢：为查找一个文件的目录，平均需查找目录表的一半，若是大型目录表，则搜索效率低下。
- 不允许重名：文件名和文件体有一一对应关系
- 不便于实现文件共享

5.3.3 二级目录

- 二级目录：在根目录下，每个用户对应一个目录（第二级目录）；在用户目录下是该用户的文件，而不再有下级目录。适用于多用户系统，各用户可有自己的专用目录。





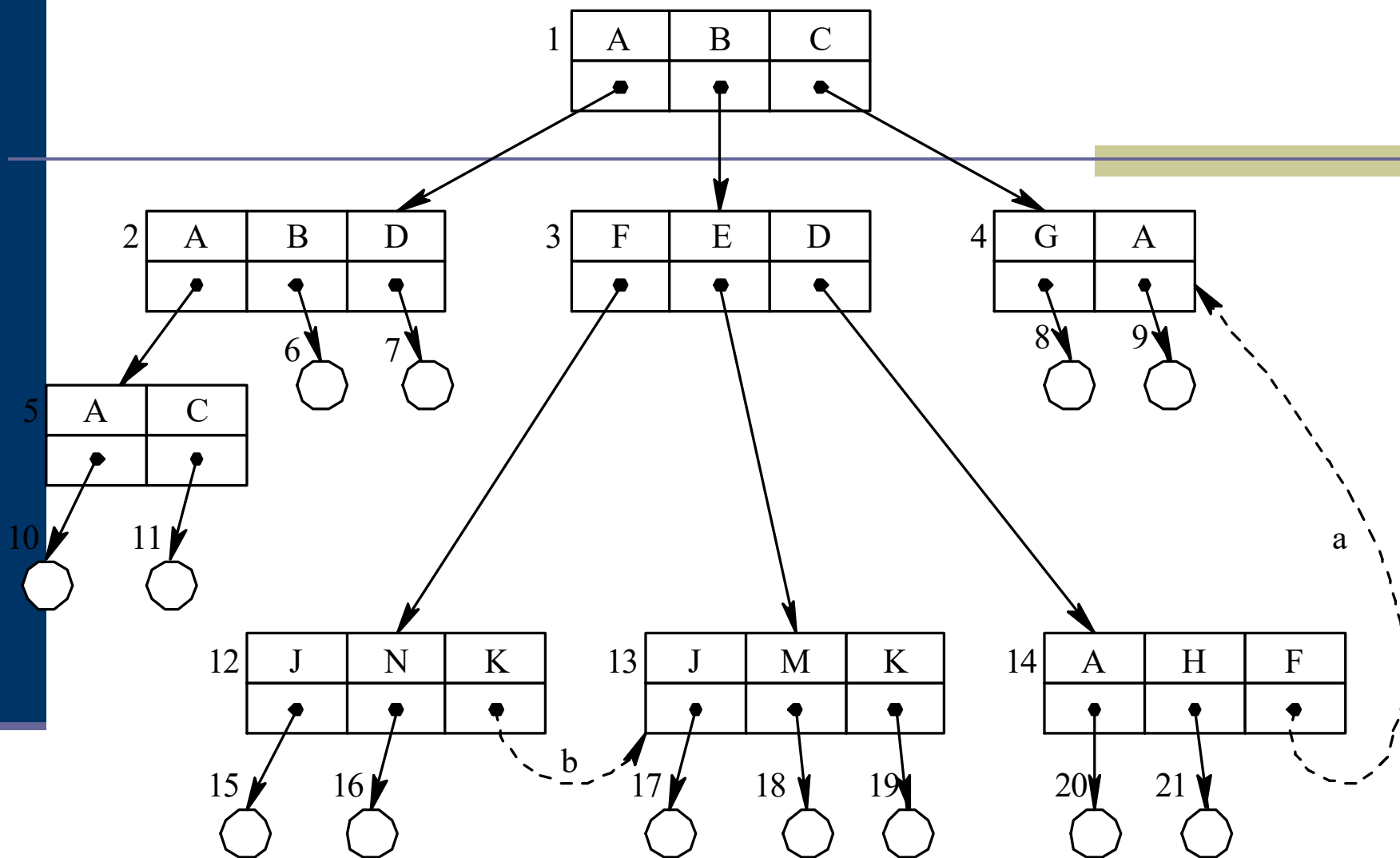
两级目录结构

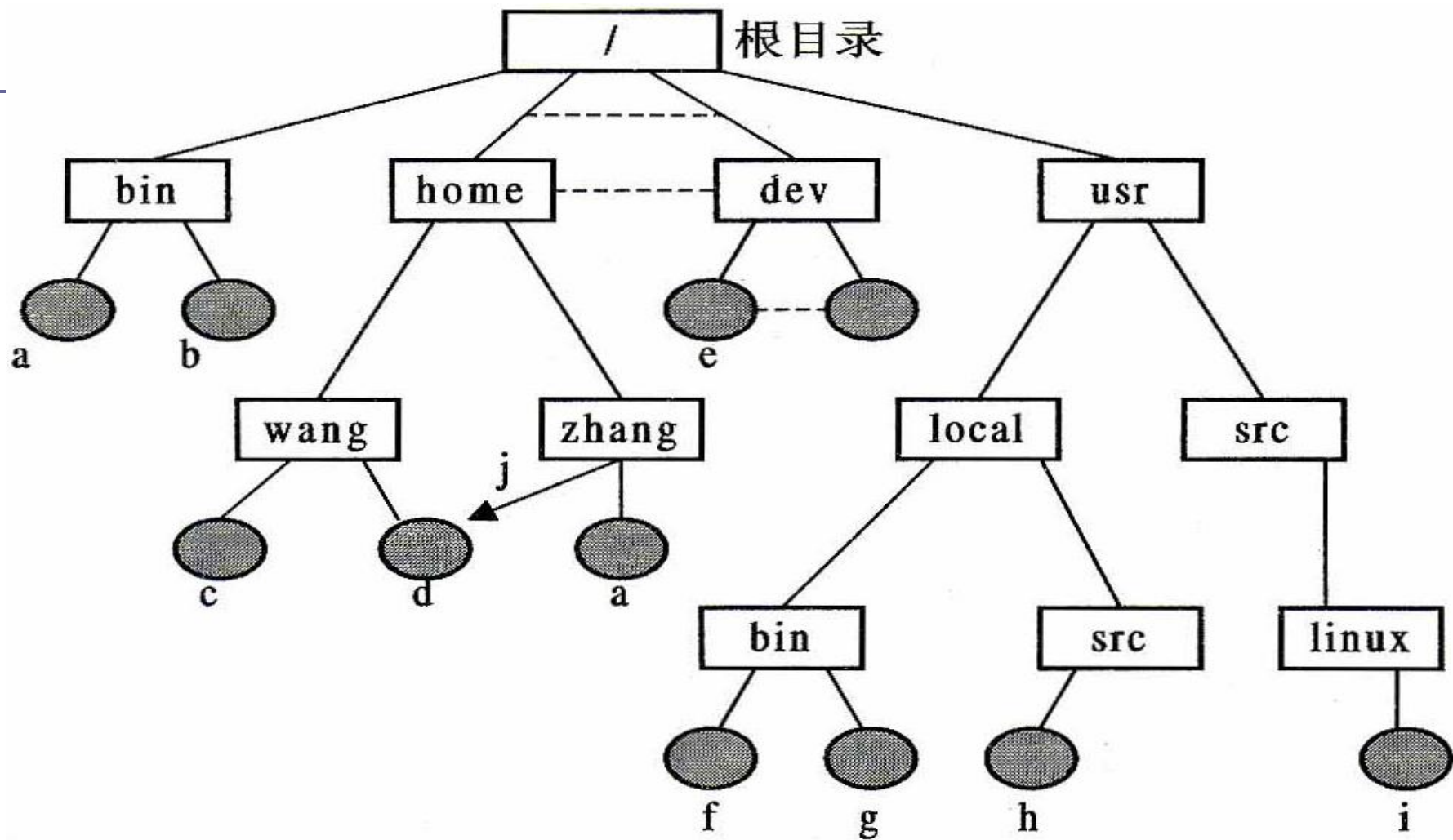
二级目录的优点

- 提高了检索目录的速度
- 在不同的用户目录中， 可以使用相同的文件名。
- 不同用户还可使用不同的文件名来访问系统中的同一个共享文件
- 缺点： 缺乏灵活性， 特别是不能反映现实世界中多层次的关系。

