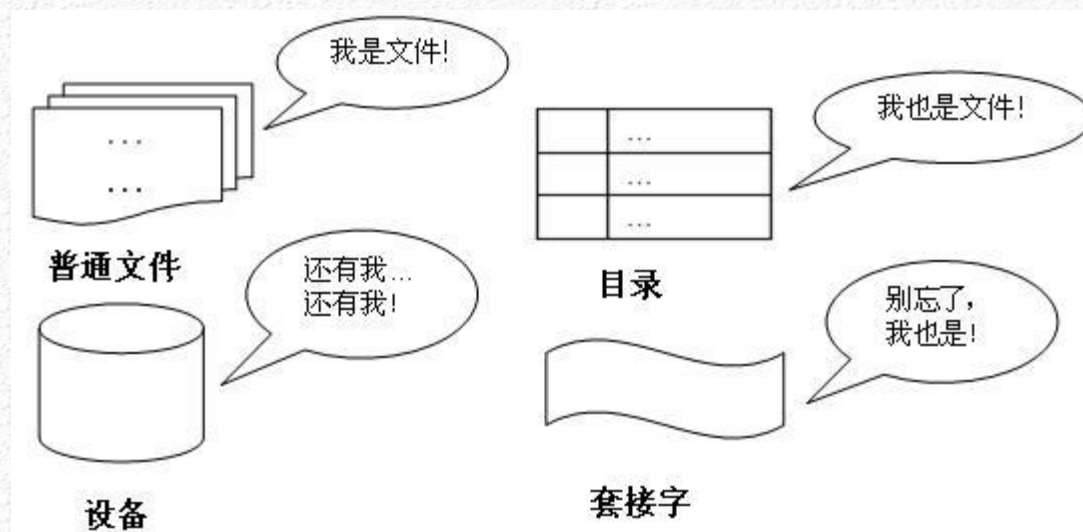


Linux操作系统编程

Linux文件系统概述

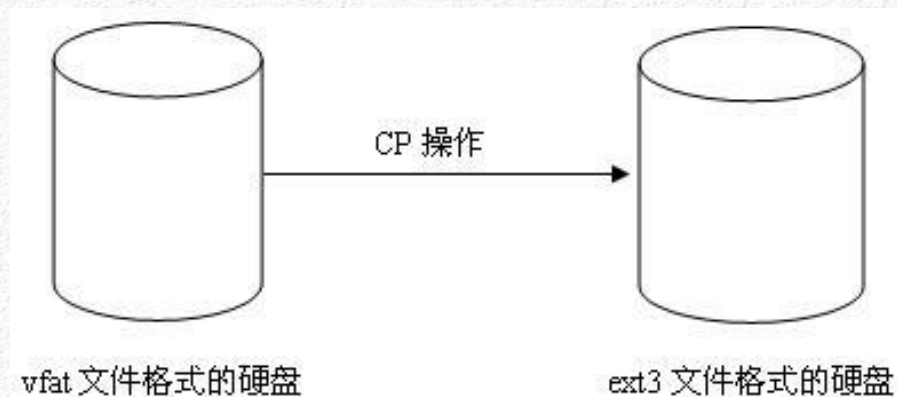
- 文件系统是以合理有效的层次结构组织的文件和目录的集合
- “一切皆是文件” 是 Unix/Linux 的基本哲学之一
- 普通文件、目录、字符设备、块设备、套接字等在 Unix/Linux 中都是文件
- 类型不同的文件都是通过相同的API对其进行操作



- Unix/Linux 中允许不同的文件系统共存，如 ext2, ext3, ext4,xfs,btrfs 等

文件系统	适用场景	原因
ext2	U盘	ext2不写日志，对安全性要求不高，兼容FAT
ext3	对稳定性要求高的地方	ext4稳定性不高
ext4	小文件较少	不支持inode动态分配
xfs	小文件多	支持inode动态分配
btrfs	没有频繁的写操作	功能众多

