

## 文件的相关概念

- 文件是以计算机硬盘为载体的存储在计算机上的信息集合，文件可以是文本文档、图片、程序等
- 系统运行时，计算机以进程为基本单位进行资源的调度和分配
- 在用户输入输出时，以文件为基本单位
- 相关定义
  - 操作系统的文件系统：用于实现文件的权限访问，修改，查询和保存等功能
- 文件的结构
  - 数据项：数据项是文件系统中最低级的数据组织形式
    - 基本数据项：用于描述一个对象的某种属性的一个值
    - 组合数据项：多个基本数据项组成
  - 记录：一组数据项的集合，用于描述一个对象在某方面的属性
  - 文件：创建者所定义的一组相关信息的集合
    - 有结构文件：相似的记录组成（记录式文件）
    - 无结构式文件：字符流组成（流式文件）
- 文件的属性
  - 名称：文件名称唯一，以容易读取的形式保存
  - 标识符：文件的唯一标签，通常为数字，是对人不可读的一种内部名称
  - 类型：被支持的不同类型的文件系统所使用
  - 位置：指向设备和设备上文件的指针
  - 大小：文件当前的大小，包含文件允许的最大值
  - 保护：对文件进行保护的访问控制信息
  - 时间、日期和用户标识：文件创建、修改和上次访问的相关信息，用于保护和跟踪文件的使用

## 文件的基本操作

- 创建文件
  - 文件系统为文件找到空间
  - 目录中为文件创建条目，该条目记录文件名称、在文件系统中的位置以及其他可能的信息
- 写文件
  - 执行系统调用，指明文件名称和写入内容，查找文件位置，为该文件维护一个写位置的指针，当发生写操作的时候更新写指针
- 读文件
  - 执行系统调用，指出文件名称和文件位置，搜索目录项，系统维护一个读指针，发生读操作就对该指针进行更新
- 文件重定位（文件寻址）：按照某种条件搜索目录，将当前文件位置设为定值。并且不会读、写文件
- 删除文件：搜索目录，找到文件的目录项，使其变为空项，然后回收目标文件占用的存储空间
- 截断文件：允许文件的所有属性不变，并删除文件内容，即将其长度设为0并释放其空间

## 文件的打开与关闭

- open请求
  - 首次使用文件，会调用open请求指明文件的属性（包括其物理位置）从外存复制到内存打开文件表的一个条目中，并将该条目的编号（索引）返回给用户
  - 操作open会根据文件名搜索目录，并将目录条目复制到打开文件
  - 调用open请求（创建、只读、读写、添加等）得到允许，进程就可以打开文件，open会返回一个指向打开文件表中的一个条目的指针
  - 通过使用该指针进行I/O操作，简化步骤并节省资源
- 文件关联信息
  - 文件指针：系统跟踪上次的读写位置作为当前文件位置的指针，这种指针对于打开文件的某个进程来说是唯一的，因此必须与磁盘文件属性分开保存
  - 文件打开计数：文件关闭时，必须重用其打开文件表条目，否则表内空间会不够用，计数器为0关闭文件，删除该条目
  - 文件磁盘位置：该信息存储在内存放，以免每个操作都要从磁盘中读取
  - 访问权限：每个进程打开文件都需要一个访问模式（创建、只读、读写、添加等）。该信息保存在进程打开的文件表中，以便操作系统能够允许或拒绝之后的I/O请求

## 文件的逻辑结构

- 最简单的文件组织形式
- 无结构文件
  - 将数据按照顺序组织记录并积累、保存、是有序相关信息项的集合
  - 由于其没有结构，所以只能采用更穷举搜索
  - 管理简单，方便用户对其操作
  - 基本信息单位操作不多的文件适合采用字符流的无结构方式
- 有结构文件（记录式文件）
  - 顺序文件
    - 文件的记录是一个接一个排列，记录通常是定长的，可以顺序存储或者链表存储
    - 批量处理时，顺序文件的效率是所有逻辑文件中效率最高的
    - 但是增删改查操作比较困难
  - 索引文件
    - 定长记录文件 按照公式 $A = i * L$ 可以直接得到文件地址（第i条记录，L是文件长度）
    - 变长记录文件
      - 查找前i-1条记录后，才能查找第i条记录
      - 通过建立索引表后可以有效提高查找速度
  - 索引顺序文件
    - 顺序和索引两种组织形式的结合。
    - 索引文件将顺序文件中的所有记录分成若干组，为顺序文件建立起一张索引表，在索引表中为每组中的第一条记录建立一个索引项，其中含有该记录得关键字值和指向该记录的指针
    - 索引顺序文件提高了查找效率，但是索引表也占用了存储空间
  - 直接文件或散列文件
    - 给定记录的键值或通过散列函数转换的键值直接决定记录的物理地址
    - 这种映射结构不同于顺序文件或者索引文件，没有顺序的特性

