Enumerating threads in a process

devblogs.microsoft.com/oldnewthing/20060223-14

February 23, 2006

Raymond Chen

The tool helper library is sort of the black sheep of Win32. It grew out of the 16-bit TOOLHELP library, which provided services for system debugging tools to do things like take stack traces and enumerate all the memory in the system. The original incarnation of Win32 didn't incorporate it; it wasn't until Windows 95 that a 32-bit version of the tool helper library sort of got bolted onto the side of Win32.

Disowned or not, the functions are still there, so let's give them a spin.

```
#include <stdio.h>
#include <windows.h>
#include <tlhelp32.h>
int __cdecl main(int argc, char **argv)
{
 HANDLE h = CreateToolhelp32Snapshot(TH32CS_SNAPTHREAD, 0);
 if (h != INVALID_HANDLE_VALUE) {
  THREADENTRY32 te;
  te.dwSize = sizeof(te);
  if (Thread32First(h, &te)) {
   do {
     if (te.dwSize >= FIELD_OFFSET(THREADENTRY32, th32OwnerProcessID) +
                      sizeof(te.th32OwnerProcessID)) {
       printf("Process 0x%04x Thread 0x%04x\n",
             te.th32OwnerProcessID, te.th32ThreadID);
     }
   te.dwSize = sizeof(te);
   } while (Thread32Next(h, &te));
  }
  CloseHandle(h);
 }
 return 0;
}
```

Running this program prints a list of all the threads in the system (or at least all the ones you have access to). This is particularly straightforward, the only subtlety being the strange check that the size returned by the `Thread32First` function is large enough to emcompass the

`th32OwnerProcessID` field that we need. This complexity is necessary due to the somewhat unorthodox way that the `Thread32First` and `Thread32Next` functions check structure sizes.

That's what happens when you're the black sheep of the Win32 API.

Raymond Chen

Follow