
Subject: Re: -32768

Posted by [Yngvar Larsen](#) on Fri, 21 Mar 2014 18:02:08 GMT

[View Forum Message](#) <> [Reply to Message](#)

On Friday, 21 March 2014 00:44:17 UTC+1, Chris Torrence wrote:

> On Thursday, March 20, 2014 3:30:49 PM UTC-6, Yngvar Larsen wrote:

>

>> IDL> help, -9223372036854775808LL

>> <Expression> LONG64 = -9223372036854775808

>> IDL> help, -9223372036854775809LL

>> <Expression> LONG64 = 9223372036854775807

>>

>> Thus, we have the same bug for 32-bit integers, but not for 64-bit!!

>

> Not to start a flame war, but I would argue that 64-bit has the bug, and that 16-bit and 32-bit are behaving correctly. It seems bad that the 64-bit case is quietly returning what could arguably be the wrong value.

For literals, I agree that this might be regarded as dubious. But in general, the fact that N-bit integer arithmetic is silently wrapped modulo 2^N should be regarded as semantically well defined. If not, I believe a serious performance hit would result.

> As an aside, in the original post, the user stated that they want to "set an integer variable..." But in IDL, there is no such thing as an "integer variable" because IDL is loosely typed. There are just variables that happen to have integer type, but only after they've been created.

>

>

> Jim P. is correct that the minus sign is really an operator. So it really is a runtime error to write "a = 32768s", regardless of whether you are then going to take the negative of it.

Fair enough. In that case, what is missing is a way to enter a literal negative number. "-376768" is a valid 16-bit signed integer after all! Not really a problem, since negation of a single number hardly is a big performance hit... Also, the OPs problem hasn't hit me at all during the 15+ years I've been using IDL. And there are at least 3 simple (bit silly) workarounds, already mentioned in this thread.

--

Yngvar
