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
 -- --------------