操作系统

Operating Systems

提纲

- 文件系统的概念
  - 文件系统和文件
  - 文件描述符
  - 目录
  - 文件别名
  - 文件系统种类
- 虚拟文件系统
- 文件缓存和打开文件
- 文件分配
- 空闲空间管理
- 冗余磁盘阵列RAID

文件系统和文件

- 文件系统是操作系统中管理持久性数据的子系统，提供数据存储和访问功能
  - 组织、检索、读写访问数据
  - 大多数计算机系统都有文件系统
  - Google 也是一个文件系统
- 文件是具有符号名，由字节序列构成的数据项集合
  - 文件系统的基本数据单位
  - 文件名是文件的标识符号

文件系统的功能

- 分配文件磁盘空间
  - 管理文件块（位置和顺序）
  - 管理空闲空间(位置)
  - 分配算法 (策略)

文件系统的功能

- 分配文件磁盘空间
- 管理文件集合
  - 定位：文件及其内容
  - 命名：通过名字找到文件
  - 文件系统结构：文件组织方式

文件系统的功能

- 分配文件磁盘空间
- 管理文件集合
- 数据可靠和安全
  - 安全：多层次保护数据安全
  - 可靠
    - 持久保存文件
    - 避免系统崩溃、媒体错误、攻击等

文件属性

- 文件属性
  - 名称、类型、位置、大小、保护、创建者、创建时间、最近修改时间、…
- 文件头：文件系统元数据中的文件信息
  - 文件属性
  - 文件存储位置和顺序

提纲

- 文件系统的概念
  - 文件系统和文件
  - 文件描述符
  - 目录
  - 文件别名
  - 文件系统种类
- 虚拟文件系统
- 文件缓存和打开文件
- 文件分配
- 空闲空间管理
- 冗余磁盘阵列RAID

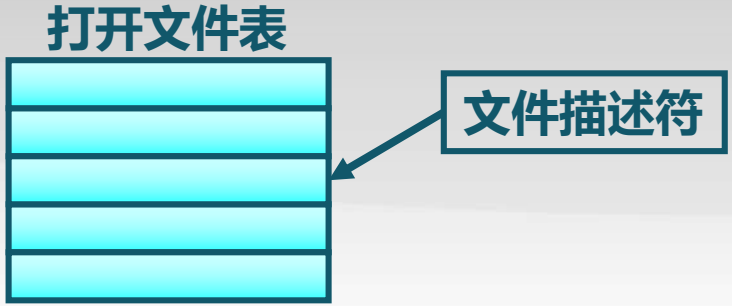
打开文件和文件描述符

- 文件访问模式
  - 进程访问文件数据前必须先“打开”文件

```
f = open(name, flag);
...
read(f, ...);
...
close(f);
```

打开文件和文件描述符

- 文件访问模式
  - 进程访问文件数据前必须先“打开”文件
- 内核跟踪进程打开的所有文件
  - 操作系统为每个进程维护一个打开文件表
  - 文件描述符是打开文件的标识



文件描述符

- 操作系统在打开文件表中维护的打开文件状态和信息
  - **文件指针**

    最近一次读写位置

    每个进程分别维护自己的打开文件指针

文件描述符

- 操作系统在打开文件表中维护的打开文件状态和信息
  - **文件指针**
  - **文件打开计数**

    当前打开文件的次数

    最后一个进程关闭文件时，将其从打开文件表中移除

文件描述符

- 操作系统在打开文件表中维护的打开文件状态和信息
  - 文件指针
  - 文件打开计数
  - 文件的磁盘位置

    缓存数据访问信息

文件描述符

- 操作系统在打开文件表中维护的打开文件状态和信息
  - 文件指针
  - 文件打开计数
  - 文件的磁盘位置
  - 访问权限

    每个进程的文件访问模式信息

文件的用户视图和系统视图

- 文件的用户视图
  - 持久的**数据结构**
- 系统访问接口
  - **字节序列**的集合(UNIX)
  - 系统不关心存储在磁盘上的数据结构
- 操作系统的文件视图
  - 数据块的集合
  - 数据块是逻辑存储单元，而扇区是物理存储单元
  - 块大小< > 扇区大小

用户视图到系统视图的转换

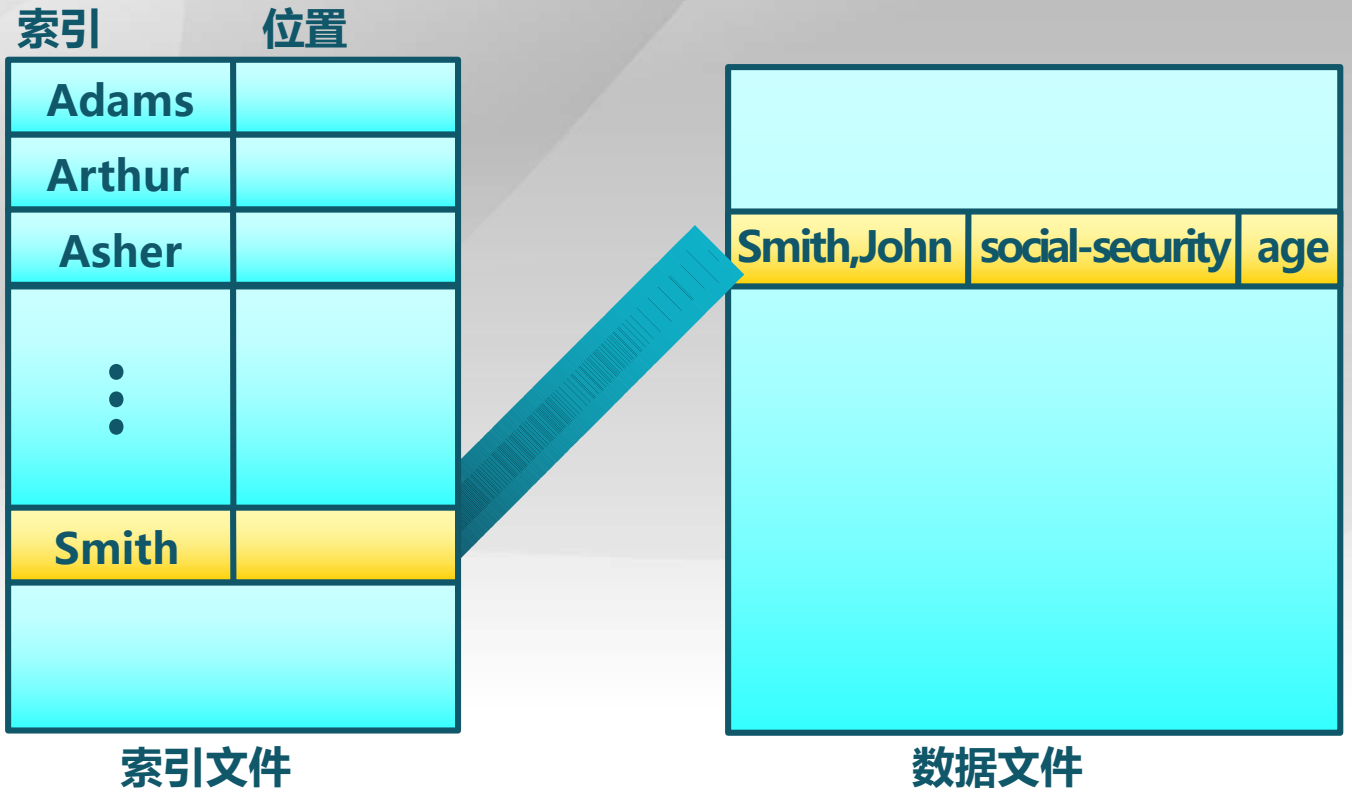
- 进程读文件
  - 获取字节所在的数据块
  - 返回数据块内对应部分
- 进程写文件
  - 获取数据块
  - 修改数据块中对应部分
  - 写回数据块
- **文件系统中的基本操作单位是数据块**
  - 例如，getc()和putc()即使每次只访问1字节的数据，也需要缓存目标数据4096字节

访问模式

- 操作系统需要了解进程如何访问文件
- **顺序访问**: 按字节依次读取
  - 大多数的文件访问都是顺序访问
- **随机访问**: 从中间读写
  - 不常用, 但仍然重要

    例如, 虚拟内存中把内存页存储在文件
- **索引访问**: 依据数据特征索引
  - 通常操作系统不完整提供索引访问
  - 数据库是建立在索引内容的磁盘访问上

索引文件示例


索引
位置
Adams
Arthur
Asher
Smith
索引文件
Smith,John
social-security
age
数据文件

文件内部结构

- 无结构
  - 单词、字节序列
- 简单记录结构
  - 分列
  - 固定长度
  - 可变长度
- 复杂结构
  - 格式化的文档(如, MS Word, PDF)
  - 可执行文件
  - …

文件共享和访问控制

- **多用户系统**中的文件共享是很必要的
- 访问控制
  - 每个用户能够获得哪些文件的哪些访问权限
  - 访问模式: 读、写、执行、删除、列表等
- 文件访问控制列表(ACL)
  - <文件实体, 权限>
- Unix模式
  - <用户|组|所有人, 读|写|可执行>
  - **用户标识ID**
    识别用户, 表明每个用户所允许的权限及保护模式
  - **组标识ID**
    允许用户组成组，并指定了组访问权限

