Subject: Re: help with structure
Posted by David Fanning on Wed, 31 Mar 2010 21:40:52 GMT
View Forum Message <> Reply to Message

Sumit writes:

> I want to create a structure with 2 fields. The fields need to be of
> variable length. I need to have 'n' such structures,where 'n' is
> scalar number. To address the issue of variable length field, I create
> pointers for each field(pointing to variable temp for initialization)
> as shown below:
> mystruct={l1:ptr_new(temp), l2:ptr_new(temp)}
> I don't know how to create copies of this structure. Replicate
> doesn't work as I guess it creates shallow copy.

What do you mean it "doesn't work"?

```
IDL> mystruct={l1:ptr_new(temp), l2:ptr_new(temp)}
IDL> a= replicate(mystruct, 100)
IDL> help, a
A               STRUCT    = -> <Anonymous> Array[100]
IDL> a[50].l2 = Ptr_New(findgen(11))
```

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: help with structure
Posted by penteado on Wed, 31 Mar 2010 21:49:13 GMT
View Forum Message <> Reply to Message

On Mar 31, 6:40 pm, David Fanning <n...@dfanning.com> wrote:
> Sumit writes:
>> I want to create a structure with 2 fields. The fields need to be of
>> variable length. I need to have 'n' such structures,where 'n' is
>> scalar number. To address the issue of variable length field, I create
>> pointers for each field(pointing to variable temp for initialization)
>> as shown below:
>> mystruct={l1:ptr_new(temp), l2:ptr_new(temp)}

>> I don't know how to create copies of this structure. Replicate
>> doesn't work as I guess it creates shallow copy.
>
> What do you mean it "doesn't work"?
>
> IDL> mystruct={l1:ptr_new(temp), l2:ptr_new(temp)}
> IDL> a= replicate(mystruct, 100)
> IDL> help, a
> A STRUCT = -> <Anonymous> Array[100]
> IDL> a[50].l2 = Ptr_New(findgen(11))

I suppose he means that by doing that, after the replicate(), all elements of a point to the same two heap variables. There is no way around it, he needs to loop on the elements to set the pointer on each one to point to something new.

Subject: Re: help with structure
Posted by Gray on Wed, 31 Mar 2010 22:12:12 GMT
View Forum Message <> Reply to Message

On Mar 31, 5:49 pm, pp <pp.pente...@gmail.com> wrote:
> On Mar 31, 6:40 pm, David Fanning <n...@dfanning.com> wrote:
>
>
>
>
>
>> Sumit writes:
>>> I want to create a structure with 2 fields. The fields need to be of
>>> variable length. I need to have 'n' such structures,where 'n' is
>>> scalar number. To address the issue of variable length field, I create
>>> pointers for each field(pointing to variable temp for initialization)
>>> as shown below:
>>> mystruct={l1:ptr_new(temp), l2:ptr_new(temp)}
>>> I don't know how to create copies of this structure. Replicate
>>> doesn't work as I guess it creates shallow copy.
>
>> What do you mean it "doesn't work"?
>
>> IDL> mystruct={l1:ptr_new(temp), l2:ptr_new(temp)}
>> IDL> a= replicate(mystruct, 100)
>> IDL> help, a
>> A STRUCT = -> <Anonymous> Array[100]
>> IDL> a[50].l2 = Ptr_New(findgen(11))
>
> I suppose he means that by doing that, after the replicate(), all
> elements of a point to the same two heap variables. There is no way

> around it, he needs to loop on the elements to set the pointer on each
> one to point to something new.

Or, he can just create the struct array without allocating heap
variables, then set that field of the array to a ptrarr, and allocate
the heaps then.

```
IDL> mystruct={11:ptr_new(),12:ptr_new()}
IDL> a=replicate(mystruct,100)
IDL> a.11 = ptrarr(100,/allocate_heap)
IDL> a.12 = ptrarr(100,/allocate_heap)
```

Since he's initializing with temporary variables anyway (at least
that's how I read the "temp" in the original post), it doesn't matter
that the heaps aren't initialized doing it this way.

