Subject: array subscripts
Posted by Martin Schultz on Thu, 17 Sep 1998 07:00:00 GMT
View Forum Message <> Reply to Message

Hi all,

now, here is something I have been wondering about several times, so I'd like to ask you for ideas *why* this is so:

It's easy to extract "rectangular domains" from an array like

```
A = FINDGEN(72,46,20)
B=A[10:20,30:32,1:10]
```

However, the following does produce an error message

```
I1 = INDGEN(11)+10
I2 = INDGEN(3)+30
I3 = INDGEN(10)+1
B=A[I1,I2,I3]
% All array subscripts must be same size. Var = A
% Execution halted at:  $MAIN$
```

Since all array subscripts are orthogonal to each other, I can't think of a logical reason why one should not be able to extract subarrays the second way. And, although I am not a programming expert, merely an "Experienced User with Medium Level Programming skills" (EUMLP), I cannot think of any real issues why it would not be possible to program the subarray extraction to allow index array arguments. One has to extract one dimension after the other anyway, so it would not be more penalty than one further index level. Am I missing something here? Would other people also like to see this feature?

Martin.

PS: For those who now think "Hey, IDL can't do that?": one can of course always extract the dimensions one after the other

```
B = A[I1,*,*]
B = B[*,I2,*]
B = B[*,*,I3]
```

PPS: And while we are at it... One of the most useful tips I got from David's book (so far) was the A[n:*] syntax to extract ranges from n to end. I am really grateful for this!

--
---------- -------
Dr. Martin Schultz

Department for Earth&Planetary Sciences, Harvard University
109 Pierce Hall, 29 Oxford St., Cambridge, MA-02138, USA

phone: (617)-496-8318
fax : (617)-495-4551

e-mail: mgs@io.harvard.edu
Internet-homepage: http://www-as.harvard.edu/people/staff/mgs/
---------- -------

Subject: Re: array subscripts
Posted by Benjamin Hornberger on Mon, 14 Nov 2005 15:26:48 GMT
View Forum Message <> Reply to Message

sebinjapan wrote:
> Hi,
>
> Hi,
>
> I found a strange feature when using out of bounds subscripts.
> I am using IDL 6.2 for Mac OSX
> there is a short example:
>
> IDL> a=indgen(4)
> IDL> print, a[-1]
> % Attempt to subscript A with <INT (-1)> is out of range.
> % Execution halted at: $MAIN$
>
> that's ok, but that happens if we use the array [-1] instead of the scalar
> -1 as the subscript address
>
> IDL> print, a[[-1]]
> 0
>
> well... looks like print, a[0 > [-1]]
>
> Same idea for "too large" subscripts
> IDL> print, a[4]
> % Attempt to subscript A with <INT (4)> is out of range.
> % Execution halted at: $MAIN$
> IDL> print, a[[4]]
> 3
>
> IDL> print, a[indgen(8)-2]
> 0 0 0 1 2 3 3 3
>
>

>

From the IDL help:

"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.

Note
Elements of the subscript array that are negative or larger than the highest subscript are clipped to the target array boundaries. Note that a common error is to use a negative scalar subscript (e.g., A[-1]). Using this type of subscript causes an error. Negative array subscripts (e.g., A[[-1]]) do not cause errors.

This clipping of out of bounds elements can be disabled within a routine by using the STRICTARRSUBS option to the COMPILE_OPT statement. (See the documentation for COMPILE_OPT for details.) If STRICTARRSUBS is in force, then array subscripts that refer to out of bounds elements will instead cause IDL to issue an error and stop execution, just as an out-of-range scalar subscript does. "

Just had to read all that myself last night ...

Benjamin

Subject: Re: array subscripts
Posted by sebinjapan on Mon, 14 Nov 2005 15:47:46 GMT
View Forum Message <> Reply to Message

thanks a lot for your quick response.
I will read carefully compile_opt manual!

seb
