Linux操作系统编程

系统调用与库函数

操作系统用户接口

- **命令接口：** 以命令形式呈现在用户面前，方便用户直接或间接控制自己的作业
- **程序接口：** 为应用程序使用系统功能而设置，是应用程序取得操作系统服务的唯一途径。由一系列系统调用组成，每一个系统调用都是一个能完成特定功能的子程序。
- **图形接口：** 采用了图形化的操作界面，将各种应用程序和文件，直观、逼真地表示出来。

系统调用

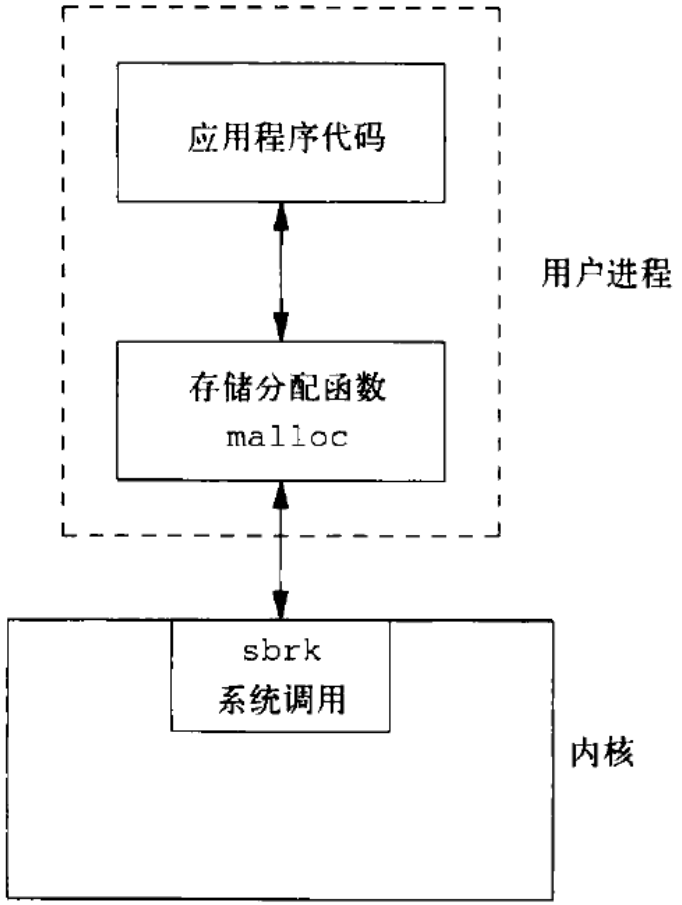
- **系统调用是内核提供的程序接口，是应用程序和硬件设备之间的中间层：**
  - 为应用程序提供了系统服务和硬件抽象能力，例如，当需要读文件时，应用程序可以不管磁盘类型和介质，甚至不用去管文件所在的文件系统到底是哪种类型；
  - 系统调用保证了系统的稳定和安全
  - 每个进程都运行在虚拟系统中
- **Research UNIX系统的第7个版本提供了大给50个系统调用，4.4BSD提供了大约110个，SVR4有大约120个。Linux根据不同的版本有240到260个。**
- **man 2 syscalls**

