Subject: Re: Proper pointer cleanup question
Posted by MKatz843 on Tue, 08 Apr 2003 17:26:17 GMT
View Forum Message <> Reply to Message

>>> 2) a and all of its dependent pointers:
>>> for i=0,n_elements((*a).p)-1 do $
>>> ptr_free, ((*a).p)(i)
>>> ptr_free, a
>>
>> #2 is the go. All the others leave you with dangling references and memory leaks. My
>> personal mantra is that when it comes to pointers, be very explicit in their garbage
>> collection i.e. don't assume freeing a pointer also frees any "child" pointers like the
>> components "p" in your example. (I actually don't know of any languages that *do* do that,
>> but I'm barely bilingual. :o)
>

Thanks! I'll write myself a full reverse-ordered cleanup routine.
I suppose this explicit cleanup is just as important for objects as
well:
Object pointer fields should be explicitly freed in the Cleanup
method.

Question 1) But what about simple scalar pointers?
a = ptr_new(fltarr(10,10))

If I set
a = 0
Will I have stranded my fltarr(), or is IDL smart enough to deallocate
it properly?

Question 2) Then how about this scenario
a = ptr_new(fltarr(10,10))
b = ptr_new(dblarr(5,5))

a = b ;--- Does this strand the original a array?
*a = *b ;--- How about this?

From looking at help, /memory and testing the above, I think the
answer to both of my questions is that memory IS stranded unless you
explicitly free it in all of the above cases.

M. Katz