If you don’t blow up a debug session every so often, you’re not debugging hard enough

devblogs.microsoft.com/oldnewthing/20161111-00

November 11, 2016

Raymond Chen

I was helping a customer live-debug an assertion failure in an automated test running in a lab. I messed up an attempt to unwind the stack to restart a call and ended up corrupting the process state. Undaunted, I realized that the issue at hand was that one specific API call was failing, so I said to myself, “That’s okay if I can’t restart the call. I can just simulate the call!” so I patched registers and manually pushed data onto the stack and all that stuff.

And then I stepped through the code, and it crashed because I messed up one detail: When virtually pushing the return address on the stack, I had a mental lapse and subtracted 4 from the stack pointer even though this was a 64-bit machine and I should have subtracted 8. Due to the stack misalignment, the code eventually crashed on a `movaps` instruction several stack frames deep into the function.

I blew up the debug session not once but twice.

If this happens to you, don’t beat yourself up. If you don’t blow up a debug session every so often, then you’re not debugging hard enough.

(That punch line is a ripoff of something I heard the Car Talk guys say: “If you don’t stall a manual transmission every so often, then you’re not driving it right.”)

Raymond Chen

Follow