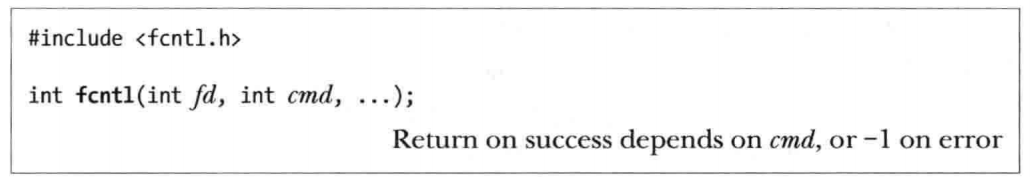
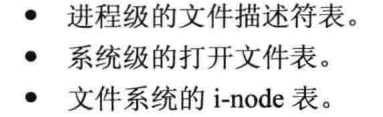
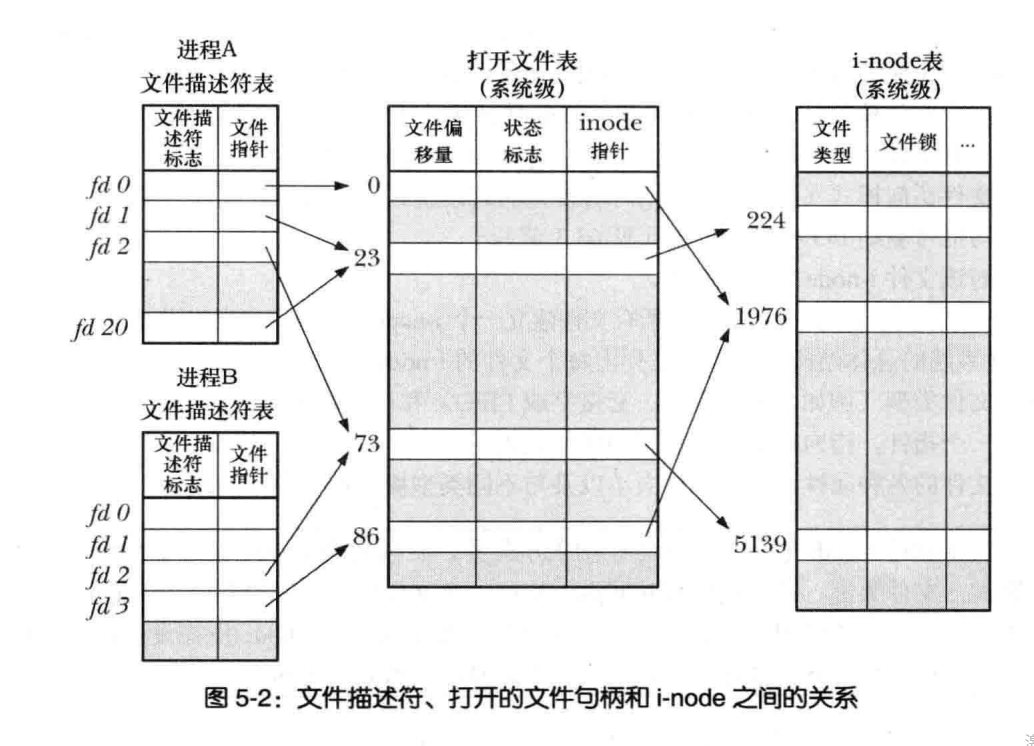
第五章

1. 内核保证所有系统调用都是以原子操作的方式执行的。
2. Open系统调用里的O\_EXCL喝O\_CREAT确保了检查文件和创建文件属于一个单一的原子操作。
3. 对于非NFS文件，open的参数O\_APPEND可以保证写入文件的原子性。
4. 文件操作控制：



