Linux 文件系统总结

1831604 张尹嘉

Linux文件系统由两个部分组成，分别是VFS和具体的文件系统。VFS是具体的文件系统的管理者，并且为上层的系统调用提供接口。

1. VFS

VFS是 Virtual Filesystem Switch 的缩写，其一切相关的数据结构都只存储在内存之中。每次系统初始化期间，Linux 都首先要在内存当中构造一棵 VFS 的目录树，对应着磁盘上的实际文件数据。在目录树中，每个结点对应一个具体的文件。每个文件都有一个unique的文件名，系统对文件按名存取。结点中还保存了指向这个文件的inode的指针，inode中记录了该文件的静态属性。同时，VFS还记录了每个具体文件系统访问文件时需要的函数。VFS在读取文件存储的数据时，并不是直接从磁盘读取的，而是从核心态缓存读取的数据。

目录树中每个结点都存储着一个目录文件，目录文件中每一项都有文件名和对应的inodeId。在进行目录搜索时，需要从树根开始，遍历父目录文件并且匹配文件名字符串，匹配成功后进入下一级目录项。如果不进行优化，目录搜索的系统开销会十分大，因此各种系统都会对目录搜索进行优化。在Windows中，每个目录都配备了一个B+树，目录文件中的每一个目录项都是有序的。 而在Linux中则采用了缓存的思路，将近期使用过的所有目录项都缓存起来。

VFS中目录项是一个叫做dentry的结构，其关键的成员包括：

```
struct dentry{
    char file_name[FILE_NAME_MAX_LEN]; // 文件名
    int dev;                           // 物理设备号
    int inode;                         // inode结点
    ListHead childDentry;              // 兄弟目录项
    ListHead sibling;                  // 父目录项
};
```

其中文件名不是文件的绝对目录，而是在当前父目录下的分量，例如 `/home/Documents` 中的 `Documents`。同时，每个文件夹下还有两个特殊的目录项，分别是 `.`和 `..`， 代表了当前目录和上一级目录。这样就可以通过这两个目录项来支持相对路径了。

每个进程的 `task_struct` 中会记录和这个进程相关的文件系统的信息。 `task_struct.fs` 记录了和文件系统相关的信息，其中包括了根目录和当前的工作目录。 `task_struct.files` 是一个 `FILE` 结构的链表，记录了该进程打开的文件。`FILE` 结构登记了进程打开的文件的动态信息， 有一个指针指向了 `inode`。`inode` 中有一根指针指向磁盘上的inode， 还有一根ops指针登记了文件访问方法, 与具体的文件系统相关。

当用户调用系统调用读取文件时，VFS会读取核心态缓存的主存页框， 如果缓存不命中，VFS会调用具体文件系统中的read方法从磁盘中读取n个4K字节到n个主存页框中。

所有的具体的文件系统都会等记载一根链表上。链表上每个结点中都有指针记录和和这个文件系统有关的所有函数。

2. 具体文件系统

具体的文件系统包括：磁盘上的文件系统，TTY/终端输入输出，Socket，除了TTY以外的所有外设以及`proc/systs`。

这里重点说磁盘上的文件系统，以`ext2`文件系统为例。磁盘上的数据布局如下：

```
[ SuperBlock | inode Set/目录项/bitmap | data ]
```

这三个部分组成了一个文件卷， 每个文件卷开头记录了一个唯一标识这个文件系统的magic number。其中SuperBlock记录了文件卷的布局； inode Set记录了这个磁盘中所有文件对应的inode； 目录项记录了这个卷中的目录项； bitmap记录了访问权限等其他信息； data区记录了所有文件的具体信息。

每个具体文件系统都要在VFS中登记具体的访问方法。这些方法的种类包括：读取SuperBlock的方法；读取inode的方法；获取根目录的位置；read/write/create方法；磁盘空闲空间管理方法；解析目录文件的方法。

`[ SuperBlock | inode Set/目录项/bitmap ]`组成了一文件系统的元数据。特别的，目录项中根目录是#2号文件， #0是指该目录项是空的， #1文件登记磁盘上坏的扇区，是Unix文件系统的隐藏文件，在数据区最开始的某个数据区。 在一个文件卷中，访问data中的数据可以通过 inode.addr + offset 访问， 但是要读取元数据的内容，是没有办法通过这种方法，因此将磁盘看作是一个文件，在/dev/sda中作为特殊类型文件，可以通过offset直接读取元数据。