Subject: Memory Headache II
Posted by David on Fri, 30 Jan 2004 21:06:43 GMT
View Forum Message <> Reply to Message

Well, I have a lovely little Mac with 8 GB of RAM, and an allegedly
64-bit OS. I have learned that OS X is not capable of giving more than
2^32 bytes of address space to a single process. I find IDL gives up on
data memory around 3.6 GB. Does IDL hold a chunk of RAM in reserve for
compiled code and temporary variables for operations?

IDL's 32 bit implementation is supposed to be capable of 2 GB arrays.
When I attempt to grab more than 2 GB using the new_ptr function I get
the expected malloc errors. However, I find I get these errors even if
I try to grab something like 1.6 or 1.7 GB; a value too far off to be
attributable to whether a GB is 10^9 bytes or 2^30 bytes. Is there some
unseen overhead at issue here? (I have experimented in detail to find
out to the byte how far I can go, but do not have that info handy.)

Another oddity occurs when I try to

a=fltarr(1024,1024,1024,/noz)

Instead of the stream of malloc errors, I get something to the effect
of "this array has too many elements" . Is there an element limit
too? When I try

a=bytarr(1024,1024,1024,/noz) the memory is allocated w/o a hitch.

My bottom line questions:
1) Why can't I get 2 GB arrays?
2) What is this "too many elements thing?" Does idl really care about
the number of elements or is this some sort of memory error anticpator
that kicks in under certain circumstances to avoid even calling malloc
with too large of a request?
3) Does anyone have a finger in the wind as to when a full 64-bit
implementation of IDL might be available for *nix distributions?

Thanks for your input

--
David Theil davidt123 ####at####sbcglobal####dot####net

Subject: Re: Memory Headache II
Posted by David T on Mon, 02 Feb 2004 06:11:17 GMT
View Forum Message <> Reply to Message

In article <MPG.1a86c395d77cb5af9897c6@news.frii.com>, David Fanning

<david@dfanning.com> wrote:

> http://www.dfanning.com/file_io/lgfiles.html

I think you meant:

http://www.dfanning.com/fileio_tips/lgfiles.html

I did read that thread, and the underlying issues described seemed to be pretty windows specific. I don't think Win32 issues will have a direct metaphor in FreeBSD, will they? I mean not beyond the inability to get a 64 bit pointer, that is. Again, I don't see what my arrays top out at 1.6 GB instead of 1.9 GB. in Mac OS X.

Subject: Re: Memory Headache II
Posted by Pavel Romashkin on Mon, 02 Feb 2004 18:07:20 GMT
View Forum Message <> Reply to Message

Have you tried to allocate smaller arrays in a loop and see how much you can allocate? IDL allocates ram for some extra stuff when using FLTARR and such, like indices.
If you weren't on OSX, I'd say your limitation is due to memory fragmentation, but I don't know how OSX handles this.

Pavel

David T wrote:
>
> In article <MPG.1a86c395d77cb5af9897c6@news.frii.com>, David Fanning
> <david@dfanning.com> wrote:
>
>> http://www.dfanning.com/file_io/lgfiles.html
>
> I think you meant:
>
> http://www.dfanning.com/fileio_tips/lgfiles.html
>
> I did read that thread, and the underlying issues described seemed to
> be pretty windows specific. I don't think Win32 issues will have a
> direct metaphor in FreeBSD, will they? I mean not beyond the inability
> to get a 64 bit pointer, that is. Again, I don't see what my arrays
> top out at 1.6 GB instead of 1.9 GB. in Mac OS X.

Subject: Re: Memory Headache II
Posted by Craig Markwardt on Tue, 03 Feb 2004 04:12:21 GMT

David T <nospam@me.or.die> writes:
> In article <MPG.1a86c395d77cb5af9897c6@news.frii.com>, David Fanning
> <david@dfanning.com> wrote:
>
>> http://www.dfanning.com/file_io/lgfiles.html
>
> I think you meant:
>
> http://www.dfanning.com/fileio_tips/lgfiles.html
>
> I did read that thread, and the underlying issues described seemed to
> be pretty windows specific. I don't think Win32 issues will have a
> direct metaphor in FreeBSD, will they? I mean not beyond the inability
> to get a 64 bit pointer, that is. Again, I don't see what my arrays
> top out at 1.6 GB instead of 1.9 GB. in Mac OS X.

Being an astronomer myself, 1.6 GB is the same as 2 GB, or even 4 GB,
in my view. You are getting close enough to the memory limit that
it's difficult to rely on anything working. You might consider some
examples:

IDL has to reserve space for its own program code.

I don't know how shared libraries work on Mac OS X, but it is common
for even shared libraries to occupy a fixed address in address space.

IDL tends to do a lot of shifting around of data during arithmetic
calculations.

Given these are true, and tend to fragment memory, I doubt you will
always get the full address space for your data. If you want perfect
memory allocation repeatability, then you can write your own operating
system in assembly language with static memory allocation.

Or, you might consider ways to divide and conquer your problem.

Good luck!
Craig

--
-- --------------
Craig B. Markwardt, Ph.D. EMAIL: craigmnet@REMOVEcow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
-- --------------