Subject: Re: IDL crashes ... short of memory?
Posted by jacobsen on Wed, 13 Oct 1993 12:40:15 GMT
View Forum Message <> Reply to Message

This is due to idiosyncracies of IDL and AIX.

IDL gobbles up memory very quickly. My understanding is that if you do
```
 IDL> a=fltarr(1024,1024)
 IDL> b=3.0
 IDL> a=0
 IDL> c=1.0
 IDL> d=fltarr(1024,1024)
```
that IDL will see 4 MBytes freed up with "a=0". However, the
"c=1.0" call will put 4 bytes in that first region of memory.
Then the "d=fltarr(1024,1024)" call will want 4 MB of contiguous
memory, and it will have to go beyond the "b=3.0" memory location
to get it. THerefore, even though you're really only using
4 MB + 8 bytes, IDL has allocated 8 MB + 8 bytes. You can
help this by being careful and using "a=fft(a,-1,/overwrite)"
or "a=3.+temporary(a)". I think my understanding of this is correct...

The real problem though is with AIX. (THis is not a flame - overall
I really like AIX, and could not imagine dealing with Unix without
SMIT). AIX has a memory allocation scheme whereby you can
malloc() [allocate for non C language types] more memory than
exists even in paging space, and you will encounter no resistance.
However, if you actually try to USE more memory than is available,
the system will go out and kill some processes (most likely but
not always yours) to free up virtual memory.

The solution is to increase your paging space and physical memory.
We routinely have 2-4 people using IDL for large array image
processing on an IBM 320H with no problems. We have 64 MB of
RAM, and 248 MB of swap space on disk. We also have four
X terminals running off the poor little machine, and another
RS/6000 using the file system through NFS. Performance
is great, though.

--
****
Chris Jacobsen, Department of Physics, SUNY at Stony Brook
Phone (516) 632-8093, FAX -8101 Bitnet: cjacobsen@sbccmail
jacobsen@xray1.physics.sunysb.edu ALL-IN_ONE: CJACOBSEN
****