文件管理

概论

文件系统是操作系统中负责存取和管理信息的模块，使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，并为用户提供一套方便有效的文件使用和操作

文件系统的基本功能有

- 文件的按名存取
- 文件目录的建立和维护
- 实现逻辑文件到物理文件的转换
- 数据保密、保护和共享
- 提供用户接口

文件

文件概念

文件是由文件名标识的一组信息的集合

按名存取可以避免直接管理物理存储地址，也可以对文件提供安全保密等措施，以及便于实现文件的共享（同名/异名共享）

文件系统中最重要的是文件名，系统按照文件名管理和控制文件

一般来说文件名由文件名和扩展名组成，中间用 "." 分割，文件名标识文件内容，扩展名标识文件特性。操作系统提供通配符用来匹配文件名

文件类型和属性

可按照多种方式对文件分类。在 UNIX 系统中有三类文件

- 普通文件：程序文件，数据文件等，存储在磁盘上
- 目录文件：由文件目录构成的用于维护文件系统结构的系统文件，普通文件的查找依赖于目录文件
- 设备文件：是对物理设备的抽象，包括块设备文件，字符设备文件，有名管道文件，socket 等

文件的属性用于管理文件和安全保护，分为

- 基本属性：文件名、所有者、授权者、长度等
- 类型属性：普通文件、目录文件、系统文件、隐式文件、设备文件等
- 保护属性：读、写、可执行、可更新、可删除、可改变保护、归档等
- 管理属性：创建时间、最后存取时间、最后修改时间等

文件存取方法

文件的存取可分为

- 顺序存取：按记录顺序读写
- 直接存取：任意次序读写
- 索引存取：基于索引文件的存取

文件的使用可分为两类，一类是在终端使用控制台命令，如 ls，touch 等，另一类是在程序中使用系统或语言提供的 API，如 open，close 等

文件目录

文件控制块

文件目录是文件的清单，每个目录项对应一个文件的信息，该目录项又称为**文件控制块**（FCB, File Control Block）。文件目录通常也以文件的形式存取，称为**目录文件**，目录文件全部由目录项构成。

文件控制块的信息包括

- 文件存取控制信息，如文件名、用户名、文件存取权限等
- 文件结构信息，文件逻辑结构、文件的物理结构等
- 文件使用信息，已打开该文件的进程数、文件的修改情况等
- 文件管理信息，文件建立日期、文件访问日期等

文件目录用于实现文件系统的按名存取，当用户要求存取某文件时，查询目录文件，获得对应的文件目录，在文件目录中根据文件名找到 FCB，根据 FCB 中的信息即可实现文件存取

在 Linux 系统中，将 FCB 除文件名以外的信息存放在一起，称为 inode，存放在磁盘的固定位置，由 inode 号标识。在目录项中只有文件名和对应的 inode 号，这样一个磁盘块可以存放更多目录信息，加快检索速度，也便于实现文件共享

层次目录结构

最简单的目录结构就是一级目录结构，所有 FCB 存放在一张线性表中。这样虽然简单，但是不能解决文件重名和文件共享问题

因此引入多级目录结构，该结构是一颗倒置的有根树，又称为树形目录结构。其根称为根目录，从上向下，每个非叶节点为子目录，叶节点为文件

层次目录结构可以反应现实世界的数据集合间的层次关系，且不在同一目录中的文件可以重名，能够实现以目录为单位的文件保护，保密和共享

层次目录结构中的文件的全名包括从根目录开始到文件为止途中遇到的所有子目录名，各个子目录名之间由斜线分割，子目录名的部分又被称为**路径名**

文件目录检索

每个文件目录创建时自动含有两个特殊目录项，分别为其自身的 inode 和其父目录的 inode，在按照文件名检索文件时，首先找到根目录的 inode ，读取其内容找到路径上下一个子目录对应的 inode，然后打开子目录的目录文件，再寻找下一个子目录的 inode ……直至最终找到文件的 inode ，然后对文件存取

