



河海大学

计算机与信息学院

第六章 文件管理



文件

文件目录

文件组织与数据存储

文件系统其他功能的实现



6.0 文件系统概述

- 文件系统是操作系统中负责存取和管理信息的模块，它用统一的方式管理用户和系统信息的存储、检索、更新、共享和保护，并为用户提供一整套方便有效的文件使用和方法



6.0 文件系统概述

- 文件不但反映了用户概念中的逻辑结构，而且和存放它的辅助存储器的存储结构紧密相关
- 一个文件必须从**逻辑文件**和**物理文件**两个侧面来观察它
- 用户只需关心文件的逻辑结构，不必涉及存储结构；而**文件系统**在**存储和检索**文件时必须实现逻辑结构到存储结构的映射，实现逻辑文件与物理文件、逻辑地址与物理地址、逻辑结构与物理结构、逻辑操作与物理操作之间的转换



6.0 文件系统的功能

- 文件的按名存取
- 文件目录建立和维护
- 实现逻辑文件到物理文件的转换
- 文件存储空间的分配和管理
- 提供合适的文件存取方法
- 实现文件的共享、保护和保密
- 提供一组可供用户使用的文件操作



河海大学

计算机与信息学院

6.1 文 件

- 文件的概念
- 文件的命名
- 文件的类型
- 文件的属性
- 文件的存取方法
- 文件的使用





文件的概念

文件的命名

文件的类型

文件的属性

文件的存取方法

文件的使用



6.1.1 文件的概念

- 文件是由文件名字标识的一组信息的集合
- 文件是一个抽象机制，提供了把文件保存在磁盘上，用户不必了解信息存储细节且便于读取的方法，这一抽象机制中最重要的是文件命名



文件的概念

文件的命名

文件的类型

文件的属性

文件的存取方法

文件的使用



6.1.2 文件的命名

- 文件名是字母或数字组成的字母数字串，它的格式和长度因系统而异
- 一般为：文件名称 + “.” + 扩展名称
- 通配符：“?”，“*”

- .txt, .html
- .c, .cpp, .h, .java, .jsp
- .class, .exe, .rar, .gz, .tar, .zip
- .mp3, .mov, .m4a, .rmvb, .mkv
- .bmp, .jpg



文件的概念

文件的命名

文件的类型

文件的属性

文件的存取方法

文件的使用



6.1.3 文件的类型

- 按用途分：系统文件、库文件、用户文件
- 按信息流向分：输入文件、输出文件
- 按保护级别分：只读文件、读写文件、不保护文件
- 按存放时限分：临时文件、永久文件、档案文件
- 按设备类型分：磁盘文件、磁带文件、软盘文件
- 还可按文件的逻辑结构或物理结构分类



6.1.3: UNIX支持的文件类型

- 普通文件: ASCII, Binary
- 目录文件
 - 由文件目录组成的系统文件
- 特别文件
 - 块设备文件: 用于磁盘、光盘或磁带等块设备的I/O操作
 - 字符设备文件: 用于终端、打印机等字符设备的I/O操作



文件的概念

文件的命名

文件的类型

文件的属性

文件的存取方法

文件的使用



6.1.4 文件的属性

- 文件属性用于文件的管理控制和安全保护，包括：
 - 文件基本属性
 - 文件类型属性
 - 文件保护属性
 - 文件管理属性
 - 文件控制属性



6.1.4: 基本属性

- 文件名称、扩展名称
- 所有者: owner
- 所有者组: group



6.1.4: 类型属性

- 普通文件
 - 魔数 (magic number)
- 目录文件
 - file 命令
- 系统文件
- 隐式文件
- 设备文件
- ASCII码文件
- 二进制文件



6.1.4: 管理属性

- 文件创建时间
- 最后存取时间
- 最后修改时间



6.1.4: 控制属性

- 逻辑记录长
- 文件当前长
- 文件最大长
- 以及允许的存取方式标志
- 关键字位置
- 关键字长度



6.1.4: 保护属性

- 可读
- 可写
- 可执行
- 可更新
- 可删除
- 可改变保护
- 档案属性



6.1.4: 保护属性（续）

- **文件保护属性**用于防止文件被破坏，称为文件保护包括两个方面：
 1. 防止系统崩溃所造成的文件破坏
 2. 防止文件主和其他用户有意或无意的非法操作所造成的文件不安全性
- 防止系统崩溃造成文件破坏的方法：
 - 定时转储
 - 多副本



6.1.4: 访问控制

- 防止文件主和其他用户有意或无意的非法操作所造成的文件不安全性
- 基本思想是建立三元组（用户、对象、存取权限）
 - 用户：每一个操作系统使用者的标识
 - 对象：一般指文件
 - 存取权限：如读、写、删除、执行等



6.1.4: 访问控制（续）

- UNIX/Linux把用户分为
 - 文件主: owner
 - 同组用户: group
 - 其他用户: others
- 定义存取权限可读r、可写w、可执行x，文件属性共有10位：

-	rwx	rwx	rwx
-	<u> </u>	<u> </u>	<u> </u>
	owner	group	others



河海大学

计算机与信息学院

文件的概念

文件的命名

文件的类型

文件的属性

文件的存取方法

文件的使用



6.1.5 文件的存取方法

- 存取方法是操作系统为用户程序提供的使用文件的技术和手段，也指读写文件存储器上的物理记录的方法
- 用户关心数据的逻辑结构，即记录及其逻辑关系，数据独立于物理环境
- 系统关心数据的物理结构，数据被文件系统按照某种规则排列和存放到物理存储介质上



6.1.5（续）

1. **顺序存取**: 按记录顺序进行读/写操作的存取方法. 主要用于磁带文件以及磁盘上的顺序文件
2. **直接存取**: 以任意次序直接读写某个记录. 用户提供相对块号给操作系统, 绝对块号由系统换算得到
3. **索引存取**: 文件专门有一个按记录关键字有序的索引表, 用户通过查找索引表定位并读出记录



文件的概念

文件的命名

文件的类型

文件的属性

文件的存取方法

文件的使用



6.1.6 文件的使用

- 用户通过两类接口与文件系统联系，获得文件系统服务：
 - 第一类是与文件有关的操作命令或作业控制语言中与文件有关的语句，构成文件系统**人机接口**
 - 第二类是提供给用户程序使用的文件类**系统调用**，构成了用户和文件系统的另一个接口



6.1.6: 基本文件类系统调用

- 建立文件 `creat`
- 打开文件 `open`, `openat`
- 读/写文件 `read`, `write`
- 文件控制 `lseek`, `fcntl`, `ioctl`
- 关闭文件 `close`
- 撤销文件 `unlink`



河海大学

计算机与信息学院

6.2 文件目录

- 文件控制块

- 层次目录结构

- 文件目录的检索





6.2.0

- 文件系统通常采用分层结构实现，大致分为三层：文件管理，目录管理和磁盘主存映射管理。
- **文件管理层**：实现文件的逻辑结构，为用户提供工作系统调用
- **目录管理层**：负责查找文件描述符，进而找到所需文件
- **磁盘主存映射管理**：将文件的逻辑地址转换成磁盘的物理地址



文件控制块

层次目录结构

文件目录的检索



6.2.1：文件控制块

- **文件控制块FCB**：文件系统给每个文件建立唯一的管理数据结构，包括：
 - **文件标识和控制信息**：如文件名，文件存取权限等
 - **文件的逻辑结构信息**：如记录类型，记录个数等
 - **文件物理结构信息**：文件所在的设备名，存放位置等
 - **文件使用信息**：共享进程数，修改情况，文件最大长度等
 - **文件管理信息**：文件的建立时间，最近修改日期等



