Subject: JULDAY 5.4 not same as 5.3?
Posted by Don Woodraska on Thu, 01 Mar 2001 23:00:14 GMT

View Forum Message <> Reply to Message

```
<!doctype html public "-//w3c//dtd html 4.0 transitional//en">
<html>

<pre>Has anyone else noticed a bug in JULDAY that appeared in 5.4?</pre>

<pre>I tried this in IDL 5.3:</pre>

<pre>IDL> help,umm,udd,uyear,uhr,umin,usec
 UMM&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nb sp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nb sp;
LONG&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;  =&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp
;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp; 2
 UDD&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nb sp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nb sp;
LONG&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;  =&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp
;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp; 16
 UYEAR&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;& nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
LONG&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;  =&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp
;&nbsp;&nbsp; 2001
 UHR&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nb sp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nb sp;
ULONG&nbsp;&nbsp;&nbsp;&nbsp;  =&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp
;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp; 0
 UMIN&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&n bsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
ULONG&nbsp;&nbsp;&nbsp;&nbsp;  =&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp
;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp; 0
 USEC&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&n bsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
ULONG&nbsp;&nbsp;&nbsp;&nbsp;  =&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp
;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp; 0
IDL> in_jday = julday( umm, udd, uyear, uhr, umin, usec )
IDL> gps0_jday = julday(1,6,1980,0,0,0)
</pre>

<pre>IDL> help,in_jday,gps0_jday
 IN_JDAY&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&am p;nbsp;&nbsp;&nbsp;
DOUBLE&nbsp;&nbsp;&nbsp;  =&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp ;&nbsp; 2451956.5
 GPS0_JDAY&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;& amp;nbsp; DOUBLE&nbsp;&nbsp;&nbsp;
=&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp ;&nbsp; 2444244.5
</pre>

<pre></pre>

<pre>I did the same thing in IDL 5.4:</pre>

<pre>IDL> in_jday = julday( umm, udd, uyear, uhr, umin, usec )
IDL> gps0_jday = julday(1,6,1980,0,0,0)
IDL> help,in_jday,gps0_jday
```

 IN_JDAY&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&am p;nbsp;&nbsp;&nbsp;
DOUBLE&nbsp;&nbsp;&nbsp; =&nbsp;&nbsp;&nbsp; 1.8140893e+08
 GPS0_JDAY&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;& amp;nbsp; DOUBLE&nbsp;&nbsp;&nbsp;
=&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp ;&nbsp; 2444244.5
</pre>

<pre>Perhaps someone with more expertise/patience/time could figure out</pre>

<pre>what is really going on here. Anyone interested?</pre>

<pre>Cheers,</pre>

<pre>Don</pre>

<pre>--&nbsp;
Donald Woodraska
Laboratory for Atmospheric and Space Physics, University of Colorado
Campus Box 590, 1234 Innovation Drive, Boulder, CO 80303</pre>
&nbsp;</html>

Subject: Re: JULDAY 5.4 not same as 5.3?
Posted by Mark Hadfield on Sun, 04 Mar 2001 20:14:39 GMT

View Forum Message <> Reply to Message

"Don Woodraska" <don.woodraska@lasp.colorado.edu> wrote in message
news:3A9FE8D9.8E9BE7DA@lasp.colorado.edu...
> I've pinpointed the problem.
>
> SUMMARY:
> ***
> The bug appears only with unsigned-long and unsigned-64-bit-long hour
argument
> to JULDAY.
>
> The bug appears on lines 178-179 of JULDAY.PRO. Here it is:
> ...

Well done, Dan. To correct my earlier post, this DOES also occur on windows.
In the list of data values in your original post, i.e...

```
>>>  UMM           LONG     =         2
>>>  UDD           LONG     =        16
>>>  UYEAR          LONG     =      2001
>>>  UHR           ULONG    =         0
>>>  UMIN          ULONG    =         0
>>>  USEC          ULONG    =         0
```

...I missed the fact that the hour minute & second values are ULONG and not LONG. (Though I did wonder why you put a "U" prefix on the variable names!)