5.3.4 多级目录

- **多级目录**：或称为树状目录(tree-like)。在文件数目较多时，便于系统和用户将文件分散管理。
- 多级目录结构由**根目录**和**各级目录**组成，为管理上的方便，除根目录外，其它各级目录均以文件的形式组成目录文件。
- 目录文件中的每个目录项可以对应一个目录文件。也可以对应一个数据文件。如此类推，就形成多级目录结构。
- 各级目录文件称**中间结点**，用**方框**表示。**数据文件**称为**叶结点**，用**圆圈**表示。



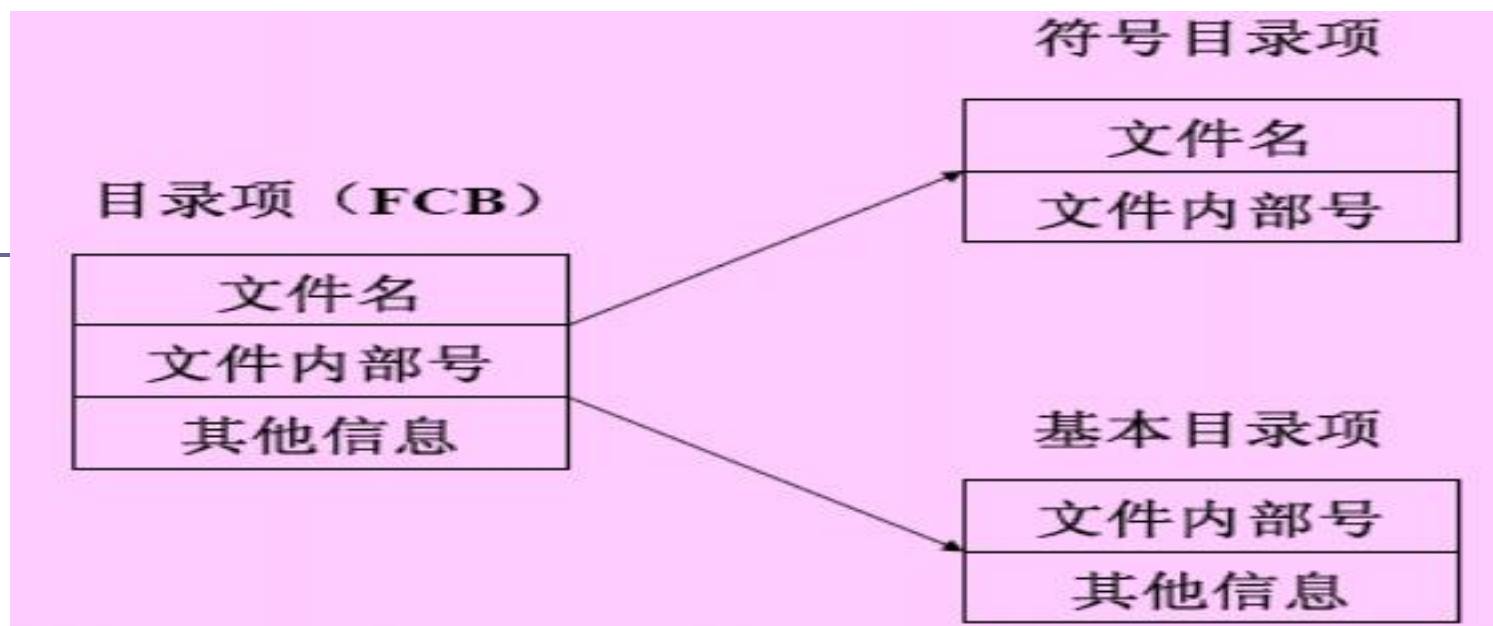


5.3.5 路径和工作目录

- **路径名(path name)**: 在多级目录结构中一个文件的唯一标识不再是文件名, 而是从根结点开始, 经过一个或多个中间结点, 到达某个叶结点的一条路径。称这条路径为文件的路径名, 它是文件的唯一标识。
- 路径名由根目录和所经过的**目录名**和**文件名**以及**分隔符**组成, 通常使用分隔符 /。
- **当前目录(Current Directory)**: 把当前使用的文件所在的目录指定为工作目录(或称当前目录)。
- **绝对路径名(absolute path name)**: 指由根目录开始的路径名。
- **相对路径名(relative path name)**: 指从当前工作目录开始的路径名。

