Subject: Re: Array of associated variables?
Posted by marc schellens[1] on Thu, 20 Jan 2005 11:25:06 GMT
View Forum Message <> Reply to Message

An array of associated variables is not possible.
You could make a poitner to your associated variable and store it in
a pointer array.

```
IDL> openu,1,'aaa.tmp'
IDL> a=assoc(1,intarr(10))
IDL> print,a
IDL> p=ptr_new(a)
IDL>
```

Cheers,
marc

sso@nilu.no wrote:
> A brief question: Is it in some way or another possible to make an
> array of associated variables? The examples below don't work but
> perhaps there is some other way?
>
> -------
> for i=1,n do begin
> a=assoc(...)
> if n_elements(all) le 0 then all=[a] else all=[all,a]
> endfor
> -------
> a=assoc(...)
> all=replicate(a,n)
> -------
>
> Thanks for any help
> Sverre Solberg
>

Subject: Re: Array of associated variables?
Posted by sso on Thu, 20 Jan 2005 14:25:08 GMT
View Forum Message <> Reply to Message

Thanks!
However, I dont get this really to work. It's probably a basic error I
am doing, but a code like this will crash.

```
all = ptrarr(n, /allocate_heap)

FOR i = 0, n-1 DO BEGIN
```

```
;..define a, the associated variable (that works)
p = ptr_new(a)
all(i) = p
ENDFOR

;..extract values back:
a = all(0)
pval = *a

----------
I'm not that into pointers so it may be a simple error
```

Sverre

Subject: Re: Array of associated variables?
Posted by David Fanning on Thu, 20 Jan 2005 14:52:09 GMT
View Forum Message <> Reply to Message

sso@nilu.no writes:

```
>  However, I dont get this really to work. It's probably a basic error I
>  am doing, but a code like this will crash.
>
>  all = ptrarr(n, /allocate_heap)
>
>  FOR i = 0, n-1 DO BEGIN
>  ;..define a, the associated variable (that works)
>  p = ptr_new(a)
>  all(i) = p
>  ENDFOR
>
>  ;..extract values back:
>  a = all(0)
>  pval = *a
>
>  ----------
>  I'm not that into pointers so it may be a simple error
```

Don't know. Looks right to me. Here is my example:

```
n=3
filename = 'junk'
all = ptrarr(n)

FOR I = 0, n-1 DO BEGIN
   Openw, lun, filename + String(I, format='(i1)') +'.pro', /Get_Lun
   a = Assoc(lun, Bytarr(200))
```

```
   p = ptr_new(a)
   all[I] = p
ENDFOR

;..extract values back:
a = all[0]
pval = *a
help, pval
END
```

And here is what I get:

```
IDL> .COMPILE "C:\RSI\David\assoc_ptr.pro"
% Compiled module: $MAIN$.
IDL> .go
PVAL            BYTE      = File<C:\RSI\David\junk0.pro> Array[200]
```

Cheers,

David

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

Subject: Re: Array of associated variables?
Posted by sso on Fri, 21 Jan 2005 08:58:46 GMT
View Forum Message <> Reply to Message

Thanks! This works, but my code still doesn't. The difference which makes my program crash is the definition of the a=assoc(...) which in my code is a structure. It's like this:

```
openu, inun, filename, /swap_if_little_endian, /get_lun
a = assoc(inun, strucval, headpos)
```

where strucval is a structure (consisting of various types of float/long data) and headpos is the offset. There's another difference also, and that's the open statement above, but I guess that shouldnt cause the program to crash.

So it seems as the use of pointer arrays for associated variables doesnt work when the associated variable is a structure? Hmm. I must admit I get somewhat confused with this. Perhaps it's what one would call a bug? At least it doesnt only halt. The whole idl session

crashes, returns to the Unix environment and give me the Unix message
"Segmentation fault" which normally indicates a rather serious error.

Guess I will have to leave the array of associated vars. then and try
to circumvent this in some way(?)

Sverre

David Fanning wrote:
> sso@nilu.no writes:
>
>> However, I dont get this really to work. It's probably a basic
error I
>> am doing, but a code like this will crash.
>>
>> all = ptrarr(n, /allocate_heap)
>>
>> FOR i = 0, n-1 DO BEGIN
>> ;..define a, the associated variable (that works)
>> p = ptr_new(a)
>> all(i) = p
>> ENDFOR
>>
>> ;..extract values back:
>> a = all(0)
>> pval = *a
>>
>> ----------
>> I'm not that into pointers so it may be a simple error
>
> Don't know. Looks right to me. Here is my example:
>
> n=3
> filename = 'junk'
> all = ptrarr(n)
>
> FOR I = 0, n-1 DO BEGIN
> Openw, lun, filename + String(I, format='(i1)') +'.pro', /Get_Lun
> a = Assoc(lun, Bytarr(200))
> p = ptr_new(a)
> all[I] = p
> ENDFOR
>
> ;..extract values back:
> a = all[0]

> pval = *a
> help, pval
> END
>
> And here is what I get:
>
> IDL> .COMPILE "C:\RSI\David\assoc_ptr.pro"
> % Compiled module: $MAIN$.
> IDL> .go
> PVAL BYTE = File<C:\RSI\David\junk0.pro> Array[200]
>
>
> Cheers,
>
> David
>
> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: Array of associated variables?
Posted by David Fanning on Fri, 21 Jan 2005 14:29:37 GMT
View Forum Message <> Reply to Message

sso@nilu.no writes:

> Thanks! This works, but my code still doesn't. The difference which
> makes my program crash is the definition of the a=assoc(...) which in
> my code is a structure. It's like this:
>
> openu, inun, filename, /swap_if_little_endian, /get_lun
> a = assoc(inun, strucval, headpos)
>
> where strucval is a structure (consisting of various types of
> float/long data) and headpos is the offset. There's another difference
> also, and that's the open statement above, but I guess that shouldnt
> cause the program to crash.

I would suspect the OPEN statement before I suspected the
structure in the ASSOC command. But you are right, it shouldn't
crash. Probably a bug somewhere. If you have a small reproducible
program, send it to RSI and find out.

> So it seems as the use of pointer arrays for associated variables
> doesnt work when the associated variable is a structure? Hmm. I must
> admit I get somewhat confused with this.

I don't think this is true. In any case, it is WAY to early
to jump to this conclusion. :-)

> Guess I will have to leave the array of associated vars. then and try
> to circumvent this in some way(?)

Store the filenames and just do the ASSOC whenever you need something
from the file.

Cheers,

David

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