6.2.1: 文件目录

- 依靠FCB, 实现文件的“按名存取”
- 为了加快文件查找速度, 需把FCB集中起来管理, 组成文件目录
- 文件目录包含两种目录项分别用于描述子目录和文件
- 文件目录的功能是将文件名转换成文件信息在磁盘上的物理位置
- 文件和目录的层次结构和集合就构成文件系统, 它负责文件目录的建立, 维护和检索, 并且要求所编排的目录便于查找, 防止冲突, 检索方便迅速。



6.2.1: UNIX 下的文件目录

- UNIX目录项建立方法

文件名	inode节点号
-----	----------

14个字节

2个字节

- 目录中的文件名和管理信息分开，后者单独组成数据结构，称索引节点（inode）
- 文件目录项中仅剩14个字节的文件名和2字节的inode号
- 目录项组成的目录文件和普通文件一样，均存放在文件存储器中
- 加快了目录检索速度，且便于实现文件共享，



6.2.1: 外存inode

- 文件存储设备上的每个文件，都有一个外存文件控制块（外存索引节点）inode与之对应
- 文件控制块inode对于文件的作用，犹如进程控制块PCB、user对于每个进程的作用，集中了文件的属性及有关的一切必要信息



6.2.1: 内存inode表

- 内存索引节点inode，使用某文件的信息时，申请一个内存inode，把外存inode内容拷贝到内存inode中
- 增加部分信息：
 - 活动标识：i_flag(占用、修改、安装点、上锁位)
 - 文件链接数：i_count
 - inode所在设备名：i_dev
 -



河海大学

计算机与信息学院

文件控制块

层次目录结构

文件目录的检索



6.2.2 层次目录结构

- 一级目录结构
- 二级目录结构
- 树形目录结构



6.2.2: 一级目录结构

- 系统构造一张线性表，每个文件有关的属性占用一个目录项就成了一级目录结构
- 容易实现
- 存在“命名冲突”

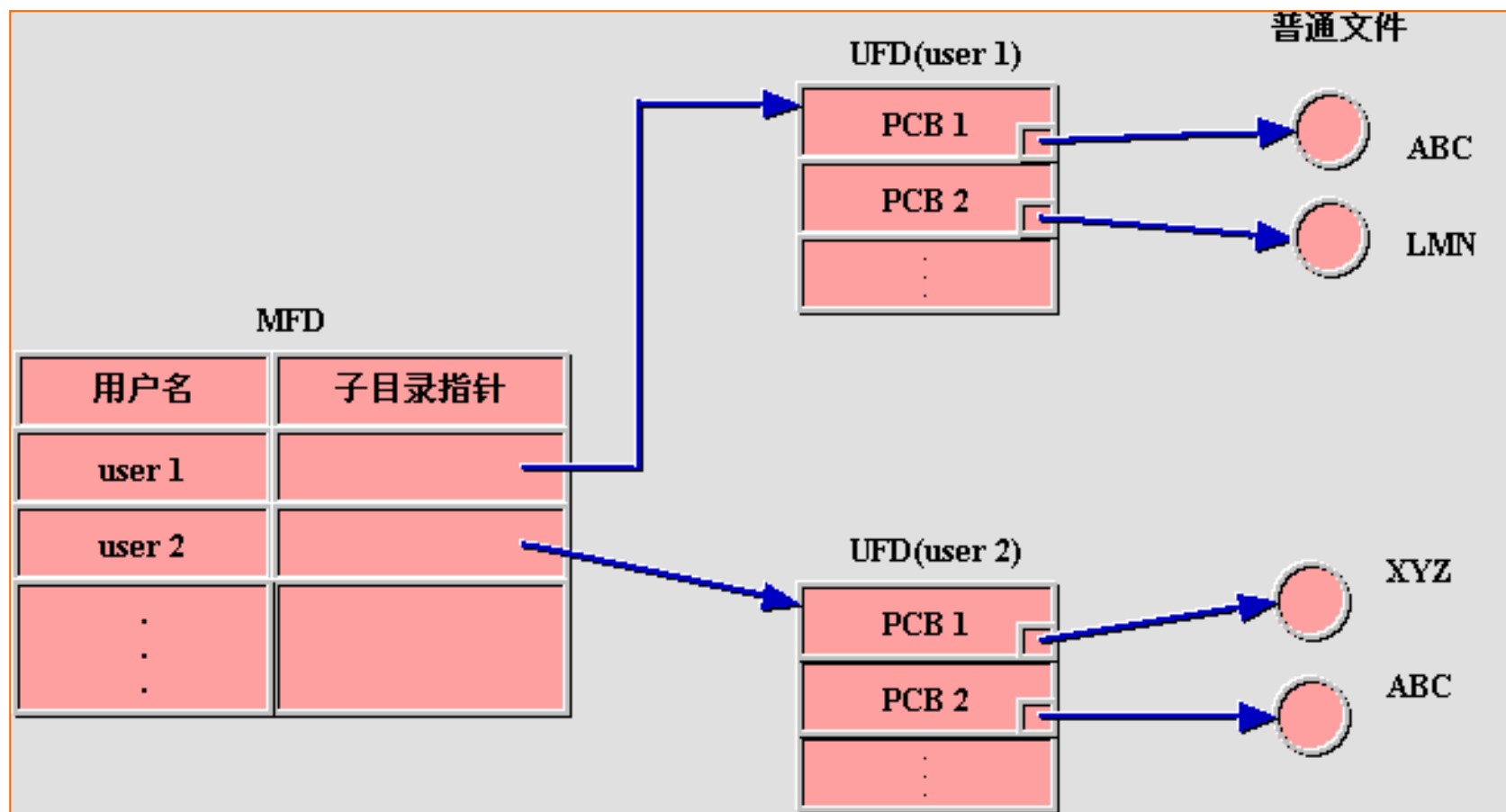


6.2.2: 二级目录结构

- **第一级为主文件目录**，用于管理所有用户文件目录，登记系统接受的用户名字及该用户文件目录的地址
- **第二级为用户文件目录**，为该用户的每个文件保存一登记栏，其内容与一级目录的目录项相同



6.2.2: 二级目录结构





6.2.2: 二级目录结构

- 优点：解决了文件的重名问题和文件共享问题，即允许在不同用户目录下，文件名可以相同。

方法：存取一个文件时要给出文件路径名：

用户名|文件名

如用户wang的文件beta，其路径名为：wang: beta



6.2.2: 树形目录结构

- 每一级目录可以是下一级目录的说明，也可以是文件的说明，形成层次关系
- 多级目录结构采用树形结构，是一棵倒向有根树，树根是根目录
- 从根向下，每个树枝是一个子目录
- 而树叶是文件



6.2.2: 树形目录结构

- 树型多级目录优点:

- 避免了重名
- 便于分类组织
- 便于高效查找
- 便于共享



6.2.2: 树形目录结构

- **路径名**: 子目录名组成的部分称为路径名
- **绝对路径名**: 从根目录开始的路径名称为绝对路径名
- **相对路径名**: 从当前目录开始直到数据文件为止所构成的路径名称为相对路径名

