

然后它再提问是否还有别的文件。最后，它总结需要执行的任务并且询问是否正确。尽管这样的用户界面可能适合初学者，但它会把有经验的程序员逼疯。他们想要的是一个佣人，不是一个保姆。

### 10.2.2 到Linux的接口

一个Linux系统可被看成一座金字塔，如图10-1所示。最底层的是硬件，包括CPU、内存、磁盘、显示器、键盘以及其他设备。运行在硬件之上的是操作系统。它的作用是控制硬件并且为其他程序提供系统调用接口。这些系统调用允许用户程序创立并管理进程、文件以及其他资源。

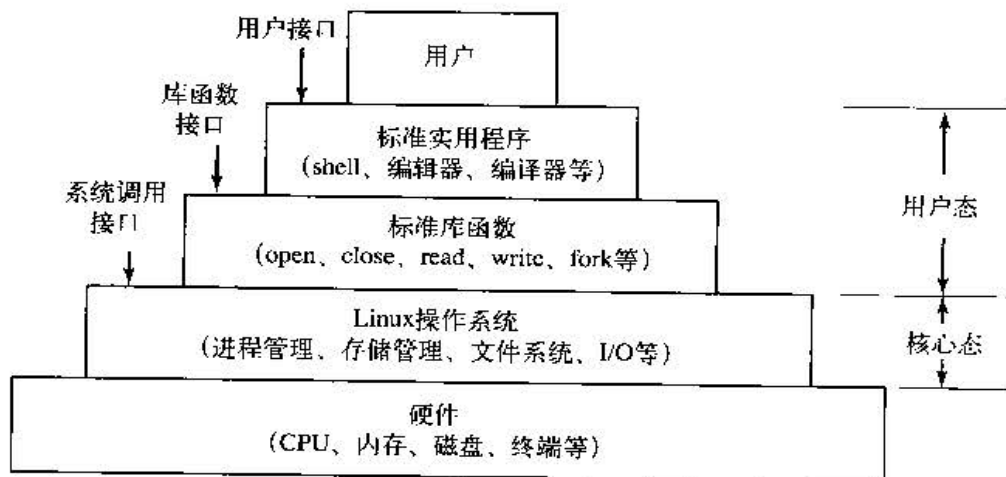


图10-1 Linux系统中的层次结构

程序通过把参数放入寄存器（有时是栈）来调用系统调用，并发出陷阱指令从用户模式切换到内核模式。由于不能用C语言写一条陷阱指令，因此系统提供了一个库，每个函数对应一个系统调用。这些函数是用汇编语言写的，不过可以从C中调用。每一个函数首先将参数放到合适的地方，然后执行陷阱命令。因此，为了执行read系统调用，一个C程序需要调用read库函数。值得一提的是，由POSIX指定的是库接口，而不是系统调用接口。换句话说，POSIX规定哪些库函数是一个符合标准规范的系统必须提供的，它们的参数是什么，它们的功能是什么，以及它们返回什么样的结果。POSIX根本没有提到真正的系统调用。

除了操作系统和系统调用库，所有版本的Linux必须提供大量的标准程序，其中一些是由POSIX 1003.2标准指定的，其他的根据不同版本的Linux而有所不同。它们包括命令处理器（shell）、编译器、编辑器、文本处理程序以及文件操作工具。用户使用键盘调用的是上述这些程序。因此，我们可以说Linux具有三种不同的接口：真正的系统调用接口、库函数接口和由标准应用程序构成的接口。

大多数的Linux个人计算机发行版都把上述的面向键盘的用户界面替换为面向鼠标的图形用户界面，而根本没有修改操作系统本身。正是这种灵活性让Linux如此流行并且在经历了如此多的技术革新后存活下来。

Linux的GUI和最初在20世纪70年代为UNIX系统开发的、后来由于Macintosh和Windows变得流行的GUI非常相似。这种GUI创建一个桌面环境，包括窗口、图标、文件夹、工具栏和拖拽功能。一个完整的桌面环境包含一个窗口管理器（负责控制窗口的摆放和外观），以及各种应用程序，并且提供一个一致的图形界面。比较流行的Linux桌面环境包括GNOME（GNU网络对象模型环境）和KDE（K桌面环境）。

Linux上的GUI由X窗口系统（常常称为X11或者X）所支持，它负责定义用于UNIX和类UNIX系统中基于位图显示的操作窗口的通信和显示协议。其主要组成部分X服务器，控制键盘、鼠标、显示器等设备，并负责输入重定向或者从客户程序接受输出。实际的GUI环境通常构建在一个包含与X服务器进行交互功能的低层库xlib上。图形界面将X11的基本功能进行拓展，丰富了窗口的显示，提供按钮、菜单、图标以及其他选项。X服务器可以通过命令行手动启动，不过通常在启动过程中由一个负责显示用户登录图形界面的显示管理器启动。

当在Linux上使用图形界面时，用户可以通过鼠标点击运行程序或者打开文件，通过拖拉将文件从