Subject: Negative array indices
Posted by b_gom on Thu, 02 Dec 2010 21:54:41 GMT
View Forum Message <> Reply to Message

I've been stumbling around trying to figure out uses for the negative array indexing in IDL 8. As usual, the documentation doesn't fully describe the logic behind the syntax rules.

One rule that I've discovered is that negative array indices shouldn't be thought of as indices 'rolling over' as you cross zero. For example:
If x=findgen(10), then x[-1] is the same as x[9], and x[-10] is the same as x[0], but x[-11] isn't allowed. I can see why this follows from not allowing positive indices greater than the number of elements in x.

I guess this leads to this behaviour:
IDL> x=findgen(10)
IDL> print,x[-4:0]
% Illegal subscript range: X.

because this gets translated to:
IDL> print,x[6,0]
which isn't allowed.

Now, on to the real question. Why does the following not work:

```
IDL> print,x[1:3]
      1.00000      2.00000      3.00000
IDL> print,x[[1,2,3]]
      1.00000      2.00000      3.00000
IDL> print,x[-3:-1]
      7.00000      8.00000      9.00000
IDL> print,x[[-3,-2,-1]]
     0.000000     0.000000     0.000000
```

Subject: Re: Negative array indices
Posted by David Fanning on Thu, 02 Dec 2010 22:50:40 GMT
View Forum Message <> Reply to Message

b_gom writes:

> I've been stumbling around trying to figure out uses for the negative
> array indexing in IDL 8. As usual, the documentation doesn't fully
> describe the logic behind the syntax rules.

It's always a bad sign when you are stumbling around trying

to figure out how to use a feature you never realized you
needed before. Or are you trying to write a book, too?

Cheers,

David

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

Subject: Re: Negative array indices
Posted by b_gom on Fri, 03 Dec 2010 00:35:56 GMT
View Forum Message <> Reply to Message

On Dec 2, 3:50 pm, David Fanning <n...@dfanning.com> wrote:
> It's always a bad sign when you are stumbling around trying
> to figure out how to use a feature you never realized you
> needed before. Or are you trying to write a book, too?

No, that market has already been cornered.

It was more a case of: 'Ah, this might be a situation where those new negative indices I read about might simplify things', followed quickly by the realisation that I didn't really understand what they were good for.

In general, it would be helpful if the IDL documentation went a bit further than just describing the syntax and calling sequence for the operators, commands and routines, and added a paragraph or two for each saying "And here's what it is\isn't good for:"

Another example is the new list and hash objects. The documentation does a good job describing what their methods do, but doesn't really explain what they are better for than arrays and structures, and what performance trade-offs to be aware of. I guess that is where your new book comes in..

Subject: Re: Negative array indices
Posted by penteado on Fri, 03 Dec 2010 01:28:07 GMT
View Forum Message <> Reply to Message

On Dec 2, 10:35 pm, b_gom <b_...@hotmail.com> wrote:
> Another example is the new list and hash objects. The documentation
> does a good job describing what their methods do, but doesn't really
> explain what they are better for than arrays and structures, and what
> performance trade-offs to be aware of. I guess that is where your new
> book comes in..

I had not even noticed that the documentation did not mention motivation. Maybe all the many requests for lists, hashes and negative indices over the years (mine, for one) left them with the impression that everyone already wanted them. Negative indices are a good syntatic sugar (basically and abbreviation for things like a[n_elements(a)-1] and b[0,(size(b,/dimensions))[1]-1]), and I find them very useful, but they are not even close in importance to lists and hashes. I was leaving IDL for Python because of the lack of lists and hashes.

In nearly every program I write I need a dynamic, possibly heterogeneous container. Without lists, it is too much work to have to predict or alter the dimensions of arrays, or handle (particularly, read) several layers of indirection through nested pointer arrays. And I usually have a bunch of different variables to keep associated, where structures are too awkward to use because their are static. Usually because the code needs to deal with data that can come in varied forms, not known at compile time (being a dynamically typed language is one the main factors in my choice to use IDL).

Subject: Re: Negative array indices
Posted by penteado on Fri, 03 Dec 2010 01:40:58 GMT
View Forum Message <> Reply to Message

On Dec 2, 7:54 pm, b_gom <b_...@hotmail.com> wrote:
> I guess this leads to this behaviour:
> IDL> x=findgen(10)
> IDL> print,x[-4:0]
> % Illegal subscript range: X.
>
> because this gets translated to:
> IDL> print,x[6,0]
> which isn't allowed.

Not allowed just because you are trying to decrease the index with a positive stride (it will never get to 0 starting from 5 and adding increments of 1). This does work:

IDL> x=findgen(10)
IDL> print,x[-4:0:-1]

```
      6.00000      5.00000      4.00000      3.00000      2.00000
1.00000      0.00000
IDL> print,x[6:0:-1]
      6.00000      5.00000      4.00000      3.00000      2.00000
1.00000      0.00000
```
