Subject: Re: referencing structure arrays
Posted by nivedita.raghunath on Thu, 10 Sep 2009 16:12:53 GMT
View Forum Message <> Reply to Message

Sorry, looks like what I typed looks gibberish in places. I hope this
one is more readable:

Hi All,

I have a problem referencing structure arrays and assigning values to
them. So here is a snippet of my code:

junk1=update_state(junk) ;creates a structure called junk1
statearr=replicate(junk1,3) ;creates an array of structures

infoptr=ptr_new({statearr:statearr}) ;need this to pass variables
between programs

align_images,state=state1 ;some program that aligns 2 images and
stores the alignment results in state1

;assign values from the state1 structure to infoptr.statearr[0]
structure
 (*infoptr).statearr[0].resvol_file_ptr=state1.resvol_file_pt r
(*infoptr).statearr[0].air_file_ptr=state1.air_file_ptr

Now, here is the problem. With the above code, it turns out that...

(*infoptr).statearr[0].resvol_file_ptr, (*infoptr).statearr
[1].resvol_file_ptr, (*infoptr).statearr[2].resvol_file_ptr
all have the same values. Similarly with air_file_ptr. Even though I
haven't assigned anything to the statearr 1 and 2 yet.

Is there a specific way to reference these structure arrays? Any help
is greatly appreciated!

Thanks
Niv

Subject: Re: referencing structure arrays
Posted by penteado on Thu, 10 Sep 2009 16:36:21 GMT
View Forum Message <> Reply to Message

On Sep 10, 1:12 pm, niv <nivedita.raghun...@gmail.com> wrote:
> Sorry, looks like what I typed looks gibberish in places. I hope this
> one is more readable:

>
> Hi All,
>
> I have a problem referencing structure arrays and assigning values to
> them. So here is a snippet of my code:
>
> junk1=update_state(junk) ;creates a structure called junk1
> statearr=replicate(junk1,3) ;creates an array of structures
>
> infoptr=ptr_new({statearr:statearr}) ;need this to pass variables
> between programs
>
> align_images,state=state1 ;some program that aligns 2 images and
> stores the alignment results in state1
>
> ;assign values from the state1 structure to infoptr.statearr[0]
> structure
> (*infoptr).statearr[0].resvol_file_ptr=state1.resvol_file_pt r
> (*infoptr).statearr[0].air_file_ptr=state1.air_file_ptr
>
> Now, here is the problem. With the above code, it turns out that...
>
> (*infoptr).statearr[0].resvol_file_ptr, (*infoptr).statearr
> [1].resvol_file_ptr, (*infoptr).statearr[2].resvol_file_ptr
> all have the same values. Similarly with air_file_ptr. Even though I
> haven't assigned anything to the statearr 1 and 2 yet.
>
> Is there a specific way to reference these structure arrays? Any help
> is greatly appreciated!
>
> Thanks
> Niv

If I read it right, it sounds like state1.resvol_file_ptr is not a
scalar, but is an array with 3 elements. Check it with n_elements
(state1.resvol_file_ptr). Put it another way, I think what is
happening is the same as

```
IDL> a=intarr(3)
IDL> print,a
       0       0       0
IDL> b=[1,1,1]
IDL> a[0]=b
IDL> print,a
       1       1       1
```

Which is that way because if you use an array element on the left side
of an assignment, and assign to it an array with N elements, then N

elements are assigned, starting at the one given on the left side.

Subject: Re: referencing structure arrays
Posted by Jean H. on Thu, 10 Sep 2009 16:53:39 GMT
View Forum Message <> Reply to Message

niv wrote:
> Sorry, looks like what I typed looks gibberish in places. I hope this
> one is more readable:
>
>
> Hi All,
>
> I have a problem referencing structure arrays and assigning values to
> them. So here is a snippet of my code:
>
> junk1=update_state(junk) ;creates a structure called junk1
> statearr=replicate(junk1,3) ;creates an array of structures
>
> infoptr=ptr_new({statearr:statearr}) ;need this to pass variables
> between programs
>
> align_images,state=state1 ;some program that aligns 2 images and
> stores the alignment results in state1
>
> ;assign values from the state1 structure to infoptr.statearr[0]
> structure
> (*infoptr).statearr[0].resvol_file_ptr=state1.resvol_file_pt r
> (*infoptr).statearr[0].air_file_ptr=state1.air_file_ptr
>
> Now, here is the problem. With the above code, it turns out that...
>
> (*infoptr).statearr[0].resvol_file_ptr, (*infoptr).statearr
> [1].resvol_file_ptr, (*infoptr).statearr[2].resvol_file_ptr
> all have the same values. Similarly with air_file_ptr. Even though I
> haven't assigned anything to the statearr 1 and 2 yet.
>
> Is there a specific way to reference these structure arrays? Any help
> is greatly appreciated!
>
> Thanks
> Niv

Hi Niv,
the "problem" is with replicate (or your structure). If you have a
pointer in it, you copy the pointer N times, so each of the copies still

point to the same memory space... here is an example of what is happening:

IDL> a = {b:ptr_new(indgen(10))}
IDL> c = replicate(a,5)
IDL> help,c[0].b
<Expression>   POINTER   = <PtrHeapVar2>
IDL> help,c[3].b
<Expression>   POINTER   = <PtrHeapVar2>

so if you modify one, you modify all!
What you can do is to create your structure with empty pointers,
replicate the structure and set the pointer's value then.

