

Ch 09 文件系统

1. 文件系统的管理

1.1 文件系统的可靠性

可靠性：抵御和预防各种物理性破坏和人为性破坏的能力

坏块问题

备份：通过转储操作，形成文件或文件系统的多个副本

文件系统备份

全量转储

- 定期将所有文件拷贝到后援存储器

增量转储

- 只转储修改过的文件，即两次备份之间的修改，减少系统开销

物理转储

- 从磁盘第 0 块开始，将所有磁盘块按序输出到磁带

逻辑转储

- 从一个或几个指定目录开始，递归地转储自给定日期后所有更改的文件和目录

3.2 文件系统的一致性

若在写回之前，系统崩溃，则文件系统出现不一致

设计一个实用程序，当系统再次启动时，运行该程序，检查磁盘块和目录系统

文件系统写入方式

- 通写

内存中的修改立即写到磁盘

缺点：性能差

- 延迟写

利用回写缓存的方法得到高速

可恢复性差

- 可恢复写

采用事务日志来实现文件系统的写入

既考虑安全性，又考虑速度性能

3.3 文件系统的安全性

安全性：确保未经授权的用户不能存取某些文件

数据丢失

- 硬件或软件故障
- 人为事故

入侵者

- 积极的消极的

文件保护机制

用于提供安全性、特定的操作系统机制

对拥有权限的用户，应该让其进行相应操作，否则，应禁止

防止其他用户冒充对文件进行操作

实现：对用户进行身份验证、访问控制

用户身份验证：

- 口令、密码
- 物理鉴定
- 基于生物特征信息的认证

访问控制

- 主动控制：访问控制表
每个文件一张表
- 能力表（权限表）
每个用户一张表

UNIX 采用文件的二级存取控制

审查用户的身份、审查操作的合法性

- 第一级：对访问者的识别
对用户分类
- 第二级：对操作权限的识别
对操作分类

数据恢复技术

数据恢复的原理

当磁盘、分区、文件遭到破坏时，其数据未真正被覆盖，只是数据在磁盘上的组织形式被破坏，以至于操作系统或用户不能访问

4. 文件系统的性能

设计文件系统应尽可能减少磁盘访问次数

提高文件系统性能的方法：

目录项（FCB）分解、当前目录、磁盘碎片整理、磁盘（块）高速缓存、磁盘调度、提前读取、合理分配磁盘空间、信息的优化分布、RAID 技术

4.1 磁盘高速缓存

内存中为磁盘块设置的一个缓冲区，保存了磁盘中某些块的副本——磁盘高速缓存

- 当出现一个对某一个特定块的 I/O 请求时，首先检测以确定该块是否在磁盘高速缓存中
- 如果在，则可直接进行读操作；否则，先要将数据块读入磁盘高速缓存中，再拷贝到所需的地方
- 由于访问的局部性原理，当一数据块被读入磁盘高速缓存以满足一个 I/O 请求时，很有可能将来还会再访问到这块数据
- 有些系统称为文件缓存、块高速缓存、缓冲区高速缓存