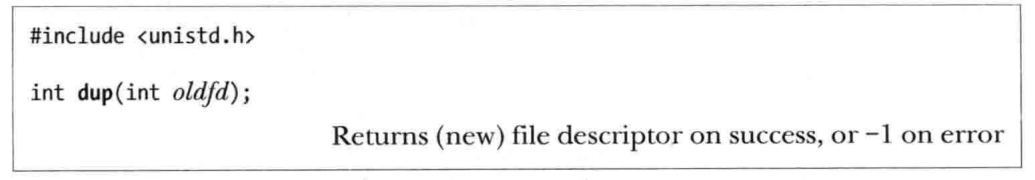
1. 文件描述符的close-on-exec机制：在fork子进程后执行exec时，相应的文件描述符对应于打开文件表会被关掉。
2. 文件描述符和打开文件之间的关系：内核维护的三个数据结构：想要理解他们的关系，就要知道这三个数据结构分别维护着什么内容。

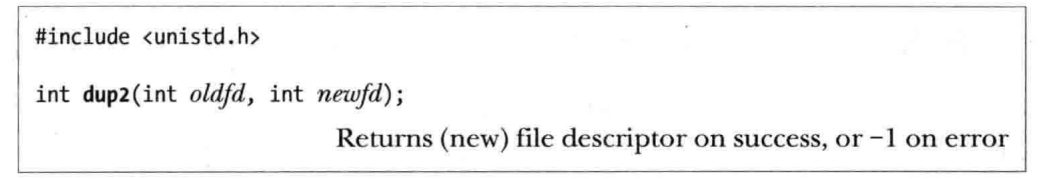


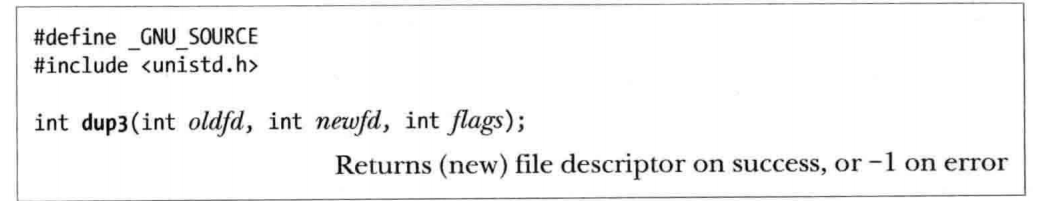


其中每个进程文件描述符表里的文件描述符标志目前只有close-on-exec。

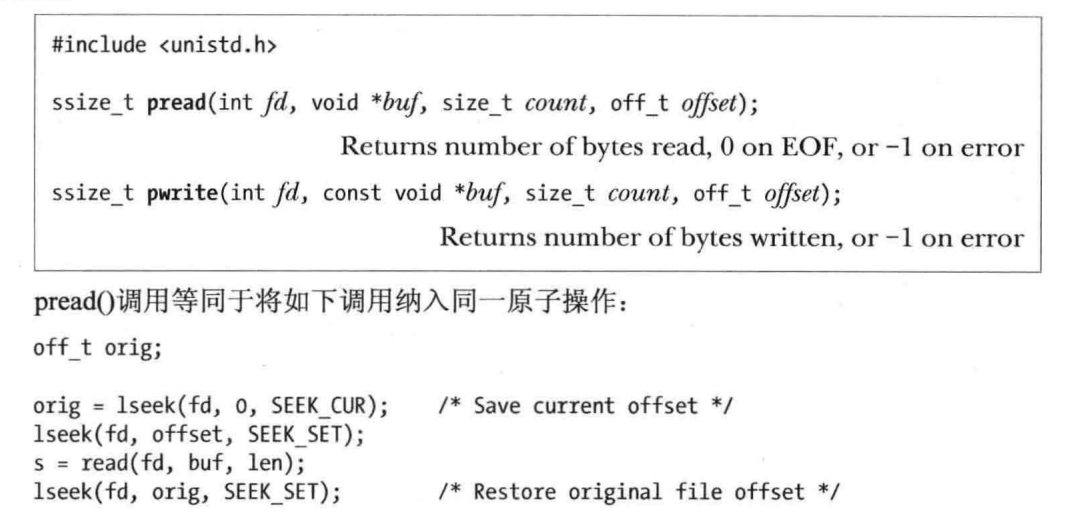
1. 复制文件描述符：怎么理解？复制oldfd文件描述符，并给newfd，也就是说这两个文件描述符指向打开文件表里面的句柄。



