Subject: Re: IDL - freeing up used memory?
Posted by David Fanning on Wed, 29 May 2013 17:20:46 GMT
View Forum Message <> Reply to Message

AMS writes:

> I'm running into a memory issue which I am not certain whether is related to my IDL code or the machine the code is running on itself, and am wondering whether someone might be able to help me out. To start with, I'm using IDL 7.1.1 on CentOS, invoking via command line.
>

I think your memory is getting fragmented. I would think about doing two things. First, I wouldn't set your unused variables to 0, as I think that promotes fragmentation. I would use the Coyote Library routine UNDEFINE to get rid of them:

   http://www.idlcoyote.com/programs/undefine.pro

But, maybe better than this, I would try (assuming they are always the same size) to reuse the memory you have already set aside for reading in the variables:

```
   FOR j=0,n DO ....
      IF N_Elements(inputVar) EQ 0 THEN $
         inputVar = ... ELSE $
         inputVar = Temporary(inputVar) * 0
         ReadU, lun, inputvar
```

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thue. ("Perhaps thou speakest truth.")

Subject: Re: IDL - freeing up used memory?
Posted by Andy Sayer on Wed, 29 May 2013 17:43:51 GMT
View Forum Message <> Reply to Message

Hi David,

Thanks for the tip--I'll try using undefine and see whether that helps. Unfortunately, the data volume in the files is quite variable so the arrays will be quite different sizes.

As an update, the same code runs without issue on IDL 7.1.1 on Mac OS 10.8. So perhaps it's in part related to CentOS?

Thanks,

Andy

On Wednesday, May 29, 2013 1:20:46 PM UTC-4, David Fanning wrote:
> AMS writes:
>
>
>
>
>
>> I'm running into a memory issue which I am not certain whether is related to my IDL code or the machine the code is running on itself, and am wondering whether someone might be able to help me out. To start with, I'm using IDL 7.1.1 on CentOS, invoking via command line.
>
>>
>
> I think your memory is getting fragmented. I would think about doing two
>
> things. First, I wouldn't set your unused variables to 0, as I think
>
> that promotes fragmentation. I would use the Coyote Library routine
>
> UNDEFINE to get rid of them:
>
>
>
>    http://www.idlcoyote.com/programs/undefine.pro
>
>
>
> But, maybe better than this, I would try (assuming they are always the
>
> same size) to reuse the memory you have already set aside for reading in
>
> the variables:
>
>
>
>    FOR j=0,n DO ....
>
>       IF N_Elements(inputVar) EQ 0 THEN $
>
>          inputVar = ... ELSE $
>
>          inputVar = Temporary(inputVar) * 0

>
>            ReadU, lun, inputvar
>
>
>
> Cheers,
>
>
>
> David
>
> --
>
> David Fanning, Ph.D.
>
> Fanning Software Consulting, Inc.
>
> Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
>
> Sepore ma de ni thue. ("Perhaps thou speakest truth.")

Subject: Re: IDL - freeing up used memory?
Posted by Andy Sayer on Fri, 31 May 2013 13:32:46 GMT
View Forum Message <> Reply to Message

As an update, in case anyone else has a similar issue in the future:

Using undefine helped somewhat (it got me further along before running into the issue).

However, and this may be something else I should have mentioned, I was using HDF5 files. Google searches suggest potential memory leaks with IDL and HDF5 (although I don't know how version/OS specific it is), e.g.:  https://groups.google.com/forum/#!msg/comp.lang.idl-pvwave/9QJ_tcK7E2k/CLVPmPNVXJwJ

I had previously been using h5f_close to close each HDF5 file after I was done with it. Adding an additional h5_close statement (after h5f_close) resulted in my memory problem vanishing. From the documentation I don't know whether calling both h5f_close and h5_close are required, or what. I'm just happy that the code is working now.

Andy

On Wednesday, May 29, 2013 1:43:51 PM UTC-4, AMS wrote:
> Hi David,
>
>
>
> Thanks for the tip--I'll try using undefine and see whether that helps. Unfortunately, the data

volume in the files is quite variable so the arrays will be quite different sizes.
>
>
>
> As an update, the same code runs without issue on IDL 7.1.1 on Mac OS 10.8. So perhaps it's in part related to CentOS?
>
>
>
> Thanks,
>
>
>
> Andy
>
>
>
> On Wednesday, May 29, 2013 1:20:46 PM UTC-4, David Fanning wrote:
>
>> AMS writes:
>
>>
>
>>
>
>>
>
>>
>
>>
>
>>> I'm running into a memory issue which I am not certain whether is related to my IDL code or the machine the code is running on itself, and am wondering whether someone might be able to help me out. To start with, I'm using IDL 7.1.1 on CentOS, invoking via command line.
>
>>
>
>>>
>
>>
>
>> I think your memory is getting fragmented. I would think about doing two
>
>>
>
>> things. First, I wouldn't set your unused variables to 0, as I think
>
>>

>
>> that promotes fragmentation. I would use the Coyote Library routine
>
>>
>
>> UNDEFINE to get rid of them:
>
>>
>
>>
>
>>
>
>>   http://www.idlcoyote.com/programs/undefine.pro
>
>>
>
>>
>
>>
>
>> But, maybe better than this, I would try (assuming they are always the
>
>>
>
>> same size) to reuse the memory you have already set aside for reading in
>
>>
>
>> the variables:
>
>>
>
>>
>
>>
>
>>   FOR j=0,n DO ....
>
>>
>
>>       IF N_Elements(inputVar) EQ 0 THEN $
>
>>
>
>>         inputVar = ... ELSE $
>
>>

```
>
>>          inputVar = Temporary(inputVar) * 0
>
>>
>
>>          ReadU, lun, inputvar
>
>>
>
>>
>
>>
>
>>  Cheers,
>
>>
>
>>
>
>>
>
>>  David
>
>>
>
>>  --
>
>>
>
>>  David Fanning, Ph.D.
>
>>
>
>>  Fanning Software Consulting, Inc.
>
>>
>
>>  Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
>
>>
>
>>  Sepore ma de ni thue. ("Perhaps thou speakest truth.")
```

