

每个文件系统包含三部分：

**superblock:** 记录此 filesystem 的整体信息，包括 inode/block 的总量、使用量、剩余量，以及文件系统的格式与相关信息等；

**inode:** 记录文件的属性，一个文件占用一个 inode，同时记录此文件的数据所在的 block 号码；

**block:** 实际记录文件的内容，若文件太大时，会占用多个 block 。

由于 Ext2 是索引式文件系统，基本上不太需要常常进行碎片整理的。

文件系统一开始就将 inode 与 block 规划好了，除非重新格式化(或者利用resize2fs 等指令变更文件系统大小)，否则 inode 与 block 固定后就不再变动。

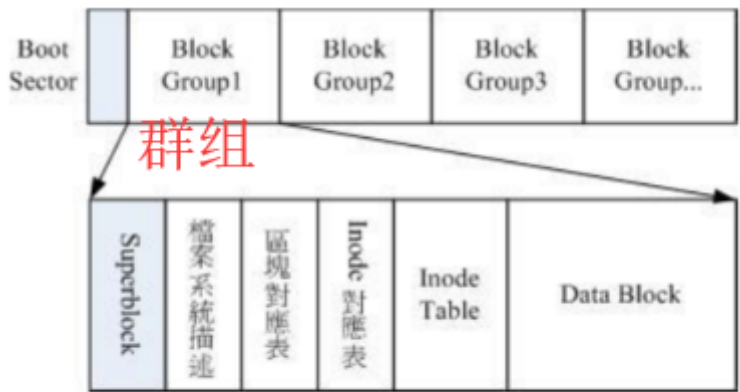


图 7.1.3、ext2 文件系统示意图 (注1)

文件系统最前面有一个启动扇区(boot sector)，这个启动扇区可以安装开机管理程序，这是个非常重要的设计，因为如此一来我们就能够将不同的开机管理程序安装到个别的文件系统最前端，而不用覆盖整颗磁盘唯一的 MBR。

每个区块群组 (block group) 的6个主要内容说明如下：

1. data block：存放真正文件内容的地方

在 Ext2 文件系统中所支持的 block 大小有 1K, 2K 及4K 三种而已，这些在格式化时就固定了，且每个都有编号，方案inode进行记录。

Block 大小	1KB	2KB	4KB
最大单一文件限制	16GB	256GB	2TB
最大文件系统总容量	2TB	8TB	16TB

- 原则上，block 的大小与数量在格式化完就不能够再改变了(除非重新格式化)；
- 每个 block 内最多只能够放置一个文件的数据；
- 承上，如果文件大于 block 的大小，则一个文件会占用多个 block 数量；
- 承上，若文件小于 block ，则该 block 的剩余容量就不能够再被使用了(磁盘空间会浪费)。

