**获取进程的当前目录**

GetCurrentDirectory函数//获取进程当前目录

**DWORD WINAPI GetCurrentDirectory(**

**\_\_in DWORD nBufferLength,**

**\_\_out LPTSTR lpBuffer**

**);**

nBufferLength：lpBuffer指针指向内存块的大小（单位TCHAR）；

lpBuffer：接收当前路径的内存块。

SetCurrentDirectory函数//设置进程当前目录

\_chdir：设置当前目录

获取制定文件的当前路径

GetFullPathName(L"E:",MAX\_PATH,str,NULL);

::::这个函数为什么能获得当前目录？因为这个函数，就是用当前目录来补齐你的文件名，不管你这个文件真正在那里！！！没啥用！！！

transacted operation：事件操作，指这个操作不能被打断。

**DWORD WINAPI GetFullPathName(**

**\_\_in LPCTSTR lpFileName,**

**\_\_in DWORD nBufferLength,**

**\_\_out LPTSTR lpBuffer,**

**\_\_out LPTSTR \*lpFilePart**

**);**

lpFileName：文件名

nBufferLength：获取全路径的内存大小（TCHAR）

lpBuffer：内存指针

lpFilePart：文件名最后一个元素，在lpBuffer中的位置。

**注意：这个函数，只是将当前路径，粘贴到你给的文件上，其他什么也没有做**。

**系统版本：**

DWORD GetVersion();

HIBYTE(GetVersion());

LOBYTE(GetVersion());

BOOL WINAPI GetVersionEx(

\_\_inout LPOSVERSIONINFO lpVersionInfo

);

// 获取当前系统版本信息

LpVersionInfo：这个参数执行[**OSVERSIONINFO**](http://127.0.0.1:47873/help/1-7900/ms.help?method=page&id=A173DF17-DAD2-4330-AA66-4FF789FD7CC2&product=VS&productVersion=100&topicVersion=85&locale=ZH-CN&topicLocale=EN-US)或者**[OSVERSIONINFOEX](http://127.0.0.1:47873/help/1-7900/ms.help?method=page&id=4AB07A72-404D-459B-B061-B3B06B5DB37E&product=VS&productVersion=100&topicVersion=85&locale=ZH-CN&topicLocale=EN-US" \t ")**这个结构体。

OSVERSIONIINFOEX结构体是OSVERSIONINFO结构体的扩展，所以这个函数可以传递这两个参数的任意一个，前提条件是，必须设置结构体的大小，让GetVersionEx函数能够知道你传递的是那个结构体。

typedef struct \_OSVERSIONINFOEX {

DWORD dwOSVersionInfoSize;

DWORD dwMajorVersion;

DWORD dwMinorVersion;

DWORD dwBuildNumber;

DWORD dwPlatformId;

TCHAR szCSDVersion[128];

WORD wServicePackMajor;

WORD wServicePackMinor;

WORD wSuiteMask;

BYTE wProductType;

BYTE wReserved;

} OSVERSIONINFOEX, \*POSVERSIONINFOEX, \*LPOSVERSIONINFOEX;

typedef struct \_OSVERSIONINFO {

DWORD dwOSVersionInfoSize;

DWORD dwMajorVersion;

DWORD dwMinorVersion;

DWORD dwBuildNumber;

DWORD dwPlatformId;

TCHAR szCSDVersion[128];

} OSVERSIONINFO;

BOOL WINAPI VerifyVersionInfo(

\_\_in LPOSVERSIONINFOEX lpVersionInfo,

\_\_in DWORD dwTypeMask,

\_\_in DWORDLONG dwlConditionMask

);

确认当前系统版本是否满足版本需求，其中lpVersionInfo是需要的系统版本，

dwTypeMask测试项，也就是说，lpVersioinInfo结构体中，那些参数被设置；

dwlConditionMask：比较条件

ULONGLONG VER\_SET\_CONDITION(

ULONGLONG dwlConditionMask,

DWORD dwTypeBitMask,

BYTE dwConditionMask

);

创建比较测试条件的宏。