Subject: Re: IDL Memory Leaks
Posted by Myron Brown on Mon, 05 Nov 2001 20:57:41 GMT
View Forum Message <> Reply to Message

Thanks for the reply. Here's more information...

1. One source of leaks was widgets, so as a first cut, I removed that code. I can later go back and fix that problem, but for now, widgets are not needed or used.
2. I have no calls to the PTR routines in my code at all.
3. Put simply, my code consists of an IDL routine that calls another IDL routine over and over in a for loop. The routine that is run over and over (1) reads data, (2) performs some processing of that data and (3) writes results to file. This continues until memory is exhausted.
4. The code includes repetitive reassignments of variables to large arrays, even within a routine (still in scope). I tried to assign variables to zero each time, but it doesn't seem to make a difference. Should this really be necessary?
5. File I/O is done using ASSOC.
6. Structures are used.

It "appears" that something isn't being deallocated somewhere and that these things add up to consume all of the available memory. Since I'm not using widgets anymore and I'm not using pointers directly, the only thing I can guess is that IDL may have issues with memory leaks under "some" circumstances, or perhaps there are routines that, when used, one should be careful with. If that's true, then I'd just like to program around these circumstances. Otherwise, I'm at a loss.

Thanks again.

Myron

David Fanning wrote:

> Myron Brown (brownmz1@jhuapl.edu) writes:
>
>> Recently, I have noticed that my IDL programs leak memory, but I never
>> use pointers directly. This is true when running with IDL on a Windows
>> PC or on an SGI workstation. Widgets seem to be one source of
>> problems. File I/O seems to be another, but I'm not yet sure. Due to
>> the problems I'm having with memory leaks, my long runs eventually die
>> when memory is exhausted.
>>
>> Does anyone have any hints on ways to avoid memory leaks in IDL?
>
> Well, be *very* careful whose programs you use.
> Those ones from that Coyote site are notorious

> for having memory leaks. :-)
>
> Having embarrassed myself with memory leaks far too
> often for it to be amusing anymore, I have learned
> a couple of things about the subject. Here are some
> rules of thumb I use.
>
> 1. In widget programs put your clean-up routines
> in a procedure that is called when the top-level
> base dies. (In other words, use the CLEANUP keyword
> to the XMANAGER call.) *Don't* put your clean-up
> routines in a QUIT button event handler. People
> don't exit your programs with the QUIT button!
>
> 2. Put your CLEANUP procedure (in widget or object
> programs) VERY close to the GUI or INIT procedures
> in your program file.
>
> Most memory leaks come from adding a pointer to the
> program somewhere during development and forgetting
> to put the complementary cleanup in the CLEANUP
> procedure. Having it very close by helps a lot.
>
> 3. Don't create a pointer without *immediately*
> adding the line that cleans it up in the CLEANUP
> routine (see 2, above).
>
> 4. If you are putting something new into a pointer,
> always do this:
>
> *ptr = somethingNew
>
> Or, if you have to, this:
>
> Ptr_Free, ptr
> ptr = Ptr_New(somethingNew)
>
> But, never this:
>
> ptr = Ptr_New(somethingNew)
>
> 5. Before you show your program to *anyone*,
> (a) start a new IDL session, (b) run your program,
> (c) exit the program, and (d) type "HELP, /HEAP".
> If anything exists on the heap, immediately read
> steps 1-4 again.
>
> That (and damn careful programming) should help. :-)

>
> Cheers,
>
> David
> --
> David W. Fanning, Ph.D.
> Fanning Software Consulting
> Phone: 970-221-0438, E-mail: david@dfanning.com
> Coyote's Guide to IDL Programming: http://www.dfanning.com/
> Toll-Free IDL Book Orders: 1-888-461-0155
