Subject: Re: IDL Memory Management
Posted by Craig Markwardt on Fri, 07 Jan 2000 08:00:00 GMT
View Forum Message <> Reply to Message

wbiagiot@suffolk.lib.ny.us writes:
> In article <onso0brt0c.fsf@cow.physics.wisc.edu>,
>   craigmnet@cow.physics.wisc.edu wrote:
>>
> If you do a lot of creating and destroying of IDL
>> variables, especially with big variables, you can end up with a lot of
>> unused *HOLES* in memory, and unfortunately holey memory can't be
>> returned to the system. Windows may or may not be different in this
>> respect.
>>
>
> Craig,
>
> Out of curiosity, are you saying that these memory "holes" are never
> reused by UNIX or IDL? (i.e. every time you enter an IDL subroutine,
> create an array, leave the subroutine (destroying the array), that it
> acts like a memory leak?)
>
> BTW, I am a Windows user.

The holes can be reused by IDL. The point however is that once the
memory has been allocated by IDL, it's very hard to unallocated it.
The original question involved a person who was working on very large
images but only returning a small subimage, and the person complained
that their machine began to swap after running the analysis. I
presume that the operations on the large images consumed a lot of
memory which was never returned to the system.

Here is an example of three commands to IDL which can show the
fragmentation problem. The horizontal bar is meant to represent the
linear memory of the IDL process.

```
a = findgen(5000)   ;; Allocates first available memory
|******************* a **************************|

b = findgen(300)    ;; Allocates next available memory
|******************* a **************************|*b*|

a = 0               ;; A large hole is left!
|------------------ Unused ------------------------|*b*|
```

In this case, the memory that "a" occupied can't be returned to the
system because "b" is still being used. IDL can still reuse that
unused block for other new variables, as long as it fits in the space.

In principle IDL could occasionally "compact" the memory space to remove the holes, but this would probably have a large impact on performance and also cause a lot of internal programming headaches.

Windows used to have a non-linear addressing model when physical system memory was scarce. Memory was allocated in chunks which could be moved around by the Windows as needed, thus alleviating the above fragmentation problem. I think however that more recent versions of Windows have accepted the linear "Unix" model of memory architecture that I have described.

Craig

--
-- --------------
Craig B. Markwardt, Ph.D.   EMAIL: craigmnet@cow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
-- --------------