## 注意：

若block大小较大，则会造成更多的空间浪费，若过小，则每个文件会占用较多的block，inode记录的更多，造成文件系统的读写不良性能。

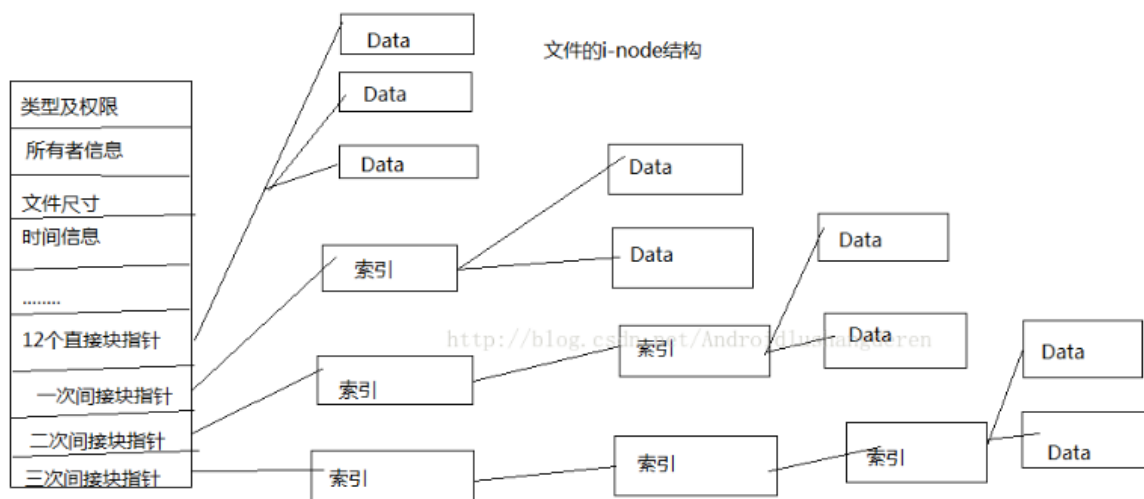
## 2. node table (inode 表格)：记录文件内容信息

主要记录一下文件的内容：

- 该文件的存取模式(read/write/excute)；
- 该文件的拥有者与群组(owner/group)；
- 该文件的容量；
- 该文件建立或状态改变的时间(ctime)；
- 最近一次的读取时间(ctime)；
- 最近修改的时间(mtime)；
- 定义文件特性的旗标(flag)，如 SetUID...；
- 该文件真正内容的指向 (pointer)

inode主要作用：

- 每个 inode 大小均固定为 128 bytes (新的 ext4 与 xfs 可设定到 256 bytes)；inode 记录一个 block 号码要花费 4byte
- 每个文件都仅会占用一个 inode 而已；
- 承上，因此文件系统能够建立的文件数量与 inode 的数量有关；
- 系统读取文件时需要先找到 inode，并分析 inode 所记录的权限与用户是否符合，若符合才能够开始实际
- 读取 block 的内容。



因此对于block为1K的时：

1. 12 个直接指向：  $12 \times 1K = 12K$  由于是直接指向，所以总共可记录 12 笔记录，因此总额大小为如上所示；

2. 间接：  $256 \times 1K = 256K$

每笔 block 号码的记录会花去 4bytes，因此 1K 的大小能够记录 256 笔记录，因此一个间接 可以记录的文件大小如上；

3. 双间接：  $256 \times 256 \times 1K = 256^2 K$

第一层 block 会指定 256 个第二层，每个第二层可以指定 256 个号码，因此总额大小如上；

4. 三间接：  $256 \times 256 \times 256 \times 1K = 256^3 K$

第一层 block 会指定 256 个第二层，每个第二层可以指定 256 个第三层，每个第三层可以指定 256 个号码，因此总额大小如上；

总额：将直接、间接、双间接、三间接加总，得到  $12 + 256 + 256 \times 256 + 256 \times 256 \times 256 (K) = 16GB$

### 3. 超级区块 (superblock)

Superblock 是记录整个 filesystem 相关信息的地方， 没有 Superblock ，就没有这个 filesystem 了。

他记录的信息主要有：

- block 与 inode 的总量；

- 未使用与已使用的 inode / block 数量;
- block 与 inode 的大小 (block 为 1, 2, 4K, inode 为 128bytes 或 256bytes);
- filesystem 的挂载时间、最近一次写入数据的时间、最近一次检验磁盘 (fsck) 的时间等文件系统的相关信息;
- 一个 valid bit 数值, 若此文件系统已被挂载, 则 valid bit 为 0, 若未被挂载, 则 valid bit 为 1。

每一个群组都可能含有超级区块, 但是后面的仅是第一个的备份而已。

#### 4. 文件系统描述说明 Filesystem Description

这个区段可以描述每个 block group 的开始与结束的 block 号码, 以及说明每个区段 (superblock, bitmap, inode map, data block) 分别介于哪一个 block 号码之间。这部份也能够用 dumpe2fs 来观察的

5. block bitmap 记录哪个block是空的。

6. node bitmap (inode 对照表)