**BOOL CreateProcess**

(

**LPCTSTR**lpApplicationName,

**LPTSTR**lpCommandLine,

**LPSECURITY\_ATTRIBUTES**lpProcessAttributes。

**LPSECURITY\_ATTRIBUTES**lpThreadAttributes,

**BOOL**bInheritHandles,

**DWORD**dwCreationFlags,

**LPVOID**lpEnvironment,

**LPCTSTR**lpCurrentDirectory,

**LPSTARTUPINFO**lpStartupInfo,

**[LPPROCESS\_INFORMATION](http://baike.haosou.com/doc/5111965-5340758.html" \t "_blank)**lpProcessInformation

);

参数

**lpApplicationName**

指向一个NULL结尾的、用来指定可执行模块的字符串。

这个字符串可以是可执行模块的[绝对路径](http://baike.haosou.com/doc/2383240-2519937.html" \t "_blank)，也可以是[相对路径](http://baike.haosou.com/doc/5717230-5929956.html" \t "_blank)，在后一种情况下，函数使用当前[驱动器](http://baike.haosou.com/doc/6309459-6523046.html" \t "_blank)和目录建立可执行模块的路径。

这个参数可以被设为NULL，在这种情况下，可执行模块的名字必须处于 lpCommandLine 参数最前面并由[空格符](http://baike.haosou.com/doc/1865296-1972893.html" \t "_blank)与后面的字符分开。

**lpCommandLine**

指向一个以NULL结尾的字符串，该字符串指定要执行的命令行。

这个参数可以为空，那么函数将使用**lpApplicationName**参数指定的字符串当做要运行的程序的命令行。

如果lpApplicationName和lpCommandLine参数都不为空，那么lpApplicationName参数指定将要被运行的模块，lpCommandLine参数指定将被运行的模块的命令行。新运行的进程可以使用[GetCommandLine](http://baike.haosou.com/doc/828190-875871.html" \t "_blank)函数获得整个命令行。C语言程序可以使用argc和argv参数。

**lpProcessAttributes**

指向一个SECURITY\_ATTRIBUTES结构体，这个[结构体](http://baike.haosou.com/doc/6031580-6244581.html" \t "_blank)决定是否返回的句柄可以被子进程继承。如果lpProcessAttributes参数为空(NULL)，那么句柄不能被继承。

**在Windows NT中**:[SECURITY\_ATTRIBUTES](http://baike.haosou.com/doc/1090506-1153943.html" \t "_blank)结构的**lpSecurityDescriptor**成员指定了新进程的[安全描述符](http://baike.haosou.com/doc/7701624-7975719.html" \t "_blank)，如果参数为空，新进程使用默认的安全描述符。

**lpThreadAttributes**

同lpProcessAttribute,不过这个参数决定的是线程是否被继承.通常置为NULL.

**bInheritHandles**

指示新进程是否从调用进程处继承了句柄。

如果参数的值为真，调用进程中的每一个可继承的打开句柄都将被子进程继承。被继承的句柄与原进程拥有完全相同的值和访问权限。

**dwCreationFlags**

指定附加的、用来控制优先类和进程的创建的标志。以下的创建标志可以以除下面列出的方式外的任何方式组合后指定。

⑴值:**CREATE\_DEFAULT\_ERROR\_MODE**

含义:新的进程不继承调用进程的错误模式。**CreateProcess**函数赋予新进程当前的默认错误模式作为替代。[应用程序](http://baike.haosou.com/doc/3417785-3597266.html" \t "_blank)可以调用SetErrorMode函数设置当前的默认错误模式。

这个标志对于那些运行在没有硬件错误环境下的多线程[外壳程序](http://baike.haosou.com/doc/7538542-7812635.html" \t "_blank)是十分有用的。

对于**CreateProcess**函数，默认的行为是为新进程继承调用者的错误模式。设置这个标志以改变默认的处理方式。

⑵值:**CREATE\_NEW\_CONSOLE**

含义:新的进程将使用一个新的控制台，而不是继承[父进程](http://baike.haosou.com/doc/3943920-4138829.html" \t "_blank)的控制台。这个标志不能与DETACHED\_PROCESS标志一起使用。

⑶值:**CREATE\_NEW\_PROCESS\_GROUP**

含义:新进程将使一个[进程树](http://baike.haosou.com/doc/2940946-3102990.html" \t "_blank)的根进程。[进程树](http://baike.haosou.com/doc/2940946-3102990.html" \t "_blank)中的全部进程都是根进程的子进程。新[进程树](http://baike.haosou.com/doc/2940946-3102990.html" \t "_blank)的[用户标识符](http://baike.haosou.com/doc/3330643-3507705.html" \t "_blank)与这个进程的标识符是相同的，由lpProcessInformation参数返回。[进程树](http://baike.haosou.com/doc/2940946-3102990.html" \t "_blank)经常使用GenerateConsoleCtrlEvent函数允许发送CTRL+C或CTRL+BREAK信号到一组控制台进程。

