**句柄复制**

进程间如何共享内核对象？句柄复制

**BOOL WINAPI DuplicateHandle(**

**\_\_in HANDLE hSourceProcessHandle,**

**\_\_in HANDLE hSourceHandle,**

**\_\_in HANDLE hTargetProcessHandle,**

**\_\_out LPHANDLE lpTargetHandle,**

**\_\_in DWORD dwDesiredAccess,**

**\_\_in BOOL bInheritHandle,**

**\_\_in DWORD dwOptions**

**);**

hSourceProcessHandle：源进程的进程句柄；

hSourceHandle：源内核对象句柄；

hTargetProcessHandle：目标进程的进程句柄；

lpTargetHandle：指向接收目标内核对象句柄内存的指针；

dwDesiredAccess：访问权限；

bInheritHandle：是否可被继承；

dwOptions：

|  |  |
| --- | --- |
| **Value** | **Meaning** |
| DUPLICATE\_CLOSE\_SOURCE  0x00000001 | 复制完毕后，将源句柄关闭，无论发生什么错误，源句柄都将被关闭。 |
| DUPLICATE\_SAME\_ACCESS  0x00000002 | 忽视dwDesiredAccess参数。复制的句柄具有和源句柄同样的权限。 |

**伪句柄的含义！**

我们通过GetCurrentProcess函数，获得的句柄都是伪句柄！伪句柄的值为-1，其实是一个固定的值！如果你在使用句柄的地方，直接写上-1，那么就代表本进程句柄，但不建议这么做，也就是说，GetCurrentProcess函数，在目前，总是返回-1

有两种方法获得真实的进程句柄。

一种是使用OpenProcess函数，另外一种是使用DuplicateHandle函数！

**HANDLE WINAPI OpenProcess(**

**\_\_in DWORD dwDesiredAccess,**

**\_\_in BOOL bInheritHandle,**

**\_\_in DWORD dwProcessId**

**);**

获得进程内核句柄；

dwDesiredAccess：希望获得的进程内核对象的句柄，具有什么样的权限。

bInheritHandle：这个句柄是否可以被继承。

dwProcessID：要获得进程内核对象句柄的进程ID。

如果成功，返回句柄，否则返回NULL。

**DWORD WINAPI GetProcessId(**

**\_\_in HANDLE Process**

**);**

通过进程句柄，获得进程Id；

Process：进程句柄。

句柄是0x30，在本进程内复制后，句柄变为0x34，是新增加了一个句柄！此时，要关闭复制后的句柄，也必须关闭源句柄！！