
Subject: IDL - freeing up used memory?

Posted by [Andy Sayer](#) on Wed, 29 May 2013 16:55:21 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi all,

I'm running into a memory issue which I am not certain whether is related to my IDL code or the machine the code is running on itself, and am wondering whether someone might be able to help me out. To start with, I'm using IDL 7.1.1 on CentOS, invoking via command line.

I've got a long piece of code which loops over a bunch of files, and for each file:

- reads the data in
- performs some operations
- stores the results in my 'output arrays'

and then moves onto the next file in the list. I define all my 'output' arrays up-front at the start of the code (before the loop). After dealing with a file, the associated interim variables (i.e. the data read directly in from the file, which consist of several hundreds of thousands of floating-point values) are set to an integer value of zero to save memory. I am also careful to close file units after they're opened.

If I do a `help,/memory` between dealing with files in my loop, the heap memory used is fairly constant (at ~130,000,000) and the maximum is typically 10-20% higher than that. This system has a lot of RAM and I don't think this process can be using anywhere near that limit. So, there is no apparent accumulation of junk in between reading files in the loop.

However, despite this, after some number of files I get this type of message:

```
% Unable to allocate memory: to make array.  
Cannot allocate memory
```

At the line this error is issued at, the code is trying to create the structure to store the several hundreds of thousands of floating-point values to be read in from the file (i.e. at the 'read the data in' stage rather than the 'perform some operations on it' stage).

If I start my process from the offending file, it works fine (and if I start a few files earlier or later in the chain, the code instead falls over correspondingly earlier or later). This suggests that, despite the `help,/memory` output, there is some issue with the available memory decreasing as I go from file to file.

I've tried using the heap memory diagnosis/cleanup tools like `heap_gc` (e.g. http://www.idlcoyote.com/fileio_tips/memleak.html) but am not using any pointers, and so that doesn't seem to do anything.

I've found posts (such as <http://cow.physics.wisc.edu/~craigm/idl/archive/msg01590.htm>) saying that these issues can be the result of available memory being fragmented. That sounds more plausible to me. Various posts report issues with various IDL version/OS combinations. David has a related page on memory issues in IDL 8, using the Workbench. However, I'm not

using the Workbench, or IDL 8, and so I don't think that is applicable to my situation.

So... does anyone know what could be going on here? Is my memory getting fragmented, and if so, is there any way to fix that? Is there some way besides help,/memory to see whether my memory usage is really stable between iterations? Any suggestions would be appreciated!

As an addendum, while I've been using IDL a while, I am not a computer scientist and have neither root access nor physical access to the machines this code is being executed on, both of which may limit my ability to poke into some aspects of the problem.

Thanks,

Andy

Subject: Re: IDL - freeing up used memory?
Posted by [Fabzi](#) on Fri, 31 May 2013 13:42:03 GMT
[View Forum Message](#) <> [Reply to Message](#)

On 05/31/2013 03:32 PM, AMS wrote:
> As an update, in case anyone else has a similar issue in the future:

Thanks for the feedback. I think I can notice similar behavior when working with NetCDF files. I am not sure if I can call it "memory leak" but after working a long time on large files, my RAM becomes more and more full. When it's getting too annoying I restart the workbench.

I have the feeling that IDL is caching previous data-access to increase speed in a future call.
Dear IDL-experts, is it possible that IDL does something like this or is it fantasy?

Subject: Re: IDL - freeing up used memory?
Posted by [chris_torrence@NOSPAM](#) on Mon, 03 Jun 2013 05:11:22 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi all,

IDL doesn't do any special memory caching. It is probably the HDF5 library, which does cache a lot of internal state information. That H5_CLOSE frees up all that cached memory.

We are currently upgrading our HDF5 to the latest version of the library, which *might* help. But in the meantime, I would recommend just doing the H5_CLOSE and allowing HDF5 to release all of its memory.

Cheers,
Chris
ExelisVIS