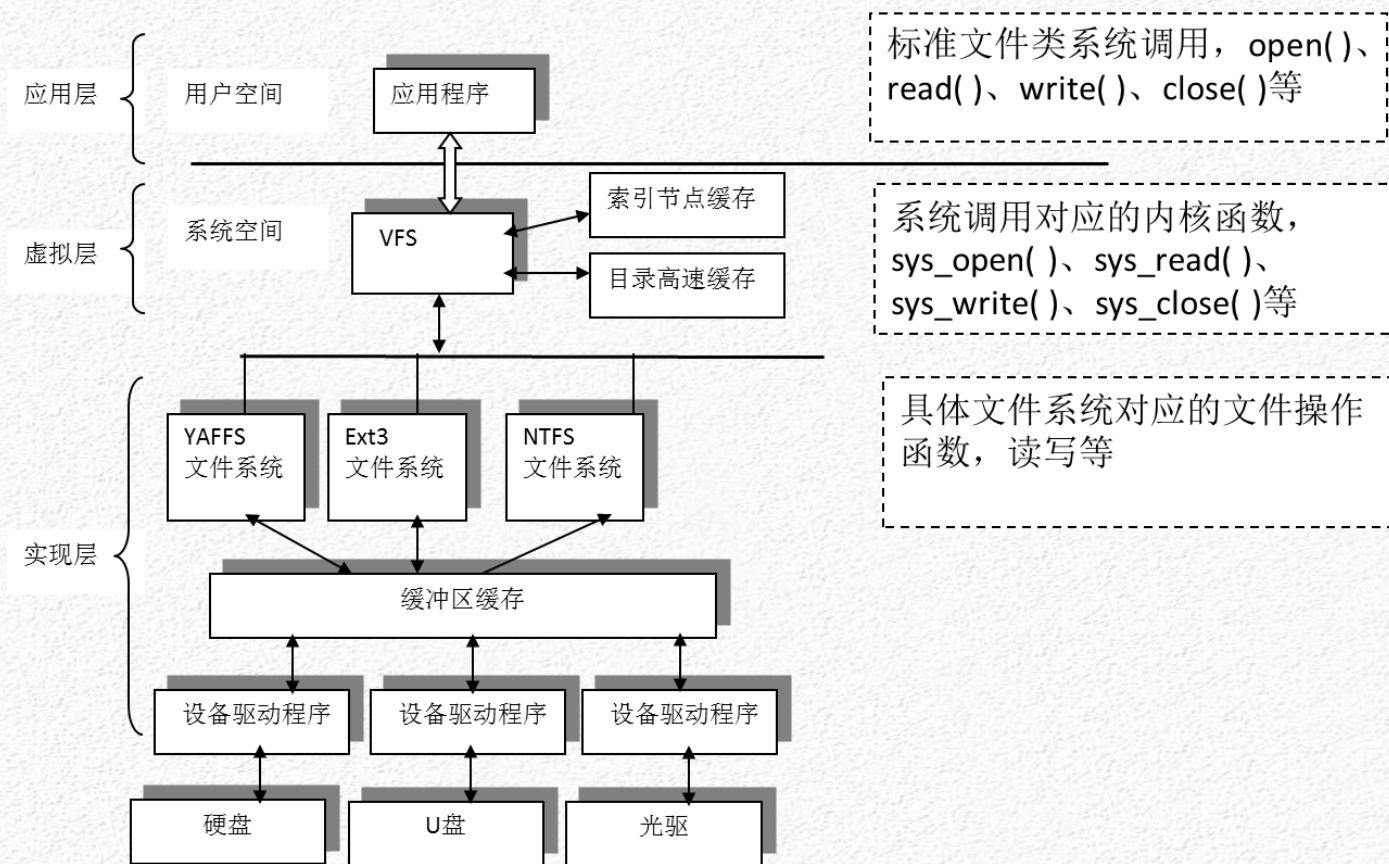
- 通过统一的文件 I/O 系统调用API即可对系统中的任意文件进行操作而无需考虑其所在的具体文件系统格式
- 文件操作可以跨文件系统执行



