Subject: Re: Proper pointer cleanup question
Posted by JD Smith on Wed, 09 Apr 2003 23:29:00 GMT
View Forum Message <> Reply to Message

On Tue, 08 Apr 2003 18:49:40 -0700, Mark Hadfield wrote:

> "Paul van Delst" <paul.vandelst@noaa.gov> wrote in message
> news:3E93649D.C9FBEBA0@noaa.gov...
>> JD Smith wrote:
>>> Another option is:
>>>
>>> heap_free,a
>>>
>>> which will accomplish the same as #2 (avoiding a memory leak), and is
>>> only slightly slower. When you're feeling truly lazy, it's quite a
>>> blessing. You have no flexibility to pick and choose what to parts
>>> of a data structure to free, but often this isn't an issue.
>>
>> Wha..? Is that another one of those undocumented IDL routines? It works
>> on
> my current
>> version, but bugger me if I can find it documented anywhere.
>
> It's in the 5.6 documentation, which says it's been in the language
> since 5.3.
>
> But I'd never heard of it either.

It's rather nice. Think of it as a light-weight HEAP_GC, and you'll
understand the time-savings it can offer you. At first I felt a bit
like I was cheating by using it, often inserting self-chiding comments
like "Replace with a real Cleanup!", but gradually I realized I didn't
really *need* to go to this trouble. The only time it's not helpful
is when you want to trim various parts of your data structure
selectively. This is an uncommon case though.

As far as the one remaining reason not to prefer HEAP_FREE, I decided
to test the notion that it is slower than doing it "the proper way"
yourself. I created an arbitrary, complex, nested structure. I spent
15 minutes figuring out how to un-roll all my pointers to structures
with pointer arrays and objects so I could free them, and here's what
I got:

"Correct" full loop free:

TIME: 0.15468192

HEAP_FREE time:

TIME: 0.11966300

Oh my. The supposedly cheating way is faster! Presumably this is because the internal HEAP_FREE doesn't have to pay the IDL looping penalty, which more than makes up for the dynamic data structure search it must perform to find all the heap data. Just to give you a flavor of which looks better in your Cleanup code, here are the two versions:

```
 if ptr_valid(a.a) then begin
    for i=0,n_elements(*a.a)-1 do begin
       obj_destroy,(*(*a.a)[i].dr).is
       ptr_free,(*a.a)[i].dr
    endfor
    ptr_free,(*a.a).dr,a.a
 endif
 if ptr_valid(a.b) then begin
    for i=0,n_elements(*a.b)-1 do ptr_free,*(*a.b)[i],(*a.b)[i]
    ptr_free,*a.b,a.b
 endif
```

vs.

```
 heap_free,a
```

I think I know which method I'll be preferring in the future.

JD