5.3.6 索引结点

- 索引结点的引入：改进的多级目录
- 为了提高目录检索速度，可把目录中的文件说明（文件描述符）信息分成两个部分：
 - 符号文件目录：由文件名和文件内部标识组成的树状结构，按文件名排序；
 - 基本文件目录（索引节点目录）：由其余所有文件说明信息组成的线性结构，按文件内部标识排序；
- UNIX：i节点（索引节点）



fl.c	4

x	5

0	
1	
2	
3	
4	
5	
6	

基本目录表

例子

- 一个FCB有48个字节
- 符号目录项占 8字节
- 文件名6字节，文件号2字节
- 基本目录项占 $48-6=42$ 字节
- 假设，物理块大小512字节
- 讨论：对查找文件时平均访盘次数的影响？

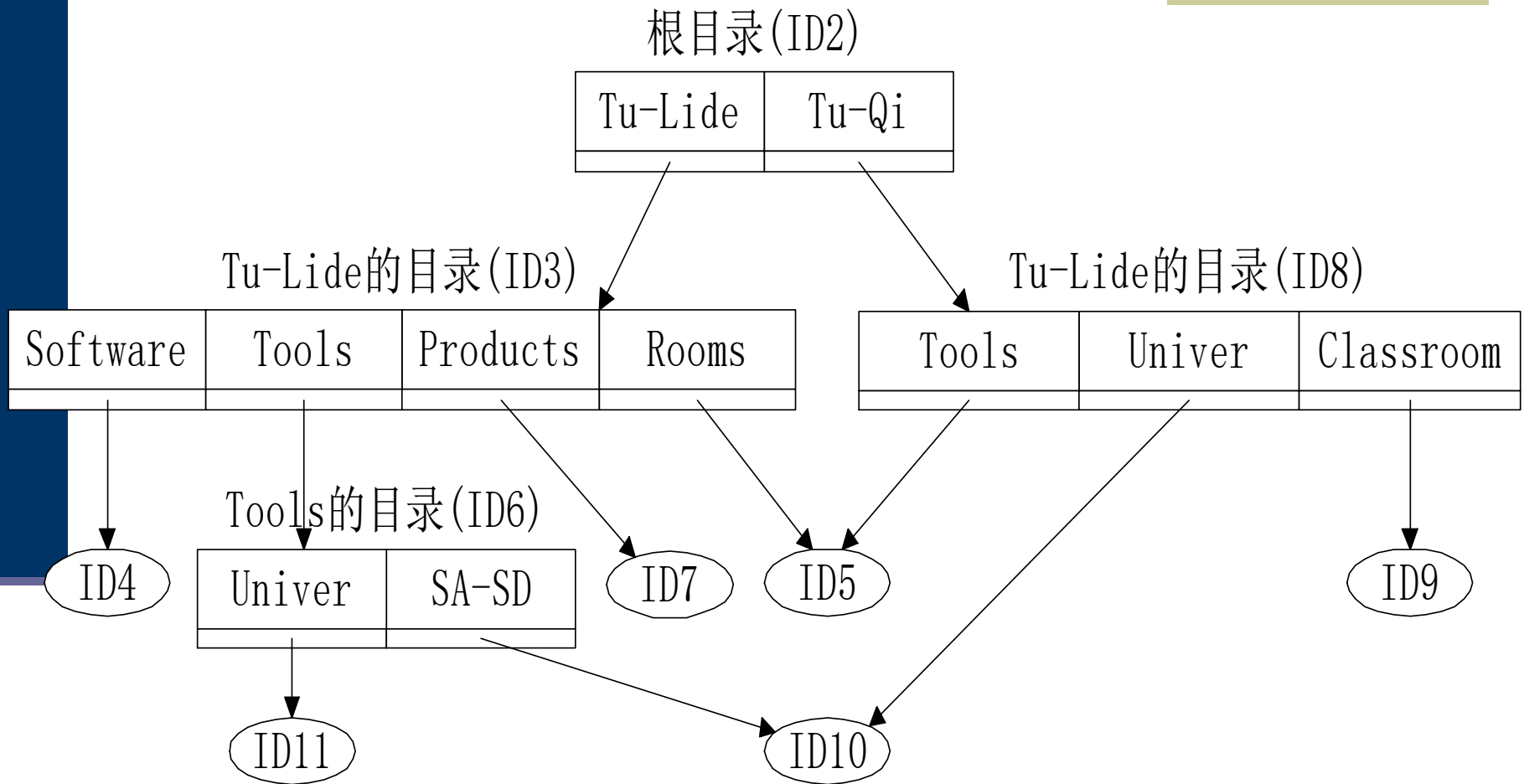
解

- 分解前：占 $512/48=10$ 个FCB
- 分解后：占 $512/8=64$ 个符号目录项或 $512/42=12$ 个基本目录项
-
- 假设：目录文件有128个目录项
- 分解前：占13块
- 分解后：符号文件占2块
- 基本文件占11块

