## 内核brief：

内核的本质：控制计算机硬件资源，为上层的应用程序提供**运行环境**（就是提供接口）。公用函数库构建在系统调用接口上。

Shell为运行其他程序提供接口（个人理解就是程序调用程序或是os调用程序）。

UNIX文件系统是目录和文件的集合，目录的起点是根（root）。

每个进程都有一个工作目录（当前工作目录），所有相对路径名都从工作目录开始解释。//其实这样说来电脑os的文件系统真的不是完全面向消费者的，还是**太底层**了些。相反iPhone的操作系统还是完全面向消费者的。

文件描述符，一个小的非负整数，内核依据此标识一个特定进程访问的文件。当内核打开或是创建一个新文件，返回一个文件描述符。**读写时**，以此标识文件。

运行一个新程序，各种shell都会为其分配三个文件描述符：标准输入0，标准输出1，标准出错2。

如果命令不做特殊处理，**默认定位**至当前Shell环境即输入，输出以及出错信息在Shell进行操作。如ls > file.list 即将输出重定向至file.list文件。

程序在硬盘上是个可执行文件，由6个**exec函数**中的一个由内核将程序存入内存，并执行。

程序执行的实例叫进程，每个进程带有一个数字标识符，叫做**进程ID**。

进程控制主要用三个函数：**Fork，Exec和Waitpid。**

一个进程只有一个控制线程，一个进程里，所有线程共享一个地址空间、文件描述符、栈、以及进程相关的属性和资源。

**一个进程只有一个控制线程**，线程有线程ID，线程ID只在同一个进程内有效。

信号（signal）是通知进程已经发生某种情况的一种技术。

进程处理信号有三种方法：忽略，按照系统默认方式处理，进程提供专门函数处理信号。

所有os都提供多种服务的入口点，程序由此向内核请求服务。

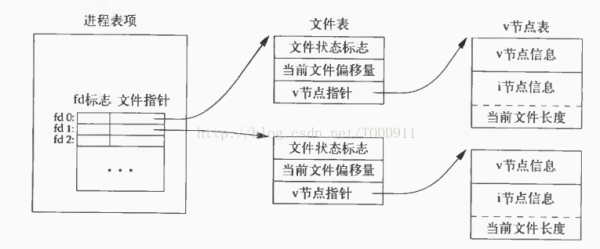
## 文件I/O

文件描述符：一个整数，当打开一个现有文件或者创建一个新文件的时候，内核向进程返回一个文件描述符。当读或写一个文件时，使用open()和create()返回的文件描述符标识该文件，将其作为参数传递给read()或者write()。

无缓冲I/O与缓冲I/O：无缓冲I/O在用户态没有缓冲，在内核态还是有的。当一个进程终止时，内核会自动关闭所有为它打开的文件。

UNIX系统支持不同进程间共享打开的文件，内核用于I/O的数据结构有三种。

每个进程一张进程表项，记录进程打开的文件，每个文件描述符表占一项，与每个文件描述符有文件描述符标志，和一个指向文件表的指针。内核为所有打开文件维持一张文件表，包含文件状态标志、当前文件偏移量、指向文件V节点表项的指针。每个打开的文件或设备都有一个类型和V节点结构。



硬链接指通过索引节点来进行链接。在Linux的文件系统中，保存在磁盘分区中的文件不管是什么类型都给它分配一个编号，称为索引节点号(Inode Index)。只删除一个链接并不影响索引节点本身和其它的链接，只有当最后一个链接被删除后，文件的数据块及目录的链接才会被释放。也就是说，文件真正删除的条件是与之相关的所有硬链接文件均被删除。