心层决定，它控制着线程的调度和同步，以及它们之间相互控制的对象，如APC。进程包含线程、地址空间和一个可以用来处理进程指定内核态对象的句柄表。进程还包括调度器进行地址空间交换和管理进程中的具体硬件信息（如段描述符）所需要的信息。我们将在11.4节研究进程和线程的管理。

执行内存管理器实现了虚拟内存架构的需求分页。它负责管理虚拟页映射到物理页帧，管理现有的物理帧，和使用备份管理磁盘上页面文件，这些页面文件是用来备份那些不再需要加载到内存中的虚拟页的私有实例。该内存管理器还为大型服务器应用程序提供了特殊功能，如数据库和编程语言运行时的组件，如垃圾收集器。我们将在11.5节中研究内存管理。

内存管理器优化I/O的性能，文件系统内核虚拟地址空间保持一个内存的文件系统页。内存管理器使用虚拟的地址进行缓存，也就是说，按照它们文件所在位置来组织缓存页。这不同于物理块内存，例如在UNIX中，系统为原始磁盘卷保持一个物理地址块的内存。

内存的管理是使用内存映射文件来实现的。实际的缓存是由内存管理器完成。内存管理器需要关心的只是文件的哪些部分需要内存，以确保缓存的数据即时地刷新到磁盘中，并管理内核虚拟地址映射缓存文件页。如果一个页所需的I/O文件在缓存中没有，该页在使用内存管理器时将会发生错误。我们会在11.6节中学习内存管理器。

安全引用监视器（security reference monitor）执行Windows详细的安全机制，以支持计算机安全要求的国际标准的通用标准（Common Critieria），一个由美国国防部的橘皮书的安全要求发展而来的标准。这些标准规定了一个符合要求的系统必须满足的大量规则，如登录验证、审核、零分配的内存等更多的规则。一个规则要求，所有进入检查都由系统中的一个模块进行检查。在Windows中此模块就是内核中的安全监视器。我们将在11.9节中更详细地学习安全系统。

执行体中包括其他一些组件，我们将简要介绍。如前所述，配置管理实现注册表的执行组件。注册表中包含系统配置数据的文件的系统文件称为储巢（hive）。最关键的储巢是系统启动时加载到内存的系统储巢。只有在执行体成功地初始化其主要组件，包括了系统磁盘的I/O驱动程序，之后才是文件系统中储巢关联的内存中的储巢副本。因此，如果试图启动系统时发生不测，磁盘上的副本是不太可能被损坏的。

LPC的组成部分提供了运行在同一系统的进程之间的高效内部通信。这是一个基于标准的远程过程调用（RPC）功能，实现客户机/服务器的处理方式的数据传输。RPC还使用命名管道和TCP/IP作为传输通道。

在Windows Vista（现在称为ALPC、高级LPC）中LPC大大加强了对RPC新功能的支持，包括来自内核态组件的RPC，如驱动。LPC是NT原始设计中的一个重要的组成部分，因为它被子系统层使用，实现运行在每个进程和子系统进程上库存例程的通信，这实现了一个特定操作系统的个性化功能，如Win32或POSIX。

Windows NT 4.0中的许多代码与Win32进入内核的图形界面相关，因为当时的硬件无法提供所需的性能。该代码以前位于csrss.exe子系统进程，执行Win32接口。以内核为基础的图形用户界面的代码位于一个专门的内核驱动win32k.sys中。这一变化预计将提高Win32的性能，因为额外的用户态/内核态的转换和转换地址空间的成本经由LPC执行通信是被清除的。但并没能像预期的那样取得成功，因为运行在内核中的代码要求是非常严格的，运行在内核态上的额外消耗抵消了因减少交换成本获得的收益。

7. 设备驱动程序

最后一部分图11-13是设备驱动程序的组成。在Windows中的设备驱动程序的动态链接库是由NTOS装载。虽然它们主要是用来执行特定硬件的驱动程序，如物理设备和I/O总线，设备驱动程序的机制也可作为内核态的一般可扩展性的机制。如上所述，大部分的Win32子系统是作为一个驱动程序被加载。

I/O管理器组织的数据按照一定的路线流经过每个设备实例，如图11-16。这个路线称为设备栈，由分配到这条路线上的内核设备对象的私有实例组成。设备堆栈中的每个设备对象与特定的驱动程序对象相关联，其中包含日常使用的I/O请求的数据包流经该设备堆栈的表。在某些情况下，堆栈中的设备驱动程序表示其唯一的目的是在某一特定的设备上过滤I/O操作目标、总线或网络驱动器。过滤器的使用是有一些原因的。有时预处理或后处理I/O操作可以得到更清晰的架构，而其他时候只是以实用为出发