

文件管理

文件概览

文件是存放于外存的、是文件系统的最大单位文件，是只有创建者所定义的、具有文件名的一组相关元素的组合。现代OS通过文件系统来组织管理计算机外存存储的程序和数据。

目录=文件夹，用于表示系统中的文件及其物理地址的数据结构，对文件进行有效管理，供以检索使用。

文件的术语

- 纪录：表述对象某方面的属性
- 关键字：唯一标识记录的数据项
- 数据项：描述一个对象某种属性的字符集。最低级的数据组织形式

文件的分类

- 按文件性质和用途分类
 - 系统文件：OS软件构成的文件
 - 用户文件：用户的源代码、目标代码、可执行文件或数据结构等构成的文件
 - 库文件：由标准子程序以及日常程序等构成的文件
- 按按文件数据形式分类
 - 源文件：用户的源程序和数据等构成的文件
 - 目标文件：用户源程序经汇编或编译后的文件
 - 可执行文件：目标代码经链接后生成的文件
- 按文件存取属性分类
 - 只执行文件：进允许核准用户调用执行的文件
 - 制度文件：进允许文件主以及核准用户读取的我呢间
 - 读写文件：进允许文件主以及核准用户读写的文件
- 按文件组织形式分类
 - 普通文件：ASCII码或二进制的文件
 - 目录文件：管理文件、实现文件系统功能的系统文件
 - 特殊文件：IO设备文件

文件的属性和操作

属性：类型、长度、物理位置、创立时间。

操作：创建、删除、读取、写入、设置文件读写位置、截断文件；

向上提供的最基本功能（调用系统）：创建create、删除delete、读read、写write、打开open、关闭close。

文件系统模型

用户（程序） -->> | **文件系统接口** | **对对象操纵和管理的软件集合** | **对象及其属性** |

1. 文件管理系统的管理对象：文件、目录、磁盘存储空间
2. 对对象操纵和管理软件的集合：对文件存储空间管理、对文件目录的管理、将文件的逻辑地址转为物理地址的基址、对文件的读写管理、对文件的共享与保护

3. 文件系统接口：命令接口、程序接口

对文件的管理

对文件的操作、逻辑结构、共享、保护、分配方式、目录，对磁盘的组织与管理

文件打开与关闭

打开：OS调用open，将指名文件属性（包括文件在外存上的物理位置）从外存靠背到内存打开文件表的一个表目中，并将该表目的编号（索引）返回给用户。

关闭：OS调用close，将指名文件关闭并把该文件从打开文件表中的表目上删除。

文件的逻辑结构

文件是由创建者所定义的、具有文件名的一组相关元素的组合，可分为无结构文件和有结构文件（顺序文件、记录寻址、索引文件、索引顺序文件、直接文件和哈希文件）。

无结构文件被大量的源程序、可执行文件、库函数所构成（流式文件，以字节为单位）。对流式文件的访问，采用读写指针指出下一个要访问的字符。

有结构文件被大量的数据结构和数据库所采用。

- 顺序文件：有一系列定长或不定长的纪录按某种顺序排列成的文件
 - 串结构：记录按照存入时间排列
 - 顺序结构：记录按关键字有序排列
 - 特点：某个记录的逻辑位置可由第1个记录的逻辑位置推算；查找和存取效率高，但增删改查单个记录难度大。
- 索引文件：问文件创建索引，索引表本身是定长记录的顺序文件，可以快速找到第i个记录对应索引项
 - 可用关键字作为索引号内容（此时还支持折半查找）。当要增删一个记录时，需要对索引表修改
- 索引顺序文件：分为若干组的文件记录，每组第一个记录在索引表中占据一个表项，最常见的逻辑文件形式
 - 索引表：一个顺序文件对应一个表
 - 含N个记录的文件，顺序文件平均查找长度为 $N/2$ ，一级索引顺序文件查找长度 \sqrt{N}
- 直接文件：根据给定记录键值直接获得指定记录的物理地址
- 哈希文件：根据键值由哈希查找来定位目录表中的位置