语义一致性

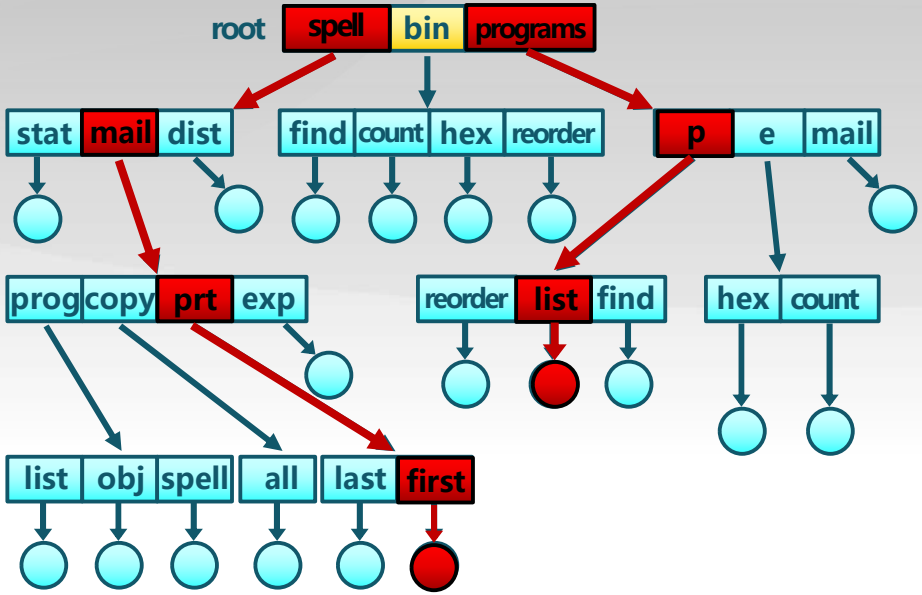
- 规定多进程如何同时访问共享文件
  - 与同步算法相似
  - 因磁盘I/O和网络延迟而设计简单
- Unix 文件系统(UFS)语义
  - 对打开文件的写入内容立即对其他打开同一文件的其他用户可见
  - 共享文件指针允许多用户同时读取和写入文件
- 会话语义
  - 写入内容只有当文件关闭时可见
- 读写锁
  - 一些操作系统和文件系统提供该功能

提纲

- 文件系统的概念
  - 文件系统和文件
  - 文件描述符
  - 目录
  - 文件别名
  - 文件系统种类
- 虚拟文件系统
- 文件缓存和打开文件
- 文件分配
- 空闲空间管理
- 冗余磁盘阵列RAID

分层文件系统

- 文件以目录的方式组织起来
- 目录是一类特殊的文件
  - 目录的内容是文件索引表<文件名, 指向文件的指针>
- 目录和文件的树型结构
  - 早期的文件系统是扁平的 (只有一层目录)



/spell/mail/prt/first

/programs/p/list

目录操作

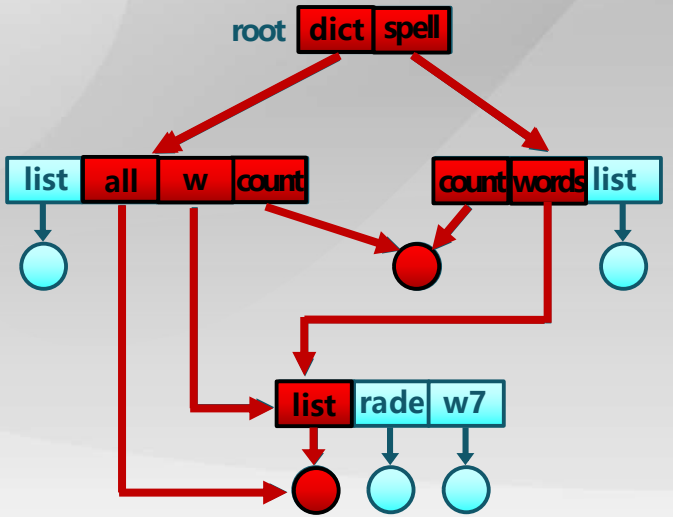
- 典型目录操作
  - 搜索文件
  - 创建文件
  - 删除文件
  - 列目录
  - 重命名文件
  - 遍历路径
- 操作系统应该只允许内核修改目录
  - 确保映射的完整性
  - 应用程序通过系统调用访问目录

目录实现

- 文件名的线性列表，包涵了指向数据块的指针
  - 编程简单
  - 执行耗时
- 哈希表 – 哈希数据结构的线性表
  - 减少目录搜索时间
  - 冲突 – 两个文件名的哈希值相同
  - 固定大小

文件别名

- 两个或多个文件名关联同一个文件

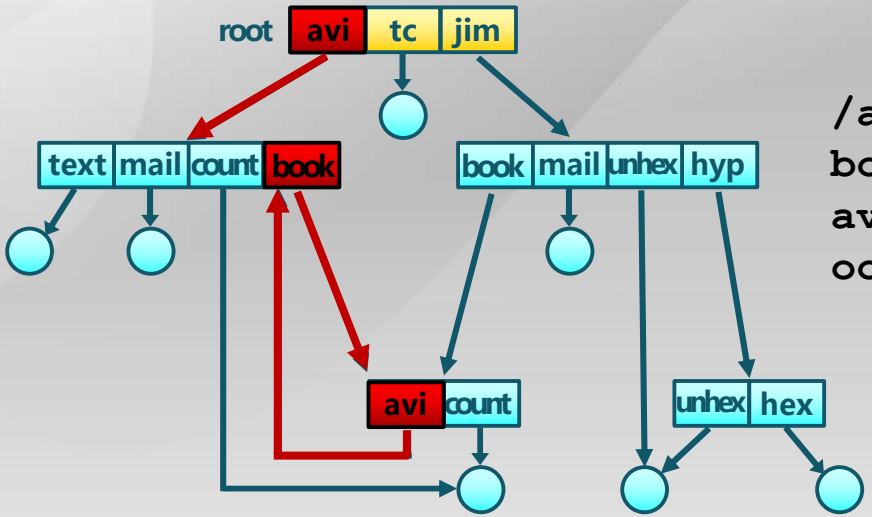



```
/dict/count
/spell/count
```

```
/dict/w/list
/dict/all
/spell/words/list
```

- **硬链接**: 多个文件项指向一个文件
- **软链接**: 以“快捷方式”指向其他文件
  - 通过存储真实文件的逻辑名称来实现

文件目录中的循环



/avi/book/avi/book/avi/
book/avi/book/avi/book/
avi/book/avi/book/avi/b
ook/avi/…

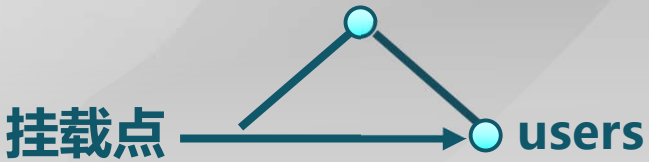
- 如何保证没有循环?
  - 只允许到文件的链接，不允许在子目录的链接
  - 增加链接时，用循环检测算法确定是否合理
- 更多实践
  - 限制路径可遍历文件目录的数量

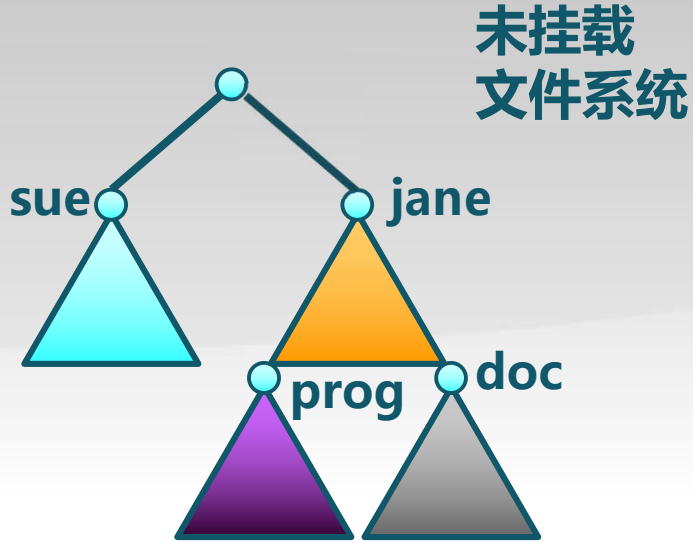
名字解析（路径遍历）

- 名字解析: 把逻辑名字转换成物理资源（如文件）
  - 依据路径名，在文件系统中找到实际文件位置
  - 遍历文件目录直到找到目标文件
- 举例: 解析“/bin/ls”
  - 读取根目录的文件头 (在磁盘固定位置)
  - 读取根目录的数据块，搜索“bin”项
  - 读取bin的文件头
  - 读取bin的数据块; 搜索“ls”顶
  - 读取ls的文件头
- 当前工作目录 (PWD)
  - 每个进程都会指向一个文件目录用于解析文件名
  - 允许用户指定相对路径来代替绝对路径
    如，用 PWD=“/bin” 能够解析 “ls”

文件系统挂载

- 文件系统需要先挂载才能被访问
- 未挂载的文件系统被挂载在挂载点上





文件系统种类

- 磁盘文件系统
  - 文件存储在数据存储设备上, 如磁盘
  - 例如: FAT, NTFS, ext2/3, ISO9660,等
- 数据库文件系统
  - 文件特征是可被寻址（辨识）的
  - 例如: WinFS
- 日志文件系统
  - 记录文件系统的修改/事件
- 网络/分布式文件系统
  - 例如: NFS, SMB, AFS, GFS
- 特殊/虚拟文件系统

网络/分布式文件系统

- 文件可以通过网络被共享
  - 文件位于远程服务器
  - 客户端远程挂载服务器文件系统
  - 标准系统文件访问被转换成远程访问
  - 标准文件共享协议
    NFS for Unix, CIFS for Windows
- 分布式文件系统的挑战
  - 客户端和客户端上的用户辨别起来很复杂
    - 例如, NFS是不安全的
  - 一致性问题
  - 错误处理模式

操作系统

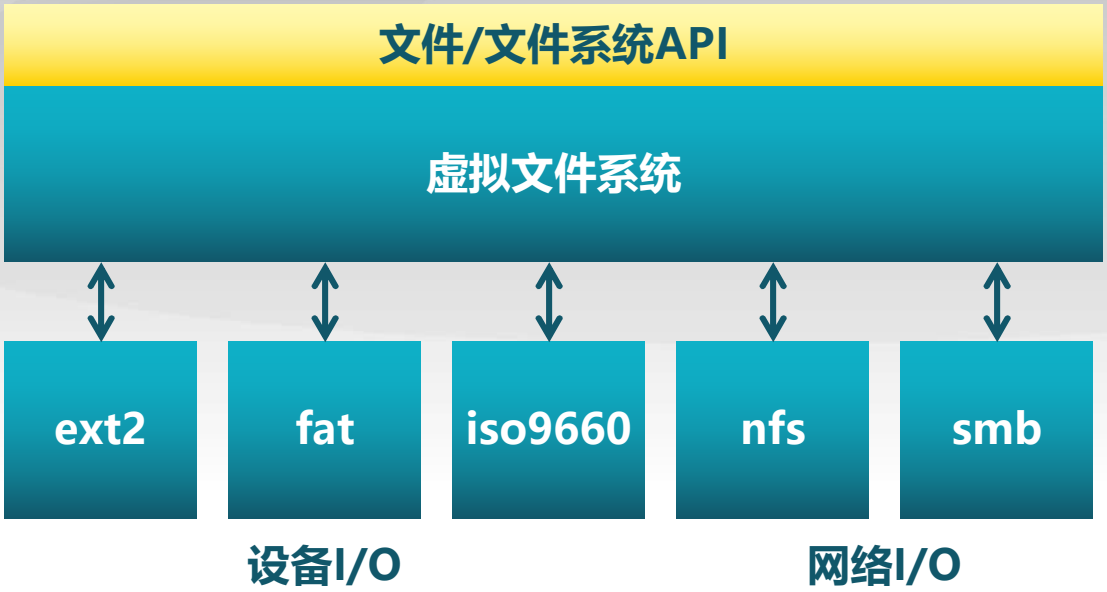
Operating Systems

操作系统

Operating Systems

文件系统的实现

- 分层结构
  - 虚拟（逻辑）文件系统(VFS, Virtual File System)
  - 特定文件系统模块



虚拟文件系统 (VFS)

