

UNIX	Win32	说 明
fork	CreateProcess	创建一个新进程
waitpid	WaitForSingleObject	可等待一个进程退出
execve	(none)	CreateProcess = fork + execve
exit	ExitProcess	终止执行
open	CreateFile	创建一个文件或打开一个已有的文件
close	CloseHandle	关闭一个文件
read	ReadFile	从一个文件读数据
write	WriteFile	把数据写入一个文件
lseek	SetFilePointer	移动文件指针
stat	GetFileAttributesEx	取得文件的属性
mkdir	CreateDirectory	创建一个新目录
rmdir	RemoveDirectory	删除一个空目录
link	(none)	Win32不支持link
unlink	DeleteFile	毁掉一个已有的文件
mount	(none)	Win32不支持mount
umount	(none)	Win32不支持umount
chdir	SetCurrentDirectory	改变当前工作目录
chmod	(none)	Win32不支持安全性(但NT支持)
kill	(none)	Win32不支持信号
time	GetLocalTime	获得当前时间

图1-23 与图1-18中UNIX调用大致对应的Win32 API调用

下面简要地说明一下图1-23中表格的内容。CreateProcess为创建一个新进程，它把UNIX中的fork和execve结合起来。它有许多参数用来指定新创建进程的性质。Windows中没有类似UNIX中的进程层次，所以不存在父进程和子进程的概念。在进程创建之后，创建者和被创建者是平等的。WaitForSingleObject用于等待一个事件，等待的事件可以是多种可能的事件。如果有参数指定了某个进程，那么调用者等待所指定的进程退出，这通过使用ExitProcess完成。

接着的六个调用进行文件操作，在功能上和它们的UNIX对应调用类似，尽管在参数和细节上它们都是不同的。和在UNIX中一样，文件可被打开、关闭和写入。SetFilePointer以及GetFileAttributesEx调用设置文件的位置并取得文件的一些属性。

Windows中有目录，目录可以分别用CreateDirectory以及RemoveDirectory API调用创建和删去。也有对当前目录的标记，这可以通过SetCurrentDirectory来设置。使用GetLocalTime可获得当前时间。

Win32接口中没有文件的链接、文件系统的安装、安全属性或信号，所以对应于UNIX中的这些调用就不存在了。当然，Win32中也有大量的在UNIX中不存在的其他调用，特别是管理GUI的种种调用。不过在Windows Vista中有了精心设计的安全系统，而且也支持文件的链接。

也许有必要对Win32做一个最后的说明。Win32并不是非常统一的或有一致的接口。其主要原因是由于Win32需要与早期的在Windows 3.x中使用的16位接口向后兼容。

1.7 操作系统结构

我们已经分析了操作系统的外部（如，程序员接口），现在是分析其内部的时候了。在下面的小节中，为了对各种可能的方式有所了解，我们将考察已经尝试过的六种不同的结构设计。这样做并没有穷尽各种结构方式，但是至少给出了在实践中已经试验过的一些设计思想。这六种设计是，单体系统、层次系统、微内核、客户机—服务器系统、虚拟机和exokernels等。

1.7.1 单体系统

到目前为止，在多数常见的组织形式的处理方式中，全部操作系统在内核态中以单一程序的方式运行。整个操作系统以过程集合的方式编写，链接成一个大型可执行二进制程序。使用这种技术，系统中