⑷值:**CREATE\_SEPARATE\_WOW\_VDM**

如果被设置，新进程将会在一个私有的虚拟DOS机(VDM)中运行。另外，默认情况下所有的16位Windows[应用程序](http://baike.haosou.com/doc/3417785-3597266.html" \t "_blank)都会在同一个共享的VDM中以[线程](http://baike.haosou.com/doc/115404-121766.html" \t "_blank)的方式运行。单独运行一个16位程序的优点是一个[应用程序](http://baike.haosou.com/doc/3417785-3597266.html" \t "_blank)的崩溃只会结束这一个VDM的运行;其他那些在不同VDM中运行的程序会继续正常的运行。同样的，在不同VDM中运行的16位Windows[应用程序](http://baike.haosou.com/doc/3417785-3597266.html" \t "_blank)拥有不同的输入队列，这意味着如果一个程序暂时失去响应，在独立的VDM中的[应用程序](http://baike.haosou.com/doc/3417785-3597266.html" \t "_blank)能够继续获得输入。

⑸值:**CREATE\_SHARED\_WOW\_VDM**

如果WIN.INI中的Windows段的DefaultSeparateVDM选项被设置为真，这个标识使得CreateProcess函数越过这个选项并在共享的虚拟DOS机中运行新进程。

⑹值:**CREATE\_SUSPENDED**

含义:新进程的[主线程](http://baike.haosou.com/doc/6649542-6863360.html" \t "_blank)会以暂停的状态被创建，直到调用[ResumeThread](http://baike.haosou.com/doc/6952955-7175359.html" \t "_blank)函数被调用时才运行。

⑺值:**CREATE\_UNICODE\_ENVIRONMENT**

含义:如果被设置，由lpEnvironment参数指定的环境块使用Unicode[字符](http://baike.haosou.com/doc/2529629-2672477.html" \t "_blank)，如果为空，环境块使用ANSI[字符](http://baike.haosou.com/doc/2529629-2672477.html" \t "_blank)。

⑻值:**DEBUG\_PROCESS**

含义:如果这个标志被设置，调用进程将被当做一个[调试程序](http://baike.haosou.com/doc/5934-6078.html" \t "_blank)，并且新进程会被当做被调试的进程。系统把被[调试程序](http://baike.haosou.com/doc/5934-6078.html" \t "_blank)发生的所有调试事件通知给调试器。

如果你使用这个标志创建进程，只有调用进程(调用CreateProcess函数的进程)可以调用WaitForDebugEvent函数。

⑼值:**DEBUG\_ONLY\_THIS\_PROCESS**

含义:如果此标志没有被设置且调用进程正在被调试，新进程将成为调试调用进程的调试器的另一个调试对象。如果调用进程没有被调试，有关调试的行为就不会产生。

⑽值:**DETACHED\_PROCESS**

含义:对于控制台进程，新进程没有访问[父进程](http://baike.haosou.com/doc/3943920-4138829.html" \t "_blank)控制台的权限。新进程可以通过[AllocConsole](http://baike.haosou.com/doc/1882592-1991734.html" \t "_blank)函数自己创建一个新的控制台。这个标志不可以与CREATE\_NEW\_CONSOLE标志一起使用。

〔11〕值:**CREATE\_NO\_WINDOW**

含义:系统不为新进程创建CUI窗口，使用该标志可以创建不含窗口的CUI程序。

**dwCreationFlags**参数

还用来控制新进程的优先类，优先类用来决定此进程的[线程调度](http://baike.haosou.com/doc/614900-650990.html" \t "_blank)的优先级。如果下面的优先级类标志都没有被指定，那么默认的优先类是**NORMAL\_PRIORITY\_CLASS**，除非被创建的进程是**IDLE\_PRIORITY\_CLASS**。在这种情况下子进程的默认优先类是**IDLE\_PRIORITY\_CLASS**。

可以选择下面的标志中的一个:

优先级:**HIGH\_PRIORITY\_CLASS**

含义:指示这个进程将执行时间临界的任务，所以它必须被立即运行以保证正确。这个优先级的程序优先于正常优先级或空闲优先级的程序。一个例子是Windows任务列表，为了保证当用户调用时可以立刻响应，放弃了对系统负荷的考虑。确保在使用高优先级时应该足够谨慎，因为一个高优先级的CPU关联[应用程序](http://baike.haosou.com/doc/3417785-3597266.html" \t "_blank)可以占用几乎全部的CPU可用时间。

优先级:**IDLE\_PRIORITY\_CLASS**

含义:指示这个进程的线程只有在系统空闲时才会运行并且可以被任何高优先级的任务打断。例如[屏幕保护程序](http://baike.haosou.com/doc/6170943-6384181.html" \t "_blank)。空闲优先级会被子进程继承。

