Subject: Large Arrays in IDL
Posted by szoonem on Fri, 12 Jan 1996 08:00:00 GMT
View Forum Message <> Reply to Message

Hi;

This is a question that concerns astronomers once in a while. I have
wondered what the limit is to the size of the arrays that IDL can create.
Some times IDL fails to read a FITS image complaining:

% Unable to allocate memory: to make array.
  not enough core

This is not so uncommon if one has to deal with even moderate size cubes.
I have tried to use .SIZE to increase the default value for data area but
it doesn't seem to work. So the question is if there is a way to open a
very large image by IDL. IRAF seems to have no problem with this. Does
this problem arise from lack of RAM on the computer?

Any advice will be appreciated.

- Saeid

  -- -----
Saeid Zoonematkermani	E-Mail: szoonem@astro.sunysb.edu
Earth and Space Sciences	Voice: (516)632-8237
State University of New York	Fax: (516)632-8742
Stony Brook, NY 11794-2100	
  -- -----
| Home Page --> http://ozone.ess.sunysb.edu/ |
  -- -----

Subject: Re: Large Arrays in IDL
Posted by David Ritscher on Sat, 20 Jan 1996 08:00:00 GMT
View Forum Message <> Reply to Message

I have worked with large arrays in memory (150 MByte) in UNIX (HP-UX) and found that
the key point is not amount of physical RAM, but rather amount of swap space. In
addition,
I found that with HP-UX 9.05 I also needed to change a kernal parameter to let the
machine
know to use the swapspace for my application.

I needed to change a kernal parameter:
maxdsiz: Max Data Segment Size (bytes)
from a smaller default value to a value appropriate to the size of my swap space. I
set it to about 300 Mbytes. (note that I have a swap space that is enough larger

than this value that
other processes and the unix kernal can continue to survive after I've grabbed all this amount
of the swap space). I can't remember what the parameter is for a SUN, but I remember doing something similar there.

One must be a bit cautious with this, and remember that, once memory has been allocated in IDL
or PV-Wave under UNIX, the program will never give it back to the operating system, even if the
variables are deleted. Also, there can be problems with spawning, due to the spawned process
inheriting the process space of the parent (see earlier postings on this theme). If more than
one person is working on the same machine, some care is needed to make sure the other person
can continue to work.

As mentioned in the other postings on this topic, ASSOC() is one approach to this problem, leaving the data on the disk. Unfortunately, indexing individual array elements using ASSOC
is very inefficient. Other programs have dealt with the problem, and handle the paper work
that allow the programmer to work with a variable as though it were all in memory, and then
deal with swapping things in and out transparently. Particularly nice are those that have a
concept of a data pipeline, where there is a source and various operations on that source;
the data is only accessed when the output of those calulations for a particle range of data is
needed. Has anyone developed anything along these lines that could be compatible with IDL/PVWave?
--

David Ritscher

Raum 47.2.279
Zentralinstitut fuer Biomedizinische Technik
Albert-Einstein-Allee 47
Universitaet Ulm
D-89069 ULM
Germany

Tel: ++49 (731) 502 5313
Fax: ++49 (731) 502 5343
or 5317
internet: david.ritscher@zibmt.uni-ulm.de