You have to free memory with the same allocator that allocated it: Logical consequences

devblogs.microsoft.com/oldnewthing/20060907-04

September 7, 2006

Raymond Chen

Everybody should know by now that you have to free memory using the same allocator that you used to allocate the memory. If you allocate with `new[]` then you have to free with `delete[]`; if you allocate with `LocalAlloc` then you have to free with `LocalFree`. Once you've internalized this rule, you can use it to draw other logical conclusions. Consider:

> When I call the `PropertySheet` function, who is responsible for freeing the memory that was allocated for the `phpage` field of the `PROPSHEETHEADER` structure?

Well, there are two candidates for this responsibility, either the `PropertySheet` function or the caller of the `PropertySheet` function. If the `PropertySheet` function was responsible for freeing the memory, it would have to make sure to use the same allocator that was used to allocate the `phpage`. But there is no requirement that that memory use any particular allocator. (In fact, a significant portion of the time, the memory is allocated from the stack, in which case there is no explicit deallocation step.) The `PropertySheet` function would now be required to be psychic and somehow "know" how the memory should be freed (or whether it should be freed at all). Since psychic powers have yet to be perfected in software, this pretty much closes off this line of reasoning.

The only remaining candidate is the caller of the `PropertySheet` function. Since that's the code that allocated the memory, it's the one who knows how to free it.

Raymond Chen

Follow