**环境变量：**

获取环境变量的一种方法：

#include <windows.h>

#include<stdio.h>

int main()

{

char\*\* p=\_environ;

while(\*p)

{

printf("%s\n",\*p);

p++;

}

getchar();

return(1);

}

#define UNICODE

#define \_UNICODE

#include <windows.h>

#include<tchar.h>

int \_tmain()

{

TCHAR\*\* p=\_wenviron;

while(\*p)

{

\_tprintf(L"%s\n",\*p);

p++;

}

\_gettchar();

return(1);

}

环境变量被保存在那里？？？

放在注册表中：

HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\Environment

这个注册表键下放的是系统环境变量。

HKEY\_CURRENT\_USER\Environment

这个注册表键下放的是当前登陆用户自己的环境变量。

注意，在你的程序中，如果你改变了环境变量，那么最好你发送一个消息：WM\_SETTINGCHANGE!!!

注意：发这个消息时，不要使用某一个窗口的句柄，应该使用：HWND\_BROADCAST作为窗口句柄！！使用SendMessage或者SendMessageTimeout函数来发送这个消息，lParam参数应该设置成 ”Environment”。

单个环境变量的获取和修改：

GetEnvironmentVariable;

SetEnvironmentVariable;//只能改变当前进程的环境变量！对系统内的环境变量没有影响。

对于%%之间的字符串，将被系统扩展！

ExpandEnvironmentStrings函数。

#include<Windows.h>

#include<tchar.h>

int \_tmain()

{

DWORD length=ExpandEnvironmentStrings(L"%SystemRoot%",NULL,0);

TCHAR\* str=new TCHAR[length+1];

ExpandEnvironmentStrings(L"%SystemRoot%",str,length);

\_tprintf(L"%s\n",str);

\_gettchar();

return 0;

}

最后一种获得环境变量的方法是：GetEnvironmentString()函数。

我们可以控制我们进程中的线程，只运行在某个CPU上，这个属性叫：Affinity

**进程的错误模式：**

一个函数：SetErrorMode(UINT uMode);

|  |  |
| --- | --- |
| **Value** | **Meaning** |
| 0 | 系统缺省模式，将所有错误都显示在错误窗口上。 |
| SEM\_FAILCRITICALERRORS  0x0001 | 系统不在显示关键错误句柄信息给用户，而是将错误信息返回给进程 |
| SEM\_NOALIGNMENTFAULTEXCEPT  0x0004 | 系统自动解决内存对齐缺陷，这种错误将不再报告给应用程序。进程的子进程也会具有这种特性。但有些CPU不支持这种方式。  这个标志一旦被设置，将不能再被取消。 |
| SEM\_NOGPFAULTERRORBOX  0x0002 | 系统不在显示general-protection-default消息框。这个框只在调试模式下被显示。 |
| SEM\_NOOPENFILEERRORBOX  0x8000 | 程序找不到文件时，不再显示消息框，这个错误被返回给进程。 |