Subject: Re: dynamic memory in call_external
Posted by David Foster on Mon, 25 Jan 1999 08:00:00 GMT
View Forum Message <> Reply to Message

Vapuser wrote:

>
> I have someone in my office who wants to know:
>
> Is it advisable to create and destroy memory within a CALL_EXTERNAL
> routine? (that is, can one safely use malloc and free?) Or must one
> make the routine(s) in question LINKIMAGE routines and the idl memory
> management routines (IDL_MEMAlloc, IDL_MEMfree and IDL_GetScratch)
> available in that environment.

I have used [m|c]alloc() and free() in CALL_EXTERNAL modules with no problems, usually for relatively small chunks of memory. But of course this is only for variables within the C module; you can't allocate memory within the C module and pass it back to IDL.

Dave
--

```
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ ~~~~~~~
  David S. Foster        Univ. of California, San Diego
   Programmer/Analyst     Brain Image Analysis Laboratory
   foster@bial1.ucsd.edu  Department of Psychiatry
   (619) 622-5892        8950 Via La Jolla Drive, Suite 2240
                         La Jolla, CA  92037
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ ~~~~~~~
```

Subject: Re: dynamic memory in call_external
Posted by Vapuser on Tue, 26 Jan 1999 08:00:00 GMT
View Forum Message <> Reply to Message

David Foster <foster@bial1.ucsd.edu> writes:

Thanks to all that answered. I should've been a little more clear in the question, however. The variables that will be created in this person's routine will be local to the CALL_EXTERNAL routine and will be passed neither into nor out of the routine. Looks like the answer to the first question is 'yes' provided that the error handling always assures that the memory is returned in case of error. I suspected the answer to the second was 'no' and am confirmed in that answer.

WHD

> Vapuser wrote:
>>
>> I have someone in my office who wants to know:
>>
>> Is it advisable to create and destroy memory within a CALL_EXTERNAL
>> routine? (that is, can one safely use malloc and free?) Or must one
>> make the routine(s) in question LINKIMAGE routines and the idl memory
>> management routines (IDL_MEMAlloc, IDL_MEMfree and IDL_GetScratch)
>> available in that environment.
>
> I have used [m|c]alloc() and free() in CALL_EXTERNAL modules with no
> problems, usually for relatively small chunks of memory. But of course
> this is only for variables within the C module; you can't allocate
> memory within the C module and pass it back to IDL.
>
> Dave
> --
>
>   ~~~~~~ ~~~~~~
>   David S. Foster   Univ. of California, San Diego
>   Programmer/Analyst   Brain Image Analysis Laboratory
>   foster@bial1.ucsd.edu Department of Psychiatry
>   (619) 622-5892   8950 Via La Jolla Drive, Suite 2240
>   La Jolla, CA 92037
>   ~~~~~~ ~~~~~~

Subject: Re: dynamic memory in call_external
Posted by David Foster on Tue, 26 Jan 1999 08:00:00 GMT

View Forum Message <> Reply to Message

Mark Rivers wrote:
>
> In article <88lnirufne.fsf@catspaw.jpl.nasa.gov>, Vapuser <vapuser@catspaw.jpl.nasa.gov>
writes:
>>
>>
>> I have someone in my office who wants to know:
>>
>> Is it advisable to create and destroy memory within a CALL_EXTERNAL
>> routine? (that is, can one safely use malloc and free?) Or must one
>> make the routine(s) in question LINKIMAGE routines and the idl memory
>> management routines (IDL_MEMAlloc, IDL_MEMfree and IDL_GetScratch)
>> available in that environment.
>
> No, you should not create and destroy memory within CAll_EXTERNAL. The reason
> is that when you use CALL_EXTERNAL you are passed only the address of the data
> storage part of the IDL variable. You are not passed other important pieces of

> information for that variable, such as how big it is, what the data type is,
> etc. If you create and destroy memory you will only change the pointer, but
> not the other descriptive information. It might work OK if you are sure you
> won't change the size or type of the IDL variable, but I would not bet on it.
>

I think we should be clear what we are talking about. I believe the question was whether one can safely use malloc() and free() to allocate and free memory *within* a CALL_EXTERNAL module, and the answer to this is yes. But you cannot allocate memory for variables that you intend to pass back to IDL (the well-known rule that you have to allocate all arguments before passing them to CALL_EXTERNAL).

