

## 系统调用（system call）

江学强，PB16120100

### 1. 什么是系统调用？

系统调用（system call）是操作系统中一个很重要的概念，Wiki 百科上的定义是：运行在使用者空间的程序向操作系统内核请求需要更高权限运行的服务。系统调用提供用户程序与操作系统之间的接口。从这样的定义我们可以看出系统调用其实就是操作系统提供的一组接口，并且这些接口之下的具体指令操作是在操作系统处于 **kernel mode** 时所执行，用户程序或者程序员不能直接进行系统调用，而是通过调用 **API**（应用程序接口）间接地进行系统调用。这样做的好处在于封装隐藏了系统调用的具体细节，程序员或者应用程序想要进行某种操作只需要关注相应地 **API** 而不需要去关注整个过程具体的实现步骤。此外，这种方式使得程序的通用性更好，因为不同系统间的系统调用往往是不同的，但是 **API** 却基本相同。现在。三种最常见的 **API** 分别是：windows 下的 win32 **API**，类 UNIX 系统下的 **POSIX API** 以及 **JAVA** 虚拟机中的 **JAVA API**。

### 2. 系统调用主要可以分为哪些种类？具体实现哪些功能？

进程控制：有些系统调用是专门用来控制进程的，例如一些系统调用用来终止进程（**terminate process**），一些是正常的终止，还有一些是异常终止，异常终止程序一般还有另外的系统调用用来保存必要的信息提供给程序员或者操作系统找出进程失败的原因。此外一些系统调用也需要将程序加载到内存中并且执行。当然在单任务系统和多任务系统中的进程控制是不同的，这里没有详述。

文件管理：文件系统是操作系统的重要组成部分，一些系统调用实现文件管理中的一些具体操作，例如新建、删除文件，打开、关闭文件，移动文件等。

设备管理：进程的进行经常需要利用计算机的资源，计算机资源是由操作系统进行调控的，很多操作系统把一些设备的操作也抽象成对文件的操作，这样很多对设备的系统调用看起来与对文件的系统调用很相似。事实上，文件本身就是一个抽象的产物，这样做方便系统的实现。

信息维护：前面已经提到当进程被异常终止时会启动系统调用保存一些必要的信息，这是所谓信息维护的一部分。此外，一些系统调用会为用户或者程序提供一些诸如日期、时间之类的信息，这也是系统维护。

通讯：这里主要说的是进程间的通讯，这里的进程可以是同一台计算机上的不同进程，也可以是不同计算机上的进程。有两种实现通信的模型，一种是基于信息传递的模型来实现通信，意思是两个（或多个）进程间实现通信需要彼此传递必要的信息，类似网络通信中的 **IP 地址**，为了标识不同的进程，这里也需要所谓的进程地址（**process name**）。操作系统提供一些系统调用来建立通信和中断通信等操作；还有一种实现通信的模型是共享空间模型，操作系统同样提供相应的系统调用来实现通信。这两种模型在很多的操作系统中都被实现，第一种模型更适用于一些数据量小的进程间的通信，这种模型一般不会产生冲突，而基于共享空间的通信机制更适用于计算机间的通信。

保护：保护提供了一种控制计算机资源访问权限的一种机制，系统调用对这些权限进行管理来保障计算机系统的安全。

### 3. 系统调用和 API 的区别与联系

系统调用通过中断向操作系统内核发送请求实现内核提供的某种服务，而 API 可以理解为只是函数的定义，规定了函数的功能，API 直接对程序员可见，而系统调用则并非如此。二者的联系简单看来是一个 API 调用一般会引起多个系统调用。

#### 参考文献格式:

- [1] Abraham Silberschatz. Operating System Concepts Essentials Second Edition Wiley 2014.
- [2] Wikipedia