类似于文件的操作，系统也提供了一系列用于操作文件目录的操作，如创建、删除、修改等

文件的组织和数据存储

基本概念

卷：物理存储介质的单位，如一盘磁带，一张光盘等，都是一卷

块：存储介质上连续信息组成的一个区域，也叫做**物理记录**，块是主存和辅存信息交换的单位

逻辑记录：文件中按信息在逻辑上的独立含义划分的一种信息单位，应用程序处理的单位

存储记录：指附加了操作系统控制信息的逻辑记录，文件管理程序处理的单位

文件的逻辑结构

文件的逻辑结构是从用户角度出发，在用户概念中的抽象信息的组织方式，以及用户可见，并可处理的数据集合。

文件的逻辑结构可分为

- 流式文件：文件中的数据不组成记录，是依次的信息集合，一般也称为字节流文件，使用读写指针访问特定字节。
- 记录式文件：有结构的文件，包含若干逻辑记录（类似数据库文件），可分为定长和变长

为了简化操作系统，大多数现代系统只提供流式文件

从操作系统角度看，逻辑记录是文件内独立的最小信息单位。记录文件的组织可分为顺序型和索引型。逻辑记录存放到存储介质上时可能占据一个或多个物理块，一个物理块也可能包含多个逻辑记录。当缓冲区中的逻辑结构达到一个物理块才写入，每次读取读取一整个物理块的逻辑结构叫做成组和分解，可以提高 IO 效率

文件的物理结构

文件物理结构是文件在物理存储空间中的存放方法和组织关系

文件物理结构涉及

- 块的划分
- 记录的排列
- 索引的组织
- 信息的检索

常见构造物理结构的方法有计算法和指针法

- 计算法：设计映射算法，如线性计算法、杂凑法将逻辑地址（记录键）转换成对应的物理地址
- 指针法：设置专门的指针，指明相应记录的物理地址或表达各记录间的关联关系

常见文件物理结构

顺序文件

顺序文件即连续的逻辑记录存储在存储介质相邻物理块上，逻辑顺序与物理顺序完全一致。FCB 中只需保存第一个物理块地址和总块数即可。

顺序文件的顺序存取速度快，但是创建文件时需要预先确定长度，且文件的插入和删除有难度，且空间利用率不高

链接文件

使用指针表示文件中各条记录间的关系，第一个物理块的地址由 FCB 给出，之后物理块的地址由上一个物理块的指针域给出。指针为 0 时指示文件结束。这种方式使得逻辑记录顺序与物理记录顺序独立，方便增删改，但是性能较低

一种变种是使用指针数组代替物理块中的指针，设指针数组 `ptr[n]` 代表指向 n 个物理块的指针，则 FCB 中给出第一个物理块为 `i`，下一个物理块的地址 `j = ptr[i]`，依次类推，思路类似静态链表

直接文件

又称散列文件，在记录的 key 和其地址间建立散列关系，需要解决冲突问题

直接散列法就是以 key 作为地址

索引文件

系统为每个文件建立一张索引表，表项为记录的 key 和其对应的存储地址，索引可分为稠密型索引与稀疏型索引

有时候文件记录数目很多，此时可采用多级索引，如某个索引项指向的物理块中存放一系列二级索引项，还可以有更多级的间接索引

文件系统的功能和实现

文件系统调用

文件系统提供给用户程序的一组系统调用，包括：创建、打开、关闭、撤销、读、写和共享等，通过这些系统调用用户能获得文件系统的各种服务。

以 UNIX 为例，内核以磁盘作为文件存储器，磁盘按扇区编号，扇区内的静态结构可分为

- 超级块（1#）：存放文件系统的结构与管理信息，包括 inode 表所占磁盘块数量，数据所占磁盘块数量，空闲块数，空闲块编号，空闲 inode 数，空闲 inode 编号。当一个设备作为文件卷安装时，超级块就要复制进内存
- inode 区（2#~k+1#）：存放 inode 表
- 数据区（k+2#~n#）：存放文件数据

