
Subject: Releasing memory in IDL

Posted by [Jaime](#) on Wed, 23 Jan 2008 14:44:11 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi,

I am trying to run a function (clumpfind) several times with different parameters, however, after I call it once not enough memory is available. I checked with 'help, /memory' to find variables that I could destroy to release memory, without luck (see below).

Is this the expected behavior of IDL? how can I release memory from the main function?

Best,
Jaime

```
IDL> help, /memory
```

```
heap memory used: 1726888467, max: 1726888502, gets: 3486942, frees: 3486559
```

```
IDL> help
```

```
% At RUN2          15 /Volumes/negra/clfind-oph/run2/run2.pro
```

```
% $MAIN$
```

```
DIR      STRING   = '../filt-0.77-1.100'
```

```
I        INT      = 9
```

```
RMS      FLOAT    = 0.110000
```

```
STEP     FLOAT    = 1.10000
```

```
THRE     FLOAT    = 0.770000
```

```
Compiled Procedures:
```

```
  $MAIN$ CHECK_FITS CLFIND  CLPROPS  CLSTATS
```

```
DEFCLUMP DEFREG  DOSTATS  EXTAST
```

```
  FITSAXES  FXADDPAR  FXADDPAR_CONTPAR
```

```
FXADDPAR_CONTWARN  MK_HDR
```

```
  MRD_HREAD  READHD  REMCHAR  RUN2
```

```
SXDELPAR TESTBAD  WCSXY2SPH
```

```
  WCS_ROTATE  WRITEFITS  XY2AD  XYAD  ZPARCHECK
```

```
Compiled Functions:
```

```
  FXPAR  FXPOSIT  GETTOK
```

```
  HEADFITS  READFITS  READ_ASCII  REVERSE
```

```
SEARCH3D  SXPARG  TAG_EXIST  UNIQ
```

```
  VALID_NUM
```

Subject: Re: Releasing memory in IDL

Posted by [David Fanning](#) on Wed, 23 Jan 2008 21:48:30 GMT

[View Forum Message](#) <> [Reply to Message](#)

Jaime writes:

- > I dug into the code and found that these variables are defined in a
- > COMMON block. Why are they still using memory after the program
- > finished? can they be destroyed altogether (e.g. common_block=0)?

Oh, dear. It gets worse and worse. Who wrote that program anyway? I'm sure there must be a good reason for writing it this way. I can't find it in the usual places.

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Releasing memory in IDL
Posted by [Jaime](#) on Wed, 23 Jan 2008 22:05:43 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Dave,

The code can be downloaded from here: <http://www.ifa.hawaii.edu/~jpw/clfind/clfind.tar>
and the description of the code is in here:
<http://www.ifa.hawaii.edu/~jpw/page16/page4/page4.html>

It is a program to define structures (clumps) in a data cube (radio astronomy).

Best,
Jaime

- >
- > Oh, dear. It gets worse and worse. Who wrote that
- > program anyway? I'm sure there must be a good reason
- > for writing it this way. I can't find it in the usual
- > places.
- >
- > Cheers,
- >
- > David
- > --
- > David Fanning, Ph.D.

- > Fanning Software Consulting, Inc.
 - > Coyote's Guide to IDL Programming:<http://www.dfanning.com/>
 - > Sepore ma de ni thui. ("Perhaps thou speakest truth.")
-

Subject: Re: Releasing memory in IDL
Posted by [Jean H.](#) on Wed, 23 Jan 2008 22:11:38 GMT
[View Forum Message](#) <> [Reply to Message](#)

Jaime wrote:

- > I dug into the code and found that these variables are defined in a
- > COMMON block. Why are they still using memory after the program
- > finished? can they be destroyed altogether (e.g. common_block=0)?
- >
- > Best,
- > Jaime

the help file says:

[...]

Variables in a common statement have a global scope within procedures defining the same common block. Unlike local variables, variables in common blocks are not destroyed when a procedure is exited.

I guess you could write a procedure to erase these values... but as David pointed out, there is probably a reason why these values are saved in a common block!... you might be using them in a different function! If you are sure you a) don't need data from another function and b) this data will not be used in another function, then I guess you could simply remove the common block... so your data will "die" at the end of the function

Jean

Subject: Re: Releasing memory in IDL
Posted by [David Fanning](#) on Wed, 23 Jan 2008 22:40:02 GMT
[View Forum Message](#) <> [Reply to Message](#)

Jaime writes:

- > The code can be downloaded from here: <http://www.ifa.hawaii.edu/~jpw/clfind/clfind.tar>
- > and the description of the code is in here:
- > <http://www.ifa.hawaii.edu/~jpw/page16/page4/page4.html>

Humm. Well, about what I expected, to be honest with you. :-)

Here is what I would do. Write a new procedure, named Cleanup_Clumpfind, and declare the two common blocks

in that. Then, just run UNDEFINE on all the variables in the common blocks. You can run them 10 at a time with the new undefine.pro:

<http://www.dfanning.com/programs/undefine.pro>

Then, when you are finished with one analysis, and before you start the next, run your Cleanup_Clumpfile program. That should make things manageable.

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Seppure ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Releasing memory in IDL

Posted by [Jaime](#) on Thu, 24 Jan 2008 05:07:47 GMT

[View Forum Message](#) <> [Reply to Message](#)

Thank you very much! this does free the memory and is very easy to implement.

Regards,

Jaime

> Humm. Well, about what I expected, to be honest with you. :-)

>

> Here is what I would do. Write a new procedure, named
> Cleanup_Clumpfind, and declare the two common blocks
> in that. Then, just run UNDEFINE on all the variables
> in the common blocks. You can run them 10 at a time
> with the new undefine.pro:

>

> <http://www.dfanning.com/programs/undefine.pro>

>

> Then, when you are finished with one analysis, and before
> you start the next, run your Cleanup_Clumpfile program.
> That should make things manageable.

>

> Cheers,

>

> David

> --

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