Subject: Re: IDL memory question
Posted by David Kastrup on Mon, 07 Sep 1998 07:00:00 GMT
View Forum Message <> Reply to Message

Alex Schuster <alex@rosa.mpin-koeln.mpg.de> writes:

> Eugenio Sansosti wrote:
>
>> Even if this operation make the required memory free for my IDL
>> application, it does not make memory free for other applicstions running
>> onmy machine. That is, other machine users cannot use the memory I have
>> allocated untill I exit IDL.
>>
>> Does any of you have a solution for that? Is it an IDL configuration
>> problem or is there any IDL command I can use?
>
> Sorry, there is no way. It seems this is a general problem for any C
> progam which uses malloc() and free() to access heap memory.
> The FAQ has an entry about this:
> http://la.znet.com/~mgs/idl_faq.html#T27

It's a general problem for any C program compiled with a stupid C
library or working on a braindead system. If your system is not
braindead and your C library is the GNU C library glibc, then large
allocations will be done in a way that allow reclaiming space
immediately by the operating system as soon as it gets released, even
if the chain of allocations would leave holes in the available memory
space.

For example, the Linux version of IDL should not exhibit this problem
(at least if it is linked with glibc instead of the old system
libraries in use over a year ago).

--
David Kastrup Phone: +49-234-700-5570
Email: dak@neuroinformatik.ruhr-uni-bochum.de Fax: +49-234-709-4209
Institut fï¿½r Neuroinformatik, Universitï¿½tsstr. 150, 44780 Bochum, Germany

Subject: Re: IDL memory question
Posted by David Kastrup on Mon, 07 Sep 1998 07:00:00 GMT
View Forum Message <> Reply to Message

Eugenio Sansosti <sansosti@irece1.irece.na.cnr.it> writes:

> Suppose I define a very big array, for instance a=fltarr(8000,8000).
> When I finish using it, I can make memory free by assigning to it a

> scalar value (for example a=0.), as also suggested in the IDL
> documentation.
>
> Even if this operation make the required memory free for my IDL
> application, it does not make memory free for other applicstions running
> onmy machine. That is, other machine users cannot use the memory I have
> allocated untill I exit IDL.
>
> Does any of you have a solution for that? Is it an IDL configuration
> problem or is there any IDL command I can use?

I would guess that IDL basically uses the malloc/free heap system from the underlying system-library/operating system. If it does, this would mean that, for example, on a standard Linux system (which allocates larger chunks of memory in a very specific way), the memory could be immediately reclaimed by other applications. Other systems where glibc is being used would show the same behaviour, but I would not guess that IDL would be compiled using glibc on other systems.

Most standard Unix (and MSDOS) libraries don't ever give back memory allocated by malloc to the system, and some do this only in very specific circumstances unlikely to occur in a complex system like IDL.

--
David Kastrup Phone: +49-234-700-5570
Email: dak@neuroinformatik.ruhr-uni-bochum.de Fax: +49-234-709-4209
Institut fi¿½r Neuroinformatik, Universitï¿½tsstr. 150, 44780 Bochum, Germany

---

Subject: Re: IDL memory question
Posted by Alex Schuster on Mon, 07 Sep 1998 07:00:00 GMT
View Forum Message <> Reply to Message

Eugenio Sansosti wrote:

> Suppose I define a very big array, for instance a=fltarr(8000,8000).
> When I finish using it, I can make memory free by assigning to it a
> scalar value (for example a=0.), as also suggested in the IDL
> documentation.

Oh, you call this very huge? :-)

> Even if this operation make the required memory free for my IDL
> application, it does not make memory free for other applicstions running
> onmy machine. That is, other machine users cannot use the memory I have
> allocated untill I exit IDL.
>

> Does any of you have a solution for that? Is it an IDL configuration
> problem or is there any IDL command I can use?

Sorry, there is no way. It seems this is a general problem for any C
progam which uses malloc() and free() to access heap memory.
The FAQ has an entry about this:
http://la.znet.com/~mgs/idl_faq.html#T27

Alex
--
Alex Schuster Wonko@weird.cologne.de PGP Key available
alex@pet.mpin-koeln.mpg.de
