800页 我真的想去si

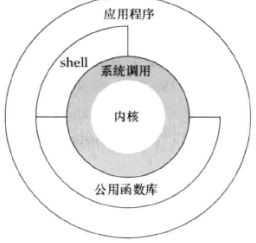
不多说了，我去看书了==。==

UNIX环境高级编程

第一章：UNIX基础知识

引言：所有操作系统都为它们所运行的程序提供服务，典型的包括执行新程序，获得时间，分配存储，打开文件等

UNIX体系结构：操作系统定义为一种控制计算机硬件资源，提供程序运行环境的软件，通常将这种软件称为内核。内核的接口被称作系统调用，公用函数库构建在系统调用接口上，应用程序即可使用公用函数库，也可使用系统调用，shell是一个特殊的应用程序，为运行其他应用程序提供了一个接口。广义上的操作系统包括内核和一些使得计算机能够发挥作用，并使计算机具有自己的特性，其他软件包括系统实用程序、应用程序、shell、公共函数库等。



登录：登录名，shell（有许多不同的种类），所有的linux系统都使用GNU shell

文件系统：为层次结构，起点为根（root “/”）,

文件名：字母，数字，句点，短横线，下划线，创建新目录时会自动创建两个新文件名. ..指向当前目录和父目录

路径名：斜线分隔的一个或多个文件名组成的序列构成路径，以斜线开头的路径名称为绝对路径名，否则为相对路径名

工作目录：每个进程都有工作目录

起始目录：登录时，工作目录设为起始目录

文件描述符：通常是一个小的非负整数，内核用来标识一个特定进程正在访问的文件，当内核打开一个现有文件或创建一个新文件，都返回这个，读写文件可以使用这个

标准输入、输出、错误：如果不做特殊处理，这三个描述符都链接向终端，大多数shell都提供一种办法让描述符重新定位到文件，如ls > file.list

不带缓冲的I.O：open、read、write、lseek以及close、

标准IO：

程序：存储在磁盘上某个目录中的可执行文件。内核使用exec函数将程序入读内存执行程序

进程：程序的执行实例

进程ID：一个非负整数，进程的唯一标识符

进程控制：fork exec waitpid

线程：通常一个进程只有一个控制线程，某一时刻执行的一组机器指令。一个进程内的所有线程共享同一地址空间、文件描述符、栈以及进程相关的属性。因为它们能访问同一存储区，所以要各线程访问共享数据时需要采取同步措施避免不一致性

线程ID：只在它所属的进程内起作用。

用户ID：一个数值向系统标识各个不同的用户。系统管理员在确定一个用户的登录名的同时，确定其用户ID，用户不能更改用户ID

组ID：口令文件登录项包括用户的组ID，它是一个数值。

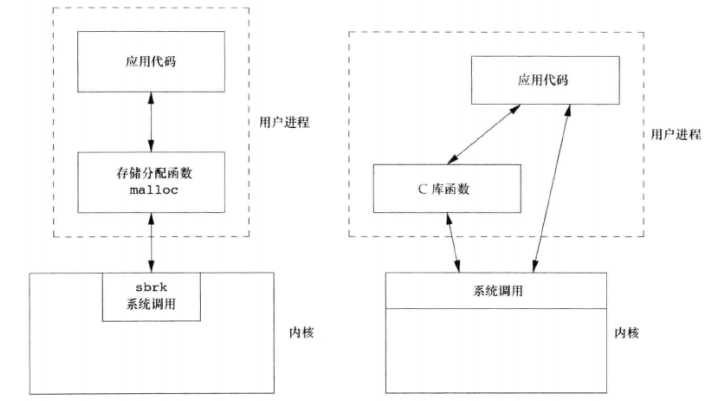
附属组ID：除了在口令文件中对一个登录名指定一个组ID外，大多数UNIX系统版本还允许一个用户属于另外一些组

信号：信号通知进程发生了某种情况。例如若某一进程执行除法操作，除数为0，则将名为SIGFPE的信号发送给该进程。进程有一下三种处理信号的方式1、忽略哦2、按照默认方式处理3、提供一个函数信号发生时调用这个函数。

