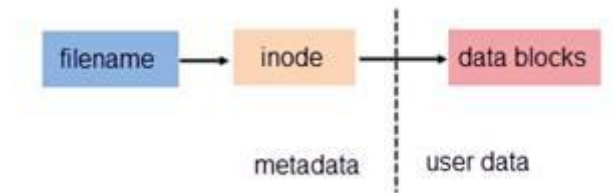


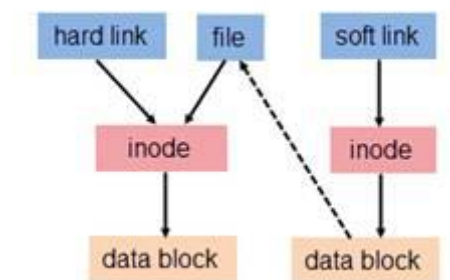
# Linux 文件系统相关

## 硬链接与软链接

通过文件名打开文件



## 硬链接与软链接



由于硬链接是有着相同 inode 号仅文件名不同的文件,因此硬链接存在以下几点特性:

- 文件有相同的 inode 及 data block;
- 只能对已存在的文件进行创建;
- 不能交叉文件系统进行硬链接的创建;
- 不能对目录进行创建,只可对文件创建;
- 删除一个硬链接文件并不影响其他有相同 inode 号的文件。

软链接与硬链接不同,若文件用户数据块中存放的内容是另一文件的路径名的指向,则该文件就是软连接。软链接就是一个普通文件,只是数据块内容有点特殊。软链接有着自己的 inode 号以及用户数据块(见图 2.)。因此软链接的创建与使用没有类似硬链接的诸多限制:

- 软链接有自己的文件属性及权限等;
- 可对不存在的文件或目录创建软链接;
- 软链接可交叉文件系统;
- 软链接可对文件或目录创建;
- 创建软链接时,链接计数 i\_nlink 不会增加;

删除软链接并不影响被指向的文件,但若被指向的原文件被删除,则相关软连接被称为死链接(即 dangling link,若被指向路径文件被重新创建,死链接可恢复为正常的软链接)。

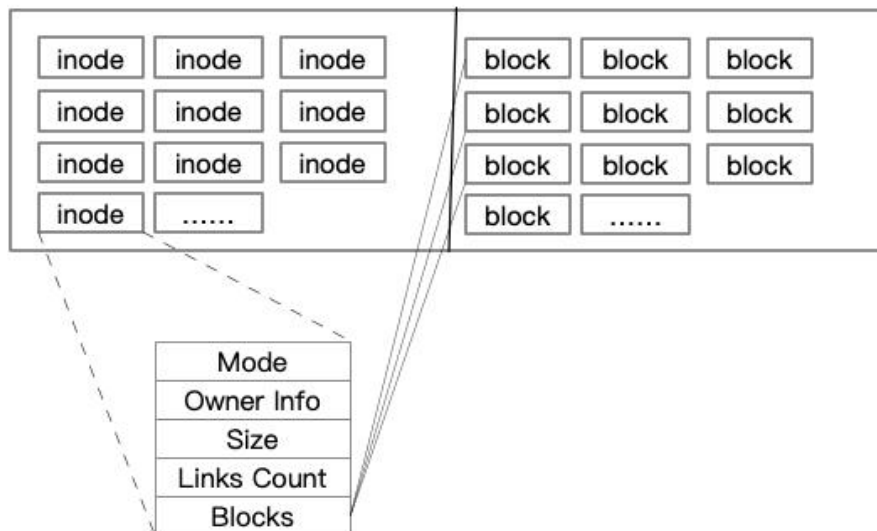
## Linux 顶层目录结构

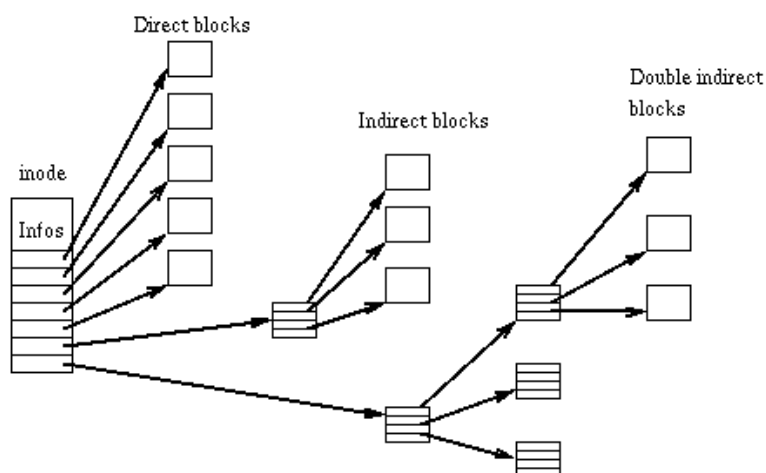
/ 根目录

— bin	存放用户二进制文件
— boot	存放内核引导配置文件
— dev	存放设备文件
— etc	存放系统配置文件
— home	用户主目录
— lib	动态共享库
— lost+found	文件系统恢复时的恢复文件
— media	可卸载存储介质挂载点
— mnt	文件系统临时挂载点
— opt	附加的应用程序包
— proc	系统内存的映射目录，提供内核与进程信息
— root	root 用户主目录
— sbin	存放系统二进制文件
— srv	存放服务相关数据
— sys	sys 虚拟文件系统挂载点
— tmp	存放临时文件
— usr	存放用户应用程序
— var	存放邮件、系统日志等变化文件

文件系统 inode 与 data block 等

## filesystem





## 寻找文件的过程

当文件系统拿到一个路径，是怎么找到对应的内容的呢？如 `/home/root/a.txt`。

首先文件系统会从 #2 inode 开始（注意这里说的经典的文件系统，不同的文件系统开始的 inode 可能不同，但为啥是 2 呢？），也就是 `/`，找到 #2 inode 指向的 block 读到 `/` 下面的文件列表。然后找到 `home` 对应的 inode。然后通过 inode 指向的 block，打开 `/home` 下的文件列表，找到 `root` 对应的 inode，找到 inode 指向的 block，读到 `a.txt` 对应的 block，最后读到 block 存储的内容，就是 `a.txt` 的内容啦。

## Inode 的一些 Tips

移动文件、重新名等操作，实际上只是修改了 parent directory 的 block 中存储的 filename  $\rightarrow$  inode 映射，所以跟文件多大是没有关系的，都是  $O(1)$  复杂度；而且这也不影响 inode 的号码。

理论上，当文件创建，inode 分配，就不会再变了。移动、重命名、写入文件、截断文件都只是修改 inode 的元信息或者修改 parent directory 的内容，inode 总是不变。所以像日志收集程序，使用 inode 来区分日志文件，是非常可靠的。日志被 rotate 了也可以做到不重不漏。

如果你知道一个 inode，怎么找出这个文件呢？`$ find . -inum 23423 -print`

但是直接操作 inode 是不允许的，Kernel 提供的 syscall 只能通过 filename 来操作文件。理由同上，文件系统 corrupt 了是很危险的。

文件系统里面有一个文件带着古怪的名字，怎么删除？可以通过 inode 删啊。  
`$ find . -inum 234234 -delete .`