
Subject: CALL_EXTERNAL failure.

Posted by Octavi Fors on Tue, 20 Jul 1999 07:00:00 GMT

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

```
<!doctype html public "-//w3c//dtd html 4.0 transitional//en">
```

```
<html>
```

Hi everybody,

<p>I'm trying to execute a FORTRAN subroutine (example1.for) with CALL_EXTERNAL

using a C wrapper function (example.c), but I'm having problems. I

guess they're related to

compiler option but I'm not sure.

<p>Here you have both C wrapper and FORTRAN subroutine code (they are simplistic
just

for test purpose):

```
<p>example.c (C wrapper function) -----
```

```
<p><font size=-1>#include &lt;stdio.h></font>
```

```
<p><font size=-1>int example(argc, argv)</font>
```

```
<br><font size=-1>int argc;</font>
```

```
<br><font size=-1>void *argv[];</font>
```

```
<br><font size=-1>{</font>
```

```
<br><font size=-1>void example1_();</font>
```

```
<p><font size=-1>int j;</font>
```

```
<br><font size=-1>double *x1,*y1,*x2,*y2;</font>
```

```
<br><font size=-1>double *rlimit1,*rlimit2,*var1,*var2;</font>
```

```
<br><font size=-1>int *n1,*n2,*nmatch;</font>
```

```
<br><font size=-1>int *match1,*match2;</font>
```

```
<p><font size=-1>n1=(int *) argv[0];</font>
```

```
<br><font size=-1>x1=(double *) argv[1];</font>
```

```
<br><font size=-1>y1=(double *) argv[2];</font>
```

```
<br><font size=-1>var1=(double *) argv[3];</font>
```

```
<br><font size=-1>rlimit1=(double *) argv[4];</font>
```

```
<br><font size=-1>n2=(int *) argv[5];</font>
```

```
<br><font size=-1>x2=(double *) argv[6];</font>
```

```
<br><font size=-1>y2=(double *) argv[7];</font>
```

```
<br><font size=-1>var2=(double *) argv[8];</font>
```

```
<br><font size=-1>rlimit2=(double *) argv[9];</font>
```

```
<p><font size=-1>example1_(n1,x1,y1,var1,rlimit1,n2,x2,y2,var2,rlimit2); </font>
```

```
<p><font size=-1>return(0);</font>
```

```
<br><font size=-1>}</font>
```

```
<br><font size=-1>----- </font>
```

```
<p>example1.for&nbsp; (FORTRAN subroutine) -----
```

```
<p><font size=-1>SUBROUTINE&nbsp;
```

```
<example1(N1,X1,Y1,VARXY1,RLIMIT1,N2,X2,Y2,VARXY2,RLIMIT2)</font >
```

```
<p><font size=-1> &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp; &nbsp; INTEGER*4 N1,N2</font>
```

```
<br><font size=-1> &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp; &nbsp; REAL*8 VARXY1,  
VARXY2, RLIMIT1, RLIMIT2</font>
```

```
<br><font size=-1> &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp; &nbsp; REAL*8
```

```
X1(N1),Y1(N1)</font>
```

```

<br><font size=-1> &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp; &nbsp; REAL*8
X2(N2),Y2(N2)</font>
<p><font size=-1> &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp; &nbsp; DO I=1,N1</font>
<br><font size=-1> &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp; &nbsp;&nbsp;&nbsp;&nbsp; PRINT*, X1(I)</font>
<br><font size=-1> &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp; &nbsp; ENDDO</font>
<p><font size=-1> &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp; &nbsp; RETURN</font>
<p> -----
<p>This is the compilation procedure, which I suspect is wrong:
<p><font size=-1>$ pwd</font>
<br><font size=-1>/home/octavi/idl</font>
<br><font size=-1>$ uname -a</font>
<br><font size=-1>SunOS mizar 5.5.1 Generic_103640-05 sun4u sparc
SUNW,Ultra-Enterprise</font>
<p><font size=-1>$ gcc -fPIC -c&nbsp; example.c</font>
<br><font size=-1>$ f77 -c -pic example1.for</font>
<br><font size=-1>$ /usr/ucb/ld -o example.so -z text -G example.o example1.o</font>
<p>Inside IDL I'm proceeding in the following way:
<p>a) first define input data
<br><font size=-1>IDL> n1=2L &amp; n2=2L</font>
<br><font size=-1>IDL> x1=[2.34d, 4.56d] &amp; y1=[2.5d,1.2d] &amp; x2=[5.44d,
4.86d] &amp; y2=[43.34d, 41.56d]</font>
<br><font size=-1>IDL> var1=23.4d &amp; var2=34.5d</font>
<br><font size=-1>IDL> rlimit1= 10.0d &amp; rlimit2=10.0d</font>
<p>b) call subroutine
<br><font size=-1>IDL> value = CALL_EXTERNAL('/u/imatge/ofors/idl/groth/example.so','exampl
e',
$</font>
<br><font size=-1> &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
n1, x1, y1, var1, rlimit1,n2, x2, y2, var2, rlimit2)</font>
<p>the result of last line is that IDL hangs up (idlde disappears) and
the following message
<br>appears in the shell:
<p><font size=-1>ld.so.1: /usr/local/rsi/idl_5/bin/bin.solaris2.sparc/idlde:</font>
<br><font size=-1>fatal: relocation error: symbol not found: __s_wsle_nv:
referenced in /home/octavi/idl/example.so</font>
<p>Well, any ideas are welcome.
<p>Thanks in advance,
<p>Octavi.</html>

```