- 目的
  - 对所有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

文件系统基本数据结构

- 文件卷控制块 (Unix: “superblock”)
  - 每个文件系统一个
  - 文件系统详细信息
  - 块、块大小、空余块、计数/指针等

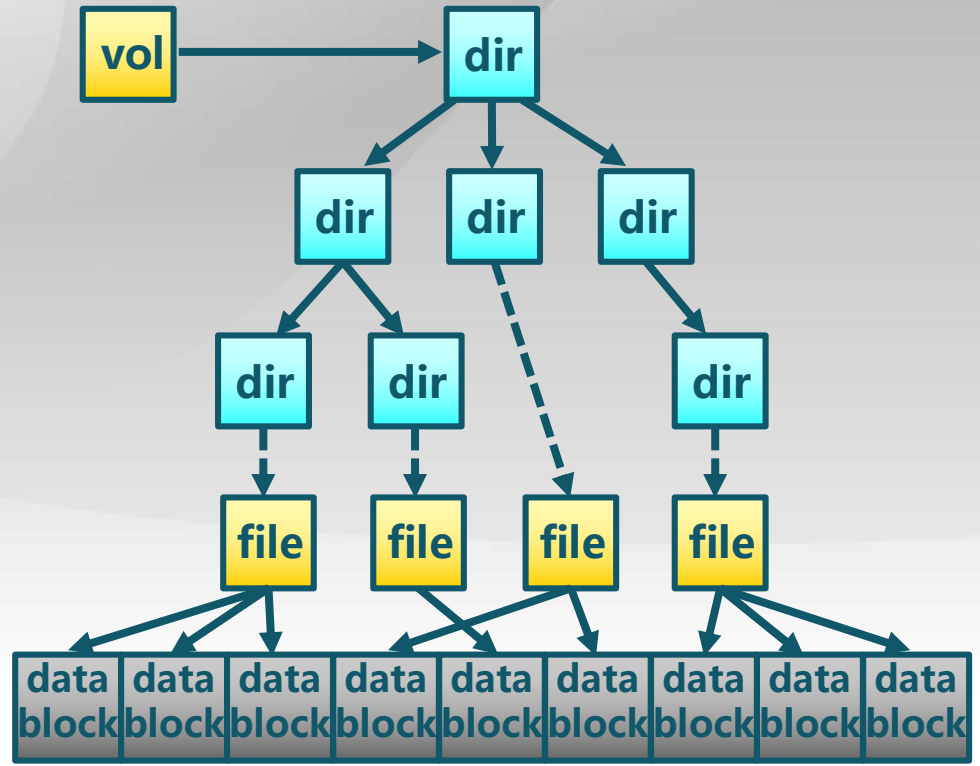
文件系统基本数据结构

- 文件卷控制块 (Unix: “superblock”)
- 文件控制块(Unix: “vnode” or “inode”)
  - 每个文件一个
  - 文件详细信息
  - 访问权限、拥有者、大小、数据块位置等

文件系统基本数据结构

- 文件卷控制块 (Unix: “superblock”)
- 文件控制块(Unix: “vnode” or “inode”)
- 目录项 (Linux: “dentry”)
  - 每个目录项一个(目录和文件)
  - 将目录项数据结构及树型布局编码成树型数据结构
  - 指向文件控制块、父目录、子目录等

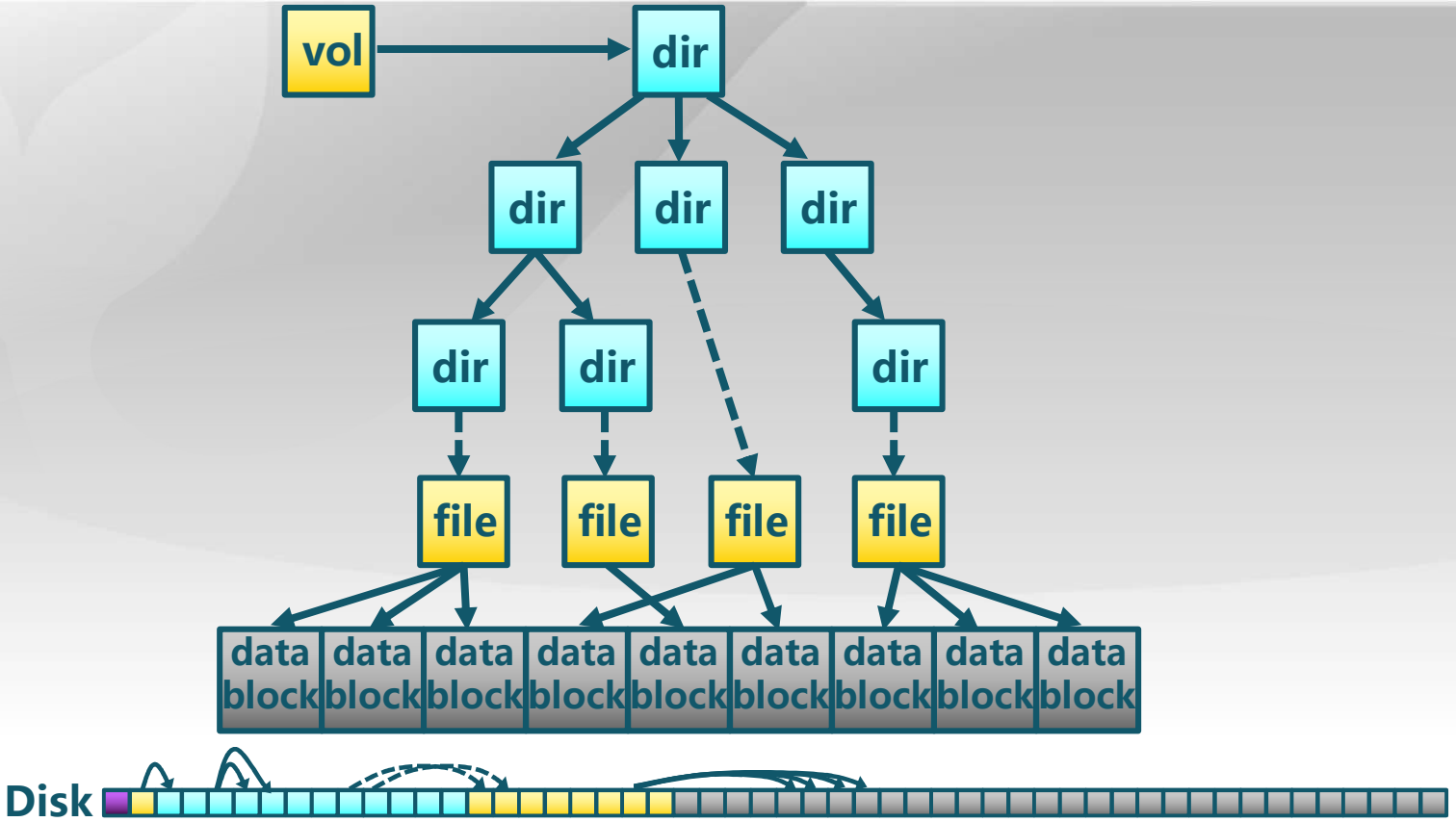
文件系统的组织视图


vol
dir
dir
dir
dir
dir
dir
dir
file
file
file
file
data block
data block
data block
data block
data block
data block
data block
data block
data block

文件系统的存储结构

- 文件系统数据结构
  - 卷控制块 (每个文件系统一个)
  - 文件控制块 (每个文件一个)
  - 目录节点(每个目录项一个)
- 持久存储在外存中
  - 存储设备的数据块中
- 当需要时加载进内存
  - 卷控制模块：当文件系统挂载时进入内存
  - 文件控制块: 当文件被访问时进入每次
  - 目录节点: 在遍历一个文件路径时进入内存

文件系统的存储视图


vol
dir
dir
dir
dir
dir
dir
dir
dir
file
file
file
file
data block
data block
data block
data block
data block
data block
data block
data block
data block
Disk