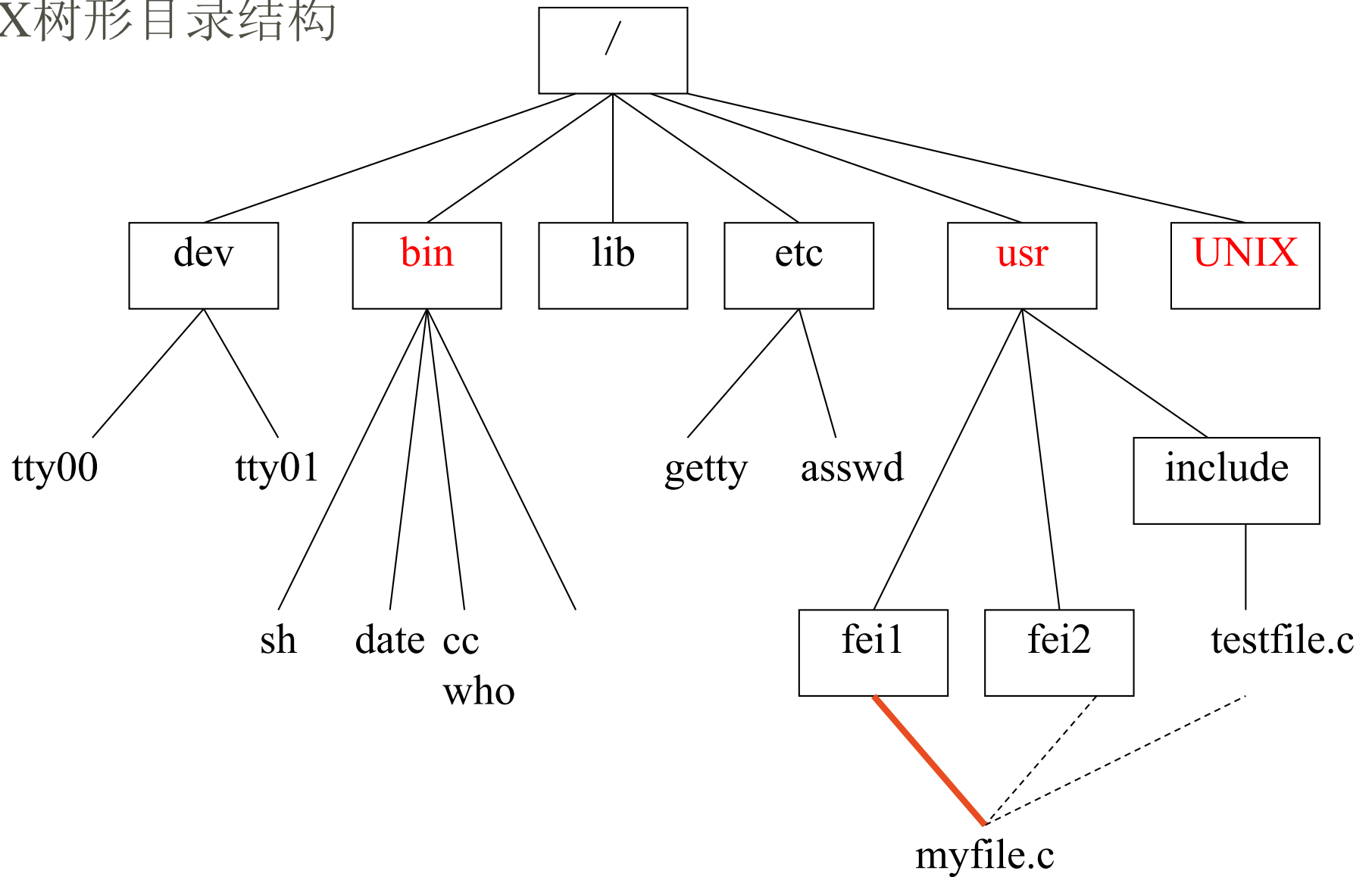


6.2.2: UNIX 树形目录结构

- dev设备子目录
- bin实用程序子目录
- lib库文件子目录
- etc基本数据和维护实用程序子目录
- tmp临时文件子目录
- usr通用目录：无需修改的命令程序文件，库程序等
- home用户文件主目录：包括：小型化的bin、小型化的tmp、小型化的文件库lib包括文件include及各用户的多种文件



