Subject: Re: How to get numbers into passed structure elements (pass-by-value/reference problem).

Posted by David Fanning on Wed, 27 Feb 2008 15:25:42 GMT

View Forum Message <> Reply to Message

Paul van Delst writes:

```
>  So, I realise this is one of those pass-by-reference or pass-by-value things, but how does
>  one get around it? Do I:
>     a) make the Type component of the Cloud structure a pointer? (Yuk!)
>     b) change the way I pass the Cloud structure into the CRTM_Read_Cloud_Record() fn?
>        i.e. not reference the cloud structure array via the index [n].
>
>  I would much prefer (b), but will that entail copying entire structures? The number of
>  "Clouds" associated with any particular "Atm[m]" profile is variable.
```

Too much misdirection for me to follow this early in the morning,
but if you pass a structure array in by reference, what are you
copying? I think you are only "copying" if you pass something in
by value. I would go with (b).

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: How to get numbers into passed structure elements (pass-by-value/reference problem).

Posted by Paul Van Delst[1] on Wed, 27 Feb 2008 15:56:22 GMT

View Forum Message <> Reply to Message

David Fanning wrote:

```
>  Paul van Delst writes:
>
>>  So, I realise this is one of those pass-by-reference or pass-by-value things, but how does
>>  one get around it? Do I:
>>     a) make the Type component of the Cloud structure a pointer? (Yuk!)
>>     b) change the way I pass the Cloud structure into the CRTM_Read_Cloud_Record() fn?
>>        i.e. not reference the cloud structure array via the index [n].
>>
>>  I would much prefer (b), but will that entail copying entire structures? The number of
>>  "Clouds" associated with any particular "Atm[m]" profile is variable.
>
```

> Too much misdirection for me to follow this early in the morning,
> but if you pass a structure array in by reference, what are you
> copying?

Well I meant if I do this:

```
  FOR n = 0, Atm.n_Clouds-1 DO BEGIN
    ; Need to store data into (*Atm.Cloud)[n]
    result = CRTM_Read_Cloud_Record( FileID, Cloud, DEBUG=Debug )
    ...Copy Cloud into (*Atm.Cloud)[n]
  ENDFOR
```

> I think you are only "copying" if you pass something in
> by value. I would go with (b).

I just completed an experiment where I passed the entire pointer array and looped over clouds in my reader,

```
  result = CRTM_Read_Cloud_Record( FileID, *Atm.Cloud, DEBUG=Debug )
```

since it was passing (*Atm.Cloud)[n] that was buggering me up.

I find it a fundamental flaw of IDL that the argument passing mechanism (reference or value) is so exposed to the user that they have to tailor their code to prevent data loss.
I understand why it's there, but even Fortran has moved beyond it (for about 18(!) years now).

cheers,

paulv

Subject: Re: How to get numbers into passed structure elements (pass-by-value/reference problem).
Posted by lecacheux.alain on Thu, 28 Feb 2008 10:23:58 GMT

View Forum Message <> Reply to Message

On 27 fév, 16:56, Paul van Delst <Paul.vanDe...@noaa.gov> wrote:
> David Fanning wrote:
>> Paul van Delst writes:
>
>>> So, I realise this is one of those pass-by-reference or pass-by-value things, but how does
>>> one get around it? Do I:
>>>   a) make the Type component of the Cloud structure a pointer? (Yuk!)
>>>   b) change the way I pass the Cloud structure into the CRTM_Read_Cloud_Record() fn?
>>>   i.e. not reference the cloud structure array via the index [n].
>
>>> I would much prefer (b), but will that entail copying entire structures? The number of
>>> "Clouds" associated with any particular "Atm[m]" profile is variable.

>
>> Too much misdirection for me to follow this early in the morning,
>> but if you pass a structure array in by reference, what are you
>> copying?
>
> Well I meant if I do this:
>
>   FOR n = 0, Atm.n_Clouds-1 DO BEGIN
>   ; Need to store data into (*Atm.Cloud)[n]
>   result = CRTM_Read_Cloud_Record(FileID, Cloud, DEBUG=Debug)
>   ...Copy Cloud into (*Atm.Cloud)[n]
>   ENDFOR
>
>> I think you are only "copying" if you pass something in
>> by value. I would go with (b).
>
> I just completed an experiment where I passed the entire pointer array and looped over
> clouds in my reader,
>
>   result = CRTM_Read_Cloud_Record(FileID, *Atm.Cloud, DEBUG=Debug)
>
> since it was passing (*Atm.Cloud)[n] that was buggering me up.
>
> I find it a fundamental flaw of IDL that the argument passing mechanism (reference or
> value) is so exposed to the user that they have to tailor their code to prevent data loss.
> I understand why it's there, but even Fortran has moved beyond it (for about 18(!) years now).
>
> cheers,
>
> paulv

Since IDL pointers point to variables which are global in scope and
IDL structures are
global too, I would suggest, in any of your function, to just pass a
pointer on the used instance
of CRTM_Atmosphere...
alx