Subject: Oh No...
Posted by mfein2 on Wed, 18 Apr 2007 11:52:15 GMT
View Forum Message <> Reply to Message

I've just discovered, after many years of using IDL, that expressions have a value:

```
IDL> print, (x = 5)
    5
```

The possibilities for Obfuscated IDL have now gone to 11 on a scale from 0 to 10.

Subject: Re: Oh No...
Posted by Foldy Lajos on Wed, 18 Apr 2007 18:44:03 GMT
View Forum Message <> Reply to Message

On Wed, 18 Apr 2007, mgalloy@gmail.com wrote:

```
>  On Apr 18, 7:45 am, Paolo Grigis <pgri...@astro.phys.ethz.ch> wrote:
>>  FÖLDY Lajos wrote:
>>
>>>  No, invalid code, with undefined result :-)
>
>  I don't think it invalid or undefined (very confusing, yes). a is
>  created and used to index itself following the "normal" rules of
>  indexing in IDL.
>
>  Remember that it is valid to use an index array that has indices out
>  of range, they are simply brought back into range:
>
>  IDL> b = findgen(10)
>  IDL> b[[-1, 20]] = 1
>  IDL> print, b
>       1.00000      1.00000      2.00000      3.00000      4.00000
>  5.00000      6.00000
>       7.00000      8.00000      1.00000
>
>  You can turn this behavior off with COMPILE_OPT:
>
>  IDL> compile_opt strictarrsubs
>  IDL> a[(a=2*findgen(10))]=1
>  % Array used to subscript array contains out of range subscript: A.
>  % Execution halted at: $MAIN$
>
```

Indexing is OK, but the result of writing to a memory location twice in a single expression is undefined.

This was the original example:

```
IDL> a[(a=2*findgen(10))]=1 ;agreed, this is a bit crazy, but hold on
IDL> print,a
      1.00000      1.00000      1.00000      6.00000      8.00000
      10.0000      1.00000      14.0000      1.00000      1.00000
```

This is the result in IDL 5.3 sun sparc:

```
IDL> a[(a=2*findgen(10))]=1 & print, a
      1.00000      2.00000      1.00000      6.00000      1.00000
      10.0000      1.00000      14.0000      1.00000      1.00000
```

Which is the correct answer? The well-defined version gives the latter:

```
IDL> a=(b=2*findgen(10)) & a[b]=1 & print, a
      1.00000      2.00000      1.00000      6.00000      1.00000
      10.0000      1.00000      14.0000      1.00000      1.00000
```

You can send a bug report to ITTVIS, and see their opinion.

regards,
lajos

Subject: Re: Oh No...
Posted by Carsten Lechte on Wed, 18 Apr 2007 19:21:35 GMT
View Forum Message <> Reply to Message

```
>  IDL> a[(a=2*findgen(10))]=1 & print, a
>       1.00000      2.00000      1.00000      6.00000      1.00000
>       10.0000      1.00000      14.0000      1.00000      1.00000
```

Isn't that what you would expect from in-place modification of an array? At first, a[a[0]]==a[0] is set to 1, then a[a[1]]==a[2] is set to 1, then a[a[2]]==a[1] is set to 1, not a[4], because a[2] has been changed by a previous operation. a[a[3]]==a[6] is set to 1, also a[a[4]]==a[8], then IDL's rules regarding array subscripts kick in and all the other subscripts cause a[9] to be set to 1.

The question is, should this kind of expression be illegal, or cause a warning, or should the language define a less confusing behaviour? Or at least be consistent across versions and platforms?

chl