Stupid memory-mapping tricks

devblogs.microsoft.com/oldnewthing/20031007-00

October 7, 2003

Raymond Chen

Shared memory is not just for sharing memory with other processes. It also lets you share memory with yourself in sneaky ways.

For example, this sample program (all error checking and cleanup deleted for expository purposes) shows how you can map the same shared memory into two locations simultaneously. Since they are the same memory, modifications to one address are reflected at the other.

```
#include <windows.h>
#include <stdio.h>
void __cdecl main(int argc, char **argv)
{
    HANDLE hfm = CreateFileMapping(INVALID_HANDLE_VALUE, NULL,
                    PAGE_READWRITE, 0, sizeof(DWORD), NULL);
    LPDWORD pdw1 = (LPDWORD)MapViewOfFile(hfm, FILE_MAP_WRITE,
                                          0, 0, sizeof(DWORD));
    LPDWORD pdw2 = (LPDWORD)MapViewOfFile(hfm, FILE_MAP_WRITE,
                                          0, 0, sizeof(DWORD));
    printf("Mapped to %x and %x\n", pdw1, pdw2);
    printf("*pdw1 = %d, *pdw2 = %d\n", *pdw1, *pdw2);
    /* Now watch this */
    *pdw1 = 42;
    printf("*pdw1 = %d, *pdw2 = %d\n", *pdw1, *pdw2);
}
```

This program prints

```
Mapped to 280000 and 290000
*pdw1 = 0, *pdw2 = 0
*pdw1 = 42, *pdw2 = 42
```

(Missing asterisks added, 8am – thanks to commenter Tom for pointing this out.)

The addresses may vary from run to run, but observe that the memory did get mapped to two different addresses, and changing one value to 42 magically changed the other.

This is a nifty consequence of the way shared memory mapping works. I stumbled across it while investigating how I could copy large amounts of memory without actually copying it. The solution: Create a shared memory block, map it at one location, write to it, then unmap it from the old location and map it into the new location. Presto: The memory instantly “moved” to the new location. This a major win if the memory block is large, since you didn’t have to allocate a second block, copy it, then free the old block – the memory block doesn’t even get paged in.

It turns out I never actually got around to using this trick, but it was a fun thing to discover anyway.

Raymond Chen

Follow