Subject: Operator precedence flipflop?
Posted by penteado on Fri, 01 Jun 2012 22:23:48 GMT

View Forum Message <> Reply to Message

Recently a user called my attention to some old and well-used software (for the FUSE space telescope) that apparently got broken. As she tracked it down, the problem was occurring when accessing an array pointed to by a structure element. This recreates the problem, IDL 8.0.1 and 8.1:

```
IDL Version 8.1 (linux x86_64 m64). (c) 2011, ITT Visual Information
Solutions
IDL> s={str:[ptr_new(['1','2']),ptr_new(['a','b'])]}
IDL> help,s.str
<Expression>    POINTER   = Array[2]
IDL> print,*(s.str(0))
1 2
IDL> print,*s.str[0]
1 2
```

So far, so good. But then:

```
IDL> print,*s.str(0)
% Expression must be a scalar in this context: S.
% Execution halted at: $MAIN$
```

So it seems that *s.str(0) means (*s.str)(0), instead of *(s.str(0)).

But in IDL 7.1.1 and 8.2 it works:

```
IDL Version 7.1.1 (linux x86_64 m64). (c) 2009, ITT Visual Information
Solutions
IDL> s={str:[ptr_new(['1','2']),ptr_new(['a','b'])]}
IDL> print,*s.str(0)
1 2
```

```
IDL Version 8.2 (linux x86_64 m64). (c) 2012, Exelis Visual
Information Solutions, Inc.
IDL> s={str:[ptr_new(['1','2']),ptr_new(['a','b'])]}
IDL> print,*s.str(0)
1 2
```

I know that none of these happen if one uses [] for array indexing, or parentheses to make the operations clear despite operator precedence. But this change in behaviour during the 8.0 to 8.1 period breaks old programs, as is the case of the mentioned FUSE software (cf_edit.pro). And I do not remember any information on operator precedence changes between versions. It seems it was a bug introduced in 8.0, that was quietly fixed in 8.2.