目录管理

目录/文件夹 是用于标识OS中的文件以及其物理地址的数据结构，是对文件进行有效管理的方法，供检索使用。目录之下是目录项。

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

一个文件控制块FCB就是一个文件目录，对应一个文件夹。目录就是FCB的有序集合。FCB包含的信息：

1. 基本信息类：文件名，文件物理位置（包括存放文件的设备名、在外存上的起始盘块号、文件占用的盘块数or字节数），文件逻辑结构，文件物理结构
2. 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

3. 使用信息类：文件建立日期和时间、上一次修改的日期和时间、当前使用信息（包括已打开该文件的进程数、是否被其它进程锁住）

文件目录的组织方式

目录管理包括有：单级文件目录，两级文件目录，树形结构目录

- **单级文件目录结构**：只建立一个目录表，每个文件占一个目录项（相当于只建一个文件夹）
 - 简单但检索慢，文件不许重名，不便于共享，适合单用户环境
- **两级文件目录**：
 - 每个用户对应一个单独用户文件目录UFD，由该用户所有文件的文件控制块组成
 - 系统建立一个主文件目录MFD，每个用户目录文件占有一个目录项，用用户名和指向用户目录文件的指针
 - 检索速度较快，不同用户可重名、可用不同名访问同一个共享文件，系统开销大。但缺乏灵活性，无法实现文件分类。不同用户可使用不同文件名来访问系统重的同一个共享文件。
- **树形结构目录**：现代操作系统中最通用且实用的文件目录结构
 - 根节点：主目录/根目录。叶子结点：数据文件。非根非终端节点：子目录
 - 从根目录到任何数据文件只有条唯一通路。对于一个文件，从根目录开始，把途径所有目录文件名与数据文件依次用“/”连接，形成路径名
 - 每访问一个文件都要使用从树根开始直到树叶为止的、包括个中间节点的全部路径名，效率较低
 - 可以把从当前目录开始直到数据文件为止的路径名，引入为一个相对路径名，简化查找层次，提高效率

目录操作

- 创建目录：用户可为自己建立UFD、创建子目录。创建新文件时，需检查所在目录有无重名文件
- 删除目录：
 - 空目录直接删
 - 非空目录：
 - 不删除非空目录，需要先删除目录中的所有文件使其成为空目录，再删除
 - 若目录中还包含子目录，采用递归调用的方式将其删除
 - 可删除非空目录：删除目录时包含目录中的所有文件和子目录也被同时删除
- 目录检索：
 1. 线性检索法：给定一个目录结构，依次逐级查找
 2. Hash检索法：引入一个Hash目录表，查找目录时查找对应的目录项。若在目录表的相应目录项中的文件名与指定文件名不匹配或发生冲突，此时将Hash值再加上一个常数，形成新的索引值，再返回第一步重新查找

寻找文件

给定一个路径：

用户视角是逐级打开目录访问文件；

OS视角是逐级指向目录，并在最后指向FCB（对应文件的文件控制块）

将目录所在盘块调入内存，逐查找文件名和目录中的文件名逐一比较：成功则将文件所在盘块调入内存/将信息写入文件所在盘块；失败则表示文件不存在

索引节点

FCB包含有：文件名，文件物理位置，文件逻辑结构，文件物理结构。

在检索信息时仅使用到了文件名，直到找到文件后，才从该目录项中读出其物理地址。于是可以把文件名与文件描述信息分开，使文件描述信息单独形成一个称为索引节点的数据结构，简称为i节点。

在文件目录中的每个目录项仅有文件名和指向该文件所对应i节点的指针构成。

磁盘索引节点：文件主标识符，文件类型，文件存取权限，文件物理地址，文件长度，文件连接计数，文件存取时间。

内存索引节点：文件被打开时，磁盘索引节点拷贝到内存中的索引节点，此时增加了：索引节点标号，状态，访问计数，逻辑设备号，链接指针。

文件的物理结构

文件在存储介质上的组织形式

- 磁盘块：外存中为了方便对文件数据管理，我们间逻辑地址空间被分为一个个文件“块”
- 内存块：内存管理中，页式管理方式将内存分为块大小
- 文件的逻辑地址可表示为(逻辑块号, 块内地址)的形式

