

inode

全称为 index node，索引节点

记录一个文件对应的所有存储块号（即存储的索引），每个 inode 对应一个文件。

如果 inode 记录所有块的索引，会导致 inode 是变长的，为了避免这样，所以限定 inode 长度，让每个 inode 中有直接指针，间接指针，二级间接指针等。

inode 中还会记录文件模式，文件链接数，文件大小等信息

inode 号与文件一一对应，通过添加一层文件名到 inode 号的映射，实现文件名与文件存储位置的解耦（inode 号可能与位置相关）

目录

目录是特殊的文件，其中保存的是目录项。

记录文件名到 inode 的映射。

硬链接 (hard link)

e.g. `ln file link`: 为文件 `file` 创建另一个名字 `link`，这里 `link` 就是 `file` 的硬链接

创建硬链接时，文件系统不会创建新的 inode，而是找到原来的 inode，同时在目标路径的父目录下添加一个指向此 inode 的新目录项。

- 通过任何一个硬链接文件名做出的修改都会影响所有
- inode 会记录链接数，链接数为 0 才能删除

当创建一个新的目录，由于 `.` 映射到自己，`..` 映射到父目录，同时父目录中还会有映射到自己的文件名，所以自己的链接数为 2，父目录的链接数加 1

软连接 (soft link) / 符号链接 (symbolic link)

e.g. `ln -s file slink`

常规文件保存数据，目录文件保持目录项。符号链接文件保存的是一个字符串，表示一个文件路径。

符号链接文件除了可以修改基本信息（文件模式，创建用户等），只支持读取操作，找到这个文件的 inode 然后返回其中的路径。

支持软连接的文件系统在查询目录时如果遇到符号链接文件需要先解析符号链接中的路径找到目标文件再继续。

有什么作用？类似重定向？可以通过软连接修改源文件吗？

可以，相当于找到那个文件再修改

区别

软连接中可以是一个不存在的路径，硬链接不行

硬链接要求目标文件不能是目录

不然可能存在 `/a/a/link` 是 `/a` 的硬链接，导致访问时出现环，导致无限递归

软连接可以跨文件系统，硬链接不行

存储布局

超级块 → 块分配信息 → inode 分配信息 → inode 表 → 文件数据块

超级块

记录整个文件系统的全局元数据

- 魔法数字 (magic number)：可以知道文件系统类型和存储布局
- 文件系统版本，文件系统管理的空间大小，最后一次挂载时间，统计信息等
- 统计信息：可支持的最大 inode 数，当前空闲 inode 数，可支持的最大块数。。。

块分配信息 & inode 分配信息

位图 (bitmap)

inode 表

数组形式保存 inode，文件系统创建时确定大小