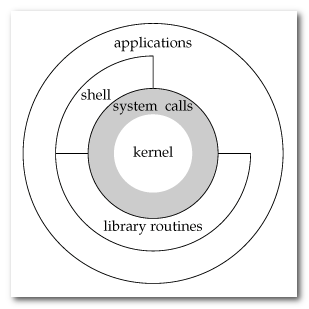
# 系统调用概述

**系统调用**，顾名思义，说的是操作系统提供给用户程序调用的一组“特殊”接口。其把用户进程的请求传达给内核，待内核把请求处理完毕后再将处理结果送回给用户空间。



# 

# 系统调用的实现

操作系统有不同的运行级别，用户态和内核态。

运行在内核态的进程可以毫无限制的访问各种资源，**如何切换到内核态呢**？

**答案是软件中断**。软件中断和我们常说的中断（硬件中断）不同之处在于，它是通过软件指令触发而并非外设引发的中断

中断有两个重要的属性，中断号和中断处理程序。在操作系统内核中维护着一个中断向量表（Interrupt Vector Table），这个数组存储了所有中断处理程序的地址，而中断号就是相应中断在中断向量表中的偏移量。

# 系统调用和库函数的区别

Linux 下对文件操作有两种方式：系统调用（system call）和库函数调用（Library functions）。

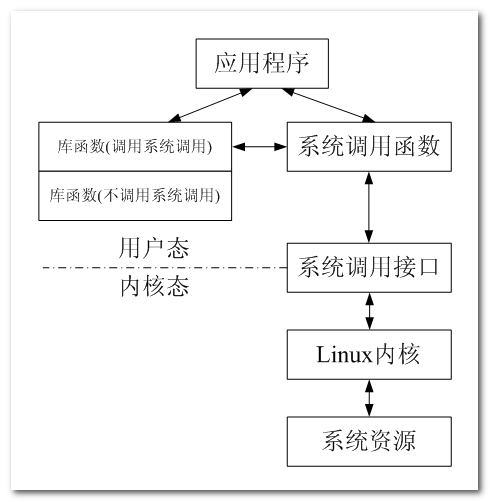
库函数由两类函数组成：

1）不需要调用系统调用

不需要切换到内核空间即可完成函数全部功能，并且将结果反馈给应用程序，如strcpy、bzero 等字符串操作函数。

2）需要调用系统调用

需要切换到内核空间，这类函数通过封装系统调用去实现相应功能，如 printf、fread等。



系统调用是需要时间的，程序中频繁的使用系统调用会降低程序的运行效率。