

文件描述符（File Descriptor）简介

[维基百科](#):文件描述符在形式上是一个非负整数。实际上，它是一个索引值，指向内核为每一个进程所维护的该进程打开文件的记录表。当程序打开一个现有文件或者创建一个新文件时，内核向进程返回一个文件描述符。在程序设计中，一些涉及底层的程序编写往往会围绕着文件描述符展开。

一、文件描述符概念

Linux 系统中，把一切都看做是文件，当进程打开现有文件或创建新文件时，内核向进程返回一个文件描述符，文件描述符就是内核为了高效管理已被打开的文件所创建的索引，用来指向被打开的文件，所有执行I/O操作的系统调用都会通过文件描述符。

二、文件描述符、文件、进程间的关系

1.描述：

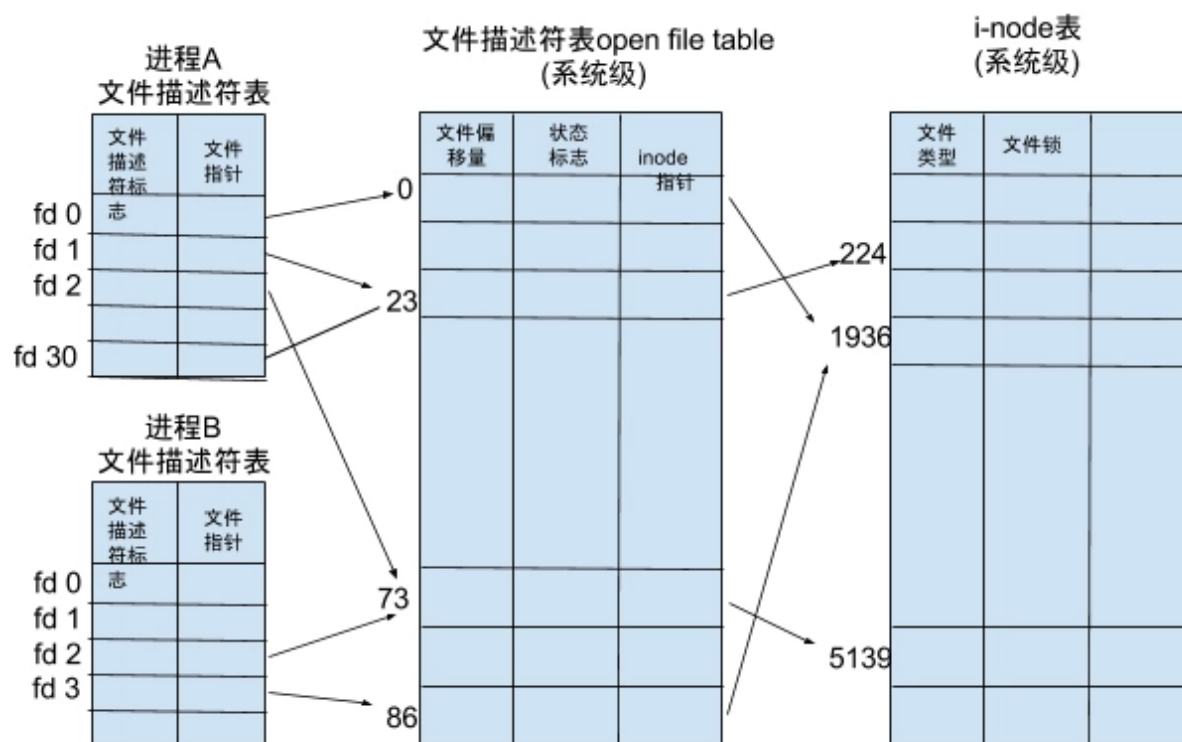
- 每个文件描述符会与一个打开的文件相对应
- 不同的文件描述符也可能指向同一个文件
- 相同的文件可以被不同的进程打开，也可以在同一个进程被多次打开

2.系统为维护文件描述符，建立了三个表

- 进程级的文件描述符表
- 系统级的文件描述符表
- 文件系统的i-node表 ([转到：阮一峰——理解inode](#))

	进程级文件描述符表	系统级文件描述符表 (open file table)	i-node表
记录内容	1.文件描述符标志 2.文件指针 (open file handle)	1.当前文件偏移量 2.状态标识 3.文件访问模式 (读，写，读和写) 4.对该文件i-node对象的引用 5.文件类型 (常规文件、套接字、FIFO...) 6.访问权限 7.一个指向该文件所持有的所累表的指针 8.文件的各种属性 9.与信号驱动相关的设置	1.文件类型 2.文件锁

3.通过这三个表，认识文件描述符



- 在进程A中，文件描述符1和30都指向了同一个打开的文件句柄（#23），这可能是该进程多次对执行 **打开** 操作
- 进程A中的文件描述符2和进程B的文件描述符2都指向了同一个打开的文件句柄（#73），这种情况有几种可能，1.进程A和进程B可能是父子进程关系;2.进程A和进程B打开了同一个文件，且文件描述符相同（低概率事件=_=）； 3.A、B中某个进程通过UNIX域套接字将一个打开的文件描述符传递给另一个进程。
- 进程A的描述符0和进程B的描述符3分别指向不同的打开文件句柄，但这些句柄均指向i-node表的相同条目（#1936），换言之，指向同一个文件。发生这种情况是因为每个进程各自对同一个文件发起了打开请求。同一个进程两次打开同一个文件，也会发生类似情况。