这里有两个简单的例子。最早的电子邮件系统通过ACSII文本发送消息。它们是完全安全的。ASCII文本不可能对计算机系统造成损失。然后人们想方设法扩展电子邮件的功能，引入了其他类型的文档，如可以包含宏程序的Word文件。读这样的文件意味着在自己的计算机上运行别人的程序。无论沙盒怎么有效，在自己的计算机上运行别人的程序必定比ASCII文本要危险得多。是用户要求从过去的文本格式改为现在的活动程序吗？大概不是吧，但系统设计人员认为这是个极好的主意，而没有考虑到隐含的安全问题。

第二个例子是关于网页的。过去的HTML网页没有造成大的安全问题（虽然非法网页也可能导致缓冲溢出攻击）。现在许多网页都包含了可执行程序（Applet），用户不得不运行这些程序来浏览网页内容，结果一个又一个安全漏洞出现了。即便一个漏洞被补上，又会有新的漏洞显现出来。当网页完全是静态的时候，是用户要求增加动态内容的吗？可能动态网页的设计者也记不得了，但随之而来是大量的安全问题。这有点像负责说“不”的副总统在车轮下睡着了。

实际上，确实有些组织认为，与非常漂亮的新功能相比，好的安全性更为重要。军方组织就是一个重要的例子。在接下来的几节中，我们将研究相关的一些问题，不过这些问题不是几句话便能说清楚的。要构建一个安全的系统，需要在操作系统的核心中实现安全模型，且该模型要非常简单，从而设计人员确实能够理解模型的内涵，并且顶住所有压力，避免偏离安全模型的要求去添加新的功能特性。

9.3.5 可信计算基

在安全领域中，人们通常讨论*可信系统*而不是安全系统。这些系统在形式上申明了安全要求并满足了这些安全要求。每一个可信系统的核心是最小的*可信计算基*（Trusted Computing Base，TCB），其中包含了实施的所有安全规则所必需的硬件和软件。如果这些可信计算基根据系统规约工作，那么，无论发生了什么错误，系统安全性都不会受到威胁。

典型的TCB包括了大多数的硬件（除了不影响安全性的I/O设备）、操作系统核心的一部分、大多数或所有掌握超级用户权限的用户程序（如在UNIX中的SETUID根程序）。必须包含在操作系统中的TCB功能有：进程创建、进程切换、内存页面管理以及部分的文件以及I/O管理。在安全设计中，为了减少空间以及纠正错误，TCB通常完全独立于操作系统的其他部分。

TCB中的一个重要组成部分是引用监视器，如图9-11所示。引用监视器接受所有与安全有关的系统请求（如打开文件等），然后决定是否允许运行。引用监视器要求所有的安全问题决策都必须在同一处考虑，而不能跳过。大多数的操作系统并不是这样设计的，这也是它们导致不安全的部分原因。

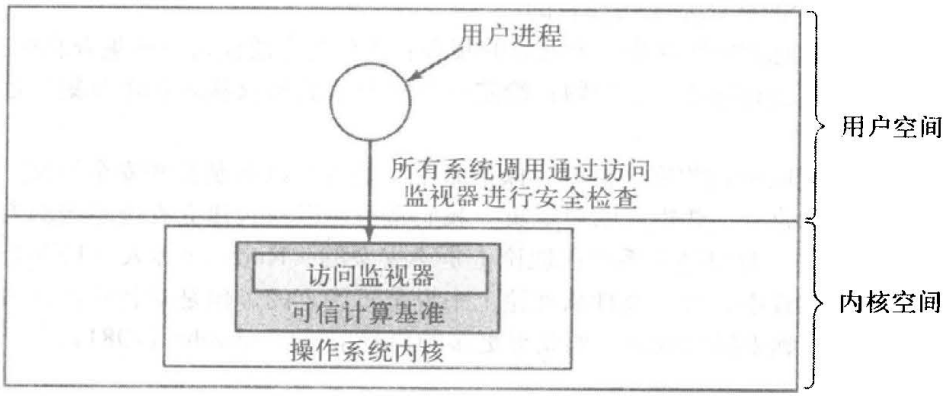

图9-11　引用监视器

现今安全研究的一个目标是将可信计算基中数百万行的代码缩短为只有数万行代码。在图1-26中我们看到了MINIX 3操作系统的结构。MINIX 3是具有POSIX兼容性的系统，但又与Linux或FreeBSD有着完全不同的结构。在MINIX 3中，只有4000行左右的代码在内核中运行。其余部分作为用户进程运行。其中，如文件系统和进程管理器是可信基的一部分，因为它们与系统安全息息相关；但是诸如打印机驱动和音频驱动这样的程序并不作为可信计算库的一部分，因为不管这些程序出了什么问题，它们的行为也不可能危及系统安全。MINIX 3将可信计算库的代码量减少了两个数量级，从而潜在地比传统系统设计提供了更高的安全性。