The x86 architecture is the weirdo

devblogs.microsoft.com/oldnewthing/20040914-00

September 14, 2004

Raymond Chen

The x86 architecture does things that almost no other modern architecture does, but due to its overwhelming popularity, people think that the x86 way is the normal way and that everybody else is weird.

Let's get one thing straight: The x86 architecture is the weirdo.

The x86 has a small number (8) of general-purpose registers; the other modern processors have far more. (PPC, MIPS, and Alpha each have 32; ia64 has 128.)

The x86 uses the stack to pass function parameters; the others use registers.

The x86 forgives access to unaligned data, silently fixing up the misalignment. The others raise a misalignment exception, which can optionally be emulated by the supervisor at an amazingly huge performance penalty.

The x86 has variable-sized instructions. The others use fixed-sized instructions. (PPC, MIPS, and Alpha each have fixed-sized 32-bit instructions; ia64 has fixed-sized 41-bit instructions. Yes, 41-bit instructions.)

The x86 has a strict memory model, where external memory access matches the order in which memory accesses are issued by the code stream. The others have weak memory models, requiring explicit memory barriers to ensure that issues to the bus are made (and completed) in a specific order.

The x86 supports single-instruction atomic load-modify-store operations. None of the others do.

The x86 passes function return addresses on the stack. The others use a link register.

Bear this in mind when you write what you think is portable code. Like many things, the culture you grow up with is the one that feels "normal" to you, even if, in the grand scheme of things, it is one of the more bizarre ones out there.

Note from the future: At the time of writing, the term "x86" was used exclusively to refer to what later became known as "x86-32". The name "x86-64" wouldn't be invented until 2006.

Raymond Chen

Follow