内存中也维护了动态结构

- 用户打开文件表：进程 PCB 中的结构，表项序号为文件描述符，指向系统打开文件表中的一个表项
- 系统打开文件表：为了解决文件共享问题，在内存中为所有打开的文件维护一个系统打开文件表，每个文件的一次打开对应一个 file 结构，多次打开生成多个 file 结构。用户打开文件表中的文件描述符指向一个 file 结构，一个文件被多次打开产生多个文件描述符，就会对应多个 file 结构，每个结构内的读写偏移量独立。file 结构包含了文件的访问信息（读/写/添加），打开计数，inode 号等，指向一个内存活动 inode 表中的 inode
- 内存活动 inode 表：当前活动的 inode 表，避免多次访问磁盘上的 inode 表。系统打开文件表中的每个表项与一个 inode 关联

三个表实现了文件描述符-file 结构-inode 的关联，均是随着一次打开行为生成，上述三个表均位于内核空间

文件创建与删除

创建新文件时需要为其分配 inode 和活动 inode，并且将文件名和 inode 号组成的目录项添加到目录中。初始化活动 inode，同时也要分配用户打开文件表项和系统打开文件表项，为各个表项赋初值，将各个表项和活动 inode 用指针相连，返回文件描述符。文件创建隐含了打开操作，创建后即可直接操作

创建文件时需要检查权限，也需要检查是否有可用 inode

文件删除需要调用者有写权限

删除时将文件的目录项从文件目录中删去，同时将 inode 中的 i_link 减一，若其为 0，则还要释放文件占用的空间

文件打开与关闭

打开文件时首先检索目录，找到 inode 号以后从磁盘将其 inode 复制到内存活动 inode 表，比较输入参数与 inode 中的权限记录，若访问非法则打开失败。合法则分配用户打开文件表项和系统打开文件表项，为各个表项赋初值，将各个表项和活动 inode 用指针相连，返回文件描述符。若在此之前已经有进程打开了该文件，则直接将活动 inode 表中的 inode 的 i_count 分量加一

关闭文件时根据传入的文件描述符找到用户打开文件表项，然后找到系统打开文件表项，将用户打开文件表项释放，然后将系统打开文件表项中的 f_count 减一，若其不为 0，说明有进程在共享此表项，直接返回。否则释放该表项，并将对应 inode 的 i_count 减一，若其不为 0，说明还有进程在使用此文件，返回，否则表示已经没有进程使用该文件，将其 inode 写回磁盘 inode 区，释放活动 inode

文件读写

读文件时，需要检查读操作是否合法。按照 inode 中的物理块信息和文件偏移量找到相应的物理地址，将物理块内容读入缓冲区，然后拷贝指定字节的数据到参数指向的内存区域