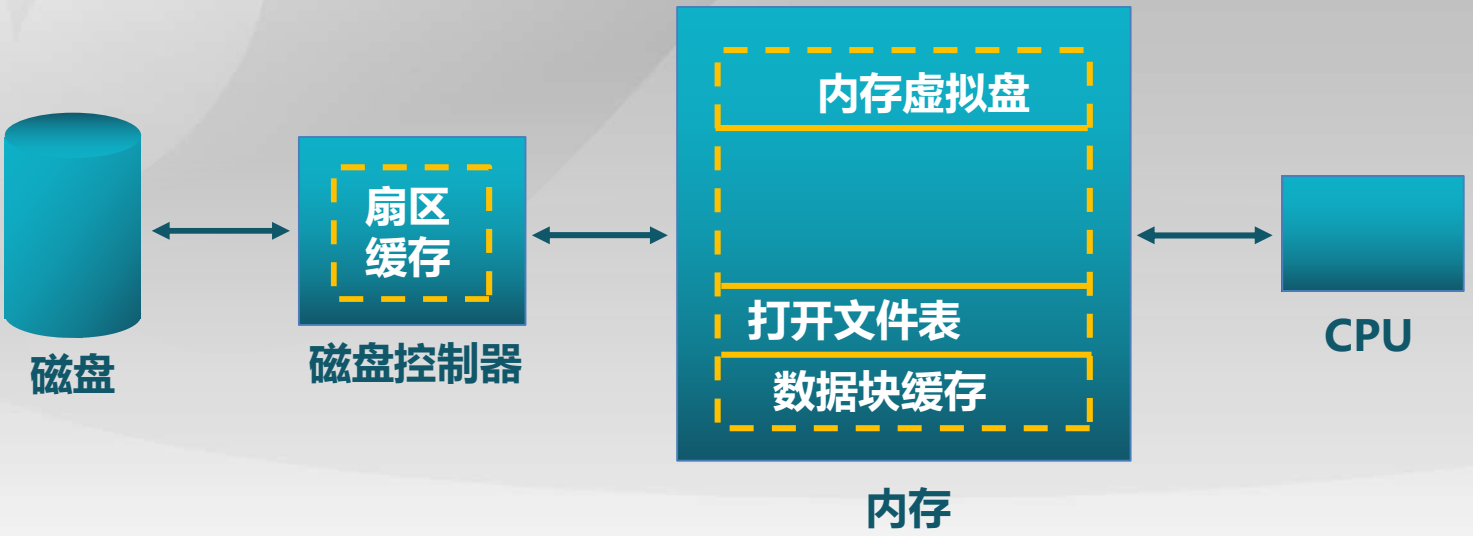
操作系统

Operating Systems

操作系统

Operating Systems

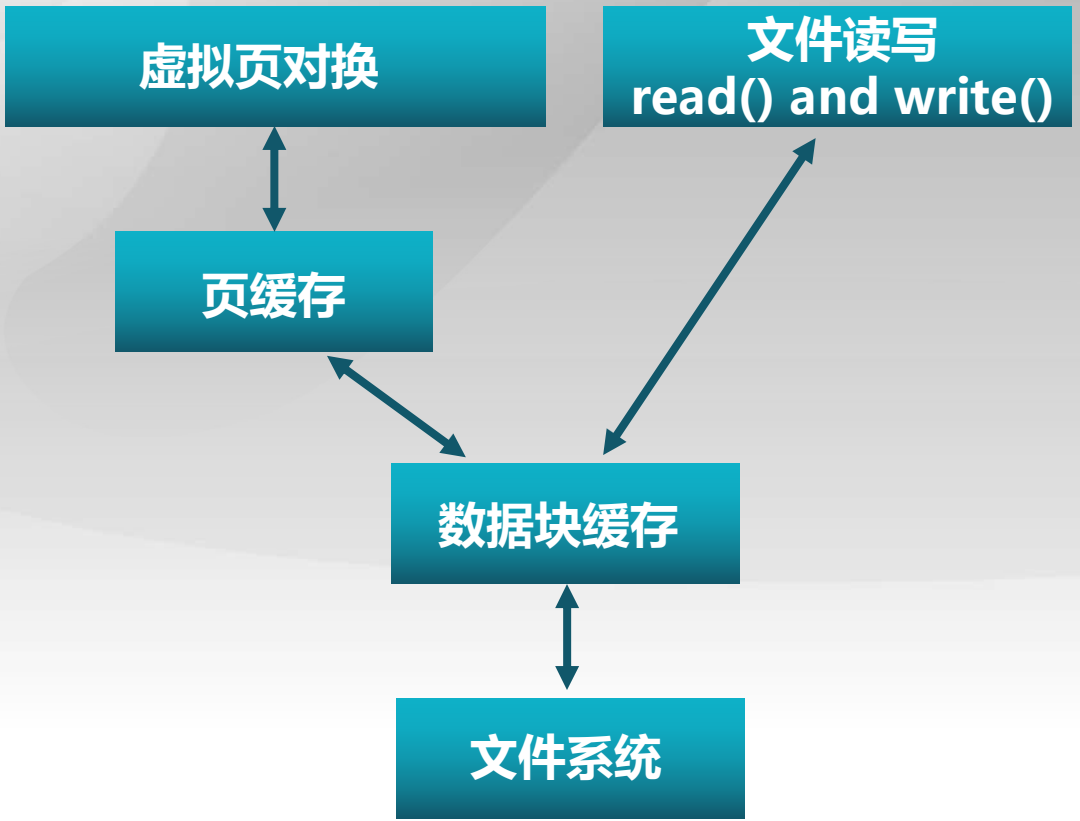
多种磁盘缓存位置



数据块缓存

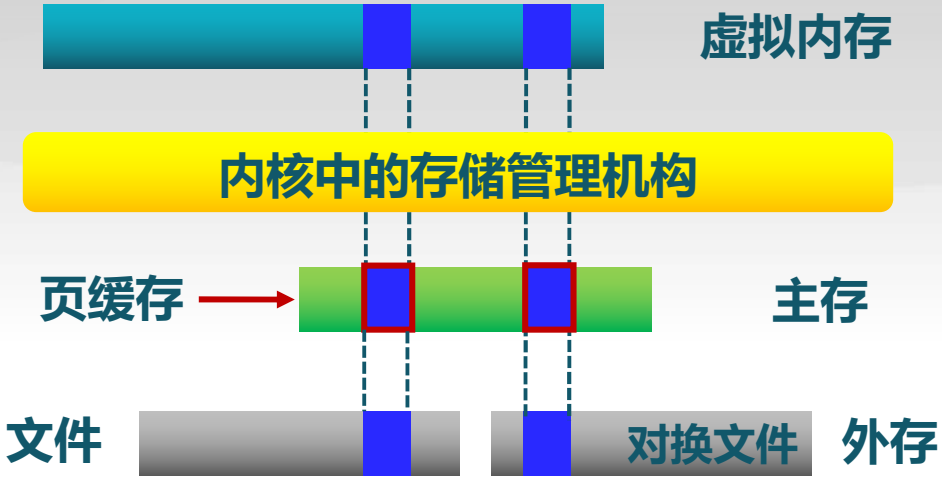
- 数据块按需读入内存
  - 提供read()操作
  - 预读: 预先读取后面的数据块
- 数据块使用后被缓存
  - 假设数据将会再次用到
  - 写操作可能被缓存和延迟写入
- 两种数据块缓存方式
  - 数据块缓存
  - 页缓存: 统一缓存数据块和内存页

数据块缓存

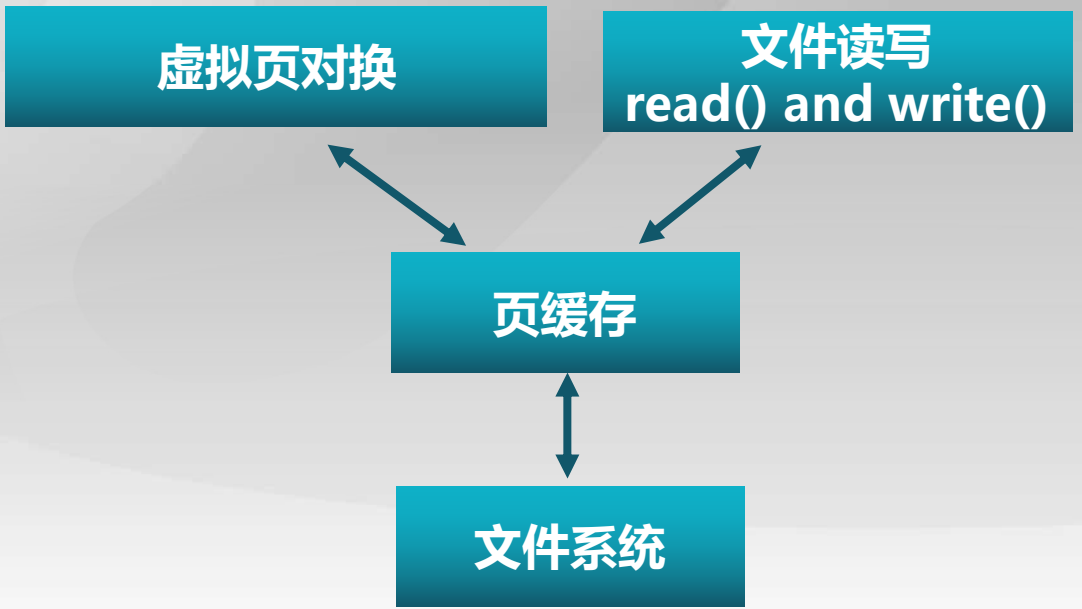


页缓存

- 虚拟页式存储
  - 在虚拟地址空间中虚拟页面可映射到本地外存文件中
- 文件数据块的页缓存
  - 在虚拟内存中文件数据块被映射成页
  - 文件的读/写操作被转换成对内存的访问
  - 可能导致缺页和/或设置为脏页
  - 问题: 页置换算法需要协调虚拟存储和页缓存间的页面数



