Subject: Re: CALL_EXTERNAL and memory allocation
Posted by Mark Rivers on Sat, 09 Mar 2002 17:42:08 GMT
View Forum Message <> Reply to Message

lyubo <lzagorch@cs.wright.edu> wrote in message
news:a6d9nn$76k$1@mercury.wright.edu...
> I have to get back a string from C, and if I have a null string
> defined in IDL and passed to C it won't work, because there
> isn't any memory allocated to the string.
>
> In general, how can I allocate memory for a string in IDL?

I would recommend not passing strings to CALL_EXTERNAL, but rather pass a byte array, it's easier and more portable, since the IDL string structure definition changed between IDL 5.4 and 5.5.

In your wrapper routine.

```
b = bytarr(256)  ; Must dimension for maximum string length
s = call_external('my_dll', 'my_funct', b)
; The external C routine places the string in the bytarr 'b', with NULL
termination.
str = string(b)  ; Convert from bytarr to string
```

Mark Rivers

Subject: Re: CALL_EXTERNAL and memory allocation
Posted by Michael Zingale on Sun, 10 Mar 2002 21:42:19 GMT
View Forum Message <> Reply to Message

I typically do something like the following:

in IDL:

```
string = ' '

ierr = call_external('lib.so', 'read_string', string)
```

and in C:

```
IDL_STRING* string = (IDL_STRING *) argv[0];
```

then fill string.s with the string.

To do an array of strings, it is a little more compilicated. If you do

unklabels = strarr(num),

you must initialize each string with a UNIQUE value, otherwise, IDL
gives them the same memory address. I do something like this:

```
  unklabels = strarr(nvar)

; this is important -- each string must be initialized to a unique
; 'word', otherwise all the unklabels will share the same address in
; memory
    for i = 0, nvar-1 do begin
        unklabels[i] = string(i, format = '(i4)')
     endfor
```

Then I can fill this in IDL w/o problems.

Mike

lyubo wrote:

> I have to get back a string from C, and if I have a null string
> defined in IDL and passed to C it won't work, because there
> isn't any memory allocated to the string.
>
> In general, how can I allocate memory for a string in IDL?
>
>   lyubo
>
>
>
>
>
>
>

Subject: Re: CALL_EXTERNAL and memory allocation
Posted by Michael Zingale on Sun, 10 Mar 2002 23:43:18 GMT
View Forum Message <> Reply to Message

I should also point out that I have an example of reading a string from an HDF5 dataset into IDL via call_external at the bottom of the following page:

http://www.ucolick.org/~zingale/io_tutorial/

in the file idl_hdf5.tar

Mike

Michael Zingale wrote:

> I typically do something like the following:
>
>
> in IDL:
>
> string = ' '
>
> ierr = call_external('lib.so', 'read_string', string)
>
>
> and in C:
>
> IDL_STRING* string = (IDL_STRING *) argv[0];
>
> then fill string.s with the string.
>
>
> To do an array of strings, it is a little more compilicated. If you do
>
> unklabels = strarr(num),
>
> you must initialize each string with a UNIQUE value, otherwise, IDL
> gives them the same memory address. I do something like this:
>
> unklabels = strarr(nvar)
>
> ; this is important -- each string must be initialized to a unique
> ; 'word', otherwise all the unklabels will share the same address in
> ; memory
> for i = 0, nvar-1 do begin
> unklabels[i] = string(i, format = '(i4)')
> endfor
>
> Then I can fill this in IDL w/o problems.
>
> Mike

>
>
>
>
>
>
> lyubo wrote:
>
>> I have to get back a string from C, and if I have a null string
>> defined in IDL and passed to C it won't work, because there
>> isn't any memory allocated to the string.
>>
>> In general, how can I allocate memory for a string in IDL?
>>
>> lyubo
>>
>>
>>
>>
>>
>>
>>
>>
>
>

Subject: Re: CALL_EXTERNAL and memory allocation
Posted by James Kuyper on Mon, 11 Mar 2002 16:43:06 GMT
View Forum Message <> Reply to Message

Michael Zingale wrote:

> I typically do something like the following:
>
>
> in IDL:
>
> string = ' '
>
> ierr = call_external('lib.so', 'read_string', string)
>
>
> and in C:
>
> IDL_STRING* string = (IDL_STRING *) argv[0];
>
> then fill string.s with the string.

Are you referring to the 'argv' that appears as a parameter in a C
main()? That's not necessarily legal. You need to first make sure that:

strlen(argv[0]) > sizeof(IDL_STRING)

Otherwise, you many be overwriting memory that you don't have permission
to access.

Subject: Re: CALL_EXTERNAL and memory allocation
Posted by Michael Zingale on Mon, 11 Mar 2002 18:03:54 GMT
View Forum Message <> Reply to Message

yes, I forgot to mention that the string that I initialize in IDL is of
length greater than or equal to the length of the string that I am
reading in in the C program. This way I know the memory is allocated.
The example on the website that I pointed to does this.

Mike

James Kuyper wrote:

> Michael Zingale wrote:
>
>> I typically do something like the following:
>>
>>
>> in IDL:
>>
>> string = ' '
>>
>> ierr = call_external('lib.so', 'read_string', string)
>>
>>
>> and in C:
>>
>> IDL_STRING* string = (IDL_STRING *) argv[0];
>>
>> then fill string.s with the string.
>
>
> Are you referring to the 'argv' that appears as a parameter in a C
> main()? That's not necessarily legal. You need to first make sure that:
>
> strlen(argv[0]) > sizeof(IDL_STRING)
>
> Otherwise, you many be overwriting memory that you don't have permission
> to access.

>