Subject: Array with float indices.
Posted by Nikola on Fri, 21 Jun 2013 12:01:20 GMT
View Forum Message <> Reply to Message

I just discovered a bug in the code that I'm writing. It was caused by a nasty feature of IDL that I wasn't aware before. This behaviour - maybe naively - I would call erratic.

Briefly, I use an array of indeces (xind) to extract a subarray (y) from an array (x). Something like:

```
x = FINDGEN(5)/5
xind =[2, 3, 4]
y = x[xind]
```

The indeces xind are computed by a subroutine that gives a FLOAT output. It may happen that the indeces are out of the range of the original array (x), e.g. xind = [3, 4, 5]. I was naively assuming that my code would crash on that. Nevertheless, it turned out that the code can live with that as long as the indeces are submitted as a float array with arbitrary values?!? Using the same example as above:

```
print, x[5]
% Attempt to subscript X with <INT       (       5)> is out of range.
% Execution halted at: $MAIN$

print, x[5.]
% Attempt to subscript X with <FLOAT     (      5.00000)> is out of range.
% Execution halted at: $MAIN$

print, x[[5.]]
    0.800000

print, x[[15.]]
    0.800000
```

Is this actually a feature of IDL? Is it documented somewhere? I found it to be quite dangerous as a potential source of bugs. I expected that

(1) x[any scalar] = x[[any scalar]]
(2) x[float scalar] = x[FLOOR(float scalar) or ROUND(float scalar)]

but not
x[[float]] = x[float < N_ELEMENTS(x)-1]

Note: It's common for 7.1 and 8.2.3.

Subject: Re: Array with float indices.
Posted by David Fanning on Fri, 21 Jun 2013 12:21:20 GMT

Nikola Vitas writes:

> Is this actually a feature of IDL? Is it documented somewhere? I found it to be quite dangerous as a potential source of bugs.

Yes, it is one of the lesser known features, but it has been a feature of IDL from forever. It is certainly documented *somewhere*. I just re-read the "Array - subscripts" section of the documentation, though, and I didn't find it there. So, who knows where it is. I can't seem to find anything in the IDL Help these days. :-(

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thue. ("Perhaps thou speakest truth.")

Subject: Re: Array with float indices.
Posted by Nikola on Fri, 21 Jun 2013 12:28:35 GMT

So untidy... It may refer to both, the documentation and to the feature itself. :(

Btw, I looked through the old help files and still cannot find any description of it.

Subject: Re: Array with float indices.
Posted by Lajos Foldy on Fri, 21 Jun 2013 12:30:38 GMT

On Friday, June 21, 2013 2:21:20 PM UTC+2, David Fanning wrote:
> Nikola Vitas writes:
>
>
>
>> Is this actually a feature of IDL? Is it documented somewhere? I found it to be quite dangerous as a potential source of bugs.
>
>
>
> Yes, it is one of the lesser known features, but it has been a feature

>
> of IDL from forever. It is certainly documented *somewhere*. I just re-
>
> read the "Array - subscripts" section of the documentation, though, and
>
> I didn't find it there. So, who knows where it is. I can't seem to find
>
> anything in the IDL Help these days. :-(
>

Language > Arrays > Using Arrays as Array Subscripts

Array Subscripts and Clipping

If an element of the subscript array is less than or equal to zero, the first element of the subscripted array is selected. If an element of the subscript array is greater than or equal to the last subscript in the subscripted array, the last element is selected.

regards,
Lajos

Subject: Re: Array with float indices.
Posted by Heinz Stege on Fri, 21 Jun 2013 12:37:28 GMT
View Forum Message <> Reply to Message

On Fri, 21 Jun 2013 05:01:20 -0700 (PDT), Nikola Vitas wrote:

> Is this actually a feature of IDL?

Yes. You can disable it with
  COMPILE_OPT STRICTARRSUBS
within your source file and/or your startup file.

> Is it documented somewhere?

Yes. See
http://www.exelisvis.com/docs/Using_Arrays_as_Array_Su.html# arrays_3727706888_753120

Cheers, Heinz

Subject: Re: Array with float indices.
Posted by Nikola on Fri, 21 Jun 2013 12:38:32 GMT
View Forum Message <> Reply to Message

So, it works for integers as well. Indeed:

x[[15]] = 0.80000

Well, I learned something new today. Thank you, guys.