Subject: ARRAY OF ARRAYS
Posted by dsheerin on Fri, 04 Sep 1998 07:00:00 GMT
View Forum Message <> Reply to Message

Hi
Can anyone tell me of a quick way of constructing an array of 2x2
arrays (such as those generated in a multiple call to the POLYWARP
procedure).
Thanks for any help received.

David

Subject: Re: Array of arrays
Posted by David Fanning on Wed, 22 Feb 2006 21:58:35 GMT
View Forum Message <> Reply to Message

Fritz writes:

> How do I define an array of arrays ?
>
> Suppose T is an array of 3 elements.
> And I want T[0] = intarr(2,5,4), T[1] = intarr(6,5,4) and T[2] =
> intarr(7,8,4).
>
> How do I declare T ?

As a pointer array. :-)

```
t = PtrArr(3)
t[0] = Ptr_New(intarr(2,5,4))
t[1] = Ptr_New(intarr(6,5,4))
t[3] = Ptr_New(intarr(7,8,4))

(*t[1])[0,*,2] = Indgen(5)+3
```

Cheers,

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

Subject: Re: Array of arrays
Posted by Fritz on Thu, 23 Feb 2006 14:31:17 GMT

Thanks.
What happens if I save the array T ?

> save, filename='Tarray.sav', T

Does the content of T is saved ? I mean do the three arrays inside T are saved ?
Does the restore command
> restore, 'Tarray.sav'
brings back the initial information ?

Francois.

"David Fanning" <davidf@dfanning.com> wrote in message news:MPG.1e668c7ae40db807989b9f@news.frii.com...
> Fritz writes:
>
>> How do I define an array of arrays ?
>>
>> Suppose T is an array of 3 elements.
>> And I want T[0] = intarr(2,5,4), T[1] = intarr(6,5,4) and T[2] =
>> intarr(7,8,4).
>>
>> How do I declare T ?
>
> As a pointer array. :-)
>
> t = PtrArr(3)
> t[0] = Ptr_New(intarr(2,5,4))
> t[1] = Ptr_New(intarr(6,5,4))
> t[3] = Ptr_New(intarr(7,8,4))
>
> (*t[1])[0,*,2] = Indgen(5)+3
>
> Cheers,
>
> David
> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: Array of arrays
Posted by David Fanning on Thu, 23 Feb 2006 14:54:34 GMT

Fritz writes:

> What happens if I save the array T ?
>
>> save, filename='Tarray.sav', T
>
> Does the content of T is saved ? I mean do the three arrays inside T are
> saved ?
> Does the restore command
>> restore, 'Tarray.sav'
> brings back the initial information ?

I take it you are not an empiricist. :-)

Yes, everything is saved and restored properly.

Cheers,

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

Subject: Re: Array of arrays
Posted by Fritz on Thu, 23 Feb 2006 16:31:38 GMT

One (hopefully) last question...

I have the following code:

```
nb_classes = 3
references = ptrarr(nb_classes)
for i=0, nb_classes-1 do begin
   nb_samples = 5
   list_samples = ptrarr(nb_samples)
    for j=0, nb_samples-1 do begin
       data = indgen(21,19,4)
        list_samples[j] = ptr_New(data)
    endfor
   references[i] = ptr_new(list_samples)
endfor
```

This code is simplified because for each class, the number of samples may

vary and also the size of data.

The question is:
How do I retrieve the data, let say, for sample 3 of class 2 ?

In the previous example,
> t = PtrArr(3)
> t[0] = Ptr_New(intarr(2,5,4))
> t[1] = Ptr_New(intarr(6,5,4))
> t[2] = Ptr_New(intarr(7,8,4))

(*t[1]) allows to acces the second element
(*t[1])[*,*,2] allows to acces the third band of the second element.

But in the code above ?
I tried
> dat = (*references[0])[(*liste_samples[1])]
but dat is not the same as data.

others...

"David Fanning" <davidf@dfanning.com> wrote in message
news:MPG.1e677a7dc69e8a26989ba0@news.frii.com...
> Fritz writes:
>
>> What happens if I save the array T ?
>>
>>> save, filename='Tarray.sav', T
>>
>> Does the content of T is saved ? I mean do the three arrays inside T are
>> saved ?
>> Does the restore command
>>> restore, 'Tarray.sav'
>> brings back the initial information ?
>
> I take it you are not an empiricist. :-)
>
> Yes, everything is saved and restored properly.
>
> Cheers,
>
> David
> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.

> Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: Array of arrays
Posted by David Fanning on Thu, 23 Feb 2006 16:53:25 GMT
View Forum Message <> Reply to Message

Fritz writes:

> One (hopefully) last question...
>
> I have the following code:
>
> nb_classes = 3
> references = ptrarr(nb_classes)
> for i=0, nb_classes-1 do begin
> nb_samples = 5
> list_samples = ptrarr(nb_samples)
> for j=0, nb_samples-1 do begin
> data = indgen(21,19,4)
> list_samples[j] = ptr_New(data)
> endfor
> references[i] = ptr_new(list_samples)
> endfor
>
> This code is simplified because for each class, the number of samples may
> vary and also the size of data.
>
> The question is:
> How do I retrieve the data, let say, for sample 3 of class 2 ?
>
> In the previous example,
>> t = PtrArr(3)
>> t[0] = Ptr_New(intarr(2,5,4))
>> t[1] = Ptr_New(intarr(6,5,4))
>> t[2] = Ptr_New(intarr(7,8,4))
>
> (*t[1]) allows to acces the second element
> (*t[1])[*,*,2] allows to acces the third band of the second element.
>
> But in the code above ?
> I tried
>> dat = (*references[0])[(*liste_samples[1])]
> but dat is not the same as data.
>

> others...

Oh, well, we are not going to embarrass ourselves by
explaining how long it takes us to understand things.
That's why there is 24 hours in every day, after all. :-)

I would access the data in the third sample in class 2 like this:

```
IDL> Help, *(*references[1])[2]
```

It will help to remember that the * operator has a lower order
of precedence than almost any other operator. So references[1]
is a reference to a pointer in the pointer array, and *references[1]
de-references that pointer, which is also a reference to a pointer
array, so (*references[1])[2] is a reference to a pointer in
that pointer array, and *(*references[1])[2] de-references that
pointer.

You might find these articles worth re-reading:

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

Cheers,

David

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

Subject: Re: Array of arrays
Posted by edward.s.meinel@aero. on Fri, 24 Feb 2006 15:46:47 GMT
View Forum Message <> Reply to Message

David Fanning wrote:

> So references[1] is a reference to a pointer
> in the pointer array, and *references[1]
> de-references that pointer, which is also
> a reference to a pointer array, so
> (*references[1])[2] is a reference to a
> pointer in *that* pointer array, and
> *(*references[1])[2] de-references that pointer.

I think I'm getting a headache.

Ed

PS: Wasn't that line in "The Princess Bride" during the poisoned drink
scene?

Subject: Re: Array of arrays
Posted by David Fanning on Fri, 24 Feb 2006 15:52:17 GMT
View Forum Message <> Reply to Message

edward.s.meinel@aero.org writes:

> PS: Wasn't that line in "The Princess Bride" during the poisoned drink
> scene?

I *knew* I had heard it somewhere! :-)

Cheers,

David

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