**第四章：进程**

**进程**

1. 什么是进程，系统如何创建一个内核对象来管理进程。
2. 如何通过进程内核对象来操作一个进程。
3. 进程的属性，以及查询和修改进程属性的函数。
4. 进程的结束。

一个进程，就是一个正在运行的程序！一个程序，可以产生多个进程。

* 一个内核对象，被系统用来管理这个进程，这个内核对象中，还包含了进程的一些策略信息。
* 一个地址空间，这个地址空间中包含了可执行代码，动态链接库模块代码，数据，程序动态内存分配获取的内存，也在这个内存地址空间中。

进程是死的，进程必须有一个线程！！

primary thread！！如果进程内所有线程退出，那么进程自动销毁！

调用ExitProcess函数，也可以结束进程。

第一个windows程序CUI和GUI。

我们所编写的windows程序中，真正第一个被执行的函数是：

图像界面的入口函数是：\_tWinMain，控制台程序的入口函数是\_tmain

WinMainCRTStartup//GUI\_ASCII字符类型的C运行时库中的入口函数

wWinMainCRTStartup// GUI\_UNICODE字符类型的C运行时库中的入口函数

mainCRTStartup// CUI\_ASCII字符类型的C运行时库中的入口函数

wmainCRTStartup// CUI\_UNICODE字符类型的C运行时库中的入口函数

这些C运行时库函数，主要完成以下任务：

1. 获取进程命令行指针；
2. 获取进程环境变量指针；
3. 初始化C/C++运行时库的全局变量，如果你包含了头Stdlib.h，那么你就可以访问这些变量！//errno这就是一个全局变量。\_enrion这是环境变量指针
4. 初始化malloc函数的内存堆；
5. 为C++全局类，调用构造函数。

malloc 函数，不要轻易使用？因为这个函数一般来说，最终会调用windows API函数，我们直接调用virtualAlloc windowsAPI函数，效率会高！

Kenerl 调用XXXCRTStartup函数；

XXXCRTStartup函数调用main函数；

main函数退出；

exit函数被调用（真正调用的是crt0dat.c中的）

static void \_\_cdecl doexit (

int code,

int quick,

int retcaller

)

这个函数主要做以下事情：

1. 调用\_onexit函数注册的所有函数；
2. 调用C++销毁函数销毁所有的全局和静态类对象；
3. 如果\_CRTDBG\_LEAK\_CHECK\_DF标志被设置，那么调用CrtDumpMemoryLeaks函数，列出泄露内存。
4. 调用ExitProcess函数，系统杀死当前进程！

获取当前操作系统版本信息：GetVersion函数

**DWORD WINAPI GetVersion(void);**

**返回操作系统的版本号。**

|  |  |  |  |
| --- | --- | --- | --- |
| Windows 8 | 6.2 |  |  |
| Windows Server 2012 | 6.2 |  |  |
| Windows 7 | 6.1 |  |  |
| Windows Server 2008 R2 | 6.1 |  |  |
| Windows Server 2008 | 6.0 |  |  |
| Windows Vista | 6.0 |  |  |
| Windows Server 2003 R2 | 5.2 |  |  |
| Windows Home Server | 5.2 |  |  |
| Windows Server 2003 | 5.2 |  |  |
| Windows XP Professional x64 Edition | 5.2 |  |  |
| Windows XP | 5.1 |  |  |
| Windows 2000 | 5.0 |  |  |