写文件过程与读文件类似，同样是检查权限，然后将缓冲区的内容写入对应物理块

文件读写都依赖于系统打开文件表项中的偏移量，故有系统调用可以修改文件的偏移量，提供随机访问的能力

DUP

dup 操作参数为一个文件描述符，将传入的文件描述符复制一个新的文件描述符，并将其返回

dup2 可以指定复制后的新文件描述符，若文件描述符已被一个打开的文件使用，则将其关闭

进程开始时已有三个打开的文件，分别为

- 0：标准输入

- 1：标准输出
- 2：标准错误输出

dup 操作可实现重定位，如打开一个文件，描述符为 fd，则

```
dup2(fd, 1);
close(fd);
```

即可实现将标准输出重定位到打开的文件

文件共享

文件共享是指不同用户（进程）共同使用一个文件，文件共享可以节省外存空间，减少文件复制增加的外存访问次数

常见共享文件方式分为静态共享（用户角度）和动态共享（进程角度）

静态共享

操作系统允许一个文件属于多个目录，但实际上只有一处物理储存。这样的关系称为链接

创建链接时找到文件的 inode 号，在新目录下创建一个对应的目录项，并将 inode 中的 i_link 分量加一

解除链接的过程就是删除文件的过程，只不过从文件所有者的角度来看是删除，从文件共享者的角度来看是解除链接

还有一种链接方式叫做**文件符号链接共享**，又被称为软链接。符号链接只有被共享的文件名，不包含 inode 号，文件的内容就是共享的文件的路径。软链接优点在于可以跨文件系统使用

动态共享

文件的动态共享是指系统中不同用户的进程或同一用户的不同进程并发访问同一文件。进程存在才有共享关系存在，一旦进程消亡则共享关系也消失。可以分为两种

- 使用同一文件偏移量，这种情况出现在父进程打开文件后 fork 产生子进程的情况，父进程和子进程的 PCB 中有相同的文件描述符，指向同一个系统打开文件表项，该表项的 f_count 加一，父子进程使用同一个文件偏移量
- 使用不同文件偏移量，这种情况即是不同进程或同一进程对于同一文件的多次打开，对应着系统打开文件表中的多个表项，这些表项的读写偏移独立，指向同一个活动 inode

其他文件系统

日志结构文件系统（Log-Structured File System）

延迟多次写盘操作，能提高I/O效率，同样也增加了文件系统不一致性风险。

集合同一次文件变动（如创建文件）所涉及的所有修改，作为一个日志记录写回磁盘

日志文件系统（Journaling File System）

每次文件变动前，先将涉及的操作记录下来，若所有操作成功，则删除操作记录，否则，根据日志记录进行恢复

虚拟文件系统

虚拟文件系统的目标是

- 同时支持多种文件系统
- 系统内可安装多个文件系统
- 为用户提供统一的接口
- **提供网络共享文件支持**
- 可扩充新的文件系统

实现思路为对文件系统的共同特征抽象，形成与具体系统无关的虚拟层，提供一致接口

文件系统的管理与优化

文件空间管理

文件系统需要解决外存空间的分配去配问题

对于管理外存空间，有以下几种方法

位示图

磁盘空间的分配去配以磁盘块为基本单位，可以使用位示图管理，其中每一位对应一个磁盘块，1/0 指示其分配/空闲。位示图占用空间少，可以完全或大部分保存在内存中，结合位操作对其修改

成组空闲块链表

将空闲块分组，每组第一块指示下一组空闲块和空闲块总数，空闲块的编号组织成链表，第一组的首节点为专用块，存放在内存中，之后的空闲块组可存放在磁盘中

分配时从内存中的空闲块链表上寻找一个空闲节点，按照编号分配对应的磁盘块，然后将空闲块计数减一，删去对应节点。若分配后空闲块数为 0，则将专用块指向的磁盘块取出，其中保存了下一组空闲块的链表，以此作为新的专用块，进行分配

去配时，将空闲的磁盘块号加入空闲块队列，若队列满，则将其整个写入新释放的磁盘块中，然后将其编号作为新的专用块

空闲块链表在磁盘和内存见多次交换会带来不必要的 IO 开销（多次分配/去配交替），解决方法为保持链表半满

文件系统优化

物理块大小的设置，块小提高磁盘利用率，块大提高磁盘的 IO 效率

高速缓存的管理可以使用 LRU 算法，但是不能完全满足要求

文件系统的备份可分为物理备份与逻辑备份

主存映射文件

结合虚存管理技术和文件管理技术实现的文件访问方法，将磁盘访问转换为内存访问

mmap 可以将文件映射到内存空间，访问对应地址时引发缺页异常，将文件内容作为页面调入内存即可

unmmap 可以解除映射并将内存数据写回磁盘文件

主存映射文件可以多进程共享文件，只要他们的虚拟页面都映射到同一物理页面