Subject: Re: help with structure
Posted by David Fanning on Wed, 31 Mar 2010 22:37:03 GMT
View Forum Message <> Reply to Message

pp writes:

> I suppose he means that by doing that, after the replicate(), all
> elements of a point to the same two heap variables.

I guess I don't follow you. Isn't this the meaning
of "replicate"?

> There is no way
> around it, he needs to loop on the elements to set the pointer on each
> one to point to something new.

Well, yes, but what else would he be doing?

Perhaps you (and he) mean to replicate a valid pointer
that points to nothing yet:

```
IDL>    struct ={field:Ptr_New(/ALLOCATE_HEAP)}
IDL>    a = Replicate(struct, 100)
IDL>    Print, Ptr_Valid(a[50].field)
  1
IDL>    *a[50].field = 5
```

Just confused, I guess. :-(

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: help with structure
Posted by David Fanning on Wed, 31 Mar 2010 22:42:15 GMT
View Forum Message <> Reply to Message

David Fanning writes:

>
Oh, wait, maybe this is what you mean:

IDL> print, *a[25].field
5

Yes, I can see how that could be a problem.

Cheers,

David

> pp writes:
>
>> I suppose he means that by doing that, after the replicate(), all
>> elements of a point to the same two heap variables.
>
> I guess I don't follow you. Isn't this the meaning
> of "replicate"?
>
>> There is no way
>> around it, he needs to loop on the elements to set the pointer on each
>> one to point to something new.
>
> Well, yes, but what else would he be doing?
>
> Perhaps you (and he) mean to replicate a valid pointer
> that points to nothing yet:
>
> IDL> struct ={field:Ptr_New(/ALLOCATE_HEAP)}
> IDL> a = Replicate(struct, 100)
> IDL> Print, Ptr_Valid(a[50].field)

> 1
> IDL> *a[50].field = 5
>
> Just confused, I guess. :-(
>
> Cheers,
>
> David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: help with structure
Posted by penteado on Wed, 31 Mar 2010 22:47:27 GMT
View Forum Message <> Reply to Message

On Mar 31, 7:37 pm, David Fanning <n...@dfanning.com> wrote:
> pp writes:
>> I suppose he means that by doing that, after the replicate(), all
>> elements of a point to the same two heap variables.
>
> I guess I don't follow you. Isn't this the meaning
> of "replicate"?

Yes, it is. But I think the question was how to make an array where
each element pointed to a different heap variable. So that when he
assigns the non-temp thing that will go into the pointer target, it
will be a different heap variable for each element of the array. And a
better answer to this problem actually is what Gray wrote above.

Subject: Re: help with structure
Posted by David Fanning on Wed, 31 Mar 2010 23:13:24 GMT
View Forum Message <> Reply to Message

pp writes:

> Yes, it is. But I think the question was how to make an array where
> each element pointed to a different heap variable. So that when he
> assigns the non-temp thing that will go into the pointer target, it
> will be a different heap variable for each element of the array. And a

> better answer to this problem actually is what Gray wrote above.

Ah, I see what you mean now. Yes, that could probably use
an article. Thanks for the clarification.

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: help with structure
Posted by penteado on Thu, 01 Apr 2010 01:50:52 GMT
View Forum Message <> Reply to Message

On Mar 31, 8:13 pm, David Fanning <n...@dfanning.com> wrote:
> pp writes:
>> Yes, it is. But I think the question was how to make an array where
>> each element pointed to a different heap variable. So that when he
>> assigns the non-temp thing that will go into the pointer target, it
>> will be a different heap variable for each element of the array. And a
>> better answer to this problem actually is what Gray wrote above.
>
> Ah, I see what you mean now. Yes, that could probably use
> an article. Thanks for the clarification.

There was a similar thread a few months ago:

http://groups.google.com/group/comp.lang.idl-pvwave/browse_t hread/thread/b8b10aaae8d514dc

---