
Subject: Call_external passing problem

Posted by [isoaga2](#) on Wed, 22 Oct 2003 05:42:12 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi everyone, recently i've been using the call_external function a bit to call fortran code through c wrappers and i thought i had it sorted however i can't seem to see my mistake in this one...

Here are the three pieces of code i'm using and the output i'm getting:

IDL CODE

```
-----
pro hyper2

a = -11.5d0
b = 20.5d0
c = 5.d0
z = double((cos(!pi/3.)))
re = 0.d0
im = 0.d0
out = call_external('/home/david/PhD/Fortran/hyper.so', 'hyper', $
    z, a, b, c, re, im)
stop
end
```

C WRAPPER

```
-----
#include <stdio.h>

void hyper_cw(int argc, void *argv[])
{
    extern void hyp_(); /* Fortran Routine */
    int *argc;
    double *z, *a, *b, *c, *re, *im;

    z = (double *) argv[0];
    a = (double *) argv[1];
    b = (double *) argv[2];
    c = (double *) argv[3];
    re = (double *) argv[4];
    im = (double *) argv[5];
    hyp_(z,a,b,c,re,im);
}
```

FORTRAN CODE (or at least the beginning of it)

```
-----  
subroutine hyp(z,a,b,c,re,im)  
  real*8  zero,one,two,half  
  parameter (zero=0.d0,one=1.d0,two=2.d0,half=0.5d0)  
  integer flag,flag2,neps  
  real*8  a,b,c,z,w,f,f1,f2,gamm,tol,test,pi,machep,re2,  
  #      alpha0,alpha1,rn,binom,eps,re,im,x1,x2,x3,x4,  
  #      coeff1,coeff2,coeff3,coeff4,temp1,temp2,term,  
  #      a1,b1,c1,a2,b2,c2,alpha2  
  logical fix  
  common /bcoeff/binom(5151)  
  
  write (6,*) z,a,b,c,re,im  
  write (6,*) zero,one,two,half
```

COMPILED AND LINKED WITH (under linux - Gentoo - PC)

```
-----  
g77 -c hyp.f  
gcc -c hyper_cw.c  
gcc -shared -o hyper_cw.so hyp.o hyper_cw.o -lg2c
```

Then when i run the idl program i get the following output:

```
IDL> hyper2  
 0.49999997 -11.5 20.5 0.49999997 20.5 -11.5  
0. 1. 2. 0.5
```

Which shows that the fortran subroutine is getting the first three arguments correctly, however the last three (c, re, im) are being mixed up somehow. Can anyone see my mistake, its driving me nutty...

Thanx.

Subject: Re: Call_external passing problem
Posted by [Rick Towler](#) on Wed, 22 Oct 2003 18:57:50 GMT
[View Forum Message](#) <> [Reply to Message](#)

"David Green" wrote...

```
> Hi everyone, recently i've been using the call_external function a bit  
> to call fortran code through c wrappers and i thought i had it sorted  
> however i can't seem to see my mistake in this one...
```

Have you tried using the AUTO_GLUE keyword to CALL_EXTERNAL?

Word on the street is that it makes life much easier. Since you are using gcc, you'll need to tweak your !MAKE_DLL environment vars a bit. All of

this is documented.

There is also Stein Vidar Hagfors Haugan's dlm form available at:
<http://www.astro.uio.no/~steinhh/idl/dlmform.html> which automagically writes
the C wrappers for F77 code if AUTO_GLUE isn't working for you.

Also, I don't use gcc to build on linux but you'll probably want to add
the -fPIC switch when compiling your wrapper.

-Rick

Subject: Re: Call_external passing problem
Posted by [isoaga2](#) on Thu, 23 Oct 2003 01:51:42 GMT
[View Forum Message](#) <> [Reply to Message](#)

"Rick Towler" <rtowler@u.washington.edu> wrote in message
news:<[bn6k2v\\$2bvs\\$1@nntp6.u.washington.edu](mailto:bn6k2v$2bvs$1@nntp6.u.washington.edu)>...
> "David Green" wrote...
>> Hi everyone, recently i've been using the call_external function a bit
>> to call fortran code through c wrappers and i thought i had it sorted
>> however i can't seem to see my mistake in this one...
>
> Have you tried using the AUTO_GLUE keyword to CALL_EXTERNAL?
>
> Word on the street is that it makes life much easier. Since you are using
> gcc, you'll need to tweak your !MAKE_DLL environment vars a bit. All of
> this is documented.
>
> There is also Stein Vidar Hagfors Haugan's dlm form available at:
> <http://www.astro.uio.no/~steinhh/idl/dlmform.html> which automagically writes
> the C wrappers for F77 code if AUTO_GLUE isn't working for you.
>
> Also, I don't use gcc to build on linux but you'll probably want to add
> the -fPIC switch when compiling your wrapper.
>
> -Rick

Well, do i feel stupid... Turns out the code i posted above is
correct, it was just my inexperience with vim that was the problem, i
was compiling the wrong piece of code :S

Oh, and thanx for the suggestions Rick, i'm now compiling and linking
with the -fPIC option and now i'm off to check out this AUTO_GLUE
stuff.
