

件管理操作系统的一部分，例如I/O、内存、进程、对象等。

对于大多数操作系统而言，许多功能在Windows上执行就像库的编码。除非在内核方式下运行，它的数据结构可以被共享和保护，以避免用户态下的编码访问，因此它具有硬件状态的访问权限，例如MMU控制寄存器。但是另一方面，执行体只是代表它的调用者简单执行操作系统的函数，因此它运行在它的调用者的线程中。

当任何执行控制操作阻塞等待与其他线程同步时，用户态线程也会阻塞。这在为一个特殊的用户态线程工作时是有意义的，但是在做一些相关的内核处理任务时是不公平的。当执行体认为一些内核处理线程是必须的时候，为了避免劫持当前的线程，一些内核态线程就会具体于特定的任务而产生，例如确保更改了的页面会被回写到硬盘上。

对于可预见的低频率任务，会有一个线程一秒运行一次而且由一个长的项目单来处理。对于不可预见的工作，有一个之前曾经提到的高优先级的辅助线程池，通过将队列请求和发送辅助线程等待的同步事件信号，可以用来运行有界任务。

对象管理器管理在执行体使用的大部分内核态对象，包括进程、线程、文件、信号、I/O设备及驱动、定时器等。就像之前提到的，内核态对象仅仅是内核分配和使用的数据结构。在Windows中，内核数据结构有许多共同特点，即它们在管理标准功能中特别有用。

这些功能由对象管理器提供，包括管理对象的内存分配和释放，配额计算，支持通过句柄访问对象，为内核态指针引用保留引用计数，在NT名字空间给对象命名，为管理每一个对象的生命周期提供可扩展的机制。需要这些功能的内核数据结构是由对象管理器来管理的。其他数据结构，例如内核层使用的控制对象，或仅仅是内核态对象的扩展对象，不由对象管理器管理。

对象管理器的每一个对象都有一个类型用来指定这种类型的对象的生命周期怎样被管理。这些不是面向对象意义中的类型，而仅仅是当对象类型产生时的一个指定参数集合。为了产生一个新的类型，一个操作元件只需要调用一个对象管理器API即可。对象在Windows的函数中很重要，在下面的章节中将会讨论有关对象管理器的更多细节。

I/O管理器为实现I/O设备驱动提供了一个框架，同时还为设备上的配置、访问和完成操作提供一些特定的运行服务。在Windows中，设备驱动器不仅仅管理硬件设备，它们还为操作系统提供可扩展性。在其他类型的操作系统中被编译进内核的功能是被Windows内核动态装载和链接的，包括网络协议栈和文件系统。

最新的Windows版本对在用户态上运行设备驱动程序有更多的支持，这对新的设备驱动程序是首选的模式。Windows Vista有超过100万不同的设备驱动程序，工作着超过了100万不同的设备。这就意味着要获取正确的代码。漏洞导致设备在用户态的进程中崩溃而不能使用，这比造成对系统进行检测错误要好得多。错误的内核态设备驱动是导致Windows可怕的BSOD（蓝屏死机）的主要来源，它是Windows检测到致命的内核态错误并关机或重新启动系统。蓝屏死机可以类比为UNIX系统中的内核恐慌。

在本质上，微软现在已经正式承认那些在microkernels研究领域的如MINIX 3和L4的研究员多年来都知道的结果：在内核中有更多的代码，那么内核中就有更多缺陷。由于设备驱动程序占了70%的内核代码，更多的驱动程序可以进入用户态进程，其中一个bug只会触发一个单一驱动器的失败（而不是降低整个系统）。从内核到用户态进程的代码移动趋势将在未来几年加速发展。

I/O管理器还包括即插即用和电源管理设施。当新设备在系统中被检测到，即插即用就开始工作。该即插即用设备的子模块首先被通知。它与服务一起工作，即用户态即插即用管理器，找到适当的设备驱动程序并加载到系统中。找到合适的设备驱动程序并不总是很容易，有时取决于先进的匹配具体软件设备特定版本的驱动程序。有时一个单一的设备支持一个由不同公司开发的多个驱动程序所支持的标准接口。

电源管理能降低能源消耗，延长笔记本电脑电池寿命，保存台式电脑和服务器的能量。正确使用电源管理是具有挑战性的，因为在把设备和buses连接到CPU和内存时有许多微妙的依赖性。电力消耗不只是由设备供电时的影响，而且还由CPU的时钟频率影响，这也是电源管理在控制。

我们会在11.7节对I/O进一步研究和以及在11.8节中介绍最重要的NT文件系统NTFS。

进程管理管理着进程和线程的创建和终止，包括建立规则和参数指导它们。但是线程运行方面由核