文件分配方式：连续分配，文件分配表FAT，索引分配

连续组织方式

连续分配方式。每个文件存放在磁盘的一个磁道或同一柱面的一组相邻盘块。目录项的“文件物理地址”字段记录该文件第一个记录所在的盘块号和文件以盘块为单位的长度信息。

这样保证了逻辑顺序和存储顺序是一致的，记录该文件第一个记录所在的号 and 文件长度。这种盘块的相邻关系，体现了逻辑上的位置相邻。

顺序访问容易，顺序访问速度快。

但要求有连续的存储空间，必须事先知道文件长度。

链接组织方式

文件逻辑结构通过链接指针体现，逻辑上相邻的位置通过物理上的上下块之间的指针来体现相邻逻辑关系。

1. 隐式链接：在每个盘块里，存储了数据、指针（用于指向下一块）
 - 只适用于顺序访问，随机访问效率低。如果一个指针出现问题，整个链会断开
 - 可将同一磁道（柱面）上相邻几个盘块组成一个簇cluster，分盘块时以簇为单位
2. 显示链接：有两个存储结构：磁盘、FAT表
 - 每个表项中存储链接指针。FAT表和磁盘项数一样多
 - 凡是属于某一文件的盘块号，每一条链的链首指针对应的盘块号，均作为文件地址被填入相应文件的FCB的“物理地址”字段中
 - 检索速度快，访问磁盘次数小。但不支持高效的直接存取，需要占用较大内存空间

单级索引分配

每个文件分配一个索引块，存放分配给该文件的所有盘块的盘块号。所以表存放的磁盘块称为索引块，文件数据存放的磁盘称为数据块。

但当文件很大时，索引块会很大。读写速度相对不快。

两级索引结构

|主索引|第二级索引|磁盘空间|

第一级索引表存放索引表块号，第二级索引表登记的是磁盘空间的内容。

先将一级索引表调入内存，查询目标所在的二级索引，再由二级索引找到目标。访问目标数据块需要三次磁盘IO

混合索引结构

将直接结构和索引结构结合。

1. 直接地址：提高文件检索速度，在索引节点中设置10个直接地址项addr(11)。每项中存放该文件数据的盘块的盘块号。
2. 一次间接地址：再利用索引节点地址项来提供一次间接地址。实质是一级索引分配方式。系统将分配给我呢间的多个盘块号计入其中。在一次间址块存放k个盘块号，因此允许文件长达4MB
3. 多次间接地址：文件长度大于4MB+40KB时，还要采用二次间址分配。此时用addr(11)提供二次间址，实质是两级所索引分配方式，最大文件长度4GB ($4KB/4B * 4KB/4B * 4KB = 4GB$)。采用addr(12)作为三次间接地址，文件最大长度可达4TB。访问磁盘次数=间址级别+1

磁盘空闲空间管理

1. 空闲表法：把每一个块的空闲空间列入空闲表中
2. 空闲链表法：把空闲磁盘块拉成一个单链表（空闲盘块链，空闲盘区链）
3. 位示图：行列中的每个交叉点表示第一个比特，1表示被分配，0表示未分配
 1. 分配：
 1. 顺序扫描位示图，找出一个或一组为0的二进制位
 2. 将该二进制位转换成与之相应的盘块号。设其位于位示图的i行列，盘块号公式： $b = n(i-1)+j$ ，n表示每行位数
 3. 修改位示图， $map[i,j] = 1$
 2. 回收：
 1. 根据行号列号，转换公式为： $j = (b-1)/n + 1, j = (b-1)\%n + 1$
 2. 修改位示图， $map[i,j] = 0$