If we don't clarify this distinction I think people will get confused.

Dave
--

```
  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ ~~~~~~
   David S. Foster         Univ. of California, San Diego
    Programmer/Analyst     Brain Image Analysis Laboratory
    foster@bial1.ucsd.edu  Department of Psychiatry
    (619) 622-5892         8950 Via La Jolla Drive, Suite 2240
                           La Jolla, CA  92037
  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ ~~~~~~
```

Subject: Re: dynamic memory in call_external
Posted by rivers on Tue, 26 Jan 1999 08:00:00 GMT

View Forum Message <> Reply to Message

In article <88lnirufne.fsf@catspaw.jpl.nasa.gov>, Vapuser <vapuser@catspaw.jpl.nasa.gov> writes:
>
>
> I have someone in my office who wants to know:
>
> Is it advisable to create and destroy memory within a CALL_EXTERNAL
> routine? (that is, can one safely use malloc and free?) Or must one
> make the routine(s) in question LINKIMAGE routines and the idl memory
> management routines (IDL_MEMAlloc, IDL_MEMfree and IDL_GetScratch)
> available in that environment.

No, you should not create and destroy memory within CAll_EXTERNAL. The reason is that when you use CALL_EXTERNAL you are passed only the address of the data storage part of the IDL variable. You are not passed other important pieces of information for that variable, such as how big it is, what the data type is, etc. If you create and destroy memory you will only change the pointer, but

not the other descriptive information. It might work OK if you are sure you
won't change the size or type of the IDL variable, but I would not bet on it.

>   Secondly, can one create an array in IDL (in the interpreter) that
> is page aligned (i.e. as if one had done it using the 'valloc' version
> of malloc in C) which can then be passed down into the CALL_EXTERNAL
> routine.

I don't think so.

Mark Rivers   (773) 702-2279 (office)
CARS   (773) 702-9951 (secretary)
Univ. of Chicago   (773) 702-5454 (FAX)
5640 S. Ellis Ave.   (708) 922-0499 (home)
Chicago, IL 60637   rivers@cars.uchicago.edu (e-mail)

or:
Argonne National Laboratory   (630) 252-0422 (office)
Building 434A   (630) 252-0405 (lab)
9700 South Cass Avenue   (630) 252-1713 (beamline)
Argonne, IL 60439   (630) 252-0443 (FAX)

Subject: Re: dynamic memory in call_external
Posted by korpela on Tue, 26 Jan 1999 08:00:00 GMT

View Forum Message <> Reply to Message

In article <88lnirufne.fsf@catspaw.jpl.nasa.gov>,
Vapuser <vapuser@catspaw.jpl.nasa.gov> wrote:
>
> Is it advisable to create and destroy memory within a CALL_EXTERNAL
> routine? (that is, can one safely use malloc and free?)

These are actually two separate questions. The answer to the second is "yes," one can safely use malloc and free from a call_external routine. IDL uses malloc and free for its memory allocations (at least in the UNIX versions). The answer to the first question is "maybe not." Some of the RSI domeumentation claims that this can lead to more memory fragmentation than using IDL's allocation routines would. However I cannot easily imagine a case in which this would happen, unless IDL depends upon behavior of realloc that is not guaranteed to occur.

>   Secondly, can one create an array in IDL (in the interpreter) that
> is page aligned (i.e. as if one had done it using the 'valloc' version
> of malloc in C) which can then be passed down into the CALL_EXTERNAL
> routine.

Not that I am aware of.

Eric
--
Eric Korpela | An object at rest can never be
korpela@ssl.berkeley.edu | stopped.
<a href="http://sag-www.ssl.berkeley.edu/~korpela">Click for home page.</a>