Subject: call_external fortran with OS X...what compiler for f95?
Posted by henrygroe on Thu, 30 Oct 2003 19:39:26 GMT
View Forum Message <> Reply to Message

I'm just switching over to mac osx from linux. I've a bunch of old f77 and f95 code that I call from idl using call_external under linux and need to get it working on the mac.

I've figured out how to do this with g77 (see my example below), but I need to find an f95 compiler that will work for this. Does anyone know how (and if) you can do this with NAG's f95 or Absoft's f95 compiler? (or is there some other f95 compiler out there?)

Thanks!
-Henry

A truly dumb little example of a f77 call_external in mac osx:

On mac osx compile the following with:

```
g77 -fPIC -c osx_test1.f
g77 -bundle -flat_namespace -dynamic -lm -lc -undefined suppress -o \
                osx_test1.so osx_test1.o
```

(NOTE: ignore warnings from compiler)

in file "osx_test1.pro":

```
------------------
function osx_test1,a,b
 a = fix(a)
 b = fix(b)
 testvar = CALL_EXTERNAL( 'osx_test1.so','call_osx_test1__', $
                    a,b, /unload)
;not sure why the extra "_" needs to be added, but presumably could
fix
;by changing compiler flags
return,1
end
------------------
```

in file "osx_test1.f":

```
------------------
      FUNCTION call_osx_test1(ARGC, ARGV)
      IMPLICIT NONE
      INTEGER*4             ARGC    !Argument count
      INTEGER*4             ARGV(*) !Vector of pointers to
argments
      INTEGER               ARG_CNT
      INTEGER*4       call_osx_test1
      ARG_CNT = LOC(ARGC)
```

```
      IF(ARG_CNT .ne. 2)THEN
        WRITE(*,*)': Incorrect number of arguments'
         call_osx_test1 = -1
        RETURN
      ENDIF
      CALL osx_test1(
     &     %val(ARGV(1)), %val(ARGV(2))
     &     )
      call_osx_test1 = 1
      RETURN
      END


      SUBROUTINE osx_test1(a,b)
c$$$C  INTEGER*2 is equivalent to IDL's normal integer type
c$$$C  INTEGER*4 is equivalent to IDL's LONG integer type
c$$$C  LOGICAL*1 is equivalent to IDL's byte type
c$$$C  REAL*4 is equivalent to IDL's float type
c$$$C  REAL*8 is equivalent to IDL's DOUBLE type
      IMPLICIT NONE
      integer*2 a,b

      b = a

      return
      end
-------------------
```