Subject: Re: IDL - freeing up used memory?
Posted by ptomar2006 on Mon, 22 Dec 2014 13:05:08 GMT
View Forum Message <> Reply to Message

On Thursday, May 30, 2013 2:20:46 AM UTC+9, David Fanning wrote:
> AMS writes:
>
>
>> I'm running into a memory issue which I am not certain whether is related to my IDL code or the machine the code is running on itself, and am wondering whether someone might be able to help me out. To start with, I'm using IDL 7.1.1 on CentOS, invoking via command line.
>>
> I think your memory is getting fragmented. I would think about doing two
> things. First, I wouldn't set your unused variables to 0, as I think
> that promotes fragmentation. I would use the Coyote Library routine
> UNDEFINE to get rid of them:
>
>    http://www.idlcoyote.com/programs/undefine.pro
>
> But, maybe better than this, I would try (assuming they are always the
> same size) to reuse the memory you have already set aside for reading in
> the variables:
>

```
>    FOR j=0,n DO ....
>       IF N_Elements(inputVar) EQ 0 THEN $
>          inputVar = ... ELSE $
>          inputVar = Temporary(inputVar) * 0
>          ReadU, lun, inputvar
```

>
> Cheers,
>
> David
> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
> Sepore ma de ni thue. ("Perhaps thou speakest truth.")

Hi David,

Hi,

I have the same problem with IDL 7.0 using on fedora 20. I have 8 GB RAM.
For example, my idl is using max memory 1.6 GB...not more than that

```
IDL> b=bytarr(1000,1000,1600)
IDL> help,/mem
heap memory used: 1601945318, max: 1601945409, gets:  2884510, frees:  2883617
```

If I give larger array, showing error

```
IDL> b=bytarr(1000,1000,1700)
% Unable to allocate memory: to make array.
```

```
  Cannot allocate memory
% Execution halted at: $MAIN$
```

I do not know why my IDL is not using the available RAM memory.
Any help will be appriciated.

Thanks
Pankaj

Subject: Re: IDL - freeing up used memory?
Posted by Helder Marchetto on Mon, 22 Dec 2014 13:53:52 GMT
View Forum Message <> Reply to Message

On Monday, December 22, 2014 2:05:10 PM UTC+1, ptoma...@gmail.com wrote:
> On Thursday, May 30, 2013 2:20:46 AM UTC+9, David Fanning wrote:
>> AMS writes:
>>
>>
>>> I'm running into a memory issue which I am not certain whether is related to my IDL code or the machine the code is running on itself, and am wondering whether someone might be able to help me out. To start with, I'm using IDL 7.1.1 on CentOS, invoking via command line.
>>>
>> I think your memory is getting fragmented. I would think about doing two
>> things. First, I wouldn't set your unused variables to 0, as I think
>> that promotes fragmentation. I would use the Coyote Library routine
>> UNDEFINE to get rid of them:
>>
>>    http://www.idlcoyote.com/programs/undefine.pro
>>
>> But, maybe better than this, I would try (assuming they are always the
>> same size) to reuse the memory you have already set aside for reading in
>> the variables:
>>

```
>>    FOR j=0,n DO ....
>>       IF N_Elements(inputVar) EQ 0 THEN $
>>          inputVar = ... ELSE $
>>          inputVar = Temporary(inputVar) * 0
>>          ReadU, lun, inputvar
```

>>
>> Cheers,
>>
>> David
>> --
>> David Fanning, Ph.D.
>> Fanning Software Consulting, Inc.
>> Coyote's Guide to IDL Programming: http://www.idlcoyote.com/

```
>>  Sepore ma de ni thue. ("Perhaps thou speakest truth.")
>
>  Hi David,
>
>  Hi,
>
>  I have the same problem with IDL 7.0 using on fedora 20. I have 8 GB RAM.
>  For example, my idl is using max memory 1.6 GB...not more than that
>
>  IDL> b=bytarr(1000,1000,1600)
>  IDL> help,/mem
>  heap memory used: 1601945318, max: 1601945409, gets:  2884510, frees:  2883617
>
>  If I give larger array, showing error
>  IDL> b=bytarr(1000,1000,1700)
>  % Unable to allocate memory: to make array.
>    Cannot allocate memory
>  % Execution halted at: $MAIN$
>
>
>  I do not know why my IDL is not using the available RAM memory.
>  Any help will be appriciated.
>
>  Thanks
>  Pankaj
```

Just a guess: could it be that your IDL version is 32-bit?

Try

```
IDL> help, !version, /struct
** Structure !VERSION, 8 tags, length=104, data length=100:
   ARCH            STRING    'x86_64'
   OS              STRING    'Win32'
   OS_FAMILY       STRING    'Windows'
   OS_NAME         STRING    'Microsoft Windows'
   RELEASE         STRING    '8.4'
   BUILD_DATE      STRING    'Sep 27 2014'
   MEMORY_BITS     INT             64
   FILE_OFFSET_BITS
                   INT             64
```

Notice that according to the help:
"MEMORY_BITS
The number of bits used to address memory. Possible values are 32 or 64. The number of bits used to address memory places a theoretical upper limit on the amount of memory available to IDL."

Cheeers