UNIX树形目录结构





河海大学

计算机与信息学院

文件控制块

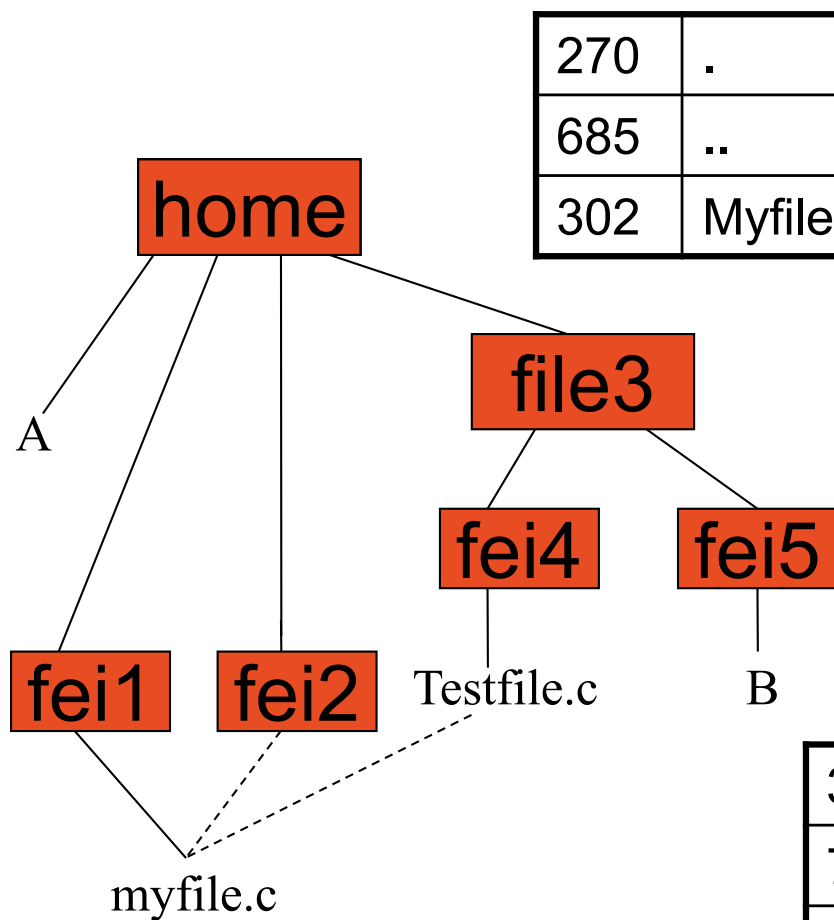
层次目录结构

文件目录的检索

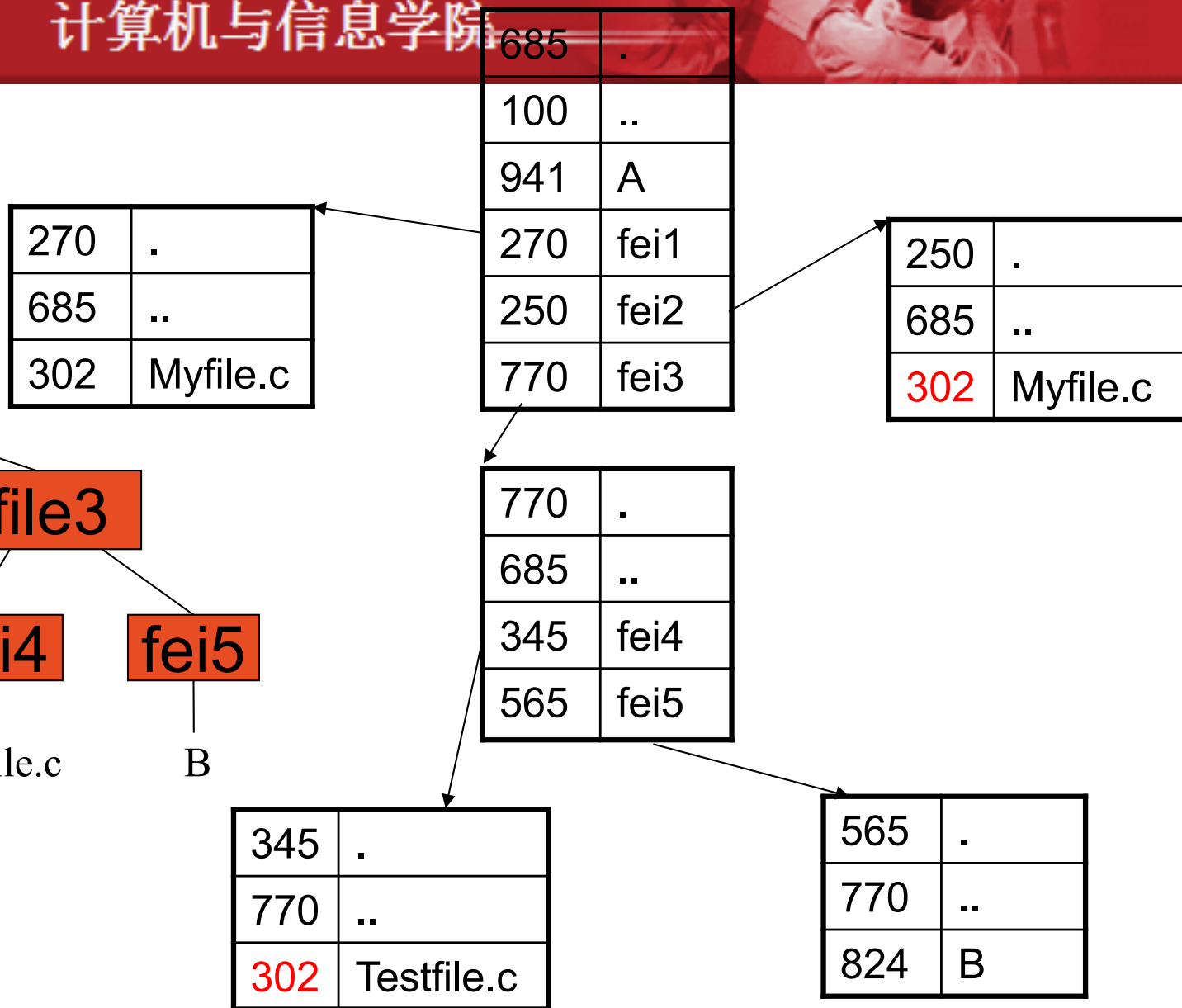


6.2.3 文件目录的检索

- 每个目录创建时都含有两个特殊的目录项
- “.” 项指出目录本身的inode入口
- “..” 项指出其父目录的inode入口
- 如果“.”和“..”都指向同一个inode，则为系统根目录



(a) 用户角度的目录结构



(b) 系统角度的目录链接



6.2.3（续）

- 执行打开文件时，根据用户所提供的文件名，从根目录开始
- 逐级查找路径名中的各子目录名，用其作索引，逐层搜索各级目录文件，最后找到相匹配的文件目录项。



河海大学

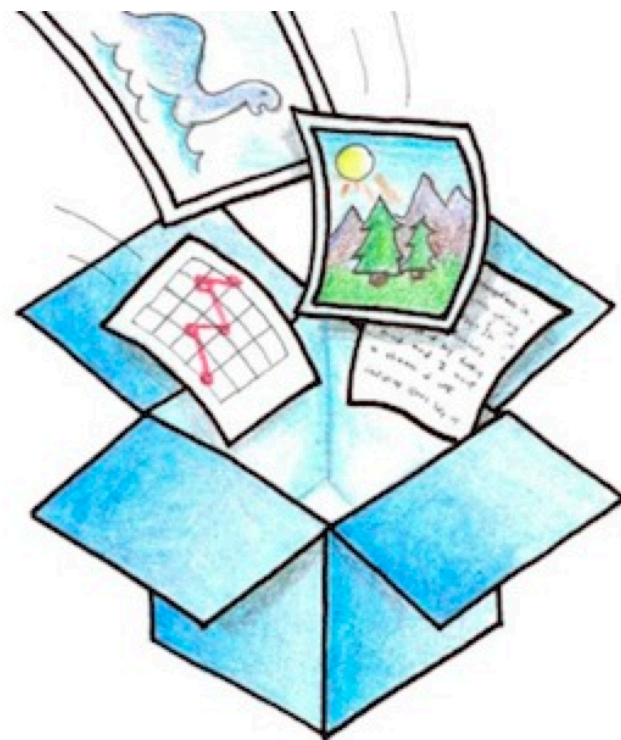
计算机与信息学院

6.3 文件组织和数据存储

- 文件的存储

- 文件的逻辑结构

- 文件的物理结构





文件的存储

文件的逻辑结构

文件的物理结构



6.3.1 文件的存储

- 一盘磁带、一张光盘片、一个硬盘分区或一张软盘片都称为一卷，**卷**是存储介质的物理单位
- 一个卷可以保存一个文件或多个文件，也可以一个文件保存在多个卷上



6.3.1 文件的存储

- **块**是存储介质上连续信息所组成的一个区域，也叫做物理记录。块是主存储器和辅助存储设备进行信息交换的物理单位，每次总是交换一块或整数块信息。不同类型的存储介质，块的长短常常各不相同；同一类型的存储介质，块的长短也可以不同



文件的存储

文件的逻辑结构

文件的物理结构



6.3.2 (1) : 流式和记录式文件

- 文件组织指文件中信息的配置和构造方式，应该从文件的逻辑结构和组织及文件的物理结构和组织两方面考虑
- 文件的逻辑结构和组织是从用户观点出发，研究用户概念中的信息组织方式，这是用户能观察到，可加以处理的数据集合



6.3.2 (1)

- **流式文件**：指文件内的数据不再组成记录，只是依次的一串信息集合，可以看成是只有一个记录的记录式文件
- **记录式文件**：有结构的文件，包含若干逻辑记录。逻辑记录是文件中按信息在逻辑上的独立含意划分的信息单位

大多数现代操作系统对用户仅提供流式文件, 记录式文件往往由高级语言或数据库管理系统提供



6.3.2 (2) : 成组和分解

- 逻辑记录 vs. 块
- 逻辑记录是按信息在逻辑上的独立含义划分的单位，块是存储介质上连续信息所组成的区域
- 一个逻辑记录被存放到文件存储器的存储介质上时，可能占用一块或多块，也可以一个物理块包含多个逻辑记录

文件比作书，逻辑记录比作书中的章节，那么，卷是册而块是页

书和章节相当于文件和逻辑记录，是逻辑概念；而册和页相当于卷和块，是物理概念



6.3.2 (2)

- 若干个记录合并成一组, 写入一个块叫做**记录成组**, 这时每块中的逻辑记录的个数称块因子
- 当存储介质上的一个物理记录读进输入缓冲区后, 把逻辑记录从块中分离出来的操作叫**记录分解**



6.3.2 (3) : 记录格式

- 记录格式就是记录内数据的排列方式，包括：
 - **定长记录**：记录文件中所有的逻辑记录具有相同的长度
 - **变长记录**：记录文件中的逻辑记录长度不相等，但每条逻辑记录的长度处理之前可以预先确定
 - **跨块记录**：当所处理的变长记录大于块长时，会发生逻辑记录跨越物理块的情形



6.3.2 (4) : 记录键

- 对每条逻辑记录至少指定一个与其对应的基本数据项，该逻辑记录的数据项就是记录键，也称关键字，简称键
- 同一个文件中，能唯一标识某条逻辑记录的记录键称为主键



河海大学

计算机与信息学院

文件的存储

文件的逻辑结构

文件的物理结构



6.3.3 文件的物理结构

- 文件和数据逻辑结构是面向用户和应用程序的
- 文件的物理结构和组织是指逻辑文件在物理存储空间中存放方法和组织关系。这时文件看作为物理文件,即相关物理块的集合
- 文件的存储结构涉及: 块的划分、记录的排列、索引的组织、信息的搜索, 其优劣直接影响文件系统的性能



6.3.3: 物理文件构造方法

- 第一类计算法，设计映射算法，通过对记录键的计算转换成对应的物理块地址，找到所需记录。直接寻址文件、计算寻址文件、顺序文件均属此类
- 第二类指针法，设置专门指针，指明相应记录的物理地址或表达各记录之间的关联。索引文件、索引顺序文件、连接文件、倒排文件等均属此类