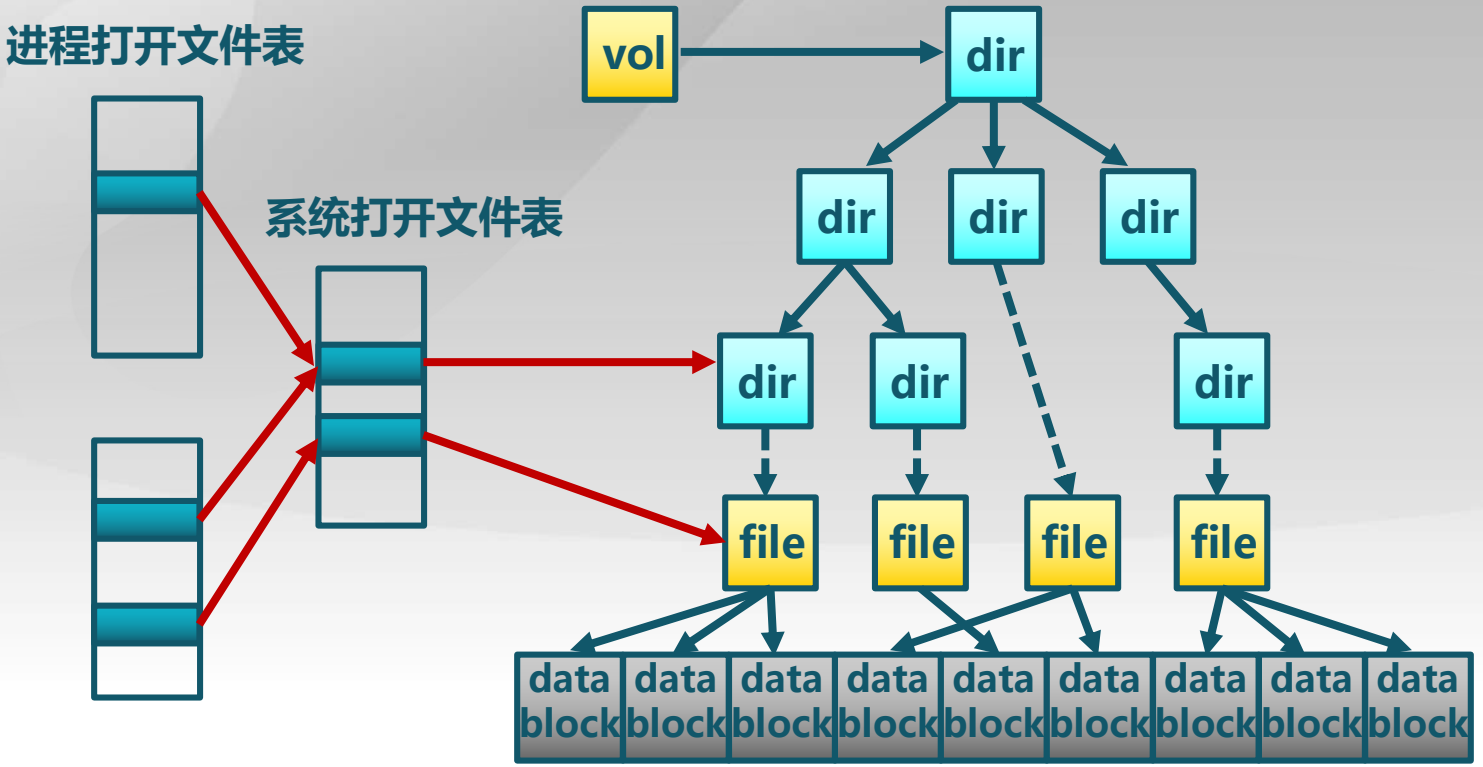
页缓存



文件系统中打开文件的数据结构

- 文件描述符
  - 每个被打开的文件都有一个文件描述符
  - 文件状态信息
    - 目录项、当前文件指针、文件操作设置等
- 打开文件表
  - 每个进程一个进程打开文件表
  - 一个系统级的打开文件表
  - 有文件被打开时，文件卷就不能被卸载

打开文件表


进程打开文件表
系统打开文件表
vol
dir
dir
dir
dir
dir
dir
dir
dir
file
file
file
file
data block
data block
data block
data block
data block
data block
data block
data block
data block

打开文件锁

- 一些文件系统提供文件锁，用于协调多进程的文件访问
  - **强制** – 根据锁保持情况和访问需求确定是否拒绝访问
  - **劝告** – 进程可以查找锁的状态来决定怎么做

操作系统

Operating Systems

操作系统

Operating Systems

文件大小

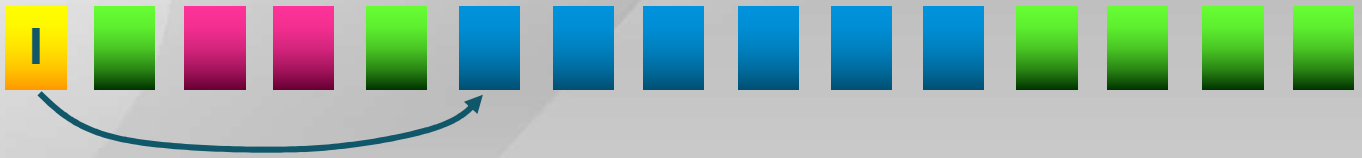
- 大多数文件都很小
  - 需要对小文件提供很好的支持
  - 块空间不能太大
- 一些文件非常大
  - 必须支持大文件 (64位文件偏移)
  - 大文件访问需要高效

文件分配

- 如何表示分配给一个文件数据块的位置和顺序
- 分配方式
  - 连续分配
  - 链式分配
  - 索引分配
- 指标
  - 存储效率：外部碎片等
  - 读写性能：访问速度

连续分配

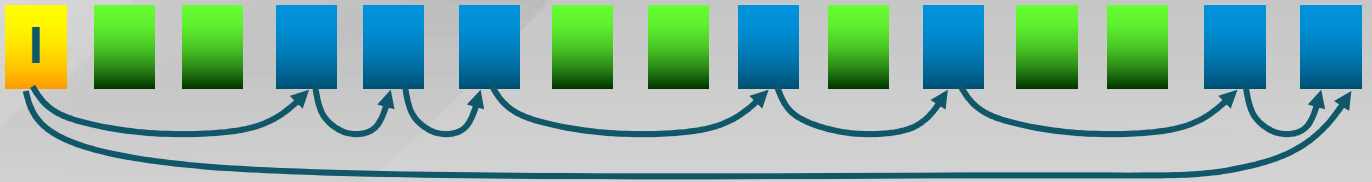
- 文件头指定起始块和长度



- 分配策略
  - 最先匹配, 最佳匹配, ...
- 优点
  - 文件读取表现好
  - 高效的顺序和随机访问
- 缺点
  - 碎片!
  - 文件增长问题
    - 预分配?
    - 按需分配?

链式分配

- 文件以数据块链表方式存储
- 文件头包含了到第一块和最后一块的指针

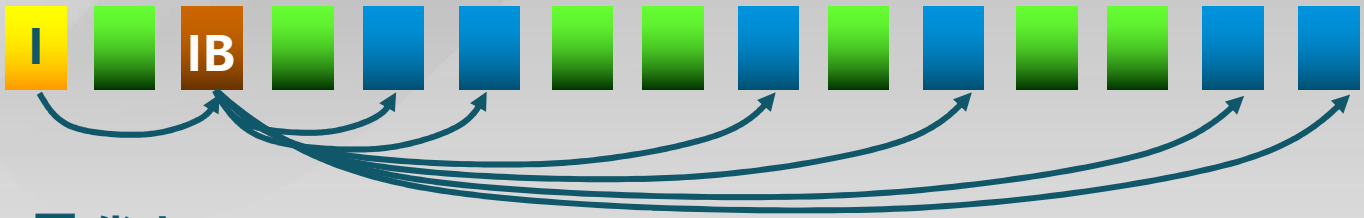


- 优点
  - 创建、增大、缩小很容易
  - 没有碎片
- 缺点
  - 无法实现真正的随机访问
  - 可靠性差

    破坏一个链，后面的数据块就丢了

索引分配

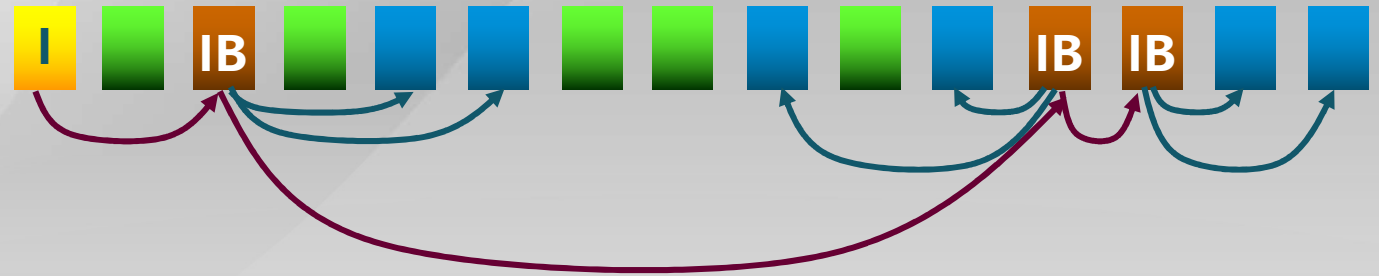
- 为每个文件创建一个索引数据块
  - 指向文件数据块的指针列表
- 文件头包含了索引数据块指针



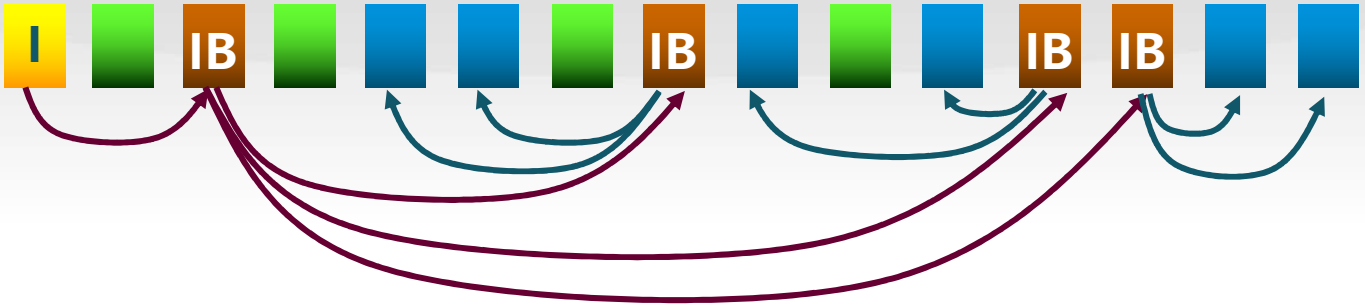
- 优点
  - 创建、增大、缩小很容易
  - 没有碎片
  - 支持直接访问
- 缺点
  - 当文件很小时，存储索引的开销
  - 如何处理大文件?

