Unix的哲学--“一切皆文件”。首先，一切都包括什么，文件指的是什么。

计算机包括cup、存储器(一般指内存)、外设。一切主要指的是外设，包括键盘、显示器、鼠标、网卡等。而文件是为了能有效、统一的使用外设而采取的一种方式(对linux而言)。

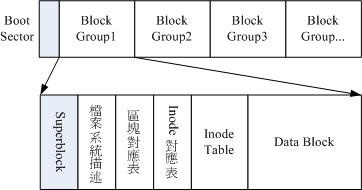
这里的文件是虚拟文件系统中的名词。虚拟文件系统的作用就是为程序员提供使用外设的统一接口。

可以认为文件就是字节数组！！即I/O设备在我们看来就是一个个的字节数组！！！虚拟文件系统实现了I/O设备到字节数组的转换，数组我们熟吧。

linux中虚拟文件系统的支持的文件类型有：普通文件[-](纯文本文件、二进制文件、数据格式文件)、目录文件[d]、连接文件[l]、设备文件(块设备[b]、字符设备[c])、套接字文件[s](基于网卡的)、管道文件[p](基于内存)。

下面以普通文件为例，说明linux文件系统的工作过程：

Linux ext2文件系统的磁盘：



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，同时记录此文件的内容所在的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 。

Inode表和Block表：记录Inode和Block的使用情况。

可以看到linux将文件的属性和文件的内容分开存储。Linux ext2期望的工作是这样的：

目录：当做文件看待，Inode存的仍然是目录信息，相应的Block存的是目录下面的文件对应的Inode号。

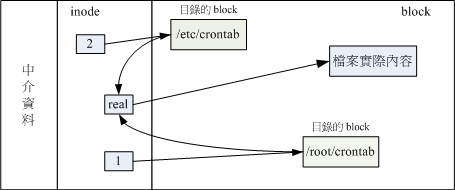
所以，对文件或目录的操作首先由根目录一层层的通过文件名找Inode，通过Inode找Block，在Block里面再通过文件名找Inode。最终找到文件。

对于符号链接：

从ext2的工作方式中可以看出：文件(或目录)的信息存于Inode，内容存于Block，而文件名是作为找到相应Inode的一个符号，存放在目录文件的Block中。

1. 硬链接：只是目录文件的Block中文件名对应的Inode是其连接文件的Inode。

ln /etc/crontab .



在目录的Block中，/etc/crontab和/root/crontab对应的Inode是同一个。这时，

Ls -il /etc/crontab /root/crontab

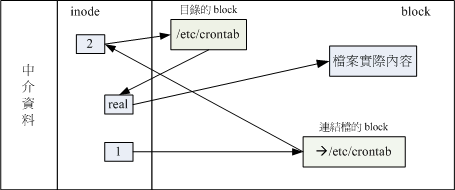
34474855 -rw-r--r--. 2 root root 451 Jun 10 2014 crontab

34474855 -rw-r--r--. 2 root root 451 Jun 10 2014 /etc/crontab

其中的2表示映射到该Inode的数量。

2、软连接：有自己单独的Inode和Block块，只不过Block块存放的是所连接文件名。Linux会根据这个文件名去找相应的文件。

Ln -s /etc/crontab crontab2



注意，这里的Inode 1对应的Block块是文件的Block块，而硬链接时，是目录的Bolck块。

这时如果删除real的话，则另一个软链接就不能打开文件了。而硬链接时，删除real，只会把链接数减一，减到0时才真正删除文件。不过硬链接不能链接目录。

下面介绍虚拟文件系统的实现方式：(以普通文件为例)

首先介绍常用到的4个数据结构：

1、文件(file)：代表一个打开的文件，不同的进程打开同一个文件，会得到各自的文件对象。

2、目录项(dentry)：表示的是文件的各个路径，有缓存，能够快速找到各个文件。其实通过iNode结点就可以找到文件，不过iNode的属性太多，通过iNode进行文件的比较、查找这些频繁的操作，效率无法接受，所以就添加目录项。

