Debugging tip: Use .frame /r to recover nonvolatile registers from the stack frame

devblogs.microsoft.com/oldnewthing/20170706-00

July 6, 2017

Raymond Chen

A customer was debugging a crash dump (using a debugger that is based on the Windows debugger engine) and observed that when they used the `k` command, the "Args to Child" values didn't make any sense, and they suspect that the values are not accurate. They were wondering if there were any tips for recovering the actual parameters.

The customer is correct. The values reported by "Args to Child" come from the stack, but the values on the stack might not be the parameters.

On x86, the values on the stack probably were the parameters at one point, but the compiler is free to reuse that space to hold other values. And on x64, the values were most likely never on the stack to begin with. In debug mode, the compiler will probably spill the register parameters into their corresponding home space, but in release mode, the compiler will most likely not bother. In all cases, the values reported as "Args to Child" are unreliable.

But all hope is not lost.

The old-fashioned way of recovering the values is to disassemble the function and follow the parameter to see where the compiler decided to save it. If the compiler saved it into memory on the stack frame, then that's great: You can inspect the value on the stack. But if the compiler enregistered the value, then you have to follow the register into the called function to see where it saved that register. And often that register got saved on the stack, which means you have to do some math to find out where it ended up.

But the debugger can help out. If the debugger has access to a PDB file, it can use that information to virtually pop the registers from the stack back into registers. And on x64, all functions must provide register unwind information, so this information is available in general.

If you use the `.frame` command to switch context to a particular stack frame, you can use the `/r` option to ask the debugger to show the register state that results from virtual register unwinding.

Note that when you issue this command, the debugger shows all registers, even the volatile registers, so don't be faked out.

Raymond Chen

Follow