大文件的索引分配

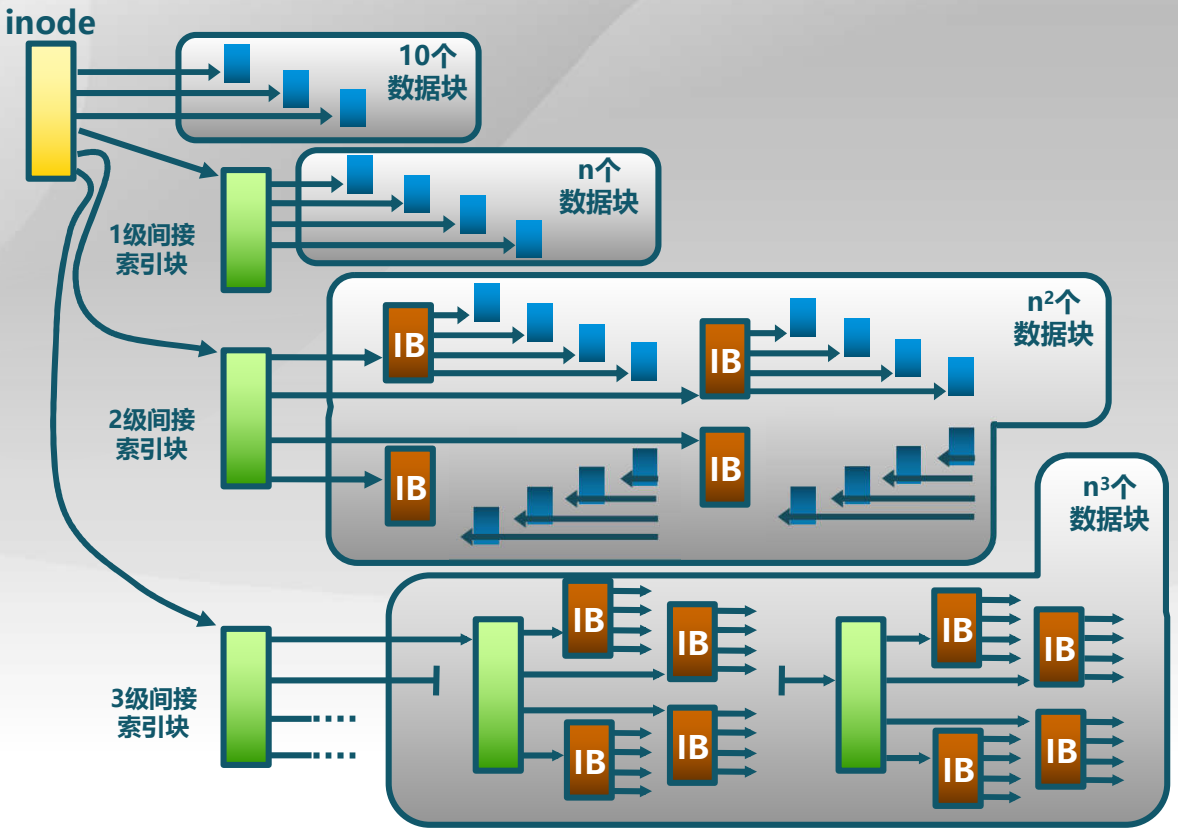
- 链式索引块 (IB+IB+...)



- 多级索引块(IB*IB *...)



UFS多级索引分配


inode
10个
数据块
n个
数据块
1级间接
索引块
n²个
数据块
IB
IB
2级间接
索引块
IB
IB
n³个
数据块
IB
IB
IB
IB
3级间接
索引块
IB
IB
IB
IB

UFS多级索引分配

- 文件头包含13个指针
  - 10 个指针指向数据块
  - 第11个指针指向索引块
  - 第12个指针指向二级索引块
  - 第13个指针指向三级索引块
- 效果
  - 提高了文件大小限制阈值
  - 动态分配数据块，文件扩展很容易
  - 小文件开销小
  - 只为大文件分配间接数据块，大文件在访问数据块时需要大量查询

操作系统

Operating Systems

操作系统

Operating Systems

空闲空间管理

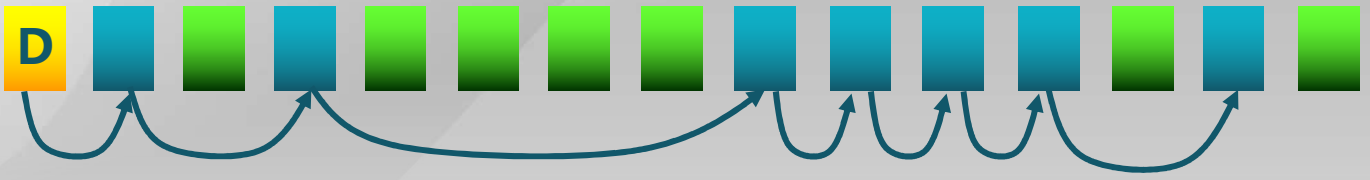
- 跟踪记录文件卷中未分配的数据块
  - 采用什么数据结构表示空闲空间列表?

空闲空间组织: 位图

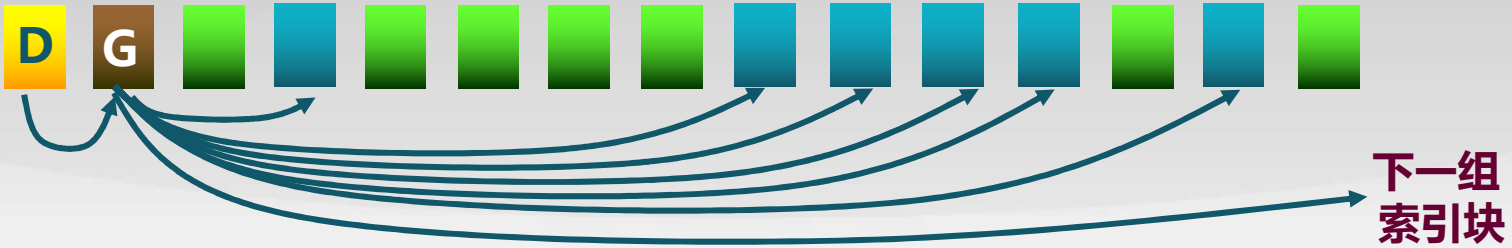
- 用位图代表空闲数据块列表
  - 1111111111111111001110101011101111...
  - D_i = 0 表明数据块i是空闲, 否则，表示已分配
- 使用简单但是可能会是一个大的很大向量表
  - 160GB磁盘-> 40M数据块-> 5MB位图
  - 假定空闲空间在磁盘中均匀分布，
    则找到“0”之前要扫描n/r
    - n = 磁盘上数据块的总数
    - r = 空闲块的数目