3、iNode结点(inode)：唯一的表示一个文件，对应硬盘中的Inode。

4、超级块(super\_block)：包含文件系统的一些信息，对应硬盘中的超级块。

值得注意的是，上面所有的数据结构，都有一个操作指针。该指针指向一个存有改数据结构所支持操作的数组。

与进程相关的数据结构：这里就说两个吧

1、files\_struct：里面包括进程所打开文件的数组。open函数的返回值就是该数组的下标。

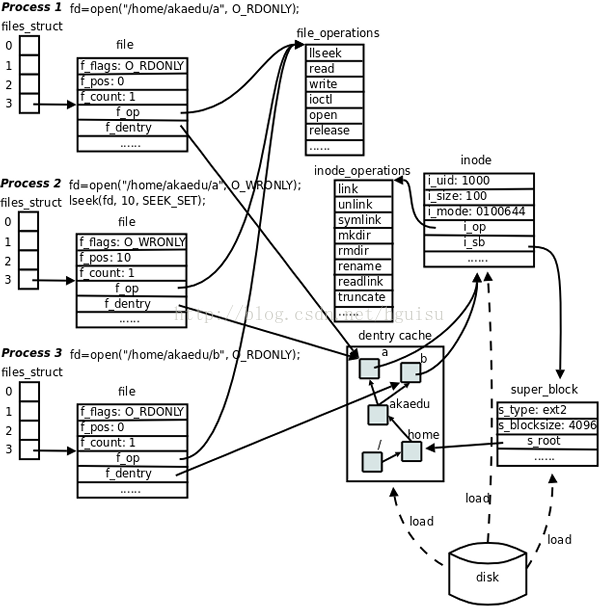
2、fs\_struct: 里面包括当前目录和根目录的目录项。

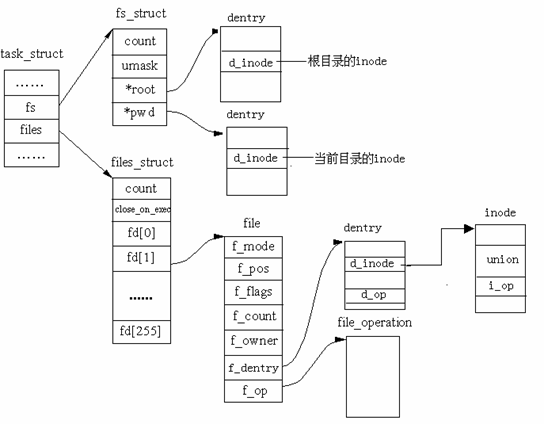
进程控制块结构：

task\_struct:里面含有进程的所有信息(files\_struct和fs\_struct就在进程控制块里面)，大小为1K多，跟版本有关。位于内核栈的末尾(每一个进程，都有一个内核栈)。

