利用进程ID获取主线程ID

转载

weixin_34218890 于 2019-01-08 00:46:23 发布 1735 收藏

利用进程ID获取主线程ID,仅适用于单线程。多线程应区分哪个是主线程,区分方法待验证

（1）好像可以用StartTime最早的，不过通过线程执行时间不一定可靠，要是在最开始就CreateThread了，线程的执行时间会相同。可以通过回溯栈上的值来判断哪个线程是主线程，主线程的栈多少有些不同。最明显就是主线程栈上的PE入口点 信息，没有这个的就是子线程。

（2）CsrProcessLink中取CsrProcessInfo->ClientId.UniqueThread即可，绝对可靠。

```
#include <iostream.h>
#include <windows.h>
#include <tlhelp32.h>
void main()
{
  DWORD dwProcessID, dwThreadID;
  while(1)
  {
   dwThreadID = 0;
   cout<< "请输入进程pID:";
   cin >>dwProcessID;
   THREADENTRY32 te32 = {sizeof(te32)};
   HANDLE hThreadSnap = CreateToolhelp32Snapshot(TH32CS_SNAPTHREAD,0);
   if( Thread32First( hThreadSnap, &te32) )
   {
    do{
      if( dwProcessID == te32.th32OwnerProcessID )
      {
       dwThreadID = te32.th32ThreadID;
       break;
      }
    }while( Thread32Next( hThreadSnap, &te32) );
   }
   if (dwThreadID != 0)
    cout<<"主线程ID："<<dwThreadID<<endl;
   else
    cout<<"没找到"<<endl;
  }
}
```

汇编代码

```
local @stProcess:PROCESSENTRY32        ;每一个进程的信息
local @hSnapShot                ;快照句柄
```

```
DWORD dwProcessID = xxxxx, dwThreadID = 0;
THREADENTRY32 te32 = {sizeof(te32)};
HANDLE hThreadSnap = CreateToolhelp32Snapshot(TH32CS_SNAPTHREAD,0);
if( Thread32First( hThreadSnap, &te32) )
{
do{
if( dwProcessID == te32.th32OwnerProcessID )
{
dwThreadID = te32.th32ThreadID;
break;
}
}while( Thread32Next( hThreadSnap, &te32) );
}
```