Subject: Memory leak in IDL hdf5 library?
Posted by Eddie Schlafly on Tue, 18 Jan 2011 19:07:06 GMT
View Forum Message <> Reply to Message

I think I've found a memory leak in the IDL hdf5 library and I wanted to know if anyone knows what I'm doing wrong or what is going on. I've appended some code that leaks memory based on the IDL hdf5 example code.

I modified the example ex_create_hdf5 routine to write a structure containing an array of structures; if the number of elements in this array is greater than 1, I see a memory leak.

Running:
ex_create_hdf5 & for i=0l,10 do ex_read_hdf5
causes the amount of memory allocated to IDL according to "top" to increase by ~700 MB every time the line is run, despite the fact that the written file is only about 40 KB.

IDL's
help, /memory
does not change and gives no indication that hundreds of MB of memory are being used by IDL. Likewise,
help, /heap
shows no pointers or objects.

I am running:
help, !version, /st
```
** Structure !VERSION, 8 tags, length=104, data length=100:
   ARCH            STRING    'x86_64'
   OS              STRING    'linux'
   OS_FAMILY       STRING    'unix'
   OS_NAME         STRING    'linux'
   RELEASE         STRING    '7.1.1'
   BUILD_DATE      STRING    'Aug 21 2009'
   MEMORY_BITS     INT           64
   FILE_OFFSET_BITS
                   INT           64
```

Any hints as to what is going on?

Thanks for your help,

Eddie Schlafly

--

PRO ex_create_hdf5

```
  file = 'hdf5_test.h5'
  fid = H5F_CREATE(file)

  ;; create data
  data0 = { a:replicate({b:0},2) }
  data = replicate(data0, 10000)

  ;; get data type and space, needed to create the dataset
  datatype_id = H5T_IDL_CREATE(data)
  dataspace_id = H5S_CREATE_SIMPLE(size(data,/DIMENSIONS))

  ;; create dataset in the output file
  dataset_id = H5D_CREATE(fid,$
    'Sample data',datatype_id,dataspace_id)
  ;; write data to dataset
  H5D_WRITE,dataset_id,data

  ;; close all open identifiers
  H5D_CLOSE,dataset_id
  H5S_CLOSE,dataspace_id
  H5T_CLOSE,datatype_id
  H5F_CLOSE,fid

END

PRO ex_read_hdf5

  ; Open the HDF5 file.
  file = 'hdf5_test.h5'
  file_id = H5F_OPEN(file)

  ; Open the image dataset within the file.
  ; This is located within the /images group.
  ; We could also have used H5G_OPEN to open up the group first.
  dataset_id1 = H5D_OPEN(file_id, 'Sample data')

  ; Read in the actual image data.
  image = H5D_READ(dataset_id1)

  ; Close all our identifiers so we don't leak resources.
  H5D_CLOSE, dataset_id1
  H5F_CLOSE, file_id

END
```