Subject: Memory leak in STRMID

Posted by Heinz Stege on Wed, 14 Oct 2015 11:26:14 GMT

View Forum Message <> Reply to Message

Hi all,

I believe that there is a bug within IDL's strmid function. However
I'm using a 5 years old version of IDL. So this bug may be fixed in
the meantime.

There seems to be a memory leak in the strmid funcion. When you
calculate a substring c of a very long string b
   c=strmid(b,1000,10)
then the short string c (10 chars in this case) needs as much memory
as the long string b.

Here is the demo (and a workaround):

First I start a fresh IDL session an look for the memory consumption:

```
|IDL> help,/mem
|heap memory used:    797011, max:    800529, gets:    1137, frees:    291
```

About 1 MB overhead. Seems to be okay for running the workbench. Then
I create a huge string with 10 million chars:

```
|IDL> b=byte((randomu(seed,10000000,/long) mod 255)+1)
|IDL> help,/mem
|heap memory used:  10797258, max:  50797356, gets:    1152, frees:    304
|IDL> b=string(b)
|IDL> help,/mem
|heap memory used:  10797072, max:  20797211, gets:    1164, frees:    316
```

The string needs 10 MB memory. So far so good. Now I create a
substring with strmid:

```
|IDL> c=strmid(b,1000,10)
|IDL> help,/mem
|heap memory used:  20796961, max:  20797050, gets:    1176, frees:    327
```

Oops, the result string with 10 chars needs 10 MB too? Is it really 10
chars long?

```
|IDL> help,strlen(b)
|<Expression>   LONG      =    10000000
|IDL> help,strlen(c)
|<Expression>   LONG      =        10
|IDL> help,/mem
```

|heap memory used: 20796822, max: 20797026, gets: 1197, frees: 348

Yes it is. 10 MB for 10 chars! Can we make a copy of c, and how much
memory will it need?

|IDL> cc=c
|IDL> help,/mem
|heap memory used: 20796753, max: 20796833, gets: 1209, frees: 359

The copy of c needs only a few byts. That's fine. How can we release
the unjustified memory usage of c? Does a simple operation help?

|IDL> c+="
|IDL> help,/mem
|heap memory used: 10796631, max: 20796759, gets: 1221, frees: 371

Aah, it helps. This is a workaround.

|IDL> print,!version
|{ x86 Win32 Windows Microsoft Windows 8.0.1 Oct 5 2010   32   64}

And now it's up to someone of you, to check verion 8.5. Unfortunately
I don't have this version.

Cheers, Heinz