从网络到图形的设备驱动程序组成。HAL对其他组件隐藏了硬件上的某些差别。内核层管理CPU以支持多线程和同步，执行体实现大多数的内核态服务。

执行体基于内核态的对象，这些对象代表了关键的执行体数据结构，包括进程、线程、内存区、驱动程序、设备以及同步对象等。用户进程通过调用系统服务来创建对象并获得句柄的引用以用于后续对执行体组件的调用。操作系统也创建一些内部对象。对象管理器维护着一个名字空间，对象可以插入该名字空间以备后续的查询。

Windows系统中最重要的对象是进程、线程和内存区。进程拥有虚拟地址空间并且是资源的容器。线程是执行的单元并被内核层使用优先级算法调度执行，该优先级算法使优先级最高的就绪线程总在运行，并且如有必要可抢占低优先级线程。内存区表示可以映射到进程地址空间的像文件这样的内存对象。EXE和DLL等程序映像用内存区来表示，就像共享内存一样。

Windows支持按需分页虚拟内存。分页算法基于工作集的概念。系统维护着几种类型的页面列表来优化内存的使用。这些页面列表是通过调整工作集来填充的，调整过程使用了复杂的规则试图重用在长时间内没有被引用的物理页面。缓存管理器管理内核中的虚拟地址并用它将文件映射到内存，这提高了许多应用程序的I/O性能，因为读操作不用访问磁盘就可被满足。

设备驱动程序遵循Windows驱动程序模型，并执行输入/输出。每个驱动程序开始先初始化一个驱动程序对象，该对象含有可被系统调用以操控设备的过程的地址。实际的设备用设备对象来代表，设备对象可以根据系统的配置描述来创建，或者由即插即用管理器按照它在枚举系统总线时所发现的设备创建。设备组织成一个栈，I/O请求包沿着栈向下传递并被每个设备的驱动程序处理。I/O具有内在的异步性，驱动程序程序通常将请求排队以便后续处理然后返回到调用者。文件系统卷作为I/O系统中的设备实现。

NTFS文件系统基于一个主文件表，每个文件或者目录在表中有一条记录。NTFS文件系统的所有元数据本身是NTFS文件的一部分。每个文件含有多个属性，这些属性或存储在MFT记录中或者不在其中（存储在MFT外部的块中）。除此之外，NTFS还支持Unicode、压缩、日志和加密等。

最后，Windows Vista拥有一个基于访问控制列表和完整性级别的成熟的安全系统。每个进程带有一个令牌，此令牌表示了用户的标识和进程所具有的特殊权限。每个对象有一个与其相关联的安全描述符。安全描述符指向一个自主访问控制列表，该列表中包含允许或者拒绝个体或者组访问的访问控制入口项，Windows在最近的发行版本中增加了大量的安全特性，包括用BitLocker来加密整个卷，采用地址空间随机化，不可执行的堆栈以及其他措施使得缓冲区溢出攻击更加困难。

习题

1. HAL可以跟踪从1601年开始的所有时间。举一个例子，说明这项功能的用途。
2. 在11.3.2节，我们介绍了在多线程应用程序中一个线程关闭了句柄而另一个线程仍然在使用它们所造成的问题。解决此问题的一种可能性是插入序列域。请问该方法是如何起作用的？需要对系统做哪些修改？
3. Win32 系统没有信号功能。如果要引入此功能，我们可以将信号设置为进程所有，线程所有，两者都有或者两者都没有。试着提出一项建议，并解释为什么。
4. 另一种使用DLL的方式是静态地将每个程序链接到它实际调用到那些库函数，既不多也不少。在客户端机器或者服务器机器上引入此方法，哪个更合理？
5. 在Windows中线程拥有独立的用户态栈和内核态栈的原因是哪些？
6. TLB对性能有重大的影响。为了提高TLB的有效性，Windows使用了大小为4MB的页，这是什么？
7. 在一个执行体对象上可定义的不同操作的数量有没有限制？如果有，这个限制从何而来？如果没有，请说明为什么。
8. Win32 API的调用WaitForMultipleObjects以一组同步对象的句柄为参数，使得线程被这组同步对象阻塞。一旦它们中的任何一个收到信号，调用者线程就会被释放。这组同步对象是否可以包含两个信号灯、一个互斥体和一个临界区？理由是什么？提示：这不是一个恶作剧的问题，但确实有必要认真考虑一番。
9. 给出三个可能会终止线程的原因。
10. 如11.4节所述，有一个特殊的句柄表用于为进程和线程分配ID。句柄表的算法通常是分配第一个可用的句柄（按照后进先出的顺序维护空