查找一个文件的平均访盘次数

- 平均访盘次数 = $(N+1) / 2$ 次
- 分解前: $(1+13)/2 = 7$ 次
- 分解后: $(1+2)/2 + 1 = 2.5$ 次
- 减少了访问硬盘的次数, 提高了检索速度

符号文件目录



增加和删除目录

(1) 不删除非空目录。当目录(文件)不空时，不能将其删除，而为了删除一个非空目录，必须先删除目录中的所有文件，使之先成为空目录，后再予以删除。如果目录中还包含有子目录，还必须采取递归调用方式来将其删除，在MS-DOS中就是采用这种删除方式。

(2) 可删除非空目录。当要删除一目录时，如果在该目录中还包含有文件，则目录中的所有文件和子目录也同时被删除。

目录查询技术

1. 线性检索法

根目录

1	.
1	..
4	bin
7	dev
14	lib
9	etc
6	usr
8	tmp

在结点 6 中查找

usr 字段

结点 6 是

/usr 的目录

132

132 号盘块是

/usr 的目录

6	.
1	..
19	dick
30	erik
51	jim
26	ast
45	bal

结点 26 是

/usr/ast 的目录

496

496 号盘块是

/usr/ast 的目录

26	.
6	..
64	grants
92	books
60	mbox
81	minik
17	src

查找/usr/ast/mbox的步骤

2. Hash方法

一种处理此“冲突”的有效规则是：

(1) 在利用Hash法索引查找目录时，如果目录表中相应的目录项是空的，则表示系统中并无指定文件。

(2) 如果目录项中的文件名与指定文件名相匹配，则表示该目录项正是所要寻找的文件所对应的目录项，故可从中找到该文件所在的物理地址。

(3) 如果在目录表的相应目录项中的文件名与指定文件名并不匹配，则表示发生了“冲突”，此时须将其Hash值再加上一个常数(该常数应与目录的长度值互质)，形成新的索引值，再返回到第一步重新开始查找。

文件别名的实现：文件共享

- 提供文件共享的方法有两种：
 - 各用户通过唯一的共享文件的路径名访问共享文件（该方法的访问速度慢，适用于不经常访问的文件共享），
 - 利用多个目录中的不同文件名来描述同一共享文件（即文件别名，该方法的访问速度快，但会影响文件系统的树状结构，适用于经常访问的文件共享，同时存在一定的限制）。
- 文件别名的实现方法有以下两种：
 - 基于索引结点
 - 基于符号链接

1. 基于索引结点(index node)的文件别名

也称为**硬链接 (hard link)**；通过多个文件名链接(link)到同一个索引结点，可建立同一个文件的多个彼此平等的别名。别名的数目记录在索引结点的链接计数中，若其减至0，则文件被删除。

- UNIX举例："ln source target ; rm source"则该文件还存在，文件名为target；
- 限制：不能跨越不同文件卷；通常不适用于目录，否则由树状变为网状。

2. 基于符号链接(symbolic link, shortcut)的文件别名

它是一种特殊类型的文件，其内容是到另一个目录或文件路径的链接。建立符号链接文件，并不影响原文件，实际上它们各是一个文件。可以建立任意的别名关系，甚至原文件是在其他计算机上。

- UNIX举例：`"ln -s a b ; rm a"`则文件a不存在，b能被控制但无法访问；
- 缺点：空间和时间开销更大。如果设置不当，上下级目录关系可能会形成环状。

在文件间建立连接ln

- ln命令用来建立硬连接和符号连接。硬连接是一个文件的额外的名字，没有源文件，硬连接便不能存在。而对于符号连接，当原文件被删除后，符号连接仍然存在但无法访问。
- -s 建立一个符号连接而不是硬连接
- -d 建立目录的硬连接

■ 硬连接：

- 原文件名和链接文件名都指向相同的物理位址。
- 目录不能有硬连接；硬连接不能跨越文件系统（不能跨越不同的分区）。
- 文件在磁盘中只有一个复制，以节省硬盘空间。
- 由于删除文件要在同一个索引节点属于唯一的连接时才能成功，因此可以防止不必要的误删除。