系统调用的类型

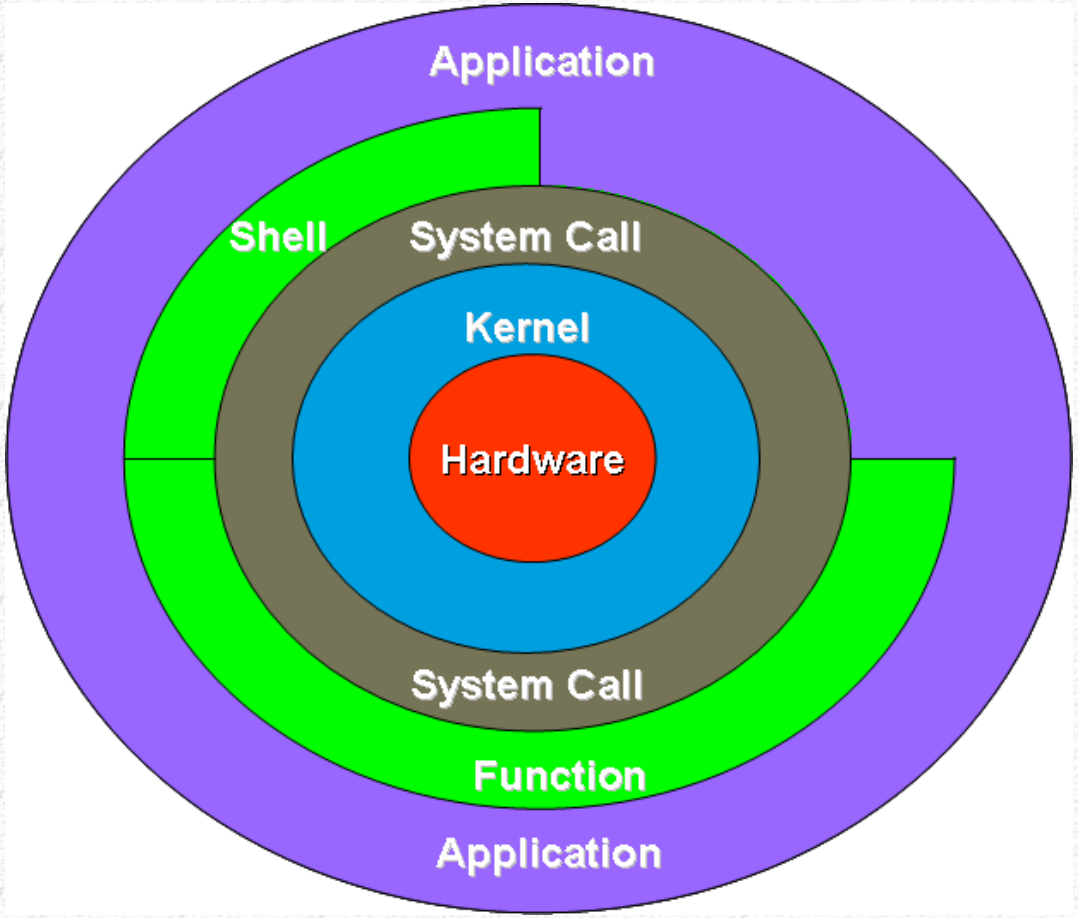
- **文件操作类系统调用：** 如打开、创建、读取、删除、修改文件；
- **进程控制类系统调用：** 如创建进程、设置或获取进程属性等；
- **通信类系统调用：** 创建进程间的通信连接，发送、接收消息，或其他的通信方式；
- **设备管理类系统调用：** 打开、关闭和操作设备；
- **信息维护类系统调用：** 在用户程序和OS之间传递信息。例如，系统向用户程序传送当前时间、日期、操作系统版本号等。

系统调用与C库函数的关系

- **系统调用和C库函数之间并不是一一对应的关系，可能几个不同的函数会调用到同一个系统调用：**
  - malloc函数和free函数都是通过sbrk系统调用来扩大或缩小进程的堆栈空间；
  - execl、execlp、execle、execv、execvp和execve函数都是通过execve系统调用来执行一个可执行文件
- **并非所有的库函数都会调用系统调用，例如，printf函数会调用write系统调用以输出一个字符串，但strcpy和atoi函数则不使用任何系统调用**



UNIX/Linux软件层次架构



API（POSIX）、C库函数、系统调用

- 一般而言，应用程序使用API而不是直接使用系统调用来编程
- 进程UNIX/Linux的C库遵循POSIX规范，以库函数的形式实现了POSIX API（在API中使用系统调用完成相应功能）。
- 参考：/usr/include/asm/unistd.h