**第十章 系统级I/O**

* 文件结尾没有明确的EOF符号
* 三种主要的文件类型
  + 普通文件
* 对内核而言文本文件和二进制文件没有区别
* Linux换行符：\n，0x0a
* 目录
  + 包含一组链接的文件
  + 每个链接把文件名映射到一个文件
  + . ..
  + mkdir, ls, rmdir
* 套接字
* 根目录：/，文件名末尾有/表示是目录
* 当前工作目录：每个进程上下文的一部分，cd修改
* 路径名：指定目录层次结构中的位置，绝对/相对
* open返回的描述符总是在进程中当前未打开的最小描述符，出错返回-1
  + int open(char \*filename, int flags, mode\_t mode)
  + flags: O\_RDONLY, O\_WRONLY, O\_RDWR, O\_CREAT, O\_TRUNC, O\_APPEND
  + mode: 访问权限位=mode&~umask，每个进程有一个umask，表示禁止的权限
* 关闭一个已关闭的描述符->出错返回-1
* 读取文件元数据
  + int stat(const char \*filename, struct stat \*buf);

int fstat(int fd, struct stat \*buf);

成功返回0，出错返回1

* + st\_size：文件的字节数大小

st\_mode：文件访问许可位、文件类型（普通文件、目录、套接字）

* + S\_ISREG(m), S\_ISDIR(m), S\_ISSOCK(m)
* 读取目录内容
  + DIR \*opendir(const char \*dirname); 成功返回指向目录流（目录项的列表）的指针，失败返回NULL
  + struct dirent \*readdir(DIR \*dirp); 成功返回指向下一个目录项的指针；出错/没有目录项返回NULL；d\_name文件名，d\_ino文件位置；出错设置errno，只能通过检查errno是否被修改过区分错误和流结束
  + int closedir(DIR \*dirp); 成功返回0，失败返回1
* 共享文件
  + 内核用三个数据结构表示打开的文件
    - 描述符表：每个进程一个，每个打开的描述符一个表项，指向文件表中的一个表项
    - 文件表：所有进程共享，打开文件的集合。表项：文件位置，引用计数，指向v-node表项的指针。引用计数为0，内核删除表项
    - v-node表：所有进程共享，打开的文件有，表项包含stat结构中的大多数信息（st\_mode, st\_size…）
  + 多个描述符可以通过不同的文件表表项来引用同一个文件
    - 以同一个filename调用open两次
    - 每个描述符都有自己的文件表表项，不同的文件表表项指向v\_node表中的一个表项
    - 两个描述符的当前文件位置是互相独立的
  + 复制进程共享文件表，引用计数>1，同一个文件位置
    - 父子进程都关闭文件，内核才能删除文件表表项
* I/O重定向
  + int dup2(int oldfd, int newfd)
    - 把描述符表oldfd表项复制到newfd，使newfd也指向oldfd指向的文件表表项
    - 如果newfd已打开，dup2在复制前关闭newfd，若文件表表项引用计数变成0，则要删除（v-node表表项也一样）
    - 如果newfd没打开，则打开
* 标准I/O
  + C语言
  + 将打开的文件模型化为一个流（指向FILE类型的结构的指针）
  + 流是对文件描述符和流缓冲区的抽象
    - 填充流缓冲区
    - 使用流缓冲区直到用完
* I/O函数的选择
  + Unix I/O在内核中实现，最底层
  + 只要有可能就用标准I/O

不要用scanf和rio\_readlineb来读二进制文件：0x0a

对网络套接字的I/O使用rio

* + 标准I/O流是全双工的，可以在同一个流上执行输入和输出
    - 对流的限制和对套接字的限制
      * 输出函数-fflush/fseek/fsetpos/rewind-输入函数，必须要清空缓冲区/ 重置文件位置
      * 输入函数-fseek/fsetpos/rewind-输出函数，除非输入遇到了EOF（缓冲区里没有东西了）
    - 对套接字使用lseek函数非法
    - 第一个限制：输出后输入前调用fflush

第二个限制：只能通过对同一个套接字打开两个流（open两次，一个读一个写），但是这样会试图关闭同一个套接字描述符，在线程化的程序中有问题

* + - 可以用sscanf/sprintf+rio