Subject: Re: Clear everything in IDL?
Posted by Matthew J. Sheats on Mon, 22 Sep 1997 07:00:00 GMT
View Forum Message <> Reply to Message

Clemens v. Mann wrote:
>
> Hi
>
> the memory will be freed when you assign a new value.
> This makes only sense when the new value is very smaller.
>
> Example :
> hugeArr = bindgen(1000,1000)
>
> hugeArr = 0
>
> cvm

I've tried this.. unfortunately (In windows NT anyway.. I haven't tried else where) It does not seem to actually release the memory as such.

Internal to IDL, it may mark that region of memory as freed, but Windows never receives that memory back until IDL is closed. It would also appear that any additional memory allocated doesn't utilize this memory which idl has already allocated, but instead pulls more off of NT's heap.

It creates a pretty big problem for some programs I'm working on.

Maybe I'm doing something wrong.

Just thought I'd add my experience.

Matthew J. Sheats

Subject: Re: Clear everything in IDL?
Posted by Alex Schuster on Tue, 23 Sep 1997 07:00:00 GMT
View Forum Message <> Reply to Message

Matthew J. Sheats wrote:

> I've tried this.. unfortunately (In windows NT anyway.. I haven't
> tried else where) It does not seem to actually release the memory as
> such.
>

> Internal to IDL, it may mark that region of memory as freed, but
> Windows never receives that memory back until IDL is closed. It would
> also appear that any additional memory allocated doesn't utilize this
> memory which idl has already allocated, but instead pulls more off of
> NT's heap.

You're right! The IDL FAQ has a topic covering this. Have a look at
http://ww2.sd.cybernex.net/~mgs/idl_faq.html#T27

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

Subject: Re: Clear everything in IDL?
Posted by gurman on Tue, 23 Sep 1997 07:00:00 GMT
View Forum Message <> Reply to Message

In article <34272161.238A9AC2@pitt.edu>, sheats@lanl.gov wrote:

> Clemens v. Mann wrote:
>>
>> Hi
>>
>> the memory will be freed when you assign a new value.
>> This makes only sense when the new value is very smaller.
>>
>> Example :
>> hugeArr = bindgen(1000,1000)
>>
>> hugeArr = 0
>>
>> cvm
>
>
> I've tried this.. unfortunately (In windows NT anyway.. I haven't
> tried else where) It does not seem to actually release the memory as
> such.
>
> Internal to IDL, it may mark that region of memory as freed, but
> Windows never receives that memory back until IDL is closed. It would
> also appear that any additional memory allocated doesn't utilize this
> memory which idl has already allocated, but instead pulls more off of
> NT's heap.
>
> It creates a pretty big problem for some programs I'm working on.

>
> Maybe I'm doing something wrong.

Matthew -

Have you tried

```
IDL> help, /mem
IDL> tempArr = temporary(hugeArr)
IDL> help, /mem
```

?

--

Preferences folder.

Subject: Re: Clear everything in IDL?
Posted by Peter Mason on Wed, 24 Sep 1997 07:00:00 GMT
View Forum Message <> Reply to Message

On Tue, 23 Sep 1997, Alex Schuster wrote:
> Matthew J. Sheats wrote:
>> I've tried this.. unfortunately (In windows NT anyway.. I haven't
>> tried else where) It does not seem to actually release the memory as
>> such.
>> Internal to IDL, it may mark that region of memory as freed, but
>> Windows never receives that memory back until IDL is closed. It would
>> also appear that any additional memory allocated doesn't utilize this
>> memory which idl has already allocated, but instead pulls more off of
>> NT's heap.
>
> You're right! The IDL FAQ has a topic covering this. Have a look at
> http://ww2.sd.cybernex.net/~mgs/idl_faq.html#T27

This is a problem that Unix IDL users know and hate - the IDL session's "data space" never gets reduced in size. (It's an operating-system issue, not an IDL issue.)
Now try not to laugh, but if this is a major issue for you then you might be better off using Windows 95 instead of NT. Win95 appears to release memory back to the system as soon as it is "freed" in IDL. (I must say that I'm disappointed to hear that NT seems to have the same dog-with-a-bone problem as Unix.)

If you are swinging huge arrays around in IDL then it'll probably be worth your while to explore the way IDL uses memory "on the sly". This is easy under Win95. Start the System Monitor and get a "Memory Manager: Allocated Memory"

display going, and start an IDL session. Get yourself a chunky array, e.g., N=1024L*1024L*32L & B=BYTARR(N). Try out some stuff. e.g., Watch your computer go to lunch when you type B(*)=1B. Enjoy the German engineering of B=TEMPORARY(B)+1B. Look on in horror at B=TEMPORARY(B)+1. See if your computer is even capable of B(*)=1L. And try out some of the operations your program does on large arrays.

Peter Mason
CSIRO division of Exploration and Mining
P.O Box 136, North Ryde, NSW, 2113, Australia
E-Mail: p.mason@syd.dem.csiro.au Tel: +61 2 9490-8883 Fax: 9490-8960/8921
Web: http://www.syd.dem.csiro.au/research/MMTG/

Subject: Re: Clear everything in IDL?
Posted by Matthew J. Sheats on Wed, 24 Sep 1997 07:00:00 GMT
View Forum Message <> Reply to Message

Peter Mason wrote:
>

> This is a problem that Unix IDL users know and hate - the IDL session's "data
> space" never gets reduced in size. (It's an operating-system issue, not
> an IDL issue.)
> Now try not to laugh, but if this is a major issue for you then you might be
> better off using Windows 95 instead of NT. Win95 appears to release memory
> back to the system as soon as it is "freed" in IDL. (I must say that I'm
> disappointed to hear that NT seems to have the same dog-with-a-bone problem as
> Unix.)
>

There is a bigger issue here unfortunately. At this time I'm not so sure if it is an OS related problem. I can write programs dynamically allocating memory from the heap and releasing it back without a hitch in C++. So I think there is more going on. Perhaps (as the FAQ says) it is because they use malloc's etc.

If the program I am writing was just for my personal use, I would just run on 95 and be happy. Unfortunately, I'm building it to be used on any architecture that IDL supports. It would be nice to have predictable behavior across architectures and OS's.

So I guess I'll write for the lowest common denominator until some kind of resolution can be found.

And thanks to everyone participating in this discussion, it's been interesting.

Matthew J. Sheats

Subject: Re: Clear everything in IDL?
Posted by Peter Mason on Fri, 26 Sep 1997 07:00:00 GMT
View Forum Message <> Reply to Message

On Wed, 24 Sep 1997, Matthew J. Sheats wrote:
> There is a bigger issue here unfortunately. At this time I'm not so
> sure if it is an OS related problem. I can write programs dynamically
> allocating memory from the heap and releasing it back without a hitch in
> C++. So I think there is more going on. Perhaps (as the FAQ says) it
> is because they use malloc's etc.

Fraying the thread...
It looks like IDL 5 is using a 3rd-party memory-allocation library called "SmartHeap" (on Win95 at least). This purports to be far quicker and more efficient than the standard malloc library you get with a compiler.
Well, apparently not always. I'd guess that it must be a help for the countless reasonably-small allocations a typical IDL program does, but it appears to get in the way when really chunky allocations (in the megs) are involved.
With IDL5/Win95 I still see "freed" memory (*large* variables) getting returned to the system. However, really large allocations seem to take a lot longer than in IDL4 (which doesn't use SmartHeap, I presume), and the ceiling for allocations appears to be significantly lower. (Well, no, it *is* lower - I can definitely get away with more in IDL4.)

Peter Mason
(Mr. Subjunctive)