Subject: Re: Calling data from a structure of pointers?
Posted by David Fanning on Thu, 10 Mar 2005 22:46:33 GMT
View Forum Message <> Reply to Message

Jonathan Greenberg writes:

> Hopefully this'll be a quicky -- I have a structure of pointers, so if I,
> for instance, print,structure.pointername I get something like:
>
> <PtrHeapVar5917><PtrHeapVar5922>
>
> I want to ACCESS the data from that structure tho, but:
>
> print,*structure.pointername doesn't work, nor does
> print,structure.*pointername nor:
>
> tempvar=structure.pointername
> print,*tempvar
>
> How do I retrieve the data the pointers are referring to?

If you really have a structure of pointers, then this *must*
work:

```
IDL> Print, *structure.pointername
```

Because structure dereferences have a higher order of precidence
than pointer dereferences.

But from your description, it seems more likely that structure
is a pointer to a structure of pointers. (I can't really tell
what you have there.)

Perhaps something like this would work with whatever it
is you have:

```
*(*structure).pointername
```

You might find this article helpful:

http://www.dfanning.com/misc_tips/precedence.html

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.

Subject: Re: Calling data from a structure of pointers?
Posted by R.G.Stockwell on Thu, 10 Mar 2005 23:09:30 GMT
View Forum Message <> Reply to Message

"Jonathan Greenberg" <usenet@estarcion.net> wrote in message
news:BE560C66.3FE3%usenet@estarcion.net...
> Hopefully this'll be a quicky -- I have a structure of pointers, so if I,
> for instance, print,structure.pointername I get something like:
>
> <PtrHeapVar5917><PtrHeapVar5922>

This is an odd result.
My guess is that structure.pointername is a structure of pointers.
(i.e. you have a stucture where the tag "pointername" is a structure
that has two pointers in it.

Anyways, here is a snippet showing how the structures and pointers work:

```
s = {p1:ptr_new(/alloc),p2:ptr_new(/alloc)}

pointtostruct = ptr_new(/alloc)
*pointtostruct = s
*s.p1 = 15
*s.p2 = 'hello'

print,'s.p1'
print,s.p1

print,'s.p2'
print,s.p2

print,'*s.p1'
print,*s.p1

print,'pointtostruct'
print,pointtostruct

print,'*pointtostruct'
print,*pointtostruct

print,'*(*pointtostruct).p1'
print,*(*pointtostruct).p1
```

end

Subject: Re: Calling data from a structure of pointers?
Posted by Antonio Santiago on Fri, 11 Mar 2005 06:57:35 GMT
View Forum Message <> Reply to Message

Jonathan Greenberg wrote:

> How do I retrieve the data the pointers are referring to?

pStr is pointer to a structure variable. This strucutre has 2 pointer
fields.

```
pStr = PTR_NEW(
 {
  a: PTR_NEW(10),
  b: PTR_NEW(INDGEN(3))
 })
```

pStr - Pointer variable to structure stored in HEAP memory.

```
IDL> help, pStr
PSTR            POINTER   = <PtrHeapVar6>
```

*pStr - The content that pStr points to, that is, the structure
stored in HEAP memory.l

```
IDL> help, *pStr, /STRUCT
** Structure <1f06f8>, 2 tags, length=8, data length=8, refs=1:
   A              POINTER   <PtrHeapVar4>
   B              POINTER   <PtrHeapVar5>
```

(*pStr).a - First "get" the strucutre and then access to the firts
pointer field 'a'.

```
IDL> help, (*pStr).a
<Expression>    POINTER   = <PtrHeapVar4>
```

(*(*pStr).a) - The same as above but obtain the content that 'a'
points (10).

```
IDL> help, (*(*pStr).a)
<PtrHeapVar4>   INT       =       10
```

And the same for 'b' field:

```
IDL> help, (*pStr).b
<Expression>    POINTER   = <PtrHeapVar5>
IDL> help, (*(*pStr).b)
<PtrHeapVar5>   INT       = Array[3]
```

```
IDL> arr = *(*pStr).b
IDL> print, arr
       0       1       2
```

Subject: Re: Calling data from a structure of pointers?
Posted by Jonathan Greenberg on Fri, 11 Mar 2005 17:16:10 GMT
View Forum Message <> Reply to Message

Hi all -- looks like I had been doing this is a bit wrong -- it did end up working right -- *structure.pointer[x] was what I needed (I didn't include the [] it turns out)...

--j

On 3/10/05 10:57 PM, in article d0rfgv$d6v$1@defalla.upc.es, "Antonio Santiago" <d6522117@est.fib.upc.es> wrote:

> Jonathan Greenberg wrote:
>
>> How do I retrieve the data the pointers are referring to?
>
> pStr is pointer to a structure variable. This strucutre has 2 pointer
> fields.
>
> pStr = PTR_NEW(
> {
> a: PTR_NEW(10),
> b: PTR_NEW(INDGEN(3))
> })
>
>
> pStr - Pointer variable to structure stored in HEAP memory.
> IDL> help, pStr
> PSTR POINTER = <PtrHeapVar6>
>
> *pStr - The content that pStr points to, that is, the structure
> stored in HEAP memory.l
> IDL> help, *pStr, /STRUCT
> ** Structure <1f06f8>, 2 tags, length=8, data length=8, refs=1:
> A POINTER <PtrHeapVar4>
> B POINTER <PtrHeapVar5>
>
> (*pStr).a - First "get" the strucutre and then access to the firts
> pointer field 'a'.
> IDL> help, (*pStr).a

> <Expression>   POINTER   = <PtrHeapVar4>
>
> (*(*pStr).a) - The same as above but obtain the content that 'a'
> points (10).
> IDL> help, (*(*pStr).a)
> <PtrHeapVar4>   INT   =   10
>
> And the same for 'b' field:
> IDL> help, (*pStr).b
> <Expression>   POINTER   = <PtrHeapVar5>
> IDL> help, (*(*pStr).b)
> <PtrHeapVar5>   INT   = Array[3]
>
>
> IDL> arr = *(*pStr).b
> IDL> print, arr
>   0   1   2
>
>
>