文件的共享与保护

现代OS必须提供文件共享手段，允许多用户/进程共享同一个文件，系统只保留该共享文件的一份副本。

基于索引节点的文件共享（硬链接方式）

给每个节点提出一个硬连接，在索引中设置一个count值，若count==x表示x个用户在共享该文件。

只要count>0就表示还有别的用户使用该文件，暂时不能把文件数据删除，否则导致指针悬空；当count==0系统负责删除该文件。

基于符号链接的文件共享

软链接，例如快捷方式。只有文件主才有只想索引节点的指针，其它用户只拥有一个指向该文件的链接。

文件保护

影响文件安全性的主要因素：人为因素，系统因素（系统异常造成数据损坏或丢失，或磁盘出现故障），自然因素（随着时间推移，硬盘消磁）

文件保护：口令保护，加密保护，访问控制。

保护域机制：进程只能在保护域内执行操作，且只能访问具有访问权限的对象。

在每个文件的FCB或索引节点中增加一个访问控制表ACL，记录了各用户可对该文件执行的操作。

磁盘结构

读写磁头在轴心上的磁盘扫描，磁盘分为各个扇区（扇形）和磁道（同心圆）。根据柱面号移动磁臂，让磁头指向指定柱面；集火指定盘面对应的磁头；磁盘旋转的过程中，指定扇区会从磁头下划过。

磁盘地址 = 柱面(磁道)号-盘面号-扇区号；

柱面号 = 簇号/每个柱面的簇数；

盘面号 = (簇号%每个柱面的簇数)/每个磁道的簇数；

扇区号 = 扇区地址%每个磁道的扇区数。

磁盘访问时间

- 寻道时间 $T_s = m \cdot n + s$
 - 磁头移动到指定磁道上的时间
- 旋转延迟时间 T_r
 - 指定扇区移到磁头下面的时间
- 传输时间 $T_t = b/(rN)$
 - 数据从磁盘读出或向磁盘写入数据所经历的时间
- 平均读取时间 = $1/r$
- 总时间 $T_a = T_s + 1/2r + b/rN$

对于7200r/min、平均寻道8ms、每个磁道1000个扇区的磁盘，访问一个扇区的平均存取时间：寻道时间8ms + 旋转延迟 $60 \cdot 1000 / 7200 \cdot (1/2)$ + 读写延迟 $60 \cdot 1000 / 7200 \cdot (1/1000)$

磁盘寻道算法

多个进程访问磁盘时，磁盘调度应使磁盘平均寻道时间最短。

- 先来先服务FCFS
 - 根据进程访问磁盘的先后次序进行进行调度。公平、简单，进程请求都能依次得到处理。适合请求磁盘IO进程数较少程序
- 最短寻道时间优先SSTF
 - 算法选择要求访问的磁道与当前磁头所在磁道距离最近的进程，使本次操作寻道时间最短。但可能导致某些进程发生饥饿
- SCAN算法（电梯调度算法）
 - 不仅考虑欲访问磁道与当前磁道的距离，更考虑磁头的当前移动方向。算法能获得较好的寻道性能，又能防止进程饥饿，广泛应用于大中小型机和网络中的磁盘调度。但对某些磁道不公平
- CSCAN循环扫描算法
 - 不仅考虑欲访问磁道与当前磁道的距离，更优先考虑磁头当前移动方向。规定只有磁头朝某个特定方向移动时才处理磁道访问请求，而返回时直接快速移动到起始端而不处理任何请求
- N-Step-SCAN算法
 - 上述算法容易“磁臂粘着”（某几个进程对某一磁道有较高访问频率）
 - N-Step-SCAN算法将磁盘请求队列分为若干长度为N的子队列，磁盘调度将按FCFS算法一次处理每个子序列，每处理一个队列时又是按照SCAN算法，对一个队列处理完后，再处理其它队列。如果处理过程中出现新的IO请求，则将其放入其它队列，避免粘着