其他空闲空间组织方式

■ 链表



■ 链式索引



已分配数据块　　空闲数据块

操作系统

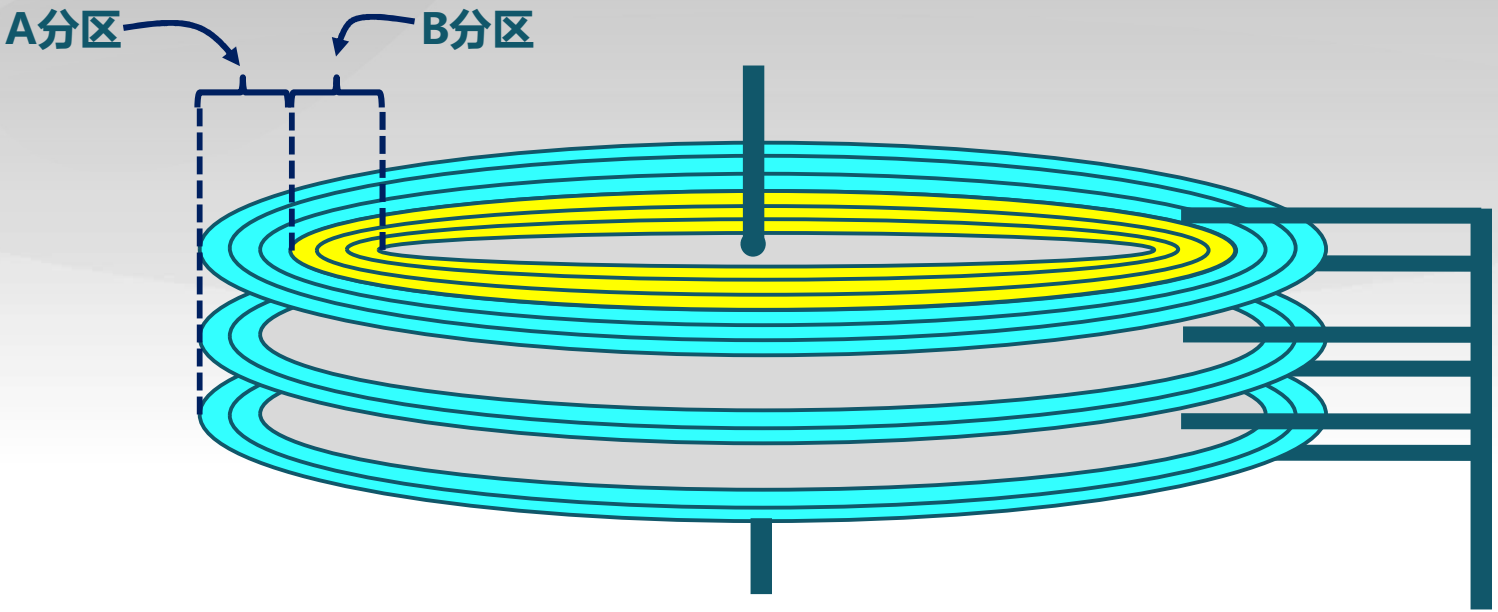
Operating Systems

操作系统

Operating Systems

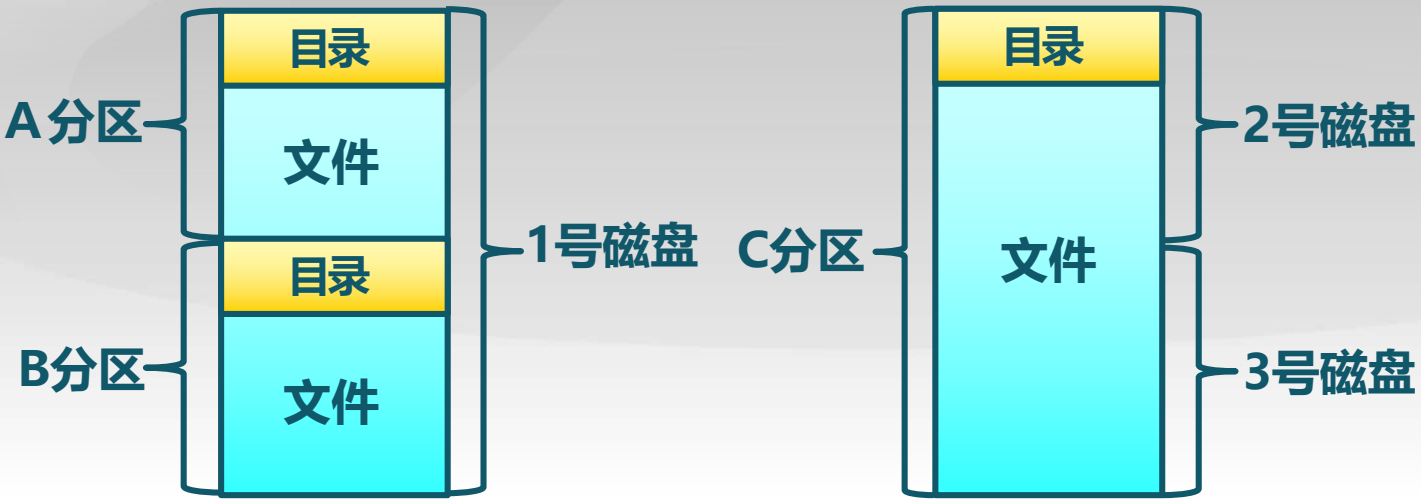
磁盘分区

- 通常磁盘通过分区来最大限度减小寻道时间
  - 分区是一组柱面的集合
  - 每个分区都可视为逻辑上独立的磁盘



一个典型的磁盘文件系统组织

文件卷：一个拥有完整文件系统实例的外存空间
通常常驻在磁盘的单个分区上

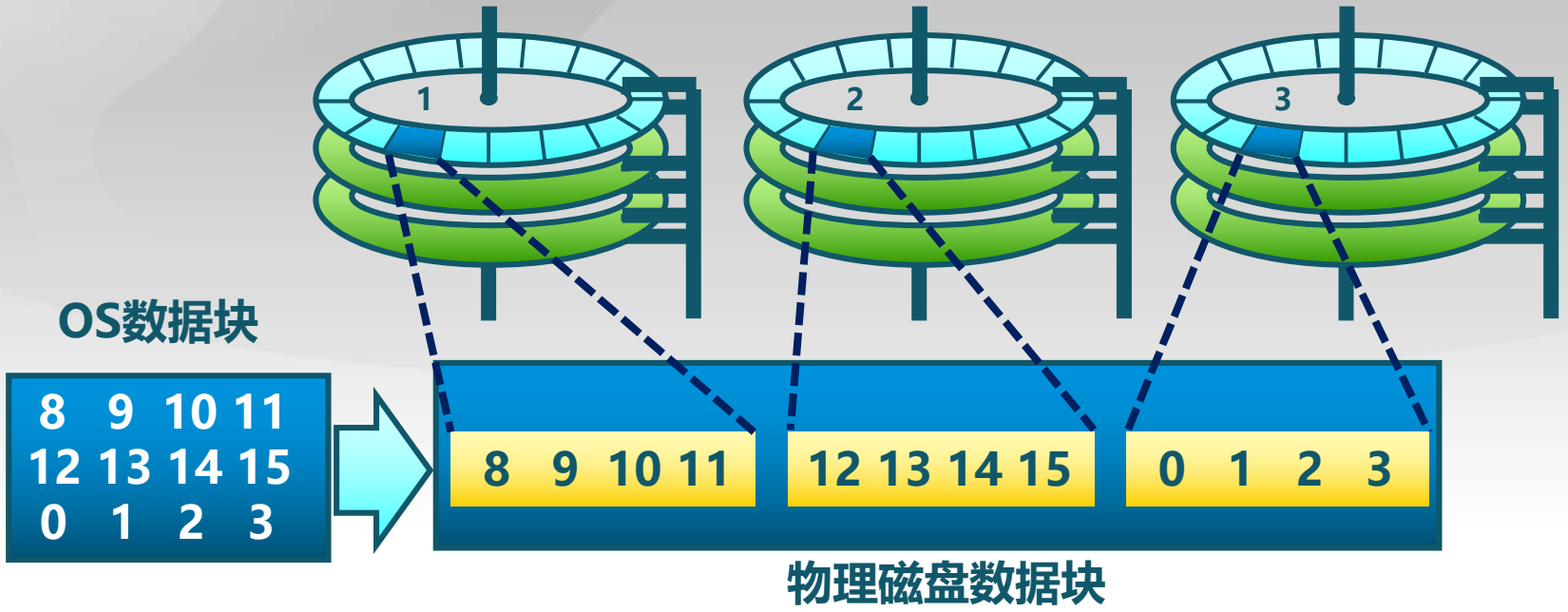


多磁盘管理

- 使用多磁盘可改善
  - 吞吐量(通过并行)
  - 可靠性和可用性 (通过冗余)
- 冗余磁盘阵列(RAID, Redundant Array of Inexpensive Disks)
  - 多种磁盘管理技术
  - RAID分类
    如, RAID-0, RAID-1, RAID-5
- 冗余磁盘阵列的实现
  - 软件：操作系统内核的文件卷管理
  - 硬件：RAID硬件控制器(I/O)