优先级:**NORMAL\_PRIORITY\_CLASS**

含义:指示这个进程没有特殊的任务调度要求。

优先级:**REALTIME\_PRIORITY\_CLASS**

含义:指示这个进程拥有可用的最高优先级。一个拥有实时优先级的进程的[线程](http://baike.haosou.com/doc/115404-121766.html" \t "_blank)可以打断所有其他进程线程的执行，包括正在执行重要任务的系统进程。例如，一个执行时间稍长一点的实时进程可能导致[磁盘缓存](http://baike.haosou.com/doc/5926231-6139154.html" \t "_blank)不足或鼠标反映迟钝。

**lpEnvironment**

指向一个新进程的环境块。如果此参数为空，新进程使用调用进程的环境。

一个环境块存在于一个由以NULL结尾的字符串组成的块中，这个块也是以NULL结尾的。每个字符串都是name=value的形式。

因为相等标志被当做[分隔符](http://baike.haosou.com/doc/6595574-6809356.html" \t "_blank)，所以它不能被[环境变量](http://baike.haosou.com/doc/5580891-5793769.html" \t "_blank)当做变量名。

与其使用[应用程序](http://baike.haosou.com/doc/3417785-3597266.html" \t "_blank)提供的环境块，不如直接把这个参数设为空，系统驱动器上的[当前目录](http://baike.haosou.com/doc/7554977-7829070.html" \t "_blank)信息不会被自动传递给新创建的进程。对于这个情况的探讨和如何处理，请参见注释一节。

环境块可以包含Unicode或ANSI字符。如果lpEnvironment指向的环境块包含Unicode[字符](http://baike.haosou.com/doc/2529629-2672477.html" \t "_blank)，那么dwCreationFlags字段的CREATE\_UNICODE\_ENⅥRONMENT标志将被设置。如果块包含ANSI[字符](http://baike.haosou.com/doc/2529629-2672477.html" \t "_blank)，该标志将被清空。

请注意一个ANSI环境块是由两个零字节结束的:一个是字符串的结尾，另一个用来结束这个快。一个Unicode环境块是由四个零字节结束的:两个代表字符串结束，另两个用来结束块。

**lpCurrentDirectory**

指向一个以NULL结尾的字符串，这个字符串用来指定子进程的[工作路径](http://baike.haosou.com/doc/288213-305114.html" \t "_blank)。这个字符串必须是一个包含驱动器名的[绝对路径](http://baike.haosou.com/doc/2383240-2519937.html" \t "_blank)。如果这个参数为空，新进程将使用与调用进程相同的驱动器和目录。这个选项是一个需要启动[应用程序](http://baike.haosou.com/doc/3417785-3597266.html" \t "_blank)并指定它们的驱动器和工作目录的[外壳程序](http://baike.haosou.com/doc/7538542-7812635.html" \t "_blank)的主要条件。

**lpStartupInfo**

指向一个用于决定新进程的主[窗体](http://baike.haosou.com/doc/2067607-2187320.html" \t "_blank)如何显示的[STARTUPINFO](http://baike.haosou.com/doc/2003283-2119863.html" \t "_blank)[结构体](http://baike.haosou.com/doc/6031580-6244581.html" \t "_blank)。

**lpProcessInformation**

指向一个用来接收新进程的识别信息的[PROCESS\_INFORMATION](http://baike.haosou.com/doc/2218886-2347843.html" \t "_blank)[结构体](http://baike.haosou.com/doc/6031580-6244581.html" \t "_blank)。

返回值

如果函数执行成功，返回非零值。

如果函数执行失败，返回零，可以使用GetLastError函数获得错误的附加信息。

例：

//创建cmd进程，并指明输入输出为管道返回的读写句柄

STARTUPINFO si;

PROCESS\_INFORMATION pi;

//取得决定新进程的主窗体如何显示的STARTUPINFO基本结构内容体并填充，使cmd的输入输出和管道关联

GetStartupInfo(&si);

si.dwFlags = STARTF\_USESHOWWINDOW | STARTF\_USESTDHANDLES;

si.hStdInput = &guandaocaozuo.hReadPipe;

si.hStdError = &guandaocaozuo.hWritePipe;

si.hStdOutput = &guandaocaozuo.hWritePipe;

si.wShowWindow = SW\_HIDE;

//得到系统路径，即cmd路径

long cmdlen;

TCHAR cmdline[256] = { 0 };

cmdlen = GetSystemDirectory(cmdline, sizeof(cmdline));

wcscat(cmdline,TEXT("\\cmd.exe"));

//创建进程

if (CreateProcess(cmdline, NULL, NULL, NULL, TRUE, NULL, NULL, NULL, &si, &pi) == 0)

{

printf("CreateProcess Error \n");

return 1;

}