这些底层程序通常是以操作系统的一部分（主要是服务和子系统），或者是内核态的设备驱动程序的形式交付的。本地的NT系统调用在版本的升级中并没有太大的改变，但是微软并没有选择公开，而Windows的应用程序都是基于Win32的，因此Win32 API在不同Windows操作系统中是通用的，从而能够让这些应用程序在基于MS-DOS和NT Windows的系统中正确运行。

对象类别	例子
同步	信号量、互斥量、时间、IPC端口、I/O完成队列
I/O	文件、设备、驱动、定时器
程序	任务、进程、线程、节、标签
Win32 GUI	桌面、应用程序回调

图11-8 内核态对象类型的普通类别

大多数内部的NT系统调用都是对内核态对象进行操作的，包括文件、线程、管道、信号量等。图11-8中给出了一些Windows Vista中NT所支持的常见内核态对象。以后，我们讨论内核对象管理器时，会讨论具体对象类型细节的。

有时使用术语“对象”来指代操作系统所控制的数据结构，这样就会造成困惑，因为错误理解成“面向对象”了。操作系统的对象提供了数据隐藏和抽象，但是缺少了一些面向对象体系基本的性质，如继承和多态性。

在本地NT API调用中存在创建新的内核态对象或操作已经存在的对象的调用。每次创建和打开对象的调用都返回一个结果叫句柄（handle）给调用者（caller）。句柄可在接下来用于执行对象的操作。句柄是特定于创建它们的具体的进程的。通常句柄不可以直接交给其他进程，也不能用于同一个对象。然而，在某些情况下通过一个受保护的方法有可能把一个句柄复制到其他进程的句柄表中进行处理，允许进程共享访问对象——即使对象在名字空间无法访问。复制句柄的进程必须有来源和目标进程的句柄。

每一个对象都有一个和它相关的安全描述信息，详细指出对于特定的访问请求，什么对象能够或者不能够针对一个特定的目标进行何种操作。当句柄在进程之间复制的时候，可添加具体的被复制句柄相关的访问限制。从而一个进程能够复制一个可读写的句柄，并在目标进程中把它改变为只读的版本。

并不是所有系统创建的数据结构都是对象，并不是所有的对象都是内核对象。那些真正的内核态对象是那些需要命名、保护或以某种方式共享的对象。通常，这些内核态对象表示了在内核中的某种编程抽象。每一个内核态的对象有一个系统定义类型，有明确界定的操作，并占用内核内存。虽然用户态的程序可以执行操作（通过系统调用），但是不能直接得到数据。

图11-9为一些本地API的示例，通过特定的句柄操作内核对象，如进程、线程、IPC端口和扇区（用来描述可以映射到地址空间的内存对象）。NtCreateProcess返回一个创建新进程对象的句柄，SectionHandle代表一个执行实例程序。当遇到异常时控制进程（例如异常、越界），DebugPort Handle用来在出现异常（例如，除零或者内存访问越界）之后把进程控制权交给调试器的过程中与调试器通信。

```
NtCreateProcess(&ProcHandle, Access, SectionHandle, DebugPortHandle, ExceptPortHandle, ...)
NtCreateThread(&ThreadHandle, ProcHandle, Access, ThreadContext, CreateSuspended, ...)
NtAllocateVirtualMemory(ProcHandle, Addr, Size, Type, Protection, ...)
NtMapViewOfSection(SectHandle, ProcHandle, Addr, Size, Protection, ...)
NtReadVirtualMemory(ProcHandle, Addr, Size, ...)
NtWriteVirtualMemory(ProcHandle, Addr, Size, ...)
NtCreateFile(&FileHandle, FileNameDescriptor, Access, ...)
NtDuplicateObject(srcProcHandle, srcObjHandle, dstProcHandle, dstObjHandle, ...)
```

图11-9 在进程之间使用句柄来管理对象的本地NT API调用示例

NtCreate线程需要ProcHandle，因为ProcHandle可以在任意一个含有句柄的进程中（有足够的访问权限）创建线程。同样，NtAllocateVirtualMemory、NtMapViewOfSection、NtReadVirtualMemory和NtWriteVirtualMemory可使进程不仅在自己的地址空间操作，也可以在分配虚拟地址和映射段，还可以读写其他进程的虚拟内存。NtCreateFile是一个内部API调用，用来创建或打开文件。NtDuplicate-Object，可以在不同的进程之间复制句柄的API调用。