

过名称打开一个对象并遭遇其管理的名字空间树的叶结点，它检查该叶结点对象类型是否指定了一个 Parse 语句。如果有，它会引用该语句，将路径名中未用的部分传给它。再以文件对象为例，叶子结点是一个表现特定文件系统卷的设备对象。Parse 语句由 I/O 管理器执行，并发起在对文件系统的 I/O 操作，以填充一个指向文件的公开实例到该文件对象，这个文件是由路径名指定的。我们将在以后逐步探索这个特殊的实例。

QueryName 语句是用来查找与对象关联的名字。Security 语句用于得到、设置或删除该安全描述符的对象。对于大多数类型的对象，此程序在运行的安全引用监视器组件里提供一个标准的切入点。

注意，在图 11-20 里的语句并不执行每种对象类型最感兴趣的操作。相反，这些程序提供给对象管理器正确实现功能所需要的回调函数，如提供对对象的访问和对对象完成时的清理工作。除了这些回调，对象管理器还提供了一套通用对象例程，例如创建对象和对象类型，复制句柄，从句柄或者名字获得引用指针，并增加和减去对象头部的参考计数。

对象感兴趣的操作都是在本地 NT API 系统调用，如 NtCreateProcess、NtCreateFile 或 NtClose（关闭句柄所有类型的通用操作），如图 11-9 所示。

虽然对象名字空间对整个运作的系统是至关重要的，但却很少有人知道它的存在，因为没有特殊的浏览工具的话它对用户是不可见的。winobj 就是一个这样的浏览工具，在 www.microsoft.com/technet/sysinternals 可免费获得。在运行时，此工具描绘的对象的名称空间通常包含对象目录，如图 11-21 列出来的及其他一些。

一个被奇怪地命名为 \?? 的目录包含用户的所有 MS-DOS 类型的设备名称，如 A: 表示软驱，C: 表示第一块硬盘。这些名称其实是在设备对象活跃的地方链接到目录 \ 装置的符号。使用名称 \?? 是因为其按字母顺序排列第一，以加快查询从驱动器盘符开始的所有路径名称。其他的对象目录的内容应该是自解释的。

目录	内容
??	查找类似 C: 的 MS-DOS 设备的查找起始位置
DosDevices	目录 ??
Device	所有 I/O 设备
Driver	每个加载的设备驱动对应的对象
ObjectTypes	如图 11-22 中列出的类型的对象
Windows	发送消息到所有 Win32 GUI 窗口的对象
BaseNamedObjects	用户创建的 Win32 对象，如信号量、互斥量等
Arcname	由启动装载器发现的分区名称
NLS	National 语言支持对象
FileSystem	文件系统驱动对象和文件系统识别对象
Security	安全系统的对象
KnownDLLs	较早开启和一直开启的关键共享库

图 11-21 在对象名字空间中的一些典型目录

如上所述，对象管理器保持一个单独的句柄为每个对象计数。这个计数是从来不会大于指针引用计数，因为每个有效的句柄对象在它的句柄表入口有一个引用指针。使用单独句柄计数的理由是，当最后一个用户态的引用消失的时候，许多类型的对象可能需要清理自己的状态，尽管它们尚未准备好让它们的内存删除。

以一个文件对象为例表示一个打开文件的实例。Windows 系统中文件被打开以供独占访问。当文件对象的最后一个句柄被关闭，重要的是在那一刻就应该删除专有访问，而不是等待任何内核引用最终消失（例如，在最后一次从内存冲洗数据之后。）否则，从用户态关闭并重新打开一个文件可能无法按预期的方式工作，因为该文件看来仍然在使用中。

虽然对象管理器在内核具有全面的管理机制来管理内核中的对象生命周期，不论是 NT API 或 Win32