- FSCAN算法
 - 将N-Step简化，只将磁盘请求队列分为两个子序列：由当前所有请求磁盘IO的进程形成的队列，和新出现的请求磁盘IO的进程队列

磁盘管理

- 提高磁盘IO速度的途径：
 - 磁盘高速缓存：内存中为磁盘设置缓冲区
 - 提前读：采用顺序访问时，可以预先将磁盘数据先读入高速缓存
 - 延迟写：应写回磁盘的缓冲数据先挂在空闲缓冲区队尾
 - 优化物理块分布：同一文件的盘块尽量处于同一磁道或相邻磁道
 - 虚拟盘：通过内存空间模拟磁盘
 - 廉价磁盘冗余阵列RAID：通过资源重复解决系统瓶颈问题
- 提高磁盘可靠性的途径
 - 磁盘容错技术：增加冗余磁盘设备
 - 后备数据：利用后备系统暂存不需要的还有用的数据
- 磁盘低级格式化
 - 将整个磁盘划分为柱面和磁道，再将磁道划分扇区（将整个磁盘重新划分各区域）。步骤如下：
 1. 会将扇区清零和重写校验值，并尝试修复坏道；
 2. 对扇区标识信息重写；
 3. 对扇区进行读写检查，并尝试替换缺陷扇区；
 4. 对所有物理扇区重新编号；
 5. 写磁道伺服信息；
 6. 写状态参数，并修改特定参数。
 - 一般分为快速格式化、高级格式化、低级格式化。低级格式化在特殊情况才使用
- 磁盘逻辑格式化（高级格式化）
 - 根据一定分区格式对磁盘进行标记，生成引导区信息、初始化空间分配表、标注逻辑坏道、校验数据等。只对磁盘进行寻常的读写操作。

文件系统挂载

磁盘格式化后，磁盘分区要能够使用，必须挂载。在挂载某个分区前要先建立一个挂载点。

虚拟文件系统VFS

用于支持大量的文件管理系统和文件结构。VFS向用户提供一个简单统一的文件系统接口，定义了一个能代表任何文件系统的通用特征和行为的通用文件模型。

VFS认为文件是计算机大容量存储器上的对象。文件可被创建、读写、删除，任何文件系统都要一个映射模块，以便将实际文件系统的特征转换为虚拟文件系统所期望的特征。

VFS等同于向用户隐藏内核空间的细节。

VFS是用C语言实现的面向对象的方案，每个对象都包含数据和函数指针。

包含的主要对象：超级块对象，索引节点对象，目录项对象，文件对象

- 超级块：一个超级块对应一个文件系统，保存了该文件系统的类型、大小、状态
- 索引节点inode：保存实际的数据信息（元数据）。创建一个文件时就给文件分配了一个inode。一个inode只对应一个实际文件，inode最大数量等同于文件数量
- 目录项：描述文件的逻辑属性，只存在于内存，相当于存在内存的目录项缓存

- 文件对象：描述进程已经打开的文件

固态硬盘SSD

以半导体闪存为介质的存储设备。主要部件为控制器和存储芯片，硬件包括主控、闪存、缓存（可选）、PCB、接口。

基本存储单元分为：SLC、MLC、TLC。

SLC的存储单元里存储1bit数据，高于4v表示0（已编程），低于4v表示1（已擦除）。

MLC存储2bit，分为00、01、10、11。与SLC采用相同电压，但阈值被分为四份。

TLC存储3bit。电压阈值分的更细致。

动态磨损：

主控会选择较新闪存颗粒进行擦除写入。算法简单粗暴，主控处理压力小，占据资源小。但精细化程度不够导致无法全面覆盖和实现所有颗粒的磨损均衡。

静态磨损：

执行擦除写入时，优先把长久不用的冷数据从较新的闪存颗粒提出并放入老颗粒，并将新写入的数据放在较新颗粒，实现均衡化。颗粒寿命高，但算法复杂、主控压力大。

补充

- 一个文件被用户进程首次打开的过程中，操作系统需要将文件控制块FCB读到内存中
- 文件被删除时，OS不可能执行的是删除此文件所在目录
- 索引节点个数与文件个数一一对应
- 索引节点所占位数可以反映最多文件个数（例如索引节点占2B，则有16位可以表示索引节点，得到最多65536个文件）
- 随机访问与动态扩张

分配方式	随机访问	动态扩张
连续分配	√	×
链接分配	×	√
索引分配	√	√

- F1的count=1，建立F1软链接文件F2，建立硬链接F3，删除F1后，F2、F3的引用计数值为1、1