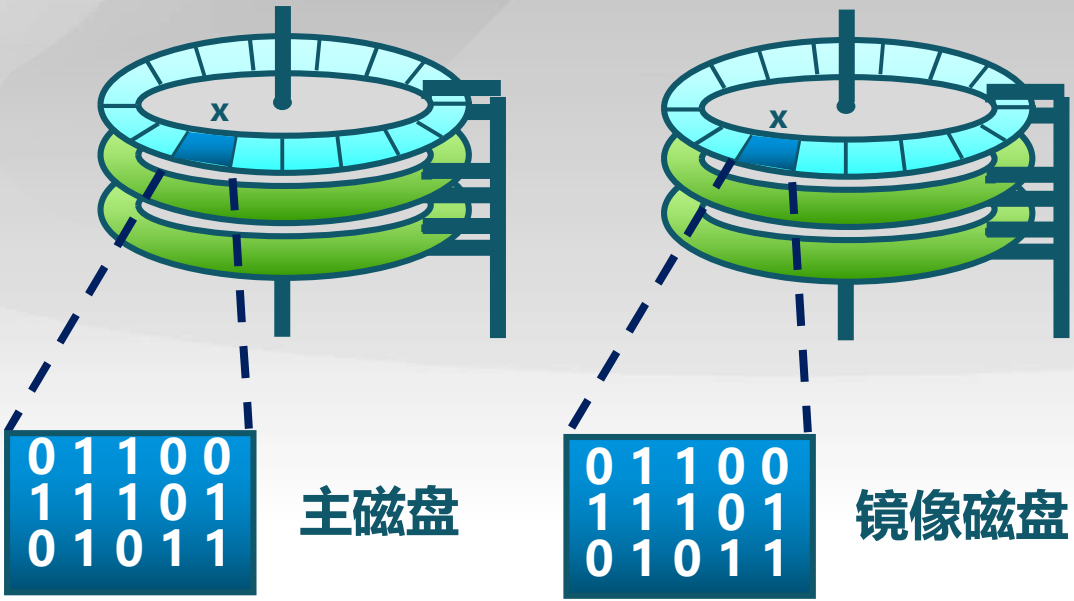
RAID-0：磁盘条带化

- 把数据块分成多个子块，存储在独立的磁盘中
  - 通过独立磁盘上并行数据块访问提供更大的磁盘带宽



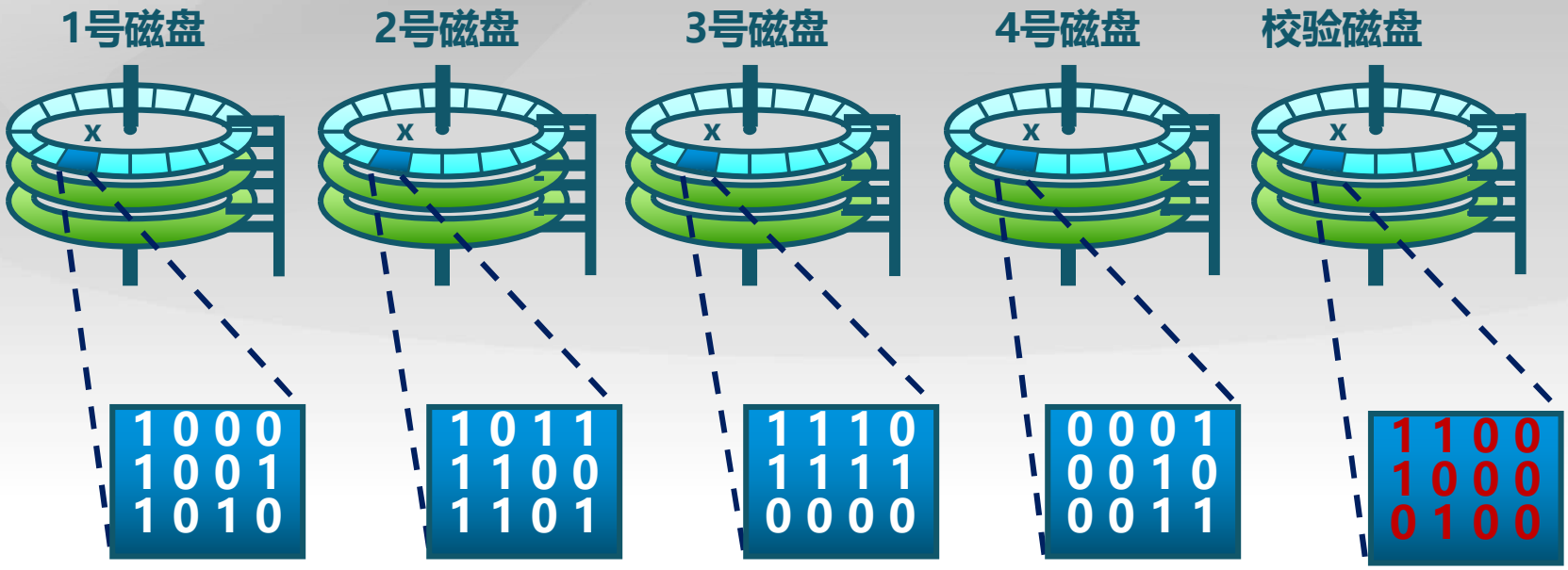
RAID-1: 磁盘镜像

- 向两个磁盘写入，从任何一个读取
  - 可靠性成倍增长
  - 读取性能线性增加

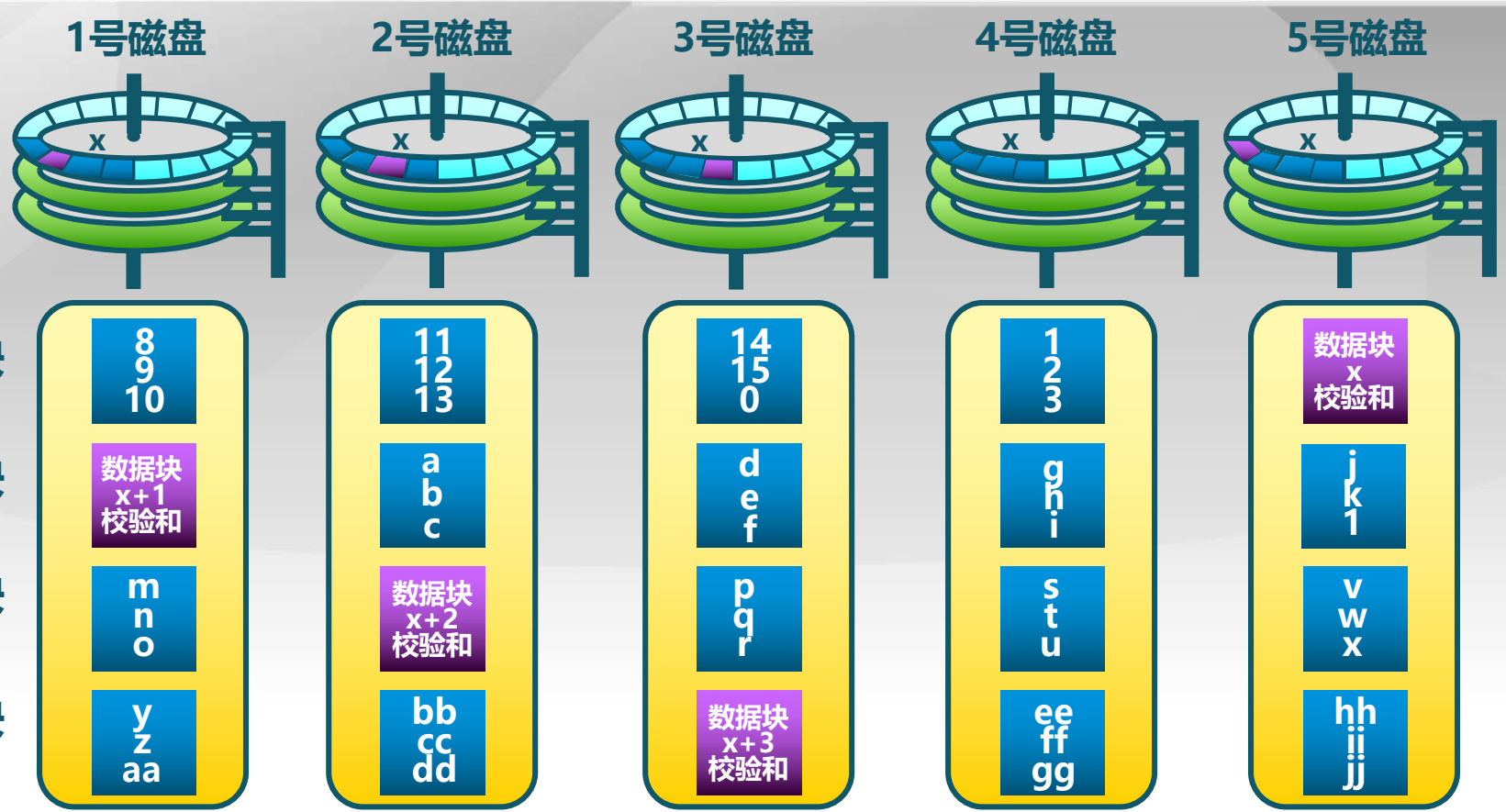


RAID-4: 带校验的磁盘条带化

- 数据块级的磁盘条带化加专用奇偶校验磁盘
  - 允许从任意一个故障磁盘中恢复

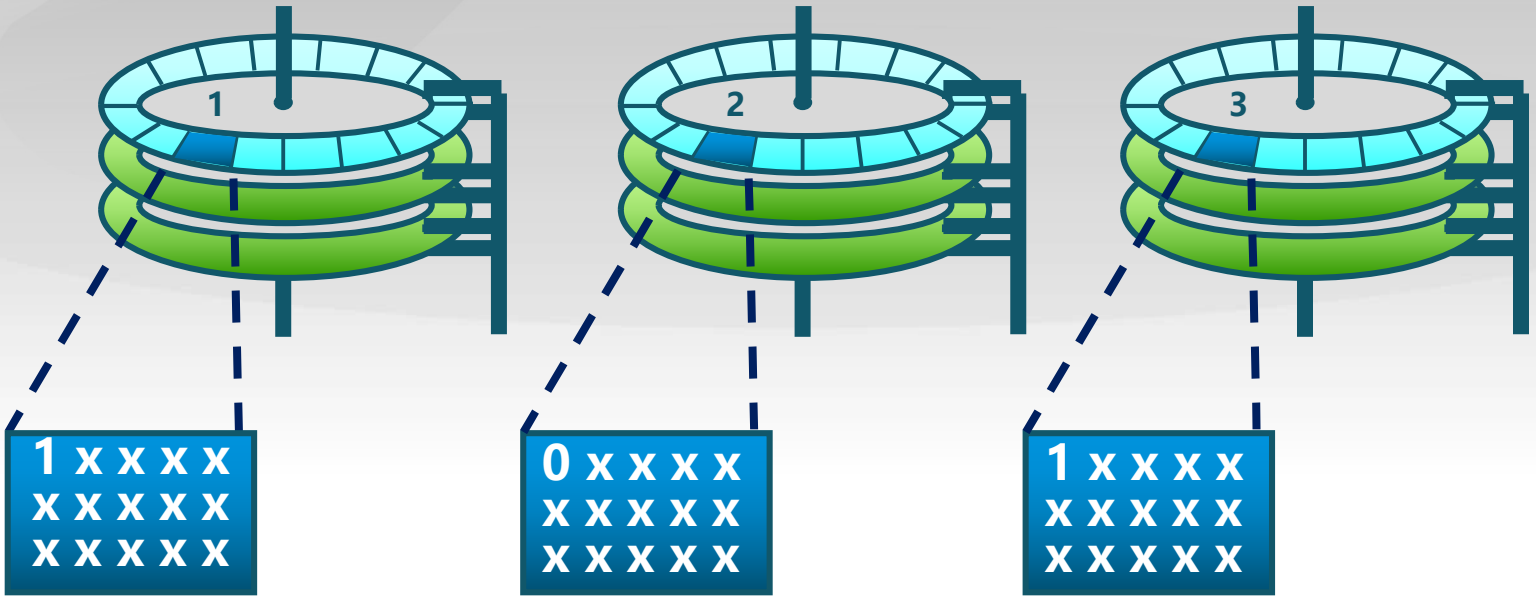


RAID-5: 带分布式校验的磁盘条带化



基于位和基于块的磁盘条带化

- 条带化和奇偶校验按“字节”或者“位”
  - RAID-0/4/5: 基于数据块
  - RAID-3: 基于位

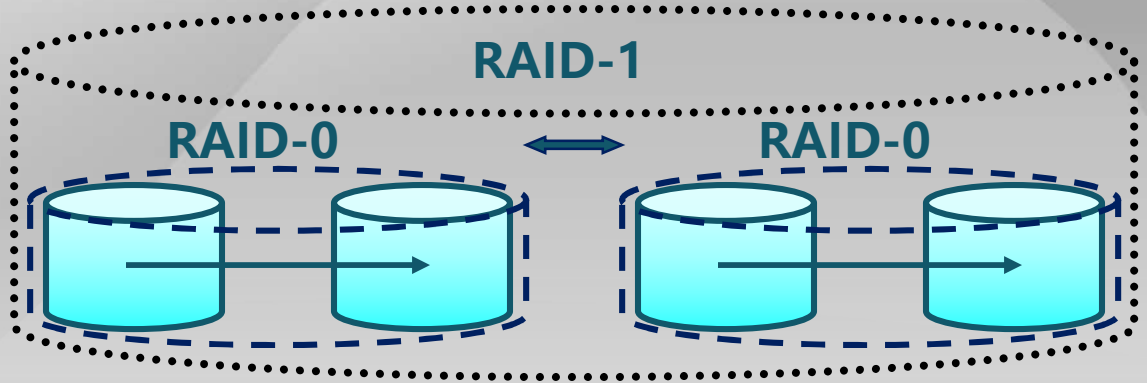


可纠正多个磁盘错误的冗余磁盘阵列

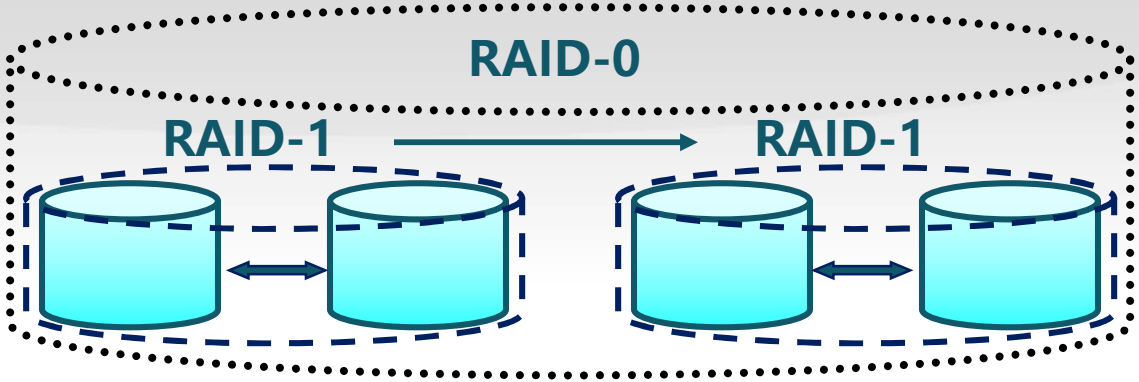
- RAID-5: 每组条带块有一个奇偶校验块
  - 允许一个磁盘错误
- RAID-6: 每组条带块有两个冗余块
  - 允许两个磁盘错误

RAID嵌套

■ RAID 0+1



■ RAID 1+0



操作系统

Operating Systems