## 4.1文件系统基础（上）

## 4.1 文件系统基础（中）

### 目录结构

包含有关文件的信息，如属性、位置和所有权等

#### 文件控制块（FCB）

用来存放控制文件需要的各种信息的数据结构，实现“按名存取”。

包含信息  
基本信息：文件名，文件的物理位置，逻辑结构、物理结构等

存取控制信息：文件存取权限

使用信息：文件建立时间 修改时间

#### 文件控制块和索引结点

检索目录文件时，不需要将文件调入内存，只是查找其目录项，文件的描述信息单独形成为索引节点的数据结构

#### 索引结点

文件主标识符：拥有该文件的个人或小组的标识符

文件类型：普通文件、目录文件、特别文件

文件存取权限：各类用户对该文件的存取权限

文件物理地址：每个索引节点中含有13个地址项，直接或者间接的方式给出数据文件所在盘块的编号

文件长度：字节为单位

文件链接计数：本文件系统中所有指向该文件的文件名的指针计数

文件存取时间：文件最近被进程存取，修改以及索引节点最近被修改的时间

索引结点编号：用于标识内存索引节点

状态：指示i节点是否被上锁或者被修改

访问计数：每当有一个进程要访问此i结点时，计数加1，访问结束减1

逻辑设备号：文件所属文件系统的逻辑设备号

链接指针：设置分别指向空闲链表和散列队列的指针

文件打开后内存索引节点增加的内容

#### 目录结构

搜索：用户使用给一个文件时，需要搜索目录，找到该文件对应的目录项

创建文件：创建一个新文件时，需要在目录中增加一个目录项

删除文件：删除一个文件时，需要在目录中删除相应的目录项

显示目录：用户可以请求显示目录的内容，显示该用户目录中的所有文件及属性

修改目录：某些文件属性保存在目录中，因而这些属性的变化需要改变相应的目录项

#### 目录结构分类

##### 单机目录结构

整个文件系统只建立一张目录表，每个文件占一个目录项

优点：实现了按名存取

缺点：查找速度慢，文件不允许重名，不便于文件共享，不适用于多用户的操作系统

将文件分为主目录和用户目录，主目录记录用户名及相应用户文件目录所在的存储位置，用户目录项记录该用户文件的FCB信息。

##### 两级目录结构

优点：解决了不同用户文件重名问题，在一定程度上保证了文件的安全

缺点：缺乏灵活性，不能对文件分类

将两级目录结构的层次关系加以推广，就形成了多级目录结构，即树形目录结构

##### 多级目录结构

进程对各文件的访问都是相对于当前目录进行的

优点：有效的对文件进行分类，文件结构层次清晰，能够有效的进行文件管理和保护

缺点：按照路径名访问中间结点，增加了磁盘访问次数，降低了查询速度

在树形目录结构基础上增加了一些指向同一结点的有向边，使整个目录称为一个有向无环图。

##### 无环图目录结构

优点：有利于实现文件共享

## 4.1 文件系统基础（下）

### 文件共享

基于索引节点的共享方式（硬链接）

文件目录中只设置文件名及指向相应索引节点的指针，在索引节点中还有一个链接计数conut，用于表示链接到本索引节点（即文件）上的用户目录项的数目。

硬链接是多个指针指向一个索引结点，保证只要还有一个指针指向索引节点，索引节点就不能删除

优点：硬链接的查找速度要比软链接快

B用户共享A用户的文件F时候，系统创建一个LINK类型的新文件，也取名F，然后将文件F写入用户B的目录中，但是新文件中知识含有被链接文件F的路径名

软链接就是把到达共享文件的路径记录下来，当要访问文件时，根据路径寻找文件

优点：网络共享只需要提供该文件所在机器的网络地址及该机器中的文件路径

缺点：由于是根据文件路径名查找文件，因此会增加时间开销并且增加了启动磁盘的频率，同时符号链的索引节点也会耗费一定的硬盘空间

利用符号链实现文件共享（软链接）

为了防止文件共享导致文件被破坏或者未经允许修改文件，文件系统必须控制用户对文件的存取，解决对文件的读、写、执行的许可问题

### 文件保护

实现方式

口令保护

加密保护

为了防止用户文件被他人窃取或者存取

访问类型

访问控制 用于控制用户对文件的访问方式

读、写、执行、添加、删除、列表清单（列出文件名和属性名）

