inode与软硬连接

学习参考阮一峰老师的blog..

文件系统的存储区域可以分为两部分连续的存储区域。一部分用来保存文件中的实际内容数据，这一部分存储区域的最小存储单位叫做扇区，每一个扇区的大小为0.5kb，每8个扇区构成一个4kb字节的块，块是文件读写时的基本单位。

另一部分用来保存文件储存的原信息，比如这个文件的数据内容实际储存在哪里，文件创建的时间，文件大小等，这类似于指针的概念。这种储存文件原信息的区域叫作inode，一个文件系统中的inode总数是固定的（给inode分配的存储空间是固定的，而一个inode的大小也是固定的），他们连续存储，构成一张“inode表”。每一个文件都对应一个inode数据，并用一个整数值来标识，这个整数被称为文件的inode号码。通过inode号码，可以索引查找inode表，进而找到文件数据实际储存的位置。由于每个文件都需要有一个inode，而inode的总数是固定的，因此可能会发生硬盘还有空间，而inode存满了导致无法创建新文件的情况。

通过`stat`命令，可以查看文件的inode信息，mac和linux略有区别（终端界面可以通过更改PS1变量来设置）：

```
(base) shen@Pinball, 1973🍄[10:29:24]:~
$ stat -x lolo.jpg
  File: "lolo.jpg"
  Size: 509217       FileType: Regular File
  Mode: (0777/-rwxrwxrwx)         Uid: (  501/santashen)  Gid: (   20/   staff)
Device: 1,4   Inode: 12895437712    Links: 1
Access: Wed May  5 14:22:25 2021
Modify: Sun Apr 18 09:42:02 2021
Change: Sun Apr 18 09:56:18 2021
```

通过df命令，可以查看文件系统磁盘空间的使用情况（mac和linux显示略有不同）

```
(base) shen@Pinball, 1973🍄[10:36:34]:~
$ df -h
Filesystem       Size   Used  Avail Capacity iused      ifree %iused  Mounted on
/dev/disk1s5s1  233Gi   14Gi   50Gi    23%  559993 2447541327    0%   /
devfs           195Ki  195Ki    0Bi   100%     674          0  100%   /dev
/dev/disk1s4    233Gi  2.0Gi   50Gi     4%       5 2448101315    0%   /System/Volumes/VM
/dev/disk1s2    233Gi  464Mi   50Gi     1%    1759 2448099561    0%   /System/Volumes/Preboot
/dev/disk1s6    233Gi  3.4Mi   50Gi     1%      19 2448101301    0%   /System/Volumes/Update
/dev/disk1s1    233Gi  166Gi   50Gi    77% 1413684 2446687636    0%   /System/Volumes/Data
map auto_home     0Bi    0Bi    0Bi   100%       0          0  100%   /System/Volumes/Data/home
```

其中以i开头的就是inode的使用情况。

在unix系统中，不使用文件名，而是使用inode号码来标识文件。通过文件名打开文件，实际上的发生的过程是：系统先根据文件名找到对应的inode号码；通过inode号码，通过inode表索引，找到文件的inode信息；最后根据inode信息，找到文件数据存储的位置。一个文件系统对象可以有多个别名，但只能有一个inode，这还是类似于指针的概念。

那么文件名和inode的对应关系是如何构建的呢？这需要理解在unix系统中，一切都是文件，为了区分文件的类型，把他们分为普通文件、目录文件、连接文件和设备文件。实际上unix系统中的目录也是一种文件，打开目录，实际上就是打开目录文件。目录文件的结构是一系列目录项的列表，每个目录项由两部分组成：所包含文件（包含的字目录也是文件）的名字，以及对应的inode。所以实际上我们所谈论的一个文件，实际上就是目录文件中的一个目录项。多个文件可以有同一个inode，他们都指向同一片内容。因此在unix系统中打开一个文件后，系统就完全用inode号码来标识这个文件，而不再考虑文件名了。

在unix系统中，如果两个文件具有相同的inode号码，那么称他们是硬连接关系。注意这里说文件A是文件B的硬连接是不准确的，因为本质上他们都是指向同一个文件数据存储区域，因此他们是平级的。连接分为硬连接和软连接两种，硬连接就如上所说的，具有相同的inode号，指向同一片数据存储区域，因此相当于指针；而软连接相当于指针的指针，如果建立文件B的软连接文件C，那么在读取文件C时，系统实际上先根据文件C找到文件B，然后根据文件B找到实际文件数据存储的位置。这意味着如果删掉文件B，那么时无法通过文件C找到实际文件数据的。可以通过`ln -s source object`来创建源文件的软连接。

创建一个目录时，实际上做了三件工作：

1. 在父目录文件中，增加一个条目
2. 为这个条目分配一个inode
3. 分配一个存储快，用来保存当前被创建目录包含的文件与子目录文件

创建的字目录文件中自动生成两个字目录文件，名称分别是"."和".."，前者具有与该目录相同的inode号码，因此实际上就是该目录的一个连接。后者的inode号码就是该目录父目录的inode号码。

```
(base) shen@Pinball, 1973[11:10:29]:~
$ stat -x dir1
  File: "dir1"
  Size: 64          FileType: Directory
  Mode: (0755/drwxr-xr-x)         Uid: (  501/santashen)  Gid: (   20/   staff)
Device: 1,4   Inode: 12898986811    Links: 2
Access: Tue May 25 11:10:29 2021
Modify: Tue May 25 11:10:29 2021
Change: Tue May 25 11:10:29 2021
```

新创建一个dir1目录，父目录中有一个dir1条目，dir内有一个"."条目，因此它的连接数为2。

在dir1中创建file1，通过`ln file1 file2`创建一个硬连接`file2`，再通过`ln -s file1 file3`创建一个软连接`file3`，再通过`cp file1 file4`复制一个`file4`，可以看到结果如下：

```
(base) shen@Pinball, 1973🍄[11:32:00]:~/dir1
$ ls -li
total 0
12898986961 -rw-r--r--  2 santashen  staff  0  5 25 11:19 file1
12898986961 -rw-r--r--  2 santashen  staff  0  5 25 11:19 file2
12898986975 lrwxr-xr-x  1 santashen  staff  5  5 25 11:20 file3 -> file1
12898987329 -rw-r--r--  1 santashen  staff  0  5 25 11:32 file4
```

`file1`和`file2`完全一致，有同一个inode号，指向同一片文件数据，相当于两个相同的指针；`file3`具有不一样的inode号，他指向`file1`；`file4`相当于进行了拷贝，拷贝了文件数据存储，因此它指向拷贝出来的新的文件数据存储区域。