UNIX使用过两种不同的时间值：1、日历时间2、进程时间。 度量进程执行时间有三个，时钟时间，用户CPU时间，系统CPU时间

系统调用：指良好定义，数量有限，直接进入内核的入口点

库函数：UNIX使用的技术是为每个系统调用在标准C库中设置一个具有同样名字的函数。



系统调用通常提供一种最小接口，而库函数通常提供比较复杂的功能。

第二章：UNIX标准及实现（回头看或许更好吧）

UNIX版本很多，因此制定了标准：ISO C标准 IEEE POSIX标准 Single UNIX Specification标准 FIPS标准

系统实现：SVR4 4.4BSD FreeBSD Linux Mac OS X Solaris

标准和实现的关系：FreeBSD Linux Mac OS X Solaris都提供UNIX编程环境。

限制：IOS C定义的限制都在头文件<limits.h>中 如整型的最大到多少，可同时打开标准I/O流的最小个数

POSIX限制定义了数值限制，最大最小值，运行时可增加的值，运行时不变值，其他不变值，路径名可变值

三个函数 sysconf pathconf fpathconf 来获取UNIX系统的各种预设的限制

限制分为三种 （1）编译时限制（头文件）（2）与文件或目录无关的运行时限制（sysconf函数）（3）与文件或目录有关的运行时限制（pathconf和fpathconf函数）

选项：和限制一样有三种看选项的方法（1）编译时选项定义在<unistd.h>中（2）与文件或目录无关的运行时选项用sysconf函数来判断（3）与文件或目录有关的运行时选项通过调用pathconf或fpathconf函数来判断

下面这部分讲的是unix的文件系统，挺深入的，重点不在函数细节，在于设计的思想和方法

第三章：文件I/O

终于开始讨论UNIX系统了，=。=

可用的文件I/O函数，包括打开文件、读文件、写文件。用到五个函数open、read、write、lseek以及close

不同缓冲长度对read和write有影响

本章的函数被称为不带缓冲的I/O，术语不带缓冲指每个read和write都调用内核中的一个系统调用。

涉及多个进程间共享资源，原子操作的概念就非常重要

文件描述符：

对于内核所有打开文件都通过文件描述符引用，当打开一个文件或创建新文件，内核向进程返回一个文件描述符，读写文件，用返回的文件描述符作为参数给read和write

文件描述符 0 1 2分别与标准输出，标准输入，标准错误关联。长替换为复航常量STDIN\_FILENO,STDOUT\_FILENO和STDERR\_FILENO