I wonder how many other routines can be broken if you pass unsigned-integer values to them. And how paranoid does a person writing a routine have to be?

Final comment: Isn't JULDAY's argument order ghastly? Month, day, year, hour, minute, second? I mean, really!

Mark Hadfield
m.hadfield@niwa.cri.nz http://katipo.niwa.cri.nz/~hadfield
National Institute for Water and Atmospheric Research

Subject: Re: JULDAY 5.4 not same as 5.3?
Posted by pit on Mon, 05 Mar 2001 18:03:06 GMT

View Forum Message <> Reply to Message

Don Woodraska <don.woodraska@lasp.colorado.edu> writes:
> I've pinpointed the problem.
>
> SUMMARY:
> ***
> The bug appears only with unsigned-long and unsigned-64-bit-long
> hour argument to JULDAY.

Hm, this is a much more general problem of unsigned data types, I stumbled across it myself just two weeks ago.

I'm still not sure who's to blame, me (i.e. the programmer) or IDL/RSI, but in general it does not make sense to take a negative of a unsigned number, and IMHO there should also occur an automatic type conversion as soon as a minus sign is involved. Currently, you get e.g.

```
IDL> x=5b
IDL> print,-x
 251
```

Any comments on that?

  Peter

--
 ~~~~~~
~~~~~~~~~~~~~~~
Dr. Peter "Pit" Suetterlin   http://www.astro.uu.nl/~suetter
Sterrenkundig Instituut Utrecht

Tel.: +31 (0)30 253 5225   P.Suetterlin@astro.uu.nl

Subject: Re: JULDAY 5.4 not same as 5.3?

Posted by Craig Markwardt on Mon, 05 Mar 2001 22:22:19 GMT

View Forum Message <> Reply to Message

pit@phys.uu.nl (Peter Suetterlin) writes:

...

> I'm still not sure who's to blame, me (i.e. the programmer) or
> IDL/RSI, but in general it does not make sense to take a negative of a
> unsigned number, and IMHO there should also occur an automatic type
> conversion as soon as a minus sign is involved. Currently, you get e.g.
>
> IDL> x=5b
> IDL> print,-x
> 251

We can argue all day about what is correct mathematically.

However I think it is correct from a microprocessor standpoint (ie, that's what the processor does). Further more it satisfies certain identities like:

 (-x) + x EQ 0

And what would you do with a number like -('ffffffff'xul), ie a number that to begin with is too large to fit into a signed type.

Finally, you would have the people (like me) who would gripe about how IDL changed the type of a variable without asking me first!

Craig

--

 -- --------------

Craig B. Markwardt, Ph.D.   EMAIL: craigmnet@cow.physics.wisc.edu

Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response

 -- --------------

Subject: Re: JULDAY 5.4 not same as 5.3?

Posted by Mark Hadfield on Mon, 05 Mar 2001 22:57:25 GMT

View Forum Message <> Reply to Message

"Craig Markwardt" <craigmnet@cow.physics.wisc.edu> wrote in message
news:onzoezonz8.fsf@cow.physics.wisc.edu...
> pit@phys.uu.nl (Peter Suetterlin) writes:
> ...
>> I'm still not sure who's to blame, me (i.e. the programmer) or
>> IDL/RSI, but in general it does not make sense to take a negative of a
>> unsigned number, and IMHO there should also occur an automatic type
>> conversion as soon as a minus sign is involved. Currently, you get e.g.
>>
>> IDL> x=5b
>> IDL> print,-x
>> 251
>
> We can argue all day about what is correct mathematically.
>
> However I think it is correct from a microprocessor standpoint (ie,
> that's what the processor does). Further more it satisfies certain
> identities like:
>
> (-x) + x EQ 0

Actually, under Peter's proposal (-x) + x would equal 0 for unsigned x.

> And what would you do with a number like -('ffffffff'xul), ie a number
> that to begin with is too large to fit into a signed type.

Now, that is a good point.

There are other situations where auto-magical type conversion could save
programmers' skins but is not done by IDL, e.g. 32767S + 1S evaluates
to -32767S, not 32768L. Hands up who hasn't been stung by that one!