6.3.3 (1) : 顺序文件

- 一个文件中逻辑上连续的信息存放到存储介质的依次相邻的块上便形成顺序文件(连续文件)
- 顺序文件是逻辑记录顺序和物理记录顺序完全一致的文件, 通常, 记录按出现的次序被读出或修改
- 一切存于磁带上的文件都只能是顺序文件, 卡片机、打印机、纸带机上的文件也属于顺序文件。存储在磁盘上的文件也可以组织成顺序文件。顺序文件可由文件目录指出存放该文件信息的第一块存储地址和文件长度



6.3.3 (1) : 顺序文件

- 顺序文件的缺点：
 - 建立文件前需要预知文件长度，修改、插入和增加文件记录有困难，对直接存储器作连续分配会造成空闲块的浪费
- 顺序文件的优点：
 - 顺序存取记录时速度较块



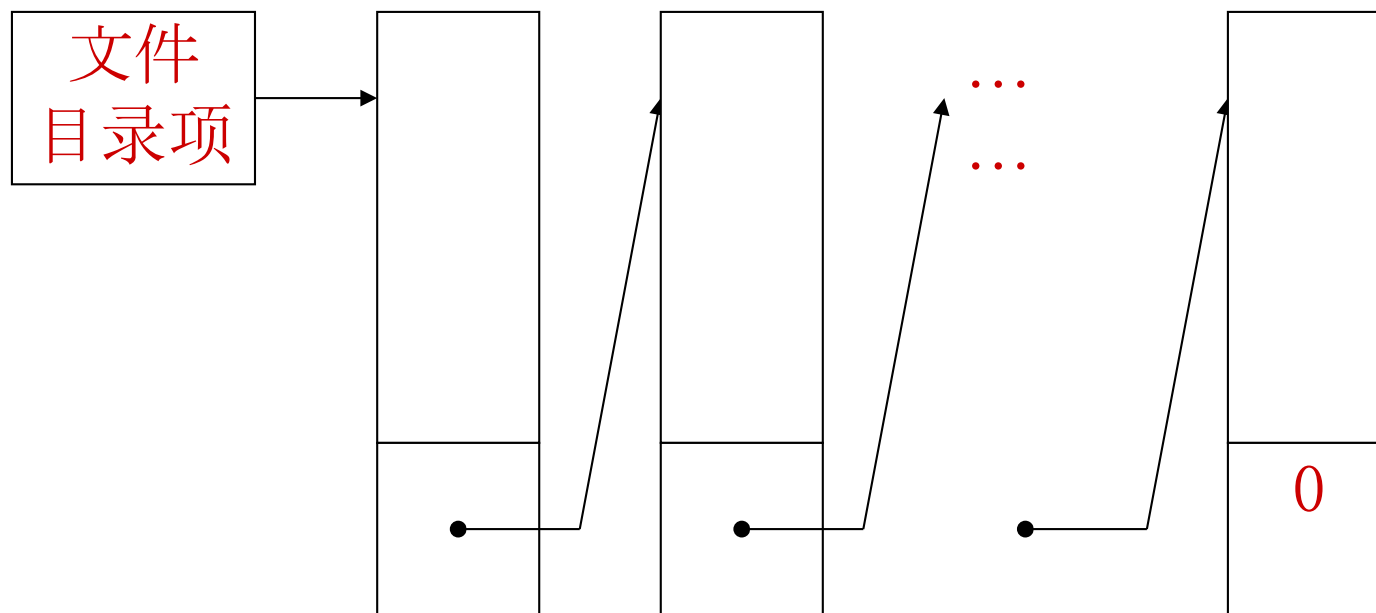
6.3.3 (1) : 顺序文件变种

- 紧凑顺序文件
 - 逻辑记录连续地存储在存储介质的相邻物理块上的文件，插入和删除要移动大量记录
- 扩展顺序文件
 - 在文件内设有空白区域以备预先估计的添加记录使用。
- 连接顺序文件
 - 文件中设置溢出区，添加的记录通过连接方法保存到溢出区中



6.3.3 (2) : 连接文件

- 连接文件使用连接字，又叫指针来表示文件中各个记录之间的关系。第一块文件信息的物理地址由文件目录给出，每一块的连接字指出文件下一个物理块位置





6.3.3 (2) : 连接文件

- 连接文件的优点：不要求物理块连续, 便于增、删、改
- 连接文件的缺点：
 - 指针与数据信息混合存放, 破坏数据完整性
 - 获得指针后, 才能获得下一物理块地址, 仅适宜于顺序存取



6.3.3 (2) : 连接文件

- OS/2 的连接文件变种：
 - 指针块（可置入高速缓存）与数据块分离存储
 - 指针块存储于磁盘首部
 - $PTRS[i] = j$



6.3.3 (3) : 直接文件

- 记录的关键字与其地址间可通过某种方式建立对应关系，利用这种关系实现存取的文件叫直接文件
- “冲突”问题：地址总数和可选择的关键字间不存在一一对应的关系。 (Bloom Filter)



6.3.3 (4) : 索引文件

- 索引结构是实现非连续存储的另一种方法，适用于数据记录保存在随机存取存储设备上的文件。系统为每个文件建立一张索引表，每个表目包含一个记录的键（或逻辑记录号）及其记录数据的存储地址，这类文件称索引文件
- 索引文件在文件存储器上分两个区：索引区和数据区



