Subject: Re: Proper pointer cleanup question
Posted by Paul Van Delst[1] on Mon, 07 Apr 2003 23:27:56 GMT
View Forum Message <> Reply to Message

Paul van Delst wrote:
>
> "M. Katz" wrote:
>>
>> I want to make sure I'm taking all the steps necessary to clean up
>> pointers and free memory when I'm done with them. Here's an example.
>>
>> Suppose I have a pointer to a structure that contains pointers.
>>
>> a = ptr_new({n:10, p:ptrarr(10)})
>>
>> So, a is a pointer, and ((*a).p)(i) are pointers as well.
>> When I'm done with a and all of it's components, I can do a few things
>> to clean it up, but I don't want to do more than what's necessary.
>> Here's a few options.
>>
>> 1) Just a:
>> ptr_free, a
>>
>> 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)

Apologies for the replying to me own post, but one thing I've noticed in using pointers in Fortran (for large arrays that are allocated and deallocated frequently) is that nullifying them in the opposite order in which they're allocated (where possible) may minimise memory fragmentation. Anecdotal evidence only on my part, though.

paulv

--
Paul van Delst
CIMSS @ NOAA/NCEP/EMC
Ph: (301)763-8000 x7748
Fax:(301)763-8545