Subject: Re: passing idl structures in call_external ?
Posted by hahn on Thu, 04 Apr 1996 08:00:00 GMT
View Forum Message <> Reply to Message

csoelle@msfd12.gwdg.de (Christian Soelle) wrote:

> As the subject already says, does anybody know how to pass idl-structures
> to a C-function using CALL_EXTERNAL. I couldn't find anything in the help
> nor in the example programs supplied.

> Did I miss something ? Is there anyone who can shed some light on this
> problem ?

See page 3-5 in IDL Advanced Development Guide, version 4.

Hope this helps
Norbert

Remember: IDL user's meeting on April 18th an 19th in Augsburg,
Germany. Have you registered?

> Regards,

>   Christian
> --
> ____
> | |
> | Christian Soeller   email: csoelle@msfd12.gwdg.de |
> | |
> | Max-Planck-Institut for Fluid Dynamics |
> | Bunsenstrasse 10 |
> | 37073 Goettingen / Germany |
> ____

Subject: Re: passing idl structures in call_external ?
Posted by David Foster on Thu, 04 Apr 1996 08:00:00 GMT
View Forum Message <> Reply to Message

csoelle@msfd12.gwdg.de (Christian Soelle) wrote:
>
>
> As the subject already says, does anybody know how to pass idl-structures
> to a C-function using CALL_EXTERNAL. I couldn't find anything in the help
> nor in the example programs supplied.

Most wouldn't consider this a problem, because an IDL structure

doesn't mean much in the context of a C function. The first
problem is how to declare the pointer argument in the C
function. Then you have to figure out how to reference the
structures using pointer offsets, and I'm not sure this would
work (the alignment might create problems). I hope someone
more familiar with this can address these problems.

I think you would be better off passing information to your
C routine using individual arguments.

Dave Foster
foster@bial1.ucsd.edu

Subject: Re: passing idl structures in call_external ?
Posted by rivers on Thu, 04 Apr 1996 08:00:00 GMT

View Forum Message <> Reply to Message

In article <CSOELLE.96Apr4151011@msfd12.gwdg.de>, csoelle@msfd12.gwdg.de (Christian Soelle) writes:
>
> As the subject already says, does anybody know how to pass idl-structures
> to a C-function using CALL_EXTERNAL. I couldn't find anything in the help
> nor in the example programs supplied.
>

The last time I looked, the means by which structures are passed was
intentionally not documented, presumably so that RSI would be free to change it
in the future.

However, I know by experience that IDL presently passes structures just like
you would expect, i.e. it passes the address of the start of the structure. All
structure elements except strings are contained in the structure itself, i.e.
the structure contains the value, not a pointer. Strings are different:
the structure contains either the descriptor or the address of the descriptor
(I forget).

I routinely pass structures to CALL_EXTERNAL, but I do so at my own risk,
since it is not guaranteed to be done the same way in future versions of IDL.

I have found that the structures will contain padding to keep the members
aligned on natural boundaries. The C compiler will normally do this on the
structures in your CALL_EXTERNAL code as well, so it has not been a problem.

Mark Rivers	(312) 702-2279 (office)
CARS	(312) 702-9951 (secretary)
Univ. of Chicago	(312) 702-5454 (FAX)

5640 S. Ellis Ave.   (708) 922-0499 (home)
Chicago, IL 60637   rivers@cars3.uchicago.edu (Internet)

Subject: Re: passing idl structures in call_external ?
Posted by Robert Cannon on Wed, 10 Apr 1996 07:00:00 GMT

View Forum Message <> Reply to Message

Christian Soelle wrote:
>
> As the subject already says, does anybody know how to pass idl-structures
> to a C-function using CALL_EXTERNAL. I couldn't find anything in the help
> nor in the example programs supplied.
>

I don't know anything official, but as the prefious reply said, they
generally turn out to be alligned the same way between c and IDL.
I use the routine below (structtoc) to produce a c header file to
declare the structure once I have defined it in IDL.

I hope it is of some use,
Robert

Neuroscience Research Group
School of Biologial Sciences
Southampton, UK   SO16 7PX
rcc@hera.neuro.soton.ac.uk

```
FUNCTION structname, a
   hhh = gethelp ('a')
   hh = hhh(0)
   fa = strpos (hh, '->')
   fb = strpos (hh, 'Arra')

   nn = strmid (hh, fa+2, fb  - (fa+2) )
   IF strlen (nn) EQ 2 THEN nn = 'annon'
   return, strlowcase (nn)
```

```
END


PRO structtoc, strin
                              ; given an idl structure str, write a c
                              ;header file declaring this struture
    str = strin

                              ; types, as returned by size
    types = ['undefined  ', $ ; 0
             'byte       ', $ ; 1
             'short int  ', $ ; 2
             'int        ', $ ; 3
             'float      ', $ ; 4
             'double     ', $ ; 5
             'complex    ', $ ; 6
             'string     ', $ ; 7
             'struct     ', $ ; 8
             'dcomplex   ']

    nt = n_tags (str)
    tn = strlowcase (tag_names (str))

    nel = n_elements (str)

    IF nel gt 1 THEN BEGIN
                              ; str is an array of structures
       subs = str(0)
       structtoc, subs
    END ELSE BEGIN

                                ; check to see if it contains any
structures
        FOR i = 0, nt-1 DO BEGIN
           sf = size (str.(i))
           sr = reverse (sf)

           IF sr(1) EQ 6 OR sr(1) EQ 7 OR sr(1) EQ 10 OR sr(1) EQ 0 THEN
BEGIN
               message, 'cant do this structure - dont like types'
           END

           IF sr(1) EQ 8 THEN BEGIN
               subs = str.(i)
               structtoc, subs
           END
       END
```

```
     stnm = structname (str)

     print, 'typedef struct '
     print, '{'

     FOR i = 0, nt-1 DO BEGIN
        sf = size (str.(i))
        sr = reverse (sf)
        vardec = types (sr(1))
        IF sr(1) EQ 8 THEN BEGIN
           ; get the name of the structure
           a = str.(i)
           vardec = structname (a)
        END


        IF sf(0) EQ 0 THEN BEGIN
           line = vardec + tn(i) + ';'
        END ELSE BEGIN
           IF sf(0) GT 1 THEN BEGIN
              print, '/* following is an idl multidimensional
array: '
              print, fix(sf(0)), ' dims: ', fix(sf(1:sf(0))), '*/'
           END

           line = vardec + tn(i) + $
             '[' + strtrim(string(sr(0)), 2) + ']' +  ';'
        END
        print, '    ' + line
     END

     print, '}  ' + stnm +  ';'
     print
     print
  END
END
```