6.3.3 (4) : 索引文件

- 访问索引文件需两步操作：
 - 第一步查找文件索引
 - 第二步以相应键登记项内容作为地址而获得记录数据



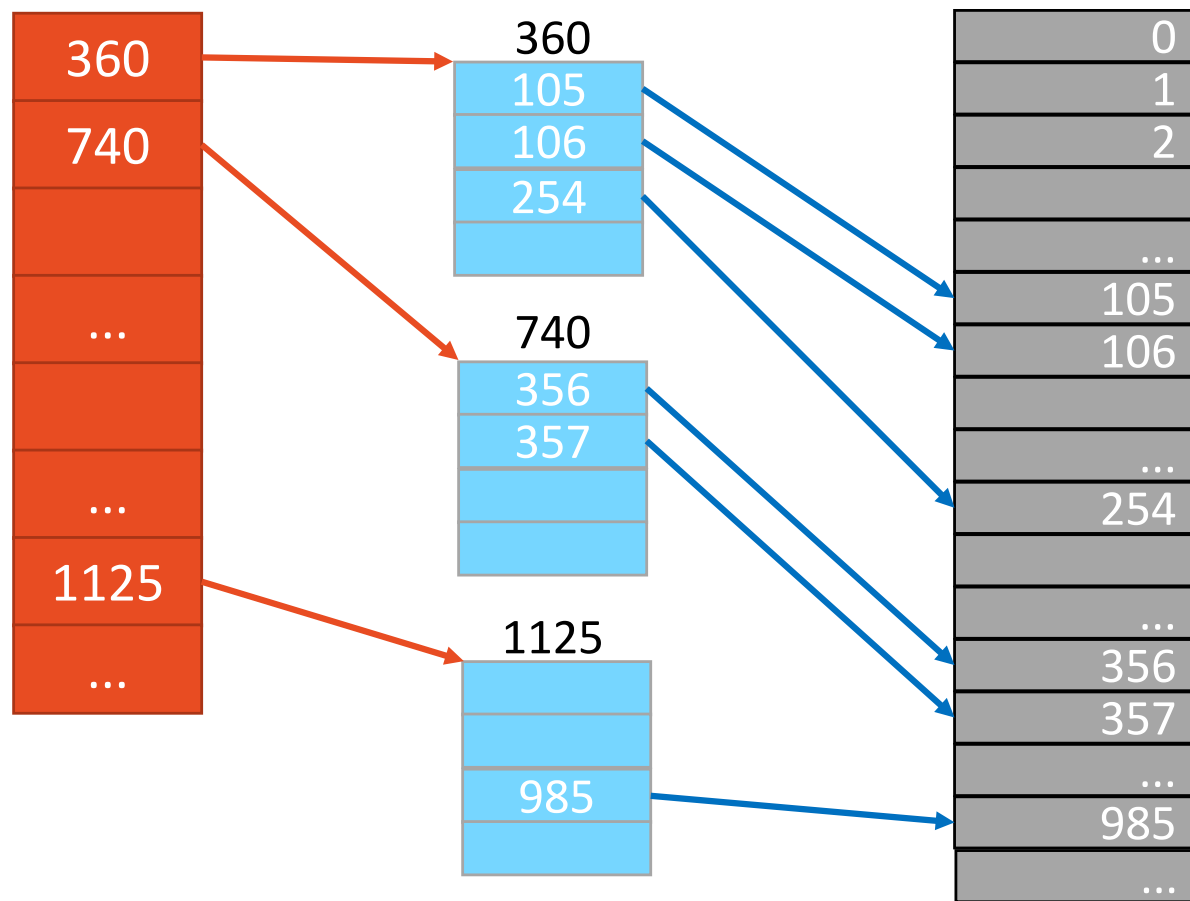
6.3.3 (4) : 索引文件

- 索引文件的优点
 - 不要求物理块连续, 便于直接存取, 便于文件的增、删、改
- 索引文件的缺点
 - 增加了索引表的空间开销和查找时间



河海大学

计算机与信息学院



多重索引结构



6.3.3: UNIX混合索引分配

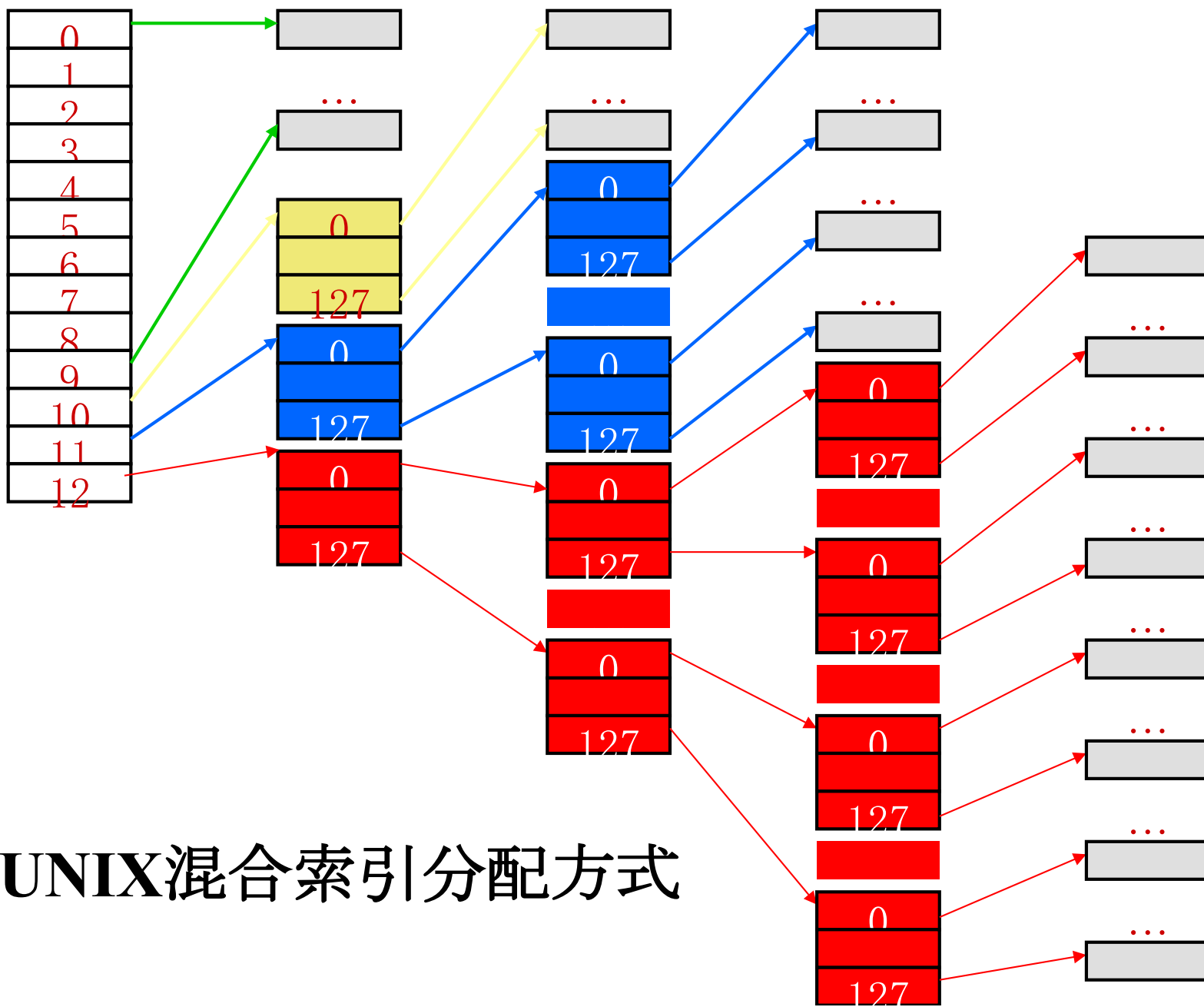
- 在UNIX系统中, 还采用了混合索引分配方式, 可以同时采用一级、二级、三级索引分配结构, 以满足不同大小文件的需要
- 每个文件的索引表为15个索引项, 每项4个字节, 登记一个存放文件信息的物理块号。最前面12项直接登记存放文件信息的物理块号, 叫直接寻址
- 如果文件大于12块, 则利用第13项指向一个物理块, 该块中最多可放256个文件物理块的块号, 叫做一次间接寻址。对于更大的文件还可利用第14和第15项作为二次和三次间接寻址, 每块大小为1KB
- UNIX中采用了三级索引结构后, 文件最大可达:

$$12KB + 256KB + 256^2KB + 256^3KB \approx 16GB$$



河海大学

计算机与信息学院



UNIX混合索引分配方式



河海大学

计算机与信息学院

6.4 文件系统其它功能实现

文件系统调用的实现

UNIX文件系统调用

文件卷的安装和使用

文件共享

辅存空间管理

内存映射文件

虚拟文件系统





文件系统调用的实现

UNIX文件系统调用

文件卷的安装和使用

文件共享

辅存空间管理

内存映射文件

虚拟文件系统



6.4.1 文件系统调用的实现

- 文件系统提供给用户程序的一组系统调用，包括：**建立、打开、关闭、撤销、读、写和控制**，通过这些系统调用用户能获得文件系统的各种服务
- 为了避免重复访问外存查找文件，**系统把常用和正在使用的那些文件目录复制进主存**。系统为每个用户进程建立一张活动文件表，用户使用一个文件之前先通过“打开”操作，把该文件的文件目录复制到指定主存区域。当不再使用该文件时，使用“关闭”操作切断用户进程和该文件目录的联系。同时，若该目录已被修改过，则应更新辅存中对应的文件目录



6.4.1 (1) : 建立文件

- 用户把信息作为一个文件存放到存储设备上时，使用建立操作向系统提出建立一个文件的要求
- 用户使用‘建立’系统调用时，通常应提供以下参数：
文件名、设备类(号)、文件属性及存取控制信息（如文件类型、记录大小、保护级别等）



