**文件系统**

：所有的计算机应用程序都需要存储的检索信息

进程的地址空间保存信息的问题

（1）进程运行时，可以在它之间的地址空间存储一定的信息，但是存储容量受虚拟地址空间的大小的限制

（2）进程终止时，保存的信息也会消失

（3）经常需要多个进程同时存取同一信息

长期存储信息的基本要求

1）能够存储大量信息

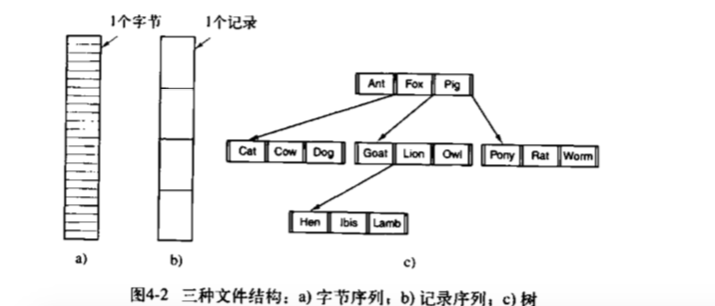
2）使用信息的进程终止时，信息仍然存在

3）必须支持多个进程并发存取有关信息

文件定义：进程创建的逻辑单元

文件系统：操作系统中处理文件的部分

**4.1文件结构**



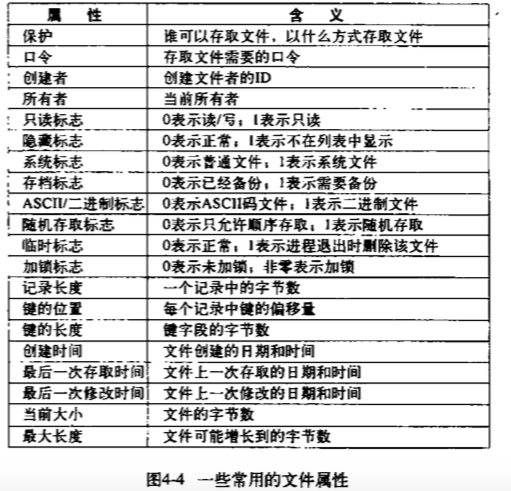
**文件类型**

unix：普通文件和目录文件；普通文件，字符特殊设备，块设备文件

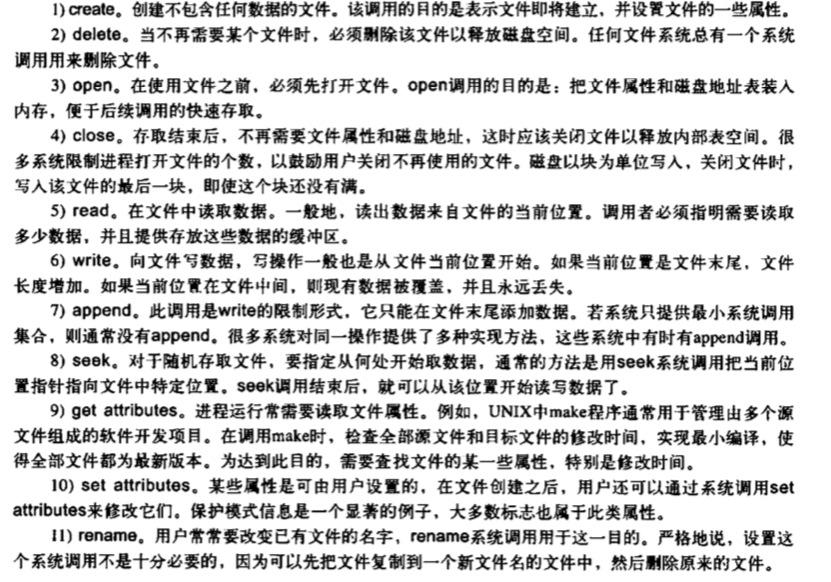
普通文件一般分为ASCII文件和二进制文件

文件存取：顺序存取

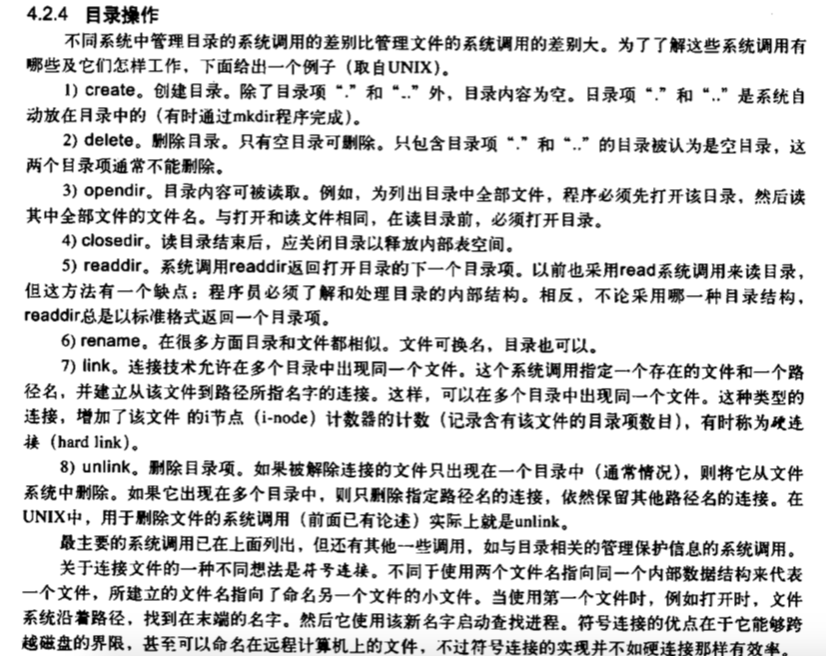
文件属性：



文件操作：（系统调用）



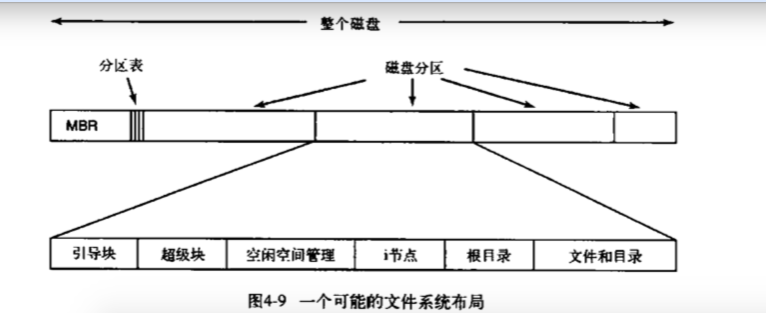
目录操作



**4.3文件系统的实现**

**4.3.1文件系统布局**

文件系统在disk上；



存储文件的第二种方法：为每个文件构造磁盘链表

**日志文件系统**

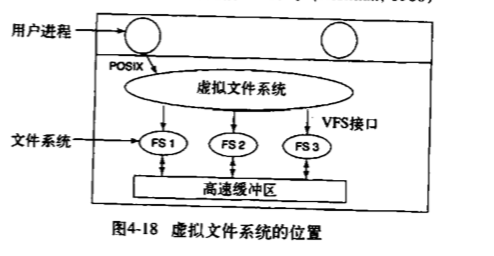
**（NTFS，EXT3，REISTERFS）**

移除文件：

1）在目录中删除文件

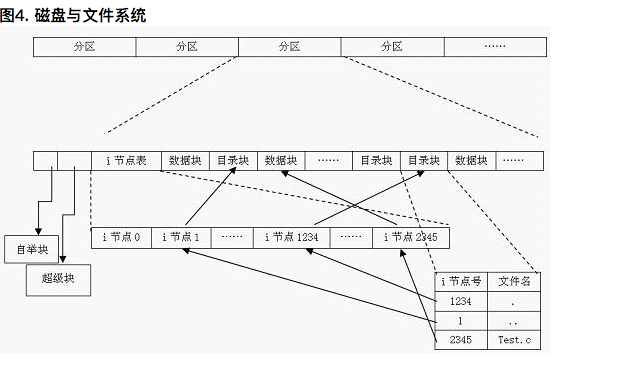
2）释放I节点到空闲I节点池

3）将所有磁盘快归还磁盘块池

**虚拟文件系统（VFS）：**

**虚拟文件系统**（Virtual File System, 简称 VFS）， 是 **Linux** 内核中的一个软件层，用于给用户空间的程序提供文件系统接口；同时，它也提供了内核中的一个 抽象功能，允许不同的文件系统共存。系统中所有的文件系统不但依赖 VFS 共存，而且也依靠 VFS 协同工作。

为了能够支持各种实际文件系统，VFS 定义了所有文件系统都支持的基本的、概念上的接口和数据 结构；同时实际文件系统也提供 VFS 所期望的抽象接口和[数据结构](http://lib.csdn.net/base/31)，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。换句话说，一个实际的文件系统想要被 **Linux** 支持，就必须提供一个符合VFS标准 的接口，才能与 VFS 协同工作。实际文件系统在统一的接口和数据结构下隐藏了具体的实现细节，所以在VFS 层和内核的其他部分看来，所有文件系统都是相同的。图3显示了VFS在内核中与实际的文件系统的协同关系。

