



链滴

文件句柄

作者: [kevin2020](#)

原文链接: <https://ld246.com/article/1515418371717>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

最近在看Lua的时候看到这么个概念，
文件句柄，之前操作系统也听说过，但还是有些模糊，查找了一下

简介

Windows中叫File Handle 译为文件句柄

句柄就是一个标识符，只要获得对象的句柄，我们就可以对对象进行任意的操作。

句柄不是指针，操作系统用句柄可以找到一块内存，这个句柄可能是标识符，map的key，也可能是针，看操作系统怎么处理的了。

在Linux里，这叫做文件描述符(File Descriptor),用户通过某个函数打开文件以获得句柄，此后用户纵文件皆通过该句柄进行。

设计这么一个句柄的原因在于句柄可以防止用户随意读写操作系统内核的文件对象。无论是Linux还Windows，文件句柄总是和内核的文件对象相关联的，但如何关联细节用户并不可见。内核可以通过句柄来计算出内核里文件对象的地址，但此能力并不对用户开放。

(FD=File Descriptor)

下面举一个实际的例子，在Linux中，值为0、1、2的fd分别代表标准输入、标准输出和标准错误输出。在程序中打开文件得到的fd从3开始增长。fd具体是什么呢？在内核中，每一个进程都有一个私有的“开文件表”，这个表是一个指针数组，每一个元素都指向一个内核的打开文件对象。而fd，就是这个的下标。当用户打开一个文件时，内核会在内部生成一个打开文件对象，并在这个表里找到一个空项让这一项指向生成的打开文件对象，并返回这一项的下标作为fd。由于这个表处于内核，并且用户无访问到，因此用户即使拥有fd，也无法得到打开文件对象的地址，只能够通过系统提供的函数来操作。

在程序设计中，句柄是一种特殊的智能指针。当一个应用程序要引用其他系统(如数据库、操作系统)管理的内存块或对象时，就要使用句柄。

句柄与普通指针的区别在于，指针包含的是引用对象的内存地址，而句柄则是由系统所管理的引用标，该标识可以被系统重新定位到一个内存地址上。这种间接访问对象的模式增强了系统对引用对象的制。

许多操作系统仍然把指向私有对象的指针以及进程传递给客户端的内部数组下标称为句柄。

总结

在文件I/O中，要从一个文件读取数据，应用程序首先要调用操作系统函数并传送文件名，并选一个该文件的路径来打开文件。该函数取回一个顺序号，即文件句柄(file handle)，该文件句柄对于打的文件是唯一的识别依据。要从文件中读取一块数据，应用程序需要调用函数ReadFile，并将文件句在内存中的地址和要拷贝的字节数传送给操作系统。当完成任务后，在通过调用系统函数来关闭该文。