

这些性能问题使得Symbian操作系统（以及其他基于微内核的操作系统）的设计者们对于设计以及实现细节给予了极大关注。设计的重点是最小化的、紧凑的集中服务。

12.2.3 Symbian操作系统纳核

Symbian操作系统的设计者在操作系统设计的核心采用了一种纳核的结构来处理微内核所具有的问题。正如在微内核结构中，某些系统功能被移到了用户空间服务器端，Symbian操作系统将需要复杂实现的功能分离到内核中，而只将最基本的功能放在系统核心的纳核中。

在Symbian操作系统中，纳核提供部分最基本的功能。在纳核中，运行在特权级别的简单线程完成着十分初级的功能。在这一层的实现中包括调度同步操作、中断处理和同步对象，如互斥变量以及信号量。这一层中的实现功能大多是可抢占的，而且是非常初级的（所以它们可以很快）。例如，动态内存分配对于纳核就是过于复杂的功能。

这种纳核的设计需要一个二级层次来实现较为复杂的内核功能。Symbian操作系统内核层提供了操作系统所需要的其他较为复杂的内核功能。每个在Symbian操作系统内核层的操作都是特权级的操作，并与纳核层的初级操作一起来完成更加复杂的内核工作。复杂的对象服务、用户态线程、进程调度以及上下文切换、动态内存、动态库加载、复杂的同步、对象及进程间通信只是在这层实现的部分操作。这层是完全可抢占式的，并且中断可以使其对任何一部分的执行进行重新调度，即使是在上下文转换的过程中也可以。

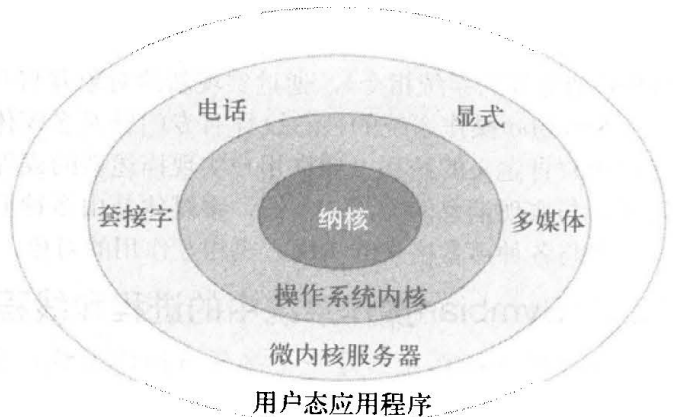


图12-1 Symbian操作系统内核结构具有多层

图12-1展示了一个完整的Symbian操作系统内核的结构。

12.2.4 客户机/服务器资源访问

正如我们所提到的那样，Symbian操作系统使用微内核设计并使用客户机/服务器模型来访问系统资源。需要访问系统资源的应用是客户端，操作系统中运行着的用来协调资源访问的程序是服务器端。在Linux中，人们可能需要调用open来打开一个文件；在Windows中，需要利用Microsoft API来创建一个窗口；而在Symbian操作系统中的过程均是相同的：首先建立一个到服务器端的连接，服务器端需要确认这个连接，然后对服务器端发出实现某个操作的请求。因此打开一个文件表示找到文件服务器端，调用connect建立与服务器端的连接，然后发送给服务器端一个附有某特定文件名字的open请求。

这样做对于保护资源有着几点好处。首先，它符合操作系统的面向对象以及微内核的设计。其次，这种结构对于管理多任务、多线程系统所需要的资源多重访问十分有效。最后，每个服务器都可以专注于它必须管理的资源，并能方便地进行升级以及替换为新的设计。

12.2.5 较大型操作系统的特点

尽管Symbian操作系统所针对的目标电脑规模较小，但它有着许多大型系统的特点。你可以在Symbian操作系统上找到大型操作系统（如Linux以及Windows）的各种特性，只是以另一种形式出现。Symbian操作系统与较大型的操作系统有一些共同的特性。

- 进程与线程：Symbian操作系统是一个多任务多线程的操作系统。许多进程可以同时运行，相互间可以进行通信，也可以在各进程内运行多个线程。
- 常见文件系统支持：Symbian操作系统利用一个文件系统模型来管理对系统存储空间的访问，正如大型操作系统一样。它具有一个与Windows兼容的默认文件系统（默认使用FAT-32文件系统），通过使用插件式接口支持其他文件系统。Symbian操作系统支持几种不同类型的文件系统，包括FAT-16、FAT-32、NTFS，以及许多存储卡格式（例如JFFS）。
- 网络：Symbian操作系统支持TCP/IP网络以及其他的通信接口，例如串行、红外和蓝牙。