a = {b:ptr_new()}
c = replicate(a,5)
...
c[0].b = state1.resvol_file_ptr

Jean

Subject: Re: referencing structure arrays
Posted by nivedita.raghunath on Thu, 10 Sep 2009 17:02:57 GMT
View Forum Message <> Reply to Message

On Sep 10, 12:36 pm, pp <pp.pente...@gmail.com> wrote:
> On Sep 10, 1:12 pm, niv <nivedita.raghun...@gmail.com> wrote:
>
>
>
>> Sorry, looks like what I typed looks gibberish in places. I hope this
>> one is more readable:
>
>> Hi All,
>
>> I have a problem referencingstructurearrays and assigning values to
>> them. So here is a snippet of my code:
>
>> junk1=update_state(junk) ;creates astructurecalled junk1
>> statearr=replicate(junk1,3) ;creates an array of structures
>
>> infoptr=ptr_new({statearr:statearr}) ;need this to pass variables
>> between programs
>
>> align_images,state=state1 ;some program that aligns 2 images and
>> stores the alignment results in state1
>
>> ;assign values from the state1structureto infoptr.statearr[0]

>> structure
>> (*infoptr).statearr[0].resvol_file_ptr=state1.resvol_file_pt r
>> (*infoptr).statearr[0].air_file_ptr=state1.air_file_ptr
>
>> Now, here is the problem. With the above code, it turns out that...
>
>> (*infoptr).statearr[0].resvol_file_ptr, (*infoptr).statearr
>> [1].resvol_file_ptr, (*infoptr).statearr[2].resvol_file_ptr
>> all have the same values. Similarly with air_file_ptr. Even though I
>> haven't assigned anything to the statearr 1 and 2 yet.
>
>> Is there a specific way to reference thesestructurearrays? Any help
>> is greatly appreciated!
>
>> Thanks
>> Niv
>
> If I read it right, it sounds like state1.resvol_file_ptr is not a
> scalar, but is an array with 3 elements. Check it with n_elements
> (state1.resvol_file_ptr). Put it another way, I think what is
> happening is the same as
>
> IDL> a=intarr(3)
> IDL> print,a
>   0   0   0
> IDL> b=[1,1,1]
> IDL> a[0]=b
> IDL> print,a
>   1   1   1
>
> Which is that way because if you use an array element on the left side
> of an assignment, and assign to it an array with N elements, then N
> elements are assigned, starting at the one given on the left side.

I checked it and thats not the case. state1.resvol_file_ptr is a
scalar. I even assigned a different value:

IDL> *(*infoptr).statearr[0].resvol_file_ptr='junk'

IDL> print,*(*infoptr).statearr[0].resvol_file_ptr
junk
IDL> print,*(*infoptr).statearr[1].resvol_file_ptr
junk
IDL> print,*(*infoptr).statearr[2].resvol_file_ptr
junk

I dont' get it.

Subject: Re: referencing structure arrays
Posted by nivedita.raghunath on Fri, 11 Sep 2009 14:50:30 GMT
View Forum Message <> Reply to Message

On Sep 10, 12:53 pm, "Jean H." <jghas...@DELTHIS.ucalgary.ANDTHIS.ca>
wrote:
> niv wrote:
>> Sorry, looks like what I typed looks gibberish in places. I hope this
>> one is more readable:
>
>> Hi All,
>
>> I have a problem referencing structure arrays and assigning values to
>> them. So here is a snippet of my code:
>
>> junk1=update_state(junk) ;creates a structure called junk1
>> statearr=replicate(junk1,3) ;creates an array of structures
>
>> infoptr=ptr_new({statearr:statearr}) ;need this to pass variables
>> between programs
>
>> align_images,state=state1 ;some program that aligns 2 images and
>> stores the alignment results in state1
>
>> ;assign values from the state1 structure to infoptr.statearr[0]
>> structure
>> (*infoptr).statearr[0].resvol_file_ptr=state1.resvol_file_pt r
>> (*infoptr).statearr[0].air_file_ptr=state1.air_file_ptr
>
>> Now, here is the problem. With the above code, it turns out that...
>
>> (*infoptr).statearr[0].resvol_file_ptr, (*infoptr).statearr
>> [1].resvol_file_ptr, (*infoptr).statearr[2].resvol_file_ptr
>> all have the same values. Similarly with air_file_ptr. Even though I
>> haven't assigned anything to the statearr 1 and 2 yet.
>
>> Is there a specific way to reference these structure arrays? Any help
>> is greatly appreciated!
>
>> Thanks
>> Niv
>
> Hi Niv,
> the "problem" is with replicate (or your structure). If you have a
> pointer in it, you copy the pointer N times, so each of the copies still

> point to the same memory space... here is an example of what is happening:
>
> IDL> a = {b:ptr_new(indgen(10))}
> IDL> c = replicate(a,5)
> IDL> help,c[0].b
> <Expression> POINTER = <PtrHeapVar2>
> IDL> help,c[3].b
> <Expression> POINTER = <PtrHeapVar2>
>
> so if you modify one, you modify all!
> What you can do is to create your structure with empty pointers,
> replicate the structure and set the pointer's value then.
>
> a = {b:ptr_new()}
> c = replicate(a,5)
> ...
> c[0].b = state1.resvol_file_ptr
>
> Jean

Worked like a charm! Thank you.