文件描述符变化范围为0~OPEN\_MAX-1

函数open和openat：可打开或创建一个文件

openat让线程可以使用相对路径名打开目录中的文件，而不再只能打开当前工作目录，避免TOCTTOU错误，

creat可以创建一个新文件，缺点是以致谢方式打开创建的文件，

close关闭一个打开文件。加上文件描述符

lseek显式的为一个打开文件设置偏移量

read从打开文件中读取数据

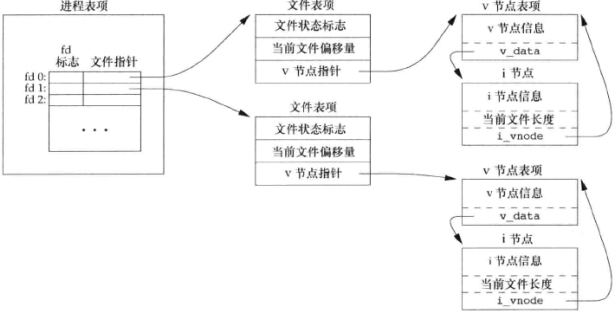
write向打开文件写数据

IO效率，预读

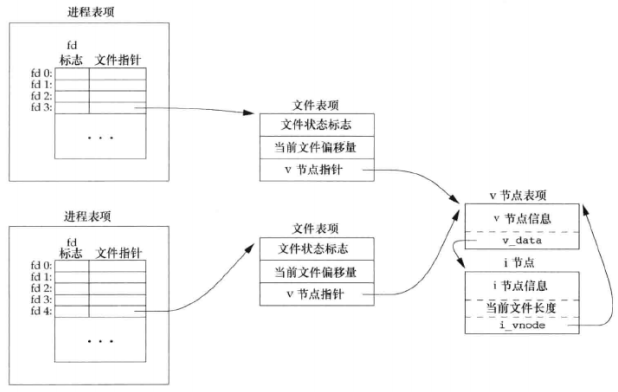
文件共享：

内核使用三种数据数据结构表示打开文件

如下图：



当两个独立进程各自打开了同一文件，



多个进程读同一文件每个进程有它自己的文件表项，其中也有他自己当前的文件偏移量。但是写同一文件，就可能产生预想不到的效果。

原子操作：追加到一个文件，原子性地定位并执行I/O，

（太细了，这本书真的太细了，关于函数部分内容不要太纠结了，太细节了，还是要宏观把握知识，看完这本书至少要知道unix这个操作系统的大致外貌，关于操作系统的四大组件他是怎么处理的。）

一般而言，原子操作指的是由多步组成的一个操作。如果该操作原子地执行，要么执行完所有步骤，要么一步也不执行，不可能只执行所有步骤的子集

ioctl函数：终端I/O使用的最多

第四章：文件和目录

引言：本章描述文件系统的其他特征和文件的性质。

Stat函数返回命名文件有关的信息结构。Fstat获得已在描述符fd上打开文件的有关信息。Lstat类似stat，但当命名的文件是一个符号链接时，lstat返回该符号链接的有关信息，而不是由该符号链接引用的文件的信息

Fstatat函数为一个相对于当前打开目录的路径名返回文件统计信息。

文件类型：普通文件和目录

普通文件：最常用的，包含了某种形式的数据，至于是文本还是二进制，对于UNIX内核并无区别。

目录文件：

块特殊文件：

字符特殊文件：

FIFO：用于进程间通信，有时也称为命名管道

套接字：用于进程间的网络通信

符号链接：这种类型的文件指向另一个文件

文件类型信息包含在stat结构的st\_mode成员中。

设置用户ID和设置组ID，一个进程相关联的ID有6个或更多，文件也有所属的用户ID，组ID，有的有权限去，有的没有权限

文件访问权限有九种

新文件和目录的所有权，新文件的用户ID设置为进程的有效用户ID。组ID可以是进程的有效组ID，所在目录的组ID

函数chmod，fchmod，fchmodat这三个函数可以更改现有文件的访问权限

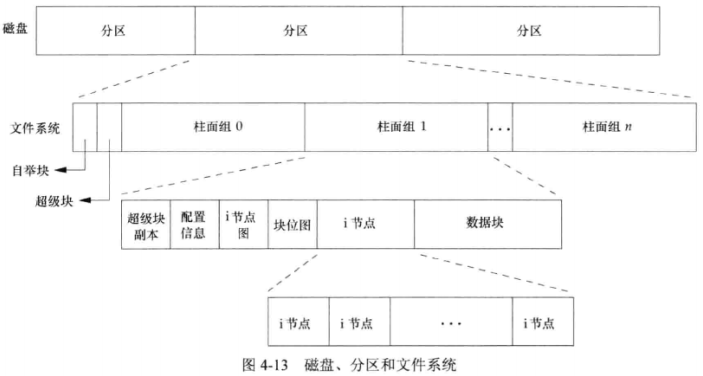
chown fchown fchownat 和 lchown可用于更改文件的用户ID和组ID。

文件长度stat结构成员st\_size表示以字节为单位的文件的长度

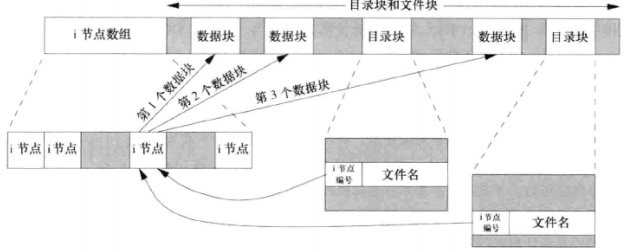
文件空洞：偏移量超过尾端并写入了数据

文件截断：文件尾端截去数据缩短文件

文件系统基本结构：正在使用的文件系统有多种实现，传统的基于BSD的UNIX文件系统（称为UFS），本节讨论这个，UFS以Berkeley快速文件系统为基础，本文讨论该文件系统



