Subject: Re: IDL Memory Management
Posted by Craig Markwardt on Thu, 06 Jan 2000 08:00:00 GMT
View Forum Message <> Reply to Message

"Myron Brown" <Myron.Brown@jhuapl.edu> writes:

> Hi. I'm having a problem that I'm not sure how to solve. I call a
> procedure I wrote which manipulates large images. When it's done, it passes
> me back a relatively small array. However, AFTER the procedure is complete,
> I have problems with swap space. It appears that something is still
> allocated (or something). I am not using pointers, just simple arrays
> (which I assume get thrown on the stack - maybe not). Any ideas on how to
> deallocate this?
>

This question comes up every so often, and there is no easy answer.
RSI has a "Tip" on its web page which treats this issue in a little
more detail.

I assume you are running under a Unix-type OS. Under Unix the memory
space of any process including IDL is treated as a one-demensional
array of cells. IDL requests more memory from the system on an
as-needed basis. If you do a lot of creating and destroying of IDL
variables, especially with big variables, you can end up with a lot of
unused *HOLES* in memory, and unfortunately holey memory can't be
returned to the system. Windows may or may not be different in this
respect.

The solutions are to:

 * avoid making extra copies of your memory hogging image variables.
   This includes the implicit copies that are made in arithmetic
   expressions.

 * read about the TEMPORARY() function, and use it.

 * consider some form of "chunking" in your processing. That is,
   operate on banded subsets of your image so that your overall memory
   footprint is smaller. I do this successfully with gigabytes worth
   of data.

Cheers, and good luck,

Craig

--

 -- --------------

Craig B. Markwardt, Ph.D.   EMAIL: craigmnet@cow.physics.wisc.edu

Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
-- --------------