

13 - 15 File System

文件：在连续逻辑地址空间上存放的文件（实际上物理地址空间大多数也是连续的）。

文件锁：一个进程正在操作文件时，需要给文件加锁；

- **强制锁** (Mandatory Lock)
- **指导锁** (Advisory Lock)

索引 (index)：保存文件的元数据 (metadata)。

文件和数据的关系：文件是数据的一个抽象的逻辑视图。

目录：通常用树形来组织目录。

- 单机：B+树；
- 分布式：基于日志结构的合并树 (Log Structured Merge Tree (LSM Tree))。

无环图目录 (Acyclic-Graph Directory)：不同的目录可以共享同一个文件，也可以共享同一个目录。目录与子目录、子文件的连接通过指针实现。

挂载 (mount)：一个文件系统必须先挂载才能被访问，挂载的地方称为挂载点。

权限列表

- 三种对文件的访问模式：
 - 读 (read)；
 - 写 (write)；
 - 执行 (execution)。
- 访问也有三种来源：
 - Owner access：文件的所有者（用户）；
 - Group access：文件的所有者所在的用户组；
 - Public access：其他用户及用户组。
- 用 `chmod` 命令来进行修改；`RWX` 依次用三位二进制表示（看成一位 0~7 十进制）；访问来源依次连接，最后组成3位 0~7 的十进制组成。

文件系统架构

- 应用程序发送请求 (Application) ；
- 逻辑（虚拟）文件系统 (VFS, Virtual File System) ；
- 文件组织模块 (File Data Structure) ；
- 真实文件系统 (NTFS/EXT) ；
 - 数据管理 (data)；
 - 元数据管理 (metadata)；
 - 卷管理 (volume)；
 - 备份/恢复/访问控制.....
- I/O 控制单元（设备驱动） ；
- 设备（如硬盘等）

关键数据结构

- 超级块 / 卷 (superblock)：NTFS/EXT 层级管理；
- 目录项 (dentry)：directory 层级管理；
- 元数据 (inode)：metadata 文件属性；

- 文件 (file): file 文件管理

目录实现方式

- 线性表, B+树;
- 哈希表: 需要考虑哈希碰撞。

连续空间分配 (Contiguous Allocation)

- 文件记录逻辑地址起始块号以及长度。
- 如何得到具体的块号以及块内偏移?

$$Q = LA / PS, \quad R = LA \bmod PS$$

LA 表示逻辑地址, Q 表示块号, P 表示块内偏移, PS 表示块大小。

链式空间分配 (Link Allocation)

- 文件记录逻辑地址起始块号以及终止块号。
- 块内需要留1个字节存指向下一个块的指针。

索引分配 (Indexed Allocation)

- 将所有块的信息记录在一个索引块中;
- 对于文件来说, 只需要记录索引块号即可;
- 直接映射, 两级映射, 三级映射 (类似页表)。

空余空间管理 (Free-Space Management):

- **位向量表 / 位图** (bit vector / bit map)
 - 对每一块用一个二进制位表示状态, 0 表示占用 (不一定占满) 而 1 表示空闲;
 - 优化: 一个二进制位表示多个块, 减小存储占用。
- **空闲页链表** (Linked Free Space List): 将空闲页用链表串联。

备份 (Backup and Recovery): 文件系统提供的备份以及恢复机制。

文件系统组织结构: 池 - 卷 - 分区 - 目录 & 文件

虚拟文件系统: 提供不同的文件系统的统一接口 (如不同磁盘以及U盘、远程文件系统等等, 不同组织形式 NTFS/EXT/NFS.....), 承上启下, 上层的 API 不需要进行修改。