■ 符号连接：

- 用ln -s命令创建文件的符号连接；
- 符号连接是Linux特殊文件的一种，作为一个文件，它的资料是它所连接的文件的路径名。类似于Windows下的快捷方式。
- 可以删除原有的文件而保存链接文件，没有防止误删除功能。

5.4 文件系统的使用

- 在文件系统中提供对文件的各种操作，这些操作方便、灵活地使用文件及文件系统，形式分别为：系统调用或命令
- 主要操作
 - 提供设置和修改对用户文件存取权限
 - 提供建立、修改、改变、删除目录的服务
 - 提供文件共享，设置访问路径的服务
 - 提供创建、打开、读、写、关闭、撤消文件等服务
 - 文件系统维护
 - 文件系统的转储和恢复

(1) 建立文件

- 实质是建立文件的FCB
- 建立必要的存储空间，分配空FCB，根据提供的参数及需要填写有关内容，返回一个文件描述
- 目的：建立系统与文件的联系
- create (文件名, 访问权限, (, 最大长度))
- ① 检查参数的合法性
 - 文件名是否符合命名规则
 - 是→②, 否则→错误返回
- ② 检查同一目录下有无重名文件
 - 无→③, 有→错误返回

■ ③ 在目录中是否有空闲位置

- 有→②，否则→不成功返回

- 有的系统可能要为此文件申请数据块空间（申请一部分或一次性全部申请）

■ ④ 填写目录项内容：

- 文件名，用户名等，存取权限，长度置零，（，首址）

■ ⑤ 返回

(2) 打开文件

- 为文件读写做准备
- 给出文件路径，获得文件句柄(file handle)，或文件描述符(file descriptor)，需将该文件的目录项读到内存
- `fd=open` (文件路径名，打开方式)
- ① 根据文件路径名查目录，找到FCB主部；
- ② 根据打开方式、共享说明和用户身份检查访问合法性；

- ③ 根据文件号查系统打开文件表，看文件是否已被打开；
 - 是→共享计数加1
 - 否则→将外存中的FCB主部等信息填入系统打开文件表空表项，共享计数置为1；
- ④ 在用户打开文件表中取一空表项，填写打开方式等，并指向系统打开文件表对应表项
- 返回信息：fd：文件描述符，是一个非负整数，用于以后读写文件

■ (3) 关闭文件

- 释放文件描述符

- 把该文件在内存缓冲区的内容更新到外存上

■ (4) 删除文件：撤销FCB

(5) 指针定位

■ seek (fd, 新指针的位置)

- 系统为每个打开文件维护一个读写指针，是相对于文件开头的偏移地址（读写指针指向每次文件读写的开始位置，在每次读写完成后，读写指针按照读写的数据量自动后移相应数值）
- ① 由fd查用户打开文件表，找到对应的入口；
- ② 将用户打开文件表中文件读写指针位置设为新指针的位置，供后继读写命令存取该指针处文件内容

(6) 读文件

- read (文件名, (文件内位置), 要读的长度, 内存目的地址)
- 隐含参数: 进程主
- ① 检查长度是否为正整数
 - 是→②, 否则→⑩
- ② 根据文件名查找目录, 确定该文件在目录中的位置 (? 文件未打开)
- ③ 根据隐含参数中的进程主和目录中该文件的存储权限数据, 检查是否有权读?
 - 是→④, 否则→⑩

- ④ 由文件内位置与要读的长度计算最末位置，将其与目录中的文件长度比较，超过否？
 - 是→⑩，否则→⑤
 - 也可将参数中的长度修正为目录中的文件长度
- ⑤ 根据参数中的位置、长度和目录中的信息，确定块号、块数、块内位移（多次读盘）
- ⑥ 根据下一块号读块至内存缓冲区
- ⑦ 根据块内位移长度取出要读的内容，送至参数中的内存目的地址
- ⑧ 确定还读下一块吗？同时确定下一块块号
 - 是→⑥，否则→⑨
- ⑨ 正常返回
- ⑩ 错误返回，返回相应错误号

- (7) 写文件
- (8) 文件连接(LINK)
- (9) 复制文件
- (10) 目录的操作

What you need to do?

- 复习课本5.2、5.3节的内容
- 课后作业：P165 习题5、9