****

**创建** 以某种方式格式化磁盘的过程就是在其之上建立一个文件系统的过程。创建文现系统时，会在磁盘的特定位置写入 关于该文件系统的控制信息。

**注册** 向内核报到，声明自己能被内核支持。一般在编译内核的时侯注册；也可以加载模块的方式手动注册。注册过程实 际上是将表示各实际文件系统的数据结构struct file\_system\_type 实例化。

**安装** 也就是我们熟悉的mount操作，将文件系统加入到**Linux**的根文件系统的目录树结构上；这样文件系统才能被访问。

VFS数据结构

struct super\_block { //超级块数据结构

struct list\_head s\_list; /\*指向超级块链表的指针\*/

……

struct file\_system\_type \*s\_type; /\*文件系统类型\*/

struct super\_operations \*s\_op; /\*超级块方法\*/

……

struct list\_head s\_instances; /\*该类型文件系统\*/

……

};

struct super\_operations { //超级块方法

……

//该函数在给定的超级块下创建并初始化一个新的索引节点对象

struct inode \*(\*alloc\_inode)(struct super\_block \*sb);

……

//该函数从磁盘上读取索引节点，并动态填充内存中对应的索引节点对象的剩余部分

void (\*read\_inode) (struct inode \*);

……

};

VFS即**虚拟文件系统**是**Linux**文件系统中的一个抽象软件层；因为它的支持，众多不同的实际文件系统才能在**Linux**中共存，跨文件系统操作才能实现。 VFS借助它四个主要的数据结构即超级块、索引节点、目录项和文件对象以及一些辅助的数据结构，向**Linux**中不管是普通的文件还是目录、设备、套接字等 都提供同样的操作界面，如打开、读写、关闭等。只有当把控制权传给实际的文件系统时，实际的文件系统才会做出区分，对不同的文件类型执行不同的操作。由此 可见，正是有了VFS的存在，跨文件系统操作才能执行，Unix/**Linux**中的“一切皆是文件”的口号才能够得以实现。

共享文件：