一个磁盘每个分区可以包含一个文件系统，i节点和数据块放大后



对于文件，有指向相同节点的硬链接，当链接计数减少到0时，才可以删除这个文件。

LINK\_MAX制定一个文件链接数的最大值

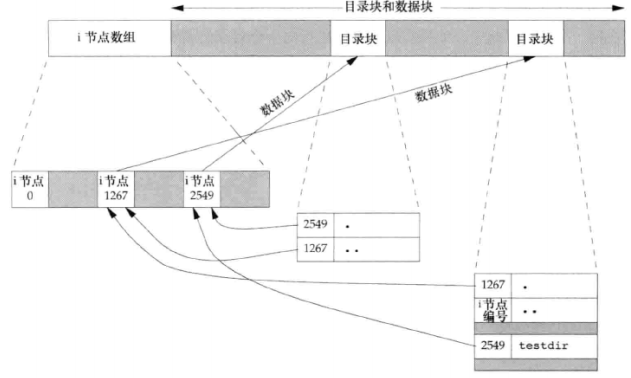
另外一种链接类型称为符号链接，实际内容包含该符号链接指向的文件的名字。软链接

i节点包含了文件类型，访问权限位，长度，指向文件数据块的指针等

目录项中的i节点编号指向同一文件系统中的相应i节点，一个目录项不能指向另一个文件系统的i节点

不更换文件系统情况下为一个文件重命名，该文件的实际内容并未移动，秩序构造一个指向现有i节点的新目录项

新建一个目录，那么文件系统中会发生的变化



创建指向现有文件的链接的方法：link和linkat函数，删除现有目录项，可以用unlink函数

文件或目录重命名可以用rename函数或者renameat函数

符号链接是对一个文件的间接指针，硬链接直接指向文件的i节点。对符号链接以及它指向何种对象并无任何系统限制，任何用户都可以创建指向目录的符号链接，符号链接一般用于将一个文件或目录结构移动到系统中的另一位置

创建符号链接symlink 或symlinkat

读取符号链接readlink和readlinkat

文件时间：修改时间，状态更改时间区别

更改文件的访问和修改时间futimens和utimensat

函数mkdir mkdirat创建目录，. ..自动创建 rmdir删除空目录

读目录：一系列的内容读目录

函数chdir fchdir可以改变当前工作目录， getcwd可以获得当前的完整工作目录，每个进程都有一个当前工作目录，

设备特殊文件，st\_dev和st\_rdev这两个字段经常引起混淆，

第五章 标准IO库

标准IO库围绕流进行，

流定向决定读写字符是单字节还是多字节的

freopen清楚一个流的定向，fwide设置一个流的定向

每个进程预定义三个流 标准输入 输出 错误

缓冲：标准IO 提供缓冲为了尽可能减少read和write调用的次数

三种 全缓冲 行缓冲 不带缓冲

默认下错误不带缓冲，指向终端设备的流是行缓冲，否则是全缓冲

打开流有三个函数 fopen freopen fdopen 返回文件指针 参数路径，不要文件描述符

fclose关闭一个打开的流

读和写流：三种 每次一个字符 一行 直接IO

一个字符 getc fgetc getchar

出错和到尾端分别用 ferror和feof来判断

ugetc可以把字符以相反的顺序压送回流中

输出 putc fputc putchar

每次一行IO

fgets gets

fputs puts

二进制IO（直接IO）

fread fwrite 适用于读写对象，可以设定一次读写的字节数用sizeof

实现细节是标准IO都要借助第三章的IO函数

tmpnam和tmpfile提供了两个函数帮助创建临时文件

内存流：fmemopen允许调用者提供缓冲区用于内存流

标准io有替代软件

第六章：系统数据文件和信息