还可以对文件重命名、复制、编辑等加以控制

访问控制

根据用户身份进行控制，为每个文件和目录增加一个访问控制列表，规定每个用户名及其所允许的访问类型

优点：可以使用复杂的访问方法

缺点：长度无法预计且可能导致复杂空间管理

精简访问列表

所有者：创建文件的用户

组：一组需要共享文件且具有类似访问的用户

其他：系统内的所有其他用户

口令

口令：用户请求访问时需要提供相应的口令

优点：时间和空间开销不多

缺点：口令直接存储在系统内部不安全

密码

密码：用户对文件进行加密，用户访问需要秘钥解密

优点：保密性强。节省了存储空间

缺点：加密和解密需要花费一定时间

口令和密码都是防止文件被他人存取或者窃取，没有控制用户对文件的访问类型

## 4.2 文件系统的实现





## 4.3 磁盘组织与管理

### 磁盘结构

表面涂有磁性物质的金属或塑料构成的圆形盘片，通过一个称为磁头的导体线圈从磁盘读取数据。

磁盘盘面上的数据存储在在一组同心圆中，称为磁道

一个盘面上有上千个磁道，磁道又划分为几百个扇区，每个扇区固定存储大小，一份扇区称为一个盘块

磁盘地址用 柱面号-盘面号-扇区号(块号) 表示

#### 磁盘分类

固定头磁盘：磁头相对于盘片的径向方向固定

活动头磁盘：每个磁道一个磁头，磁头可以移动

固定盘磁盘：磁头臂可以来回伸缩定位磁道，磁盘永久固定在磁盘驱动器内

可换盘磁盘：可以移动和替换

#### 读写时间组成

寻找时间：活动头磁盘在读写信息前，将磁头移动到指定磁道所需要的时间（跨越n条磁道+启动磁臂）： $T_s = m \cdot n + s$

延迟时间：磁头定位到某一磁道扇区所需要的时间， $T_r = 1/2r$ （转速为r）

传输时间：从磁盘读出或向磁盘写入数据经过时间， $T_r = b / rN$

### 磁盘调度算法

按照进程请求访问磁盘的先后顺序进行调度

先来先服务算法（FCFS）

优点：公平 实现简单

缺点：适用于少量进程访问，如果进程过多算法更倾向于随机调度

选择调度处理的磁道是与当前磁头所在磁道距离最近的磁道

最短寻找时间有限算法（SSTF）

优点：性能强于先来先服务算法

缺点：容易产生饥饿现象

在磁头当前移动方向上选择与当前磁头所在的磁道距离最近的请求作为下一次服务对象

扫描(SCAN)算法

优点：寻道性能好，可以避免饥饿现象

缺点：对最近扫描过的区域不公平，访问局部性方面不如FCFS和SSTF好

循环扫描算法（C-SCAN）

磁头单向移动，回返时直接回到起始端，而不服任何请求

#### 磁盘调度算法

在SCAN与C-SCAN算法的基础上规定了查看移动方向上是否有请求，如果没有就不会继续向前移动，而是直接改变方向（LOOK）或者直接回到第一个请求处（C-LOOK）

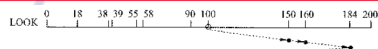


图 4.23-1 LOOK 磁盘调度算法

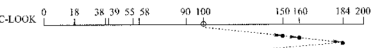


图 4.23-2 C-LOOK 磁盘调度算法

对盘面交替编号 原因：磁头在读/写一个物理块后，需要经过短暂的处理时间才能开始处理下一块

### 磁盘的管理

#### 磁盘初始化

低级格式化：磁盘分扇区，为每个扇区采用特别的数据结构（头、数据区域、尾部组成），头部含有一些磁盘控制器所使用的信息

进一步格式化处理：磁盘分区，对物理分区进行逻辑格式化（创建文件管理系统），包括空闲和已分配的空间及一个初始为空的目录

#### 引导块

计算机启动时运行自举程序，初始化CPU寄存器、设备控制器和内存等，然后启动操作系统

组局程序通常保存在ROM中，在ROM中保留很小的自举块，完整的自举程序保存在启动块上

拥有启动分区的磁盘称为启动磁盘或系统磁盘

#### 坏块

无法使用的扇区

对于简单的磁盘，可以在逻辑格式化时（建立文件系统时）对整个磁盘进行坏块检查，标明哪些扇区是坏扇区，比如：在 FAT 表上标明

#### 处理方式

简单磁盘：手动处理，对坏块进行标记，程序不会使用

复杂磁盘：控制器维护一个磁盘坏块链表，同时将一些块作为备用，用于替代坏块（扇区备用）