与windows文件系统的区别

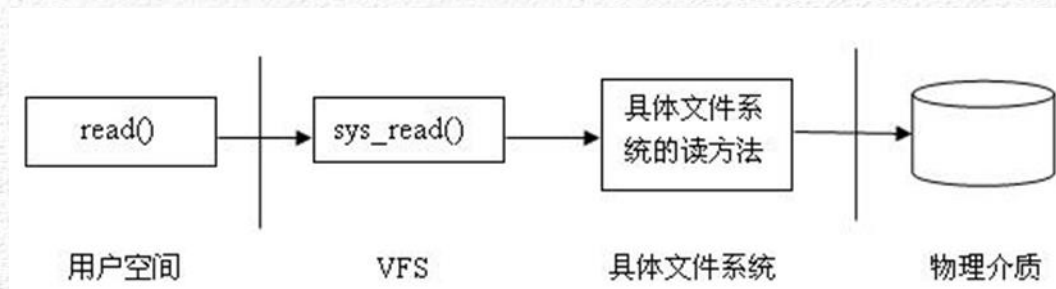
	Unix/Linux文件系统	Windows文件系统
1	根目录文件系统	多根目录文件系统
2	文件名区分大小写	文件名不区分大小写
3	目录路径使用正斜杠，如 /home/name	目录路径使用反斜线，如 C:\Users\zxy
4	没有驱动器号，一切文件都在 root目录下	驱动器号标记分区和设备
5	无独占访问权限	独占访问权限

Linux文件系统架构



虚拟文件系统 (VFS)

- 虚拟文件系统是Linux 内核中的一个软件层，对内实现文件系统的抽象，允许不同的文件系统共存，对外向应用程序提供统一的文件系统接口
- 为了能够支持不同文件系统，VFS 定义了所有文件系统都支持的基本的、抽象的接口和数据结构
- 实际文件系统实现VFS 定义的抽象接口和数据结构，将自身的诸如文件、目录等概念在形式上与VFS的定义保持一致，在统一的接口和数据结构下隐藏了具体的实现细节



- **超级块 (super block)** : 用于存储文件系统的控制信息的数据结构。描述文件系统的状态、文件系统类型、大小、区块数、索引节点数等, 存放于磁盘的特定扇区中。
- **索引节点 (inode)** : 用于存储文件的元数据 (文件的基本信息) 的一个数据结构, 包含诸如文件的大小、拥有者、创建时间、磁盘位置等信息。
- **目录项 (dentry)** : 目录被用来容纳文件, 目录可以包含子目录, 层层嵌套以形成 文件路径。
- **文件对象 (file)** : 一组在逻辑上具有完整意义的信息项的系列。

VFS-超级块 (super block)

- 超级块用来描述整个文件系统的信息。每个具体的文件系统都有各自的超级块
- VFS超级块是各种具体文件系统在安装时建立的，并在卸载时被自动删除，其数据结构是 `super_block`
- 所有超级块对象以双向环形链表的形式链接在一起

VFS-超级块 (super block)

```
struct super_block
{
    kdev_t s_dev;      /*具体文件系统的块设备标识符*/
    unsigned long s_blocksize; /*以字节为单位数据块的大小*/
    unsigned char s_blocksize_bits; /*块大小的值占用的位数*/
    ...
    struct list_head s_list; /*指向超级块链表的指针*/
    struct file_system_type *s_type;
        /*指向文件系统的file_system_type 数据结构的指针 */
    struct super_operations *s_op;
        /*指向具体文件系统的用于超级块操作的函数集合 */
    ...
    u; /*一个共用体，其成员是各种文件系统的
        fsname_sb_info数据结构 */
}
```


VFS-索引节点 (inode)

- 文件系统处理文件所需要的信息都放在称为索引节点的数据结构inode中
- 具体文件系统的索引节点是存放在磁盘上的，是一种静态结构，要使用它，必须调入内存，填写VFS的索引节点，因此，也称VFS索引节点是动态节点
- 文件名可以随时更改，但是索引节点对文件是唯一的，并且随文件的存在而存在
- 每个inode节点的大小，一般是128字节或256字节

VFS-索引节点 (inode)

```
struct inode
{
    struct list_head i_hash;    /*指向哈希链表的指针*/
    struct list_head i_list; /*指向索引节点链表的指针*/
    struct list_head i_dentry; /*指向目录项链表的指针*/
    ...
    struct inode_operations *i_op; /*指向对该节点操作的函数*/
    struct super_block *i_sb; /*指向该文件系统超级块的指针*/
    atomic_t i_count; /*当前使用该节点的进程数*/
    struct file_operations *i_fop; /*指向文件操作的指针*/
    ...
    unsigned long i_state; /*索引节点的状态标志*/
    unsigned int i_flags; /*文件系统的安装标志*/
    ...
}
```