
Subject: Re: print, long(1.0e10)
Posted by [thompson](#) on Wed, 04 Aug 1999 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Gary Fu <gfu@seadas.gsfc.nasa.gov> writes:

> Hi,

> I got different results from the following example on SGI IRIX and PC
> Linux:
> a = 1.0e10
> b = long(a)
> print, b ; 2147483647 for IRIX, -2147483648 for Linux

> It looks like that IRIX and Linux handle it differently when assigning a
> floating value greater (or less) than the boundary of a long integer to
> a long variable. Should I always check the range of 'a' before
> assigning it to a 'long' variable to make the result consistent on IRIX
> and Linux ?

I get yet a different result with IDL/v5.2 on Digital Unix.

```
IDL> a = 1.0e10
IDL> b = long(a)
% Program caused arithmetic error: Floating illegal operand
% Detected at $MAIN$
IDL> print, b
  1410065408
```

This is a completely sensible result, since it's equal to $(A \bmod 2^{32})$,
i.e. the 32 lowest most bits of A expressed as an integer.

Since different machines seem to act in different ways, it's best to catch the
error, either by testing the value beforehand, or by using something like
CHECK_MATH.

William Thompson
