

第1章 引 论

现代计算机系统由一个或多个处理器、主存、磁盘、打印机、键盘、鼠标、显示器、网络接口以及各种其他输入/输出设备组成。一般而言，现代计算机系统是一个复杂的系统。如果每位应用程序员都不得不掌握系统所有的细节，那就不可能再编写代码了。而且，管理所有这些部件并加以优化使用，是一件挑战性极强的工作。所以，计算机安装了一层软件，称为操作系统，它的任务是为用户程序提供一个更好、更简单、更清晰的计算机模型，并管理刚才提到的所有这些设备。本书的主题就是操作系统。

多数读者都会对诸如Windows、Linux、FreeBSD或Mac OS X等某个操作系统有些体验，但表面现象是会骗人的。用户与之交互的程序，基于文本的通常称为shell，而基于图标的则称为图形用户界面（Graphical User Interface, GUI），它们实际上并不是操作系统的一部分，尽管这些程序使用操作系统来完成工作。

图1-1给出了在这里所讨论主要部件的一个简化视图。图的底部是硬件。硬件包括芯片、电路板、磁盘、键盘、显示器以及类似的设备。在硬件的顶部是软件。多数计算机有两种运行模式：内核态和用户态。软件中最基础的部分是操作系统，它运行在内核态（也称为管态、核心态）。在这个模式中，操作系统具有对所有硬件的完全访问权，可以执行机器能够运行的任何指令。软件的其余部分运行在用户态下。在用户态下，只使用了机器指令中的一个子集。特别地，那些会影响机器的控制或可进行I/O（输入/输出）操作的指令，在用户态中的程序里是禁止的。在本书中，我们会不断地讨论内核态和用户态之间的差别。

用户接口程序，shell或者GUI，处于用户态程序中的最低层次，允许用户运行其他程序，诸如Web浏览器、电子邮件阅读器或音乐播放器等。这些程序也大量使用操作系统。

操作系统所在的位置如图1-1所示。它运行在裸机之上，为所有其他软件提供基础的运行环境。

操作系统和普通软件（用户态）之间的主要区别是，如果用户不喜欢某个特定的电子邮件阅读器， he 可以自由选择另一个，或者选择自己写一个，但是他不能自行写一个属于操作系统一部分的时钟中断处理程序。这个程序由硬件保护，防止用户试图对其进行修改。

然而，有时在嵌入式系统（该系统没有内核态）或解释系统（如基于Java的操作系统，它采用解释方式而非硬件方式区分组件）中，上述区别是模糊的。

另外，在许多系统中，一些在用户态下运行的程序协助操作系统完成特权功能。例如，经常有一个程序供用户修改其口令之用。但是这个程序不是操作系统的一部分，也不在内核态下运行，不过它明显地带有敏感的功能，并且必须以某种方式给予保护。在某些系统中，这种想法被推向了极致，一些传统上被认为是操作系统的部分（诸如文件系统）在用户空间中运行。在这类系统中，很难划分出一条明显的界限。在内核态中运行的当然是操作系统的一部分，但是一些在内核外运行的程序也有争议地被认为是操作系统的一部分，或者至少与操作系统密切相关。

操作系统与用户（即应用）程序的差异并不在于它们所处的地位。特别地，操作系统是大型、复杂和长寿命的程序。Linux或Windows操作系统的源代码有5百万行数量级。要理解这个数量的含义，请考虑具有5百万行的一套书，每页50行，每卷1000页（比本书厚）。为了以书的大小列出一个操作系统，需

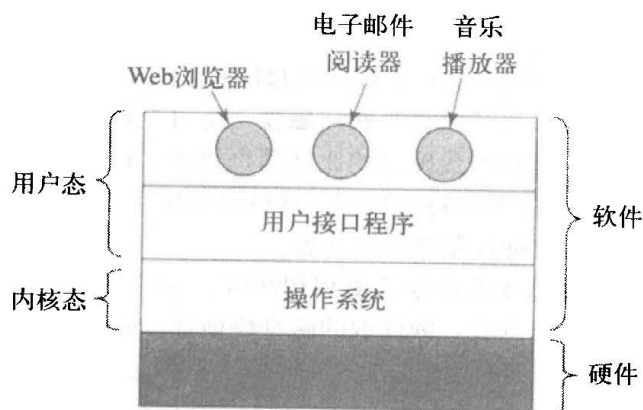


图1-1 操作系统所处的位置