4.2 提前读取

思路：每次访问磁盘，多读入一些磁盘块

依据：程序执行的空间局部性原理

开销：较小

具有针对性

Windows 的文件访问方式

不使用文件缓冲

- 普通的方式

- 通过 Windows 提供的 FlushFileBuffer 函数实现

使用文件缓冲

- 预读取。每次读取的块大小、缓冲区大小、置换方式
- 写回。

异步模式

- 不再等待磁盘操作的完成
- 使处理器和 I/O 并发工作

用户对磁盘的访问通过访问文件缓存来实现

由系统的 cache manager 来实现对缓存的控制

4.3 合理分配磁盘空间

分配块时，把有可能顺序存取的块放在一起

- 尽量分配在同一柱面上，从而减少磁盘臂的移动次数

4.4 磁盘调度

当访盘请求在等待时，采用一定的策略，对这些请求的服务顺序调整安排

- 降低平均服务时间，达到公平、高效

公平：一个 I/O 请求在有限时间内满足

高效：减少设备机械运动带来的时间浪费

一次访盘时间 = 寻道时间 + 旋转时间 + 传输时间

- 减少寻道时间
- 减少延迟时间

磁盘调度算法：

先来先服务：

FCFS：先来先服务，按照访问请求到达的先后次序服务

有点：简单、公平

缺点：效率不高，相邻两次请求可能会造成最内到最外的柱面寻道，使磁头反复移动，增加了服务时间

最短寻道时间优先：

优先选择距当前磁头最近的访问请求进行服务，寻道优先

优点：改善了磁盘平均服务时间

缺点：造成某些访问请求长期等待得不到服务

电梯算法 扫描算法：SCAN

单向扫描调度算法

N 步扫描调度算法

FSCAN 策略

旋转调度算法

旋转调度：根据延迟时间来决定执行次序的调度

4.5 信息的优化分布

记录在磁道上的排列方式也会影响输入输出的时间

4.6 记录的成组与分解

记录的成组

- 把若干个逻辑记录合成一组存放一块的动作
- 进行成组操作时必须使用内存缓冲区，缓冲区的长度等于逻辑记录长度乘成组的块因子。
- 目的：提高了存储空间的利用率；减少了启动外设的次数，提高了工作效率
- 记录的分解：从一组逻辑记录中把一个逻辑记录分离出来的操作

4.7 RAID 技术

设计时要考虑：磁盘存储系统的速度、容量、容错、数据灾难发生后的数据恢复

RAID：独立磁盘冗余阵列

多块磁盘按照一定要求构成，操作系统则将它们看成一个独立的存储设备

目标：提高可靠性和性能

通过把多个磁盘组织在一起，作为一个逻辑卷提供磁盘跨越功能

数据是如何组织存储的：

1. 通过把数据分成多个数据块，并行写入/读出多个磁盘，以提高数据传输率
2. 通过镜像或校验操作，提高容错能力（冗余）

最简单的 RAID 组织方式：镜像

最复杂的 RAID 组织方式：块交错校验

RAID 0 - 条带化

数据分布在阵列的所有磁盘上

有数据请求时，同时多个磁盘并行操作

充分利用总线带宽，数据吞吐率提高，驱动器负载均衡

RAID 1 - 镜像

最大限度保证数据安全及可恢复性

所有数据同时存在于两块磁盘的相同格式

磁盘利用率 50%

数据安全性最好

RAID 2 并行访问 - 海明码校验

将数据条块化分布于不同磁盘（字节或位为单位）

加入海明码，在磁盘阵列中间隔写入每个磁盘中

数据发生错误时可实施阵列一些动作，在各个磁盘相同位置平行存取，所以有更好的存取时间

RAID 3 交错位奇偶校验

- 以字节为单位将数据拆分，并交叉写入数据盘
- 专门设置一个存储校验盘，保存校验码（奇偶校验）

RAID 4 交错块奇偶校验

- 带奇偶校验

- 以数据块为单位

RAID 5 交错块分布式奇偶校验

- 奇偶校验分散在各个磁盘
- 数据读出效率高，写入效率一般
- 磁盘利用率好，提高了可靠性
- 有写损失

RAID 6 交错块双重分布式奇偶校验

- 设立两个校验码，并将校验码写入两个驱动器
- 数据恢复能力增强
- 磁盘利用率降低，写能力降低

RAID 7 最优化异步高 I/O 速率及高数据传输率

- 自身带有智能化实时操作系统和用于存储管理的管理工具，独立于主机运行
- 每个磁盘有独立的 I/O 通道，与主通道连接
- 操作系统直接对每个磁盘的访问进行控制，可以让每个磁盘在不同的时段进行数据读写
- 价格高

5. 文件系统的结构设计

文件系统分类

- 磁盘文件系统
- 数据库文件系统
- 日志文件系统
- 网络/分布式文件系统
- 虚拟文件系统

虚拟文件系统

- 对多个不同文件系统的抽象

功能

- 提供相同的文件和文件系统接口
- 管理所有文件和文件系统关联的数据结构
- 高效查询例程，遍历文件系统
- 与特定文件系统模块的交互

日志文件系统

- 借鉴日志结构文件系统的设计思路：鲁棒性
- 保存一个日志：记录系统下一步将要做什么
- 系统出错后，恢复时查看日志，完成所作操作

分布式文件系统

- 由多台分散的计算机互联而成
- 强调资源、任务、功能和控制的全面分布