On the whole I agree with Craig that type conversion should be done
sparingly by the core language, because it opens up a can of worms. If -5B
is promoted to an unsigned integer, what about (0B-5B)? If the latter is
promoted, then what about (6B-5B)?

Mark Hadfield
m.hadfield@niwa.cri.nz http://katipo.niwa.cri.nz/~hadfield
National Institute for Water and Atmospheric Research

Subject: Re: JULDAY 5.4 not same as 5.3?
Posted by pit on Tue, 06 Mar 2001 17:23:52 GMT
View Forum Message <> Reply to Message

"Mark Hadfield" <m.hadfield@niwa.cri.nz> writes:

>> And what would you do with a number like -('ffffffff'xul), ie a number
>> that to begin with is too large to fit into a signed type.
>
> Now, that is a good point.
>
> There are other situations where auto-magical type conversion could save
> programmers' skins but is not done by IDL, e.g. 32767S + 1S evaluates
> to -32767S, not 32768L. Hands up who hasn't been stung by that one!

That were exactly my thoughts when I was thinking about the -1b
problem lately, and was the reason I didn't ask here at that time.
Just as a similar problem showed up, I decided to shout ;^>

So the essence is boundary checks are (and should stay) a work for the
programmer (which is perfectly OK for me....).

Peter

--
~~~~~~
~~~~~~~~~~~~~~~
Dr. Peter "Pit" Suetterlin http://www.astro.uu.nl/~suetter
Sterrenkundig Instituut Utrecht
Tel.: +31 (0)30 253 5225 P.Suetterlin@astro.uu.nl

Subject: Re: unsigned int woes (was: JULDAY 5.4 not same as 5.3?)
Posted by Martin Schultz on Mon, 19 Mar 2001 14:19:15 GMT

View Forum Message <> Reply to Message

Craig Markwardt wrote:
>
> pit@phys.uu.nl (Peter Suetterlin) writes:
> ...
>> I'm still not sure who's to blame, me (i.e. the programmer) or
>> IDL/RSI, but in general it does not make sense to take a negative of a
>> unsigned number, and IMHO there should also occur an automatic type
>> conversion as soon as a minus sign is involved. Currently, you get e.g.
>>
>> IDL> x=5b
>> IDL> print,-x
>> 251
>
> We can argue all day about what is correct mathematically. [...][/color]

Well, not always. There are definitively cases where IDL produces a mathematically incorrect result. Try this:

```
print, (-1L)*2UL, 2UL*(-1L)
```

Maybe, kids don't learn this in school any more, but there are several textbooks on mathematics that indicate that you should get the same answer in both cases. Well.... the answers are -2 4294967294
;-)

Perhaps it is best not to consider unsigned types as "numbers" in the streetwise sense, i.e. entities that you can use for computations. They are good as counters or addresses, that's it. So, in the original post: what drove you to use unsigned longs as argument for julday in the first place? Question to RSI: how hard would it be to generate an overflow error whenever an unsigned type variable goes below zero? Mathematically, that would probably be the most correct solution.

Cheers,

Martin

PS: Converting the type automatically would in my opinion make things easier and more in line with how IDL handles other data types. It would make things slower but it would take away one severe possible pitfall from the users (not programmers!). Once more, it would be interesting to learn for whom IDL is really made: is it scientists who want a powerful, easy-to-use visualisation software that they can "customize" by grouping statements together as a "code", or is it programmers who are supposedly hired by scientists to develop easy-to-use visualisation software and who could not live without unsigned types, "break" statements, etc. As the answer is probably: both - I think that's exactly where the problem is. There are some IDL routines that obviously cater more to the programmer and others that are simple enough (too simple) so they can be used by scientists. And the two worlds don't always harmonize. ...and then there is history
...

--
 [[[[[[[[[[
[[Dr. Martin Schultz   Max-Planck-Institut fuer Meteorologie   [[
[[Bundesstr. 55, 20146 Hamburg   [[
[[phone: +49 40 41173-308   [[
[[fax: +49 40 41173-298   [[
[[martin.schultz@dkrz.de   [[
 [[[[[[[[[[