问权限。

还有更高级的处理文件数据的方法。除了主要的文件流，存在NTFS文件系统上的文件可以拥有额外的文件流。文件（甚至包括整个卷）可以被加密。文件可以被压缩成为一组相对稀疏的字节流，从而节省磁盘空间。不同硬盘的文件系统的卷可以通过使用不同级别的RAID存储而组织起来。修改文件或者目录可以通过一种直接通知的方式来实现，或者通过读NTFS为每个卷维护的日志来实现。

每个文件系统的卷默认挂载在NT的名字空间里，根据卷的名字来排列，因此，一个文件 \foo\bar可以命名成\Device\HarddiskVolume\foo\bar。对于NTFS的卷来说，挂载点(Windows称作再分解点）和符号链接用来帮助组织卷。

低级别的Windows I/O模式基本上是异步的。一旦一个I/O操作开始，系统调用将允许线程对I/O操作进行初始化并且开始I/O操作。Windows支持取消操作，以及一系列的不同机制来支持线程和I/O操作完成之后的同步。Windows也允许程序规定在文件打开时I/O操作必须同步，许多库函数，例如C库和许多Win32调用，也规定I/O的同步已支持兼容性或者简化编程模型。在这些情况下，执行体会在返回到用户态前和I/O操作结束时进行同步。

Win32提供的另一些调用是安全性相关的。每个线程将和一个内核对象进行捆绑，称作令牌（token），这个令牌提供关于该线程的身份和权限相关的信息。每个目标可以有一个ACL（访问权限控制列表），这个列表详细描述了哪种用户有权限访问并且对其进行操作。这种方式通过了一种细粒度的安全机制，可以指定具体哪些用户可以或者禁止访问特定的对象。这种安全模式是可以扩展的，允许应用程序添加新的安全规则，例如限制访问时间。

Win32的名字空间不同于前面描述的NT内核名字空间。NT内核空间仅仅只有一部分对Win32 API函数可见（即使整个NT名字空间可以通过Win32使用特殊字符串来访问，如“\\.”）。在Win32中，文件访问权限和驱动器号相关。NT目录\DosDevices里包含了对一个从驱动器号到实际设备对象的数个符号链接。例如，\DosDevices\C:是指向\Device\HarddiskVolume1。这个目录同样也包含了其他Win32设备的链接，如COM1:、LPT1:和NUL:(端口号和打印端口，以及非常重要的空设备)。\DosDevices是一个真正指向\??的链接，这样有利于提高效率。另外一个NT文件夹，\BaseNamedObjects用来存储各种各样的内核对象，这些文件可以通过Win32 API来访问。这些对象包括用来同步的对象，如信号、共享内存、定时器以及通信端口，MS-DOS和设备名称。

对于底层系统接口，我们额外说一下，Win32 API也支持许多GUI操作，包括系统所有图形接口的调用。有对窗口的创建、摧毁、管理和使用的调用，以及支持菜单、工具条、状态栏、滚动条、对话框、图标和许多在屏幕上显示的元素。Win32还提供调用来画几何图形、填充、使用调色板、处理文字以及在屏幕上放置图标等。也支持对键盘鼠标和其他输入设备的响应，如音频、打印等其他输出设备。

GUI操作直接使用win32k.sys驱动，这个驱动使用特殊的函数从用户态去访问内核态的接口。因为这些调用不包含NT操作系统中的系统调用，我们将不会详细讨论。

11.2.3 Windows 注册表

名字空间的根在内核中维护。存储设备，如系统的卷，附属于名字空间中。因为名字空间会因为系统的每次启动重新构建，那么系统怎么知道系统配置的细节呢？答案就是Windows会挂载一种特殊的文件系统（为小文件做了优化）到名字空间。这个文件系统称作注册表（registry）。注册表被组织成了不同的卷，称作储巢（hive）。每个储巢保存在一个单独文件中（在启动卷的目录C:\Windows\ system32\ config\下）。当Windows系统启动时，一个叫做SYSTEM的特殊储巢被装入了内存，这是由同样的装载内核和其他启动文件（例如位于启动盘的驱动程序）的程序来完成。

Windows在系统储巢里面保存了大量的重要信息，包括驱动程序去驱使什么设备工作，什么软件进行初始化，以及什么变量来控制操作系统的操作等。这些信息甚至被启动程序自己用来决定哪些驱动程序是用于启动的驱动，哪些必须立即需要启动。这些驱动包括操作系统自身来识别文件系统和磁盘驱动的程序。

其他配置储巢用在系统启动后，描述系统安装的软件的信息，特别是用户和用户态下安装在系统上的COM（Component Object-Model)。本地用户的登录信息保存在SAM（安全访问管理器）中。网络用