Subject: Integer overflow not reported
Posted by Markus Schmassmann on Tue, 13 Jun 2017 14:58:21 GMT
View Forum Message <> Reply to Message

Hello,

Why are integer overflows not reported, although according the
documentation of CHECK_MATH [1] they should be detected?

Anything I can do to activate such warnings?
(other than changing to Windows)

An example is below.

-- Markus

```
IDL> rsun=696000000
IDL> help, rsun, rsun^2, long64(rsun)^2, long64(rsun)^2 mod 2ll^31
RSUN            LONG      =    696000000
<Expression>    LONG      =    327417856
<Expression>    LONG64    =     484416000000000000
<Expression>    LONG64    =             327417856
IDL> help, 1/0
<Expression>    INT       =        1
% Program caused arithmetic error: Integer divide by 0
% Detected at $MAIN$
IDL> !version
{
    "ARCH": "x86_64",
    "OS": "linux",
    "OS_FAMILY": "unix",
    "OS_NAME": "linux",
    "RELEASE": "8.5",
    "BUILD_DATE": "Jul  7 2015",
    "MEMORY_BITS": 64,
    "FILE_OFFSET_BITS": 64
}
```

[1] http://www.harrisgeospatial.com/docs/CHECK_MATH.html

Subject: Re: Integer overflow not reported
Posted by wlandsman on Tue, 13 Jun 2017 17:17:42 GMT
View Forum Message <> Reply to Message

On Tuesday, June 13, 2017 at 10:58:25 AM UTC-4, Markus Schmassmann wrote:
> Hello,
>

> Why are integer overflows not reported, although according the
> documentation of CHECK_MATH [1] they should be detected?

The documentation on math errors says that integer overflow is not detected on all hardware.

I agree that nondetection of the overflow of long (32bit) integers is more insidious than nondetection of the overflow of short integers -- since we are now trained to always default to long integers. ---Wayne

http://www.harrisgeospatial.com/docs/Math_Errors.html

The detection of math errors, such as division by zero, overflow, and attempting to take the logarithm of a negative number, is hardware and operating system dependent. Some systems trap more errors than other systems. On systems that implement the IEEE floating-point standard, IDL substitutes the special floating-point values NaN and Infinity when it detects a floating point math error. (See Special Floating-Point Values.) Integer overflow and underflow is not detected. Integer divide by zero is detected on all platforms.