#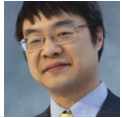 Why does my program run really slow or even crash (or stop crashing, or crash differently) if running under a debugger?

January 3, 2013

Raymond Chen

More than once, a customer has noticed that running the exact same program under the debugger rather than standalone causes it to change behavior. And not just in the "oh, the timing of various operations changed to hit different race conditions" but in much more fundamental ways like "my program runs really slow" or "my program crashes in a totally different location" or (even more frustrating) "my bug goes away".

What's going on? I'm not even switching between the retail and debug versions of my program, so I'm not a victim of changing program semantics in the debug build.

When a program is running under the debugger, some parts of the system behave differently. One example is that the `CloseHandle` function raises an exception (I believe it's `STATUS_INVALID_HANDLE` but don't quote me) if you ask it to close a handle that isn't open. But the one that catches most people is that when run under the debugger, an alternate heap is used. This alternate heap has a different memory layout, and it does extra work when allocating and freeing memory to help try to catch common heap errors, like filling newly-allocated memory with a known sentinel value.

But this change in behavior can make your debugging harder or impossible.

So much for people's suggestions to switch to a stricter implementation of the Windows API when a debugger is attached.

On Windows XP and higher, you can disable the debug heap even when debugging. If you are using a `dbgeng`-based debugger like `ntsd` or `WinDbg`, you can pass the `-hd` command line switch. If you are using Visual Studio, you can set the `_NO_DEBUG_HEAP` environment variable to 1.

If you are debugging on a version of Windows prior to Windows XP, you can start the process without a debugger, then connect a debugger to the live process. The decision to use the debug heap is made at process startup, so connecting the debugger afterwards ensures that

the retail heap is chosen.