
Subject: dynamically linking a C file
Posted by [jeff](#) on Tue, 03 Sep 2002 17:07:30 GMT
[View Forum Message](#) <> [Reply to Message](#)

IDL VERSION 5.5
Linux Red Hat 7.3 (linux x86) linux kernel-version 2.4.18
Installation Numbers :5341-1 10244
UBC/TRIUMF PET PROGRAM

I am having trouble calling an external c program from linux.
We currently have a program that runs on an old solaris machine that I am trying to update so that it will work on the current version of linux.
Within the main code their is the command CALL_EXTERNAL wich does this.

```
stat = CALL_EXTERNAL('cti_matrix_io.so',  
    'read_cti_matrix',  
        file1,  
    fr, pl, gate, data, bed,  
    image1)
```

this piece of code calls the sparc dynamically linked executable on the old solaris machine.

For the linux machine I have had an extremely tuff time trying to find any information on the subject and have tried many things without any success.

I compile the c code with gcc version 2.96 using the command.
gcc -fPIC -g -c cti_matrix_io.c

I then link the file using;
gcc -shared -o cti_matrix_io.so cti_matrix_io.o -lc
which creates the file
cti_matrix_io.so: ELF 32-bit LSB shared object, Intel 80386, version 1 (SYSV), not stripped

Now with these steps I don't get any compile errors or can't read the file cti_matrix_io.so errors when the program runs, but what I do get is:

```
IDL> .com large  
% Compiled module: MAIN_EVENT.  
% Compiled module: WIDGET.  
% Compiled module: GAUSSFUNC.  
% Compiled module: DISPLAY_MAIN_HEADER.  
% Compiled module: DISPLAY_IMAGE_SUBHEADER.  
% Compiled module: DISPLAY_SCAN_SUBHEADER.  
% Compiled module: CREATE_MAIN_HEADER.  
% Compiled module: CREATE_IMAGE_SUBHEADER.  
% Compiled module: CREATE_SCAN_SUBHEADER.  
% Compiled module: START.
```

```
% Compiled module: RUN_PROGRAM.  
IDL> start  
% Compiled module: XMANAGER.  
% Compiled module: LOADCT.  
% Compiled module: FILEPATH.  
% Compiled module: PATH_SEP.  
% LOADCT: Loading table Rainbow18  
[3]+ Done          emacs large.pro  
Segmentation fault
```

and the program crashes. I've tried to place error statement within the c code but it doesn't seem as though the program even enters the actual file.

Any help on the subject would be greatly appreciated. Thanks
Jeff Englehart
