Subject: System Resources ???
Posted by Jim Brown on Fri, 23 Feb 1996 08:00:00 GMT
View Forum Message <> Reply to Message

Hello all,

I have been digging through the IDL Manual to find some information on Memory Allocation, but I am having trouble finding the answer to my question:

I am a bit concerned with how IDL handles the allocation of system resources. Things evidently are not being freed up properly (or I do not know how to free up memory properly).

For example, when I run the following command:

```
IDL> arr = dblarr(144,73,12,20)
```

the memory appears to be allocated as expected:

Results from 'top' command:

```
  PID USERNAME PRI NICE  SIZE   RES STATE   TIME   WCPU    CPU COMMAND
25955 jtb       34    0   27M   22M sleep   0:04  4.04%  0.00% idl
```

Results from 'help, /memory':

```
IDL> help, /memory
heap memory in use:   20303069, calls to MALLOC:     162, FREE:        45
```

Now if I try to free up this space:

```
IDL> delvar, arr
```

'top' still reports SIZE=27M and RES=22M.

Another problem, is this 'delvar' command only works from the top-level routine. If I need to do some array processing, especially in some lower-level subroutines, is there some slick way to free up the system resources that IDL appears to be holding onto that is no longer needed? The only way I see that space is freed up is by running:

```
IDL> exit
```

Not quite what I want to do in the middle of a program!

I am getting pretty frustrated in that the whole scheme of memory allocation

in IDL appears to be quite inefficient. If anybody has any tips on how to work with large arrays and then freeing up their space when finished, I would appreciate the help. I have looked into the use of the 'temporary' call, but I think that just keeps extra memory from being allocated. I am not sure that addresses the issue of freeing up space that is no longer needed.

Thanks,

Jim
==========
James T. Brown
CIRES - Cooperative Institute for Research in Environmental Sciences
University of Colorado

email: jtb@cdc.noaa.gov

Subject: Re: System Resources ???
Posted by marq on Sat, 24 Feb 1996 08:00:00 GMT
View Forum Message <> Reply to Message

Jim Brown <jtb@cdc.noaa.gov> wrote
> I am a bit concerned with how IDL handles the allocation of
> system resources. Things evidently are not being freed up
> properly...

I think this question came up a lot of times - to cite from the FAQ:

The HTML version of the IDL FAQ is available at:

ftp://fermi.jhuapl.edu/www/s1r/idl/idl_faq/idl_faq.html

Ray Sterner sterner@tesla.jhuapl.edu
The Johns Hopkins University North latitude 39.16 degrees.
Applied Physics Laboratory West longitude 76.90 degrees.
Laurel, MD 20723-6099
WWW Home page: ftp://fermi.jhuapl.edu/www/s1r/people/res/res.html

<stuff deleted>

T27. Why is memory not released back to the operating system after an array is deleted?

By Eric Korpela of Berkeley

This is a result of IDL being written in C and using the C library functions (malloc and free) for memory allocation. In most C libraries, memory that is freed is NOT returned to the operating system. The C program retains this memory and will reuse it for future calls to malloc (assuming that the new allocation will fit in the freed block).

Another way of considering it is in terms of how memory allocation is done under UNIX. New memory is allocated using brk() or sbrk() which control the size of the data segment. These routines are called by malloc().

Suppose you allocate 3 1 MB regions of memory under C.

```
char *p1=(char *)malloc(3*1024*1024);
char *p2=(char *)malloc(3*1024*1024);
char *p3=(char *)malloc(3*1024*1024);
```

Here's what your data segment would look like assuming malloc had to call sbrk().

```
 ------------------------------------------------------------ ---
prev stuff | overhead | 3MB | overhead | 3MB | overhead | 3MB |
 ------------------------------------------------------------ ---
                ^           ^           ^   ^
                p1          p2          p3   end of segment.
```

Now we free(p1).

```
 ------------------------------------------------------------ ----
prev stuff | overhead | free | overhead | 3MB | overhead | 3MB |
 ------------------------------------------------------------ ----
                           ^           ^   ^
                           p2          p3   end of segment
```

Notice that the free memory is still in the data segment. If free had called brk to reduce the size of the segment, the 3MB pointed to my p3 would be outside the data segment! SIGSEGV city! If free had moved the allocated memory to lower addresses so the segment size could be reduced without losing data, then p2 and p3 would point to invalid addresses, and we'd be forced to use handles rather than pointers and call GetPointerFromHandle() every time we wanted to access the memory. Ick! Just like Windows!

Hope this helps...

Regards,

Chris Marquardt (marq@strat01.met.fu-berlin.de)

Subject: Re: System Resources ???
Posted by David Foster on Wed, 28 Feb 1996 08:00:00 GMT
View Forum Message <> Reply to Message

Jim Brown <jtb@cdc.noaa.gov> wrote:
>
> Hello all,
>
> I have been digging through the IDL Manual to find some information
> on Memory Allocation, but I am having trouble finding the
> answer to my question:
>
> I am a bit concerned with how IDL handles the allocation of
> system resources. Things evidently are not being freed up
> properly (or I do not know how to free up memory properly).
>

This has come up a number of times, and it might be answered
in detail in IDL's FAQ. But the basic gist is that the problem
is not within IDL but involves the way that the UNIX operating
system allocates memory to a program. Once the program allocates
memory, it won't give it back to the OS. Hopefully some people
will answer this again who can give you more details.

David Foster
UCSD Brain Image Analysis Lab
foster@bial1.ucsd.edu
