6) 该设备对象中遇到指向文件系统的IRP可以表示为文件系统筛选驱动程序，这可能在该操作到达对应的文件系统设备对象之前修改I/O操作。通常情况下这些中间设备代表系统扩展，例如反病毒过滤器。

7) 文件系统设备对象有一个链接到文件系统驱动程序对象，叫NTFS。因此，驱动对象包含NTFS内创建操作的地址范围。

8) NTFS将填补该文件中的对象并将它返回到I/O管理器，I/O管理器备份堆栈中的所有设备，直到IopParseDevice返回对象管理器（如11.8节所述）。

9) 在对象管理器以其名字空间中的查找结束。它从Parse程序收到一个初始化对象（这正好是一个文件对象，而不是原来对象发现的设备对象）。因此，对象管理器为文件对象在目前进程的句柄表里创建了一个句柄，并对需求者返回句柄。

10) 最后一步是返回用户态的调用者，在这个例子里就是Win32 API CreateFile，它会把句柄返回给应用程序。

可执行组件能够通过调用ObCreateObjectType接口给对象管理器来动态创建新的类型。由于每次发布都在变化，所以没有一个限定的对象类型定义表。图11-23列出了在Windows Vista中非常通用的一些对象类型，供快速参考。

类　型	描　述
Process	用户进程
Thread	进程里的线程
Semaphore	进程内部同步的信号量
Mutex	用来控制进入关键区域的二进制信号量
Event	具有持久状态（已标记信号\未标记信号）的同步对象
ALPC Port	内部进程消息传递的机制
Timer	允许一个线程固定时间间隔休眠的对象
Queue	用来完成异步I/O通知的对象
Open file	关联到某个打开的文件的对象
Access token	某个对象的安全描述符
Profile	描述CPU使用情况的数据结构
Section	表述映射的文件的对象
Key	注册表关键字，用于把注册信息关联到某个对象管理名字空间
Object directory	对象管理器中一组对象的目录
Symbolic link	通过路径名引用到另一个对象管理器对象
Device	物理设备、总线、驱动或者卷实例的I/O设备对象
Device driver	每一个加载的设备驱动都有它自己的一个对象

图11-23 对象管理器管理的一些通用可执行对象类型

进程（process）和线程（thread）是明显的。每个进程和每个线程都有一个对象来表示，这个对象包含了管理进程或线程所需的主要属性。接下来的三个对象：信号量、互斥体和事件，都可以处理进程间的同步。信号量和互斥体按预期方式工作，但都需要额外的响铃和警哨（例如，最大值和超时设定）。事件可以在两种状态之一：已标记信号或未标记信号。如果一个线程等待事件处于已标记信号状态，线程被立即释放。如果该事件是未标记信号状态，它会一直阻塞直到一些其他线程信号释放所有被阻止的线程（通知事件）的活动或只是第一个被阻止的线程（同步事件）。也可以设置一个事件，这样一种信号成功等待后，它会自动恢复到该未标记信号的状态而不是处在已标记信号状态。

端口、定时器和队列对象也与通信和同步相关。端口是进程之间交换LPC消息的通道。定时器提供一种为特定的时间区间内阻塞的方法。队列用于通知线程已完成以前启动的异步 I/O 操作，或一个端口有消息等待。（它们被设计来管理应用程序中的并发的水平，以及在使用高性能多处理器应用中使用，如SQL）。

当一个文件被打开时，Open file对象将会被创建。没打开的文件，并没有对象由对象管理器管理。访问令牌是安全的对象。它们识别用户，并指出用户具有什么样的特权，如果有的话。配置文件是线程