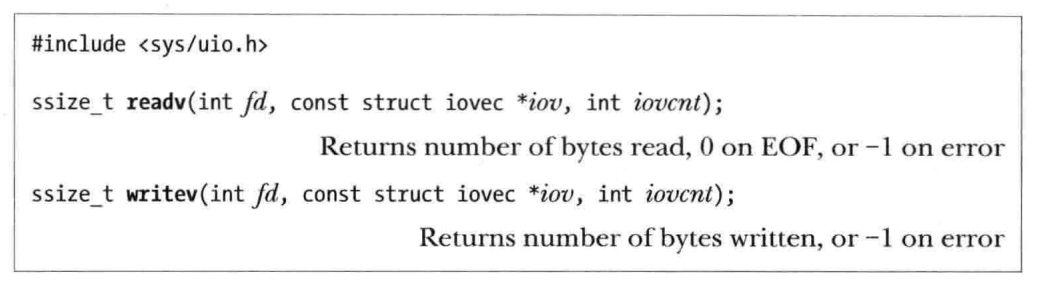


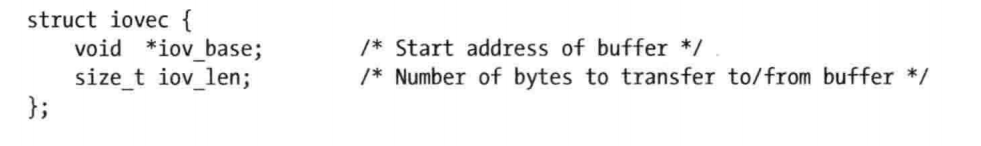


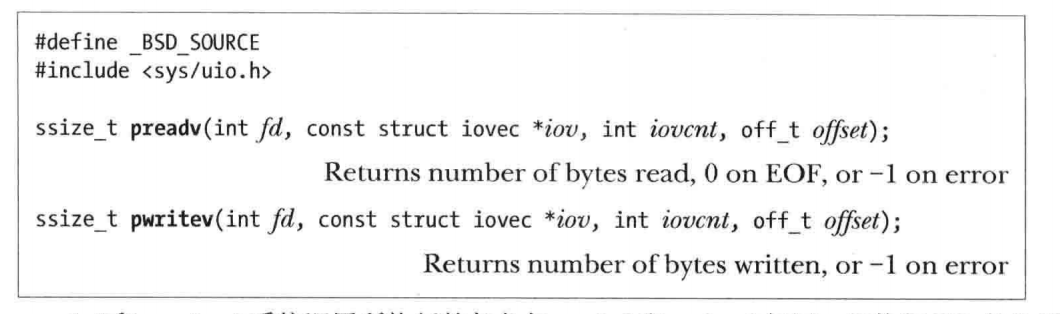
1. 在文件特定偏移量处进行IO，相对于文件开头，不改变文件偏移量，在多线程里非常好用。



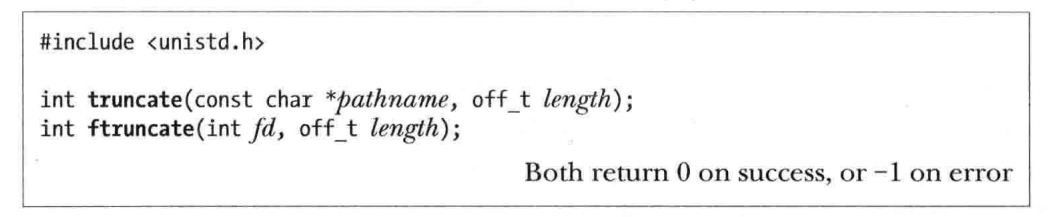
1. 分散输入和集中输出：输入输出是相对于CPU或者内存而言的。并且如下函数都具有原子性。







1. 截断文件：



1. 对于大文件而言，有两种方法：1.使用过渡型LFS API。2.将宏\_FILE\_OFFSET\_BITS定义为64（比较推荐这种用法）。注意，使用大文件扩展，参数类型比如off\_t要强制转换为long long类型输出才能保证兼容。
2. /dev/fd目录是虚拟目录，里面的文件代表本进程打开文件描述符对应的编号，主要用于shell里命令行中代替标准输入或者标准输出。
3. 介绍了一些创建临时文件的函数。