进程粗略可以认为是：进程控制块+代码段+数据段。

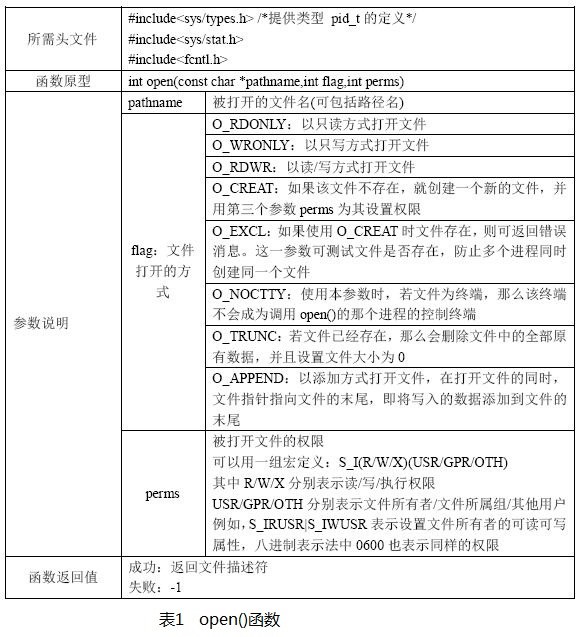
工作过程见下图。

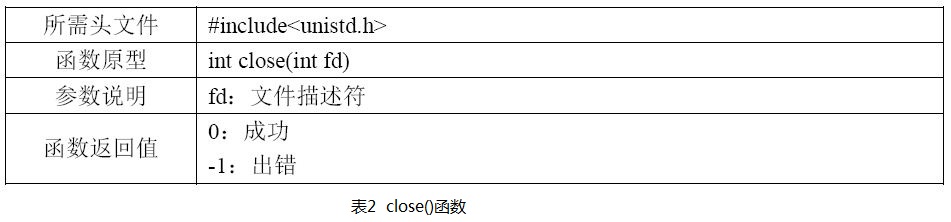


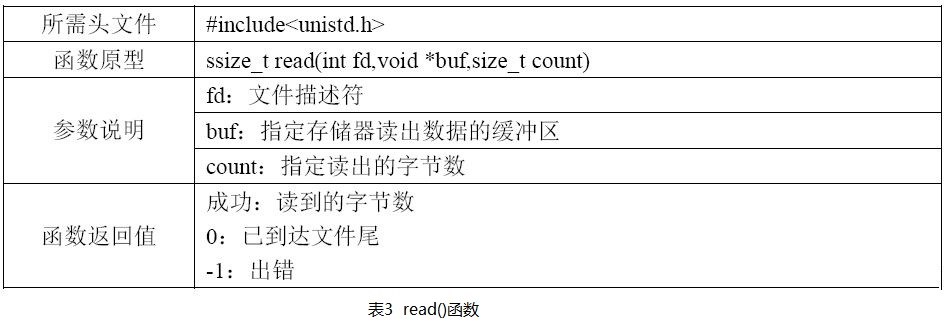


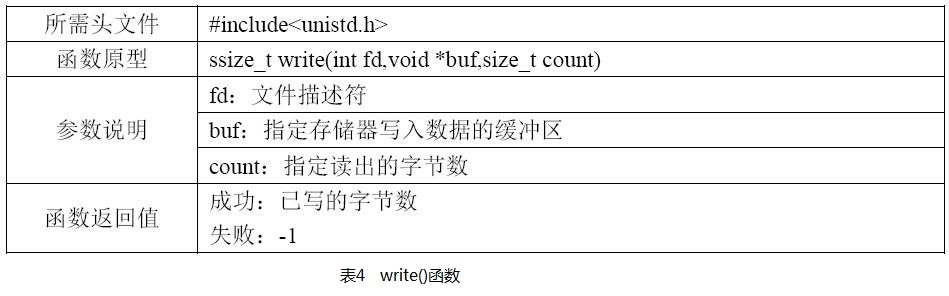
虚拟文件系统提供的统一接口是：open、read、write、lseek、close。

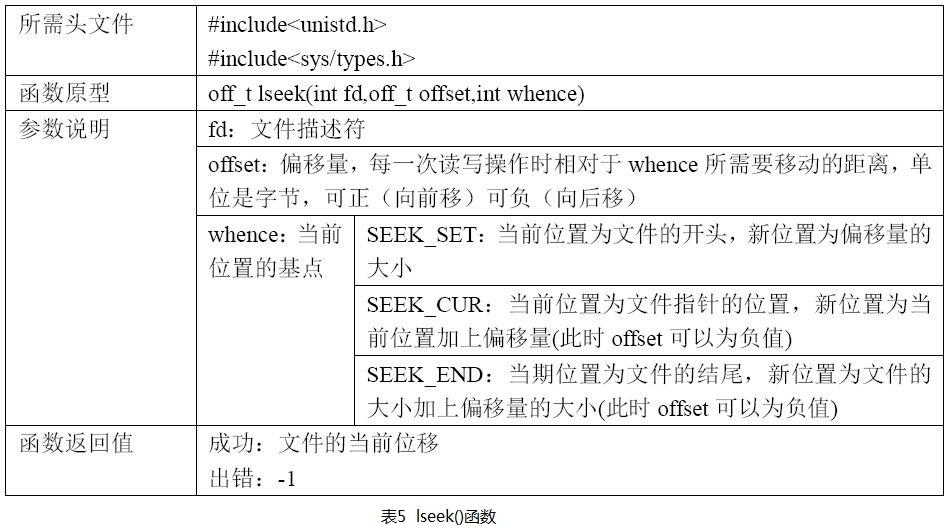
不过，有的文件有更适合自己的操作接口，比如套接字文件。虽然它也支持read、 write等接口，但还是习惯使用自己的一套方法，可能是更高效，更安全吧。下面是常用接口：











Linux文件系统还提供了很多别的系统调用，但是这些系统调用最终都是通过file、dentry、inode和super\_block的x\_op 字段调用相应的操作函数。

Linux提供的命令：df –h ; du –h;