Subject: Re: Memory Cleanup-- Messy structure
Posted by rogass on Wed, 16 Jun 2010 21:16:31 GMT
View Forum Message <> Reply to Message

On 16 Jun., 19:50, Ed Hyer <ejh...@gmail.com> wrote:
> OK, here's the sequence:
>
> IDL> STRUCT1=CREATE_STRUCT("PTR1",PTR_NEW());
> IDL> STRUCTALL=REPLICATE(STRUCT1,N)
> IDL> FOR I=0,N-1 DO STRUCTALL.PTR1=PTR_NEW(ALLDATA[*,I])
>
> Then I do what I need to do, and I'm done with it, so:
>
> IDL> UNDEFINE, STRUCTALL
>
> Except this doesn't free the memory, according to HELP,/HEAP.
>
> Since I'm not a real programmer, I still consider pointers to be kinda
> voodoo. Can someone patiently explain how I manually clean up the
> memory in this case?
>
> Thanks,
>
> --Edward H.

Hm, what's going on here. If I would do this on my PC the following
happens:

```
IDL> help,/heap
Heap Variables:
   # Pointer: 0
   # Object : 0
IDL> n=10
IDL> s={ptr1:ptr_new()}
IDL> help,/heap
Heap Variables:
   # Pointer: 0
   # Object : 0
IDL> sall=REPLICATE(S,N)
IDL> help,/heap
Heap Variables:
   # Pointer: 0
   # Object : 0
IDL> FOR I=0,N-1 DO SALL[i].PTR1=PTR_NEW(dist(i+1))
% Compiled module: DIST.
IDL> help,/heap
Heap Variables:
   # Pointer: 10
```

```
   # Object : 0

<PtrHeapVar1>   FLOAT     = Array[1]
<PtrHeapVar2>   FLOAT     = Array[2, 2]
<PtrHeapVar3>   FLOAT     = Array[3, 3]
<PtrHeapVar4>   FLOAT     = Array[4, 4]
<PtrHeapVar5>   FLOAT     = Array[5, 5]
<PtrHeapVar6>   FLOAT     = Array[6, 6]
<PtrHeapVar7>   FLOAT     = Array[7, 7]
<PtrHeapVar8>   FLOAT     = Array[8, 8]
<PtrHeapVar9>   FLOAT     = Array[9, 9]
<PtrHeapVar10>  FLOAT     = Array[10, 10]
IDL> undefine,sall
IDL> help,/heap
Heap Variables:
   # Pointer: 0
   # Object : 0
```

That's why I very like David's routines - they do always their jobs :)

Your problem might be that you don't correctly reference your pointer array. You forgot:
STRUCTALL--->[I]<-----
So, maybe you allocate some memory each time in the loop, but then you always overwrite the pointer(s).

Regards

CR

Subject: Re: Memory Cleanup-- Messy structure
Posted by MarioIncandenza on Wed, 16 Jun 2010 21:53:27 GMT
View Forum Message <> Reply to Message

CR,

Thanks much. This turned out, of course, to be PEBKAC. UNDEFINE does in fact do its job, if you know how to use it. However, I did find out this one nugget: You can see that from the way the array is created and populated, it looks like it will have nested pointers. However, if you do it the other way:

```
IDL> STRUCT1=CREATE_STRUCT("PTR1",PTR_NEW(/ALLOC));
IDL> STRUCTALL=REPLICATE(STRUCT1,N)
IDL> FOR I=0,N-1 DO *(STRUCTALL[I].PTR1)=ALLDATA[*,I]
```

You will find that there is only one pointer, which after that operation points to the last data tested = ALLDATA[*,N-1].

Later on I will post the fancy new routine that led me to have to solve this problem!

Thanks,

--Edward H.

Subject: Re: Memory Cleanup-- Messy structure
Posted by rogass on Thu, 17 Jun 2010 08:46:06 GMT
View Forum Message <> Reply to Message

I am eager to know what is your way. And, I'm also only an ordinary scientist and not a programmer. That's why I like this newsgroup ;)

Regards
CR

Subject: Re: Memory Cleanup-- Messy structure
Posted by Craig Markwardt on Thu, 17 Jun 2010 15:45:33 GMT
View Forum Message <> Reply to Message

On Jun 16, 1:50 pm, Ed Hyer <ejh...@gmail.com> wrote:

> Since I'm not a real programmer, I still consider pointers to be kinda
> voodoo. Can someone patiently explain how I manually clean up the
> memory in this case?

I still think RSI/Kodak/ITT dropped the ball on the design of this. IDL is a high level language, not a low level language - "4th generation" right? IDL should be cleaning up messes like this, not us.

Craig

Subject: Re: Memory Cleanup-- Messy structure
Posted by penteado on Thu, 17 Jun 2010 16:43:39 GMT
View Forum Message <> Reply to Message

On Jun 17, 12:45 pm, Craig Markwardt <craig.markwa...@gmail.com> wrote:
> On Jun 16, 1:50 pm, Ed Hyer <ejh...@gmail.com> wrote:

>
>> Since I'm not a real programmer, I still consider pointers to be kinda
>> voodoo. Can someone patiently explain how I manually clean up the
>> memory in this case?
>
> I still think RSI/Kodak/ITT dropped the ball on the design of this.
> IDL is a high level language, not a low level language - "4th
> generation" right? IDL should be cleaning up messes like this, not
> us.

I think it is proper for the times when IDL originated, when it was
too expensive to do automatic garbage collection. IDL 8 has it (it can
be switched off), though one could argue that it should not have taken
so long.

Subject: Re: Memory Cleanup-- Messy structure
Posted by Craig Markwardt on Fri, 18 Jun 2010 00:10:37 GMT
View Forum Message <> Reply to Message

On Jun 17, 12:43 pm, pp <pp.pente...@gmail.com> wrote:
> On Jun 17, 12:45 pm, Craig Markwardt <craig.markwa...@gmail.com>
> wrote:
>
>> On Jun 16, 1:50 pm, Ed Hyer <ejh...@gmail.com> wrote:
>
>>> Since I'm not a real programmer, I still consider pointers to be kinda
>>> voodoo. Can someone patiently explain how I manually clean up the
>>> memory in this case?
>
>> I still think RSI/Kodak/ITT dropped the ball on the design of this.
>> IDL is a high level language, not a low level language - "4th
>> generation" right? IDL should be cleaning up messes like this, not
>> us.
>
> I think it is proper for the times when IDL originated, when it was
> too expensive to do automatic garbage collection. IDL 8 has it (it can
> be switched off), though one could argue that it should not have taken
> so long.

Uh, IDL "pointers" originated in IDL 5.0, ten years ago when all the
other scripting languages did automatic garbage collection.

I don't understand how it could be too expensive. The IDL interpreter
has precise knowledge and control of when variables go out of scope.
Dealing with cycles is its own problem, but basic garbage collection
should have been implemented from the start.

Now we are stuck with many years of maintaining meticulous pointer allocation/deallocation discipline until IDL 5-6-7 are completely obsolete.

Craig

Subject: Re: Memory Cleanup-- Messy structure
Posted by rogass on Fri, 18 Jun 2010 06:47:39 GMT
View Forum Message <> Reply to Message

> I think it is proper for the times when IDL originated, when it was
> too expensive to do automatic garbage collection. IDL 8 has it (it can
> be switched off), though one could argue that it should not have taken
> so long.

When will IDL 8 be released? At the moment, Python comes closer and closer...

Regards
CR