6.4.1 (1) : 建立文件

- 建立系统调用的主要工作

1. 根据设备类（号）在相应设备上建立一个文件i-node，返回用户文件标识
2. 将文件名及属性等数据填入文件i-node
3. 调用辅存空间管理程序，为文件分配第一个物理块
4. 需要时发出装卷信息（如磁带）
5. 在活动文件表中登记该文件有关信息，文件定位，卷标处理



6.4.1 (2) : 打开文件

- 使用已建立的文件前，通过‘打开’文件操作建立文件和用户间的联系
- 用户打开文件时需要给出文件名和设备类（号）
- “打开”系统调用的主要工作



6.4.1 (2) : 打开文件

- “打开”系统调用的主要工作
 1. 在活动文件表中申请空项，存放该文件的文件目录信息
 2. 根据文件名查找目录文件，找到的文件目录信息复制到活动文件表占用栏
 3. 若打开的是共享文件，应有相应处理，共享文件用户数加 1
 4. 文件定位，卷标处理



6.4.1 (3) : 读/写文件

- 文件打开后，可用读 / 写系统调用访问文件，应给出参数：文件名、主存缓冲地址、读写的字节个数等
- 读 / 写系统调用的主要工作
 1. 按文件名从活动文件表中找到该文件的目录
 2. 按存取控制说明检查访问合法性
 3. 根据文件目录指出的该文件的逻辑和物理组织方式将逻辑记录号或字符个数转换成物理块号
 4. 向设备管理发I/O请求，完成数据传输操作



6.4.1 (4) : 关闭文件

- 文件使用完毕应关闭，以便让别的使用者用此文件。
- 此操作的主要工作
 1. 将活动文件表中该文件的‘当前使用用户’减 1；若此值为0，则撤销此表目
 2. 若活动文件表目内容已被改过，应先将表目内容写回文件存储器上相应表目中，以使文件目录保持最新状态



6.4.1 (5) : 删除文件

- 一个文件不再需要时，可向系统提出撤销文件。
- 系统的主要工作是：
 1. 若文件没有关闭，先做关闭工作；若为共享文件，应进行联访处理
 2. 在目录文件中删去相应目录项
 3. 释放文件占用的文件存储空间



河海大学

计算机与信息学院

文件系统调用的实现

UNIX文件系统调用

文件卷的安装和使用

文件共享

辅存空间管理

内存映射文件

虚拟文件系统



6.4.2: UNIX文件系统调用

- UNIX文件系统调用主要包括：打开和关闭，文件创建和删除，文件连接和解除连接，文件读和写，以及文件随机访问
- UNIX文件系统数据结构及存储器上的组织安排
 - 安装在磁盘驱动器上的一个盘组被划分成几个逻辑盘，每个构成一个文件卷（物理卷），它又被分成若干柱面、磁道和扇区。每个扇区512个字节（又称一个物理块）。系统对物理块从0开始统一编号。0#块用于系统引导过程；1#块为专用块，用于记录文件系统管理用的信息；2#~k+1#共k块，用于存放i节点（i-node）表；k+2#~n#存放文件信息，n+1#后的若干块作为交换区



6.4.2: UNIX文件系统调用

- 1# 专用块

- 包括：i节点表所占盘块数、文件卷盘块数、内存中登记的空闲盘块数(最多100)、内存中登记的空闲块的物理块号、内存中登记的空闲i节点数(最多100)、内存中登记的空闲i节点编号，及其他管理信息
- 既有盘位示图的功能，又记录了整个文件卷的控制数据。每当一个块设备作为文件卷被安装时，该设备的专用块就要复制到内存专用块中备用，而拆卸文件卷时，修改过的专用块需复制回设备的专用块中



6.4.2: UNIX文件系统调用

- $2^k \sim 2^{k+1}$ 的 i 节点 (i-node) 表
- 存放UNIX文件控制块信息，又分外存文件i节点表和内存活动i节点表，后者解决了频繁访问i节点表的效率问题



6.4.2: UNIX文件系统调用函数

- 1. 文件的创建
- 系统调用C语言格式为:

```
int fd, mode;
```

```
char *filenamep;
```

```
fd = creat(filenamep, mode);
```



6.4.2: UNIX文件系统调用函数

- 1. 文件的创建

- 系统调用C语言格式为

```
int fd, mode;
```

```
char *filenamep
```

```
fd = creat(filenamep, mode);
```

① 为新文件分配索引节点和活动索引节点，并把索引节点编号与文件分量名组成新目录项，记到目录中

② 在新文件所对应的活动索引节点中置初值，包括置存取权限*i_mode*，连接计数*i_nlink*等

③ 分配用户打开文件表项和系统打开文件表项，置表项初值。包括在*f_flag*中置“写”标志，读写位移*f_offset*清“0”。把各表项及文件对应的活动索引节点用指针连接起来，把文件描述字返回给调用者



6.4.2: UNIX文件系统调用函数

- 2. 文件的删除
- 删除把指定文件从所在的目录文件中除去。如果没有连接用户(`i_link` 为 “1”), 还要把文件占用的存储空间释放。删除系统调用形式为: `unlink (filenamep)`。在执行删除时, 必须要求用户对该文件具有 “写” 操作权



6.4.2: UNIX文件系统调用函数

- 3. 文件的打开
- 调用方式为:

```
int fd, mode;
```

```
char * filenamep;
```

```
fd = open (filenamep, mode);
```




6.4.2: UNIX文件系统调用函数

- 3. 文件的打开

- 调用方式为:

```
int fd, mode;
```

```
char * filenamep;
```

```
fd = open (filena
```

① 检索目录，把它的外存索引节点复制到活动索引节点表

②根据参数mode核对权限，如果非法，则这次打开失败

③ 当“打开”合法时，为文件分配用户打开文件表项和系统打开文件表项，并为表项设置初值。通过指针建立这些表项与活动索引节点间的联系。把文件描述字，即用户打开文件表中相应文件表项的序号返回给调用者



6. 4. 2: UNIX文件系统调用函数

- 4. 文件的关闭
- 调用方式为:

```
int fd;  
close (fd);
```



6.4.2: UNIX文件系统调用函数

- 4. 文件的关闭
- 调用方式为:

```
int fd;  
close (fd);
```

- ① 根据fd找到用户打开文件表项，继而找到系统打开文件表项。释放用户打开文件表项
- ② 把对应的系统打开文件表项中的f_count减“1”，如果不为“0”，说明还有进程共享这一表项，不用释放表项直接返回；否则释放这个表项，并找到与之连接的活动索引节点
- ③ 把活动索引节点中的i_count减“1”，若不为“0”，表明还有用户进程正在使用该文件，不用释放该而直接返回，否则在把该活动索引节点中的内容复制回文件卷上的相应索引节点中后，释放该活动索引节点



6.4.2: UNIX文件系统调用函数

- 5. 读文件
- 调用的形式为:

```
int nr, fd, count;
```

```
char buf [ ]
```

```
nr = read (fd, buf, count);
```



6.4.2: UNIX文件系统调用函数

- 5. 读文件

- 调用的形式为:

```
int nr, fd, count;
```

```
char buf [  ]
```

```
nr = read (fd, buf, count);
```

系统根据f_flag中的信息，检查读操作合法性，再根据当前位移量f_offset值，要求读出的字节数，及活动索引节点中i_addr指出的文件物理块存放地址，把相应的物理块读到缓冲区中，然后再送到bufp指向的用户内存区中。



6.4.2: UNIX文件系统调用函数

- 6. 写文件
- 调用的形式为:

`nw = write (fd, buf, count)`

buf是信息传送的源地址，即把buf所指向的用户内存区中的信息，写入到文件中



6.4.2: UNIX文件系统调用函数

- 6. 写文件
- 调用的形式为:

`nw = write (fd, buf, count)`

buf是信息传送的源地址，即把buf所指向的用户内存区中的信息，写入到文件中



6.4.2: UNIX文件系统调用函数

- 7. 文件的随机存取
- 系统调用的形式为:

