
Subject: Re: Problem of precedence with pointer and structure
Posted by [Foldy Lajos](#) on Fri, 27 Jan 2006 16:18:31 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi,

pointer dereference works only on scalar pointers (it can not be guaranteed that all the pointers point to heap variables of the same type/structure in a pointer array).

regards,
lajos

On Fri, 27 Jan 2006, L. Testut wrote:

```
> a=replicate({b:ptr_new(findgen(10)),c:ptr_new(findgen(20))}, 3)
>
> Is it possible to do the same replacement as above ???
>
> IDL> (*a[*].(0))=0.5 ;doesn't work
> % Expression must be a scalar in this context: <POINTER  Array[3]>.
> % Execution halted at: $MAIN$          1
>
> Thanks,
> Laurent
>
>
```

Subject: Re: Problem of precedence with pointer and structure
Posted by [David Fanning](#) on Fri, 27 Jan 2006 16:29:50 GMT
[View Forum Message](#) <> [Reply to Message](#)

L. Testut writes:

```
>
> Here is my problem :
> ; If i define a is a 3 elements array of structure with two fields b
> and c
> ; each field is an array
> a=replicate({b:findgen(10),c:findgen(20)},3)
>
> ; and I want to replace the first field b with the constant value 0.5
> ; for the complete array structure, I write:
>
> IDL> a[*].(0)=0.5 ;or a[*].b=0.5 ; and it works
>
> ; Suppose i have now a 3 elements array of structure with two fields b
```

```

> and c
> ; each field is *pointer* to an array
>
> a=replicate({b:ptr_new(findgen(10)),c:ptr_new(findgen(20))}, 3)
>
> Is it possible to do the same replacement as above ???
>
> IDL> (*a[*].(0))=0.5 ;doesn't work
> % Expression must be a scalar in this context: <POINTER  Array[3]>.
> % Execution halted at: $MAIN$          1

```

Yes, I think that is a problem.

```

IDL> Help, a[*].b
<Expression>  POINTER  = Array[3]

```

IDL doesn't allow you to do mathematical operations on pointer arrays:

```

IDL> *(a[*].b) = *(a[*].b) + 3
Expression must be a scalar in this context: <POINTER  Array[3]>.

```

I think a loop is the only way around this.

Cheers,

David

P.S. If I'm not mistaken, I think I recommended that structure solution. :-)

--

David Fanning, Ph.D.
 Fanning Software Consulting, Inc.
 Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Subject: Re: Problem of precedence with pointer and structure
 Posted by [L. Testut](#) on Fri, 27 Jan 2006 20:22:24 GMT
[View Forum Message](#) <> [Reply to Message](#)

```

a=ptrarr(5,/allocate_heap)
IDL> help,a
A          POINTER  = Array[5]
; A is an array of pointer OK

```

```

IDL> help,a[0]
<Expression>  POINTER  = <PtrHeapVar1>

```

; a[0] is a scalar pointer OK

```
IDL> help,/heap
```

Heap Variables:

Pointer: 5

Object : 0

```
<PtrHeapVar1> UNDEFINED = <Undefined>
```

```
<PtrHeapVar2> UNDEFINED = <Undefined>
```

```
<PtrHeapVar3> UNDEFINED = <Undefined>
```

```
<PtrHeapVar4> UNDEFINED = <Undefined>
```

```
<PtrHeapVar5> UNDEFINED = <Undefined>
```

; I've defined 5 different scalar pointers (included in an array)
pointing on 5 different heap variable

; I can affect to each of them the heap variable I want

```
IDL> *a[0]=0.
```

```
IDL> *a[1]='I hate pointers'
```

```
IDL> *a[2]=indgen(10)
```

```
IDL> *a[3]='ok ok we have understoodd !!'
```

```
IDL> *a[4]=4.
```

```
IDL> for i=0,4 do print,*a[i]
```

0.00000

I hate pointers

0 1 2 3 4 5 6 7

8 9

ok ok we understand !!

4.00000

; Now if I want to have an array of structure with a field b which is a
pointer

```
IDL> a=replicate({b:ptr_new(/allocate_heap)},5)
```

```
IDL> help,a
```

```
A STRUCT = -> <Anonymous> Array[5]
```

```
IDL> help,a.b
```

```
<Expression> POINTER = Array[5]
```

; So i think that (a.b) is equivalent to my previous a (e.g an array of
pointer pointing a 5 different heap variable
; but as i *replicate* my structure I have 5 identical pointer variable
pointing to only ONE heap variable !!!

```
IDL> *(a[0].b)=0.0
```

```
IDL> *(a[1].b]='I hate pointers'
```

```
IDL> *(a[2].b)=indgen(10)
```

```
IDL> *(a[3].b]='ok ok we have understood !!'
```

```
IDL> *(a[4].b)=4.
```

```
IDL> for i=0,4 do print,*a[i].b
```

```
4.00000
```

```
4.00000
```

```
4.00000
```

```
4.00000
```

```
4.00000
```

; Then can you confirm me that it is stupid to replicate a structure
with a pointer variable field
; or that I am stupid to think that !

> P.S. If I'm not mistaken, I think I recommended that structure
> solution. :-)

Yes you did :-)

Thanks,
Laurent
