Subject: yet another idl memory question
Posted by jw2519 on Thu, 04 Jun 2009 15:53:05 GMT
View Forum Message <> Reply to Message

How do I force that IDL set aside a chunk of contiguous memory to be refilled by a variable of equal size at each iteration of a procedure?

More specifically, I have a code which consists of three procedures. One of these serves as the wrapper, one uses the IDL OpenDAP utility to download the relevant data, and the third processes the data into the grid and format that I want. The code will iterate for as long as the wrapper indicates that processed data files need to be created. I am running this on a 64-bit linux machine (IDL 7.0).

Each iteration of the code performs four OpenDAP calls; each call downloads a large (~750MB) structure. After several iterations, the code exits with an insufficient memory to allocate array error (or sometimes "error reading value array: read is for xxx bytes, but there are only yyy in the buffer"; always this error occurs within an OpenDAP call). Using help,/memory and several of the tricks and tools posted elsewhere, I have been able to substantially reduce the max memory usage, but this only nets me another one and a half iterations or so. So my guess is that the problem is that there is no longer a sufficiently large block of contiguous memory into which to download the structure. Each of the four structures is of equal size at each iteration. How can I set aside a block of contiguous memory so that IDL will reuse the same chunk for each structure every time i call the procedure? I've tried using a solitary common block (i.e. just within the procedure in question), and I've attempted to use pointers. The program runs in both cases, but still exits with the same error. Of course, I am not at all certain that I used either of these correctly...

Any suggestions are greatly appreciated.