`long lseek;`

`long offset;`

`int whence, fd;`

`lseek (fd, offset, whence);`

- 文件描述字fd必须指向一个用读或写方式打开的文件，当whence是“0”时，则f_offset被置为offset，当whence是“1”时，则f_offset被置为文件当前位置加上offset



文件系统调用的实现

UNIX文件系统调用

文件卷的安装和使用

文件共享

辅存空间管理

内存映射文件

虚拟文件系统



6.4.3 文件卷的安装和使用

- 文件卷又称为文件子系统, 在该卷的空间中存放文件和目录信息, 也存放文件属性、空闲区域信息。一张格式化后的软盘就是一个卷; 硬盘上每个格式化了分区也是一个卷。不同卷上允许安装不同操作系统
- 在windows/DOS操作系统下, 一个盘或分区, 在物理安装和格式化后, 就可以直接使用。使用之前用户不需要运行文件卷安装操作。但是UNIX/Linux每个文件卷需要经过安装后才能使用。 UNIX的文件系统可分成基本文件系统和可装卸的子文件系统(文件卷)两部分



6.4.3（续）

- （1）基本文件系统
- 基本文件系统固定在根存储设备上，通常把硬盘作为根存储设备，系统一旦启动运行，基本文件系统就不能卸装
- （2）可装卸的子文件系统
- 存储在可装卸存储介质（如软盘）上的文件系统为可装卸的子文件系统，它可以随时更换。每个用户都可以将软盘插入软盘驱动器，然后通过系统调用命令将其与基本文件系统安装在一起，也可以使用系统调用命令使子文件系统与基本文件系统卸装



6.4.3（续）

- 可装卸的文件卷可动态地安装到一个已存在文件卷的某个节点上。通常这个节点是为文件卷安装而创建的一个空目录。文件卷的安装如同树的嫁接一样，使安装的文件卷目录树结构与原有的文件卷树形结构融为一体



河海大学

计算机与信息学院

文件系统调用的实现

UNIX文件系统调用

文件卷的安装和使用

文件共享

辅存空间管理

内存映射文件

虚拟文件系统



6.4.4 文件共享

- 文件共享指不同用户（进程）共同使用同一个文件。
- 在UNIX/Linux 系统中，文件共享形式有：
 1. 静态共享
 2. 动态共享



6.4.4 文件共享

- 1. 文件的静态共享
- 静态共享允许一个文件同时属于多个目录，但实际上文件仅有一处物理存储，这种文件在物理上一处存储，从多个目录可到达该文件的结构称为文件链接。
- 要实现文件的静态链接，只要不同目录的索引结点 i-node 号，指定为同一文件的索引结点即可



6.4.4 文件共享

- 2. 文件的动态共享
- 文件动态共享是系统中不同的用户进程或同一用户的不同进程并发地访问同一文件
- 这种共享关系只有当用户进程存在时才可能出现，一旦用户的进程消亡，其共享关系也就自动消失
- 当两个程序共享同一文件但读写指针不同时，这两个程序各自的打开文件表指向系统打开文件表中的不同表项，而这两个表项又指向内存活动i-node表中的同一表项。当两个程序共享同一文件且共用读写指针时，这两个程序的打开文件表指向系统打开文件表中的同一表项



6.4.4 文件共享

- 两个以上用户共享文件，每个希望独立地读、写文件，这时不能只设置一个读写位移指针，须为每个用户进程分别设置一个读、写位移指针
- 位移指针应放在每个进程的**用户打开文件表**的表目中。这样，当一个进程读、写文件，并修改位移指针时，另一个进程的位移指针不会随之改变，从而，使两个进程能独立地访问同一文件



6.4.4 文件共享

- 3. 文件的符号链接共享
- 又称软链接，是一个普通文件，有着自己的 inode 号以及用户数据块，只是数据块内容有点特殊，存放的内容是另一文件的路径名的指向



6.4.4 文件共享

- 3. 文件的符号链接共享
 - 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 `i_nlink` 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（即 `dangling link`，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）。



Unix 文件与链接

- `$ mkdir link`
- `$ echo "hello" > file`
- `$ ln file hardlink`
- `$ ln -s file softlink`
- `$ stat file hardlink softlink`



Unix 文件与链接

- `$ rm file`
- `$ cat softlink`
- `$ echo "hello2" > file`
- `$ cat softlink`
- `$ cat hardlink`
- `$ stat hardlink`
- `$ stat file`



河海大学

计算机与信息学院

文件系统调用的实现

UNIX文件系统调用

文件卷的安装和使用

文件共享

辅存空间管理

内存映射文件

虚拟文件系统



6.4.5 辅存空间管理

- 辅存空间分配采用两种办法
- 连续分配：文件存放在辅存空间连续存储区中，在建立文件时，用户必须给出文件大小，然后，查找到能满足的连续存储区供使用
- 非连续分配：
 - 一种方法是以块（扇区）为单位，扇区不一定要连续，同一文件的扇区按文件记录的逻辑次序用链指针连接或位示图指示
 - 另一种方法是以簇为单位，**簇是由若干个连续扇区组成的分配单位**；实质上是连续分配和非连续分配的结合。各个簇可以用链指针、索引表，位示图来管理



河海大学

计算机与信息学院

文件系统调用的实现

UNIX文件系统调用

文件卷的安装和使用

文件共享

辅存空间管理

内存映射文件

虚拟文件系统



6.4.6 内存映射文件

- 内存映射文件允许进程分配一段虚地址空间, 然后把某个盘文件映射到该地址空间中, 也即把磁盘中的文件视为进程的虚拟内存的一部分
- 内存映射文件功能是由操作系统的I/O子系统和存储管理子系统共同实现的. 每当一个应用程序访问它的虚拟地址空间的存储区域时, 存储管理子系统利用它的页面调度机制从磁盘文件中加载正确的页面. 如果应用程序向它的虚拟地址空间写入数据时, 存储管理子系统就把更改作为正常页面调度的一部分写回到文件中



河海大学

计算机与信息学院

文件系统调用的实现

UNIX文件系统调用

文件卷的安装和使用

文件共享

辅存空间管理

内存映射文件

虚拟文件系统



6.4.7 虚拟文件系统

- 虚拟文件系统要实现以下目标
- 同时支持多种文件系统
- 多个文件系统应与传统的单一文件系统没有区别，在用户面前表现为一致的接口
- 提供通过网络共享文件的支持，访问远程结点上的文件系统应与访问本地结点的文件系统一致
- 可以开发出新的文件系统，以模块方式加入到操作系统中



6.4.7 虚拟文件系统

- 虚拟文件系统（VFS）是物理文件系统与服务之间的一个接口层，它对Linux的每个文件系统的所有细节进行抽象，使得不同的文件系统在Linux核心以及系统中运行的其他进程看来，都是相同的。严格说来，VFS并不是一种实际的文件系统。它只存在于内存中，不存在于任何外存空间。VFS在系统启动时建立，在系统关闭时消亡
- VFS使Linux同时安装、支持许多不同类型的文件系统成为可能。VFS拥有关于各种特殊文件系统的公共界面，如超级块、inode、文件操作函数入口等。实际文件系统的细节，统一由VFS的公共界面来索引，它们对系统核心和用户进程来说是透明的



6.4.7 虚拟文件系统

- 虚拟文件系统设计思想
- (1) 在对多个文件系统的共同特性进行抽象基础上，形成一个与具体文件系统实现无关的虚拟层，并在此层次上定义与用户的一致性接口
- (2) 文件系统具体实现层使用类似开关表技术进行文件系统转接，每个文件系统是自包含的，实现各文件系统的具体细节



河海大学

计算机与信息学院

- 文件系统是为谁准备的?
- 是否一定需要文件?
- 信息 \neq 文件



- 作业： 3, 11