Subject: Re: pointer and structure stuff ...
Posted by David Fanning on Tue, 24 Feb 2004 15:39:29 GMT
View Forum Message <> Reply to Message

Ingo Salzmann writes:

> Can anybody please help me with the following banal problem ...
>
> IDL> info = {vectors:Ptr_New(vectors)}
> IDL> v1 = Indgen(3,30)
> IDL> *info.vectors = {v1:v1}
> IDL> help, (*info.vectors).v1
> <Expression> INT = Array[3, 30]
>
> IDL> temp = Intarr(3,20)
> IDL> (*info.vectors).v1 = temp
> IDL> help, (*info.vectors).v1
> <Expression> INT = Array[3, 30]
>
> with the first 60 elements zeroed ...
> Can I have IDL resize its variable (*info.vectors).v1 easier than by the
> following statement:
>
> IDL> v1=Intarr((SIZE(temp))[1],(SIZE(temp))[2])
> IDL> *info.vectors = {v1:v1}
> IDL> (*info.vectors).v1 = temp
> IDL> help, (*info.vectors).v1
> <Expression> INT = Array[3, 20]
>
> If dealing with a large ammount of elements stored in the info structure
> this doesn't seem handy to me :-(

No, I wouldn't think so. :-(

Is there a particular reason why you are putting your vectors
into a structure? In other words, why aren't your vectors a field
of your info structure, rather than in a structure of their own?
Pointers to structures are pretty much designed to drive you
crazy, what with all the extraneous parentheses needed to get
things to work. :-(

As a field in your info structure, you could do this:

```
info = {vectors:Ptr_New(/Allocate_Heap)}
data = Fltarr(3, 30)
*info.vectors = data
newdata = IntArr(2, 20)
*info.vectors = newdata
```

Cheers,

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

Subject: Re: pointer and structure stuff ...
Posted by Ingo Salzmann on Wed, 25 Feb 2004 08:19:02 GMT
View Forum Message <> Reply to Message

> Is there a particular reason why you are putting your vectors
> into a structure? In other words, why aren't your vectors a field
> of your info structure, rather than in a structure of their own?
> Pointers to structures are pretty much designed to drive you
> crazy, what with all the extraneous parentheses needed to get
> things to work. :-(

Well, the reason is that I would like to keep track of many different
vectors that can be categorised by their purpose ... it seemed handy to
me to have for example a structure vectors_3d_upper in which all
concerning stuff is being stored ... unfortunately the ammount of
elements is being changed sometimes and therefore I would have loved to
easily update by (*info.vectors_3d_upper).reciprocal_lattice =
uniquevecors :-(
Thanks,
Ingo

Subject: Re: pointer and structure stuff ...
Posted by btt on Wed, 25 Feb 2004 13:59:54 GMT
View Forum Message <> Reply to Message

Ingo Salzmann wrote:
>> Is there a particular reason why you are putting your vectors
>> into a structure? In other words, why aren't your vectors a field
>> of your info structure, rather than in a structure of their own?
>> Pointers to structures are pretty much designed to drive you
>> crazy, what with all the extraneous parentheses needed to get
>> things to work. :-(
>
>
> Well, the reason is that I would like to keep track of many different
> vectors that can be categorised by their purpose ... it seemed handy to

> me to have for example a structure vectors_3d_upper in which all
> concerning stuff is being stored ... unfortunately the ammount of
> elements is being changed sometimes and therefore I would have loved to
> easily update by (*info.vectors_3d_upper).reciprocal_lattice =
> uniquevecors :-(
> Thanks,
> Ingo
>

Hello,

I wonder if this might be a case for using the VECTOR object found on the RSI User Contribution web site. Or you might roll you own vector object - it's not too hard to do. These kind of objects readily accept changes in values and lengths as the need arises, but you don't have to worry about the details in you application. Then you can store all of your 'vectors' in a container and place the container in your info structure.

Cheers,
Ben
