Subject: IDL call_external and code distributed across several .so libraries
Posted by Denis Barkats on Sun, 02 Oct 2005 23:28:06 GMT
View Forum Message <> Reply to Message

Hi, I am running IDL on a mac os X (10.3) and I was happy to have gotten the "call_external" function to work under IDL using the simple exemple described in Fanning's pages (http://www.dfanning.com/tips/calling_c.html). But I'd like to use call_external with pieces of codes distributed accross several libraries. Here is my exemple: I have two C codes, generic.c and utils.c(see below) and generic.c calls a function in utils.c.

generic.c:

```
 int PowerOf2()
 {
        int output;
        output=someFunction();
        return output;
 }
```

utils.c:

```
 int someFunction()
 {
         return 32;
 }
```

Here is how I compile them

```
gcc -c utils.c
gcc -c generic.c
ld generic.o -bundle -o generic.so
ld utils.o -bundle -o utils.so ( I get the following error here  ld:
Undefined symbols: _someFunction )
```

Now how do I modify call_external to call PowerOf2, since it now depends on code in both generic.so and utils.so?

```
  IDL>

 print,call_external('/Users/denis/idl/bicep/generic.so','Pow erOf2',/i_value)
```

Thank you

Denis

Subject: Re: IDL call_external and code distributed across several .so libraries
Posted by Antonio Santiago on Mon, 03 Oct 2005 12:15:21 GMT
View Forum Message <> Reply to Message

Denis Barkats wrote:
> Hi, I am running IDL on a mac os X (10.3) and I was happy to have
> gotten the "call_external" function to work under IDL using the simple
> exemple described in Fanning's pages
> (http://www.dfanning.com/tips/calling_c.html). But I'd like to use
> call_external with pieces of codes distributed accross several
> libraries. Here is my exemple: I have two C codes, generic.c and
> utils.c(see below) and generic.c calls a function in utils.c.
>
> generic.c:
>
> int PowerOf2()
> {
> int output;
> output=someFunction();
> return output;
> }
>
> utils.c:
>
> int someFunction()
> {
> return 32;
> }
>
> Here is how I compile them
>
> gcc -c utils.c
This step "transforms" the C file to an "intermediate" format utils.o

> gcc -c generic.c
This has the same for generic.o

> ld generic.o -bundle -o generic.so
Now you are going to link, and so that you need the two *.o files.
The linker want to resolve symbols. That is the name "someFunction"
inside utils.c is a symbol that must be resolved. You need to provide
the two files so that linker now it.

This explanation is pretty poor but more or less this is the problem :)

> ld utils.o -bundle -o utils.so (I get the following error here ld:
> Undefined symbols: _someFunction)
>

> Now how do I modify call_external to call PowerOf2, since it now
> depends on code in both generic.so and
> utils.so?
>
> IDL>
>
> print,call_external('/Users/denis/idl/bicep/generic.so','Pow erOf2',/i_value)
>
>
> Thank you
>
> Denis
>

--

Antonio Santiago Pï¿½rez
(email: santiago<<at>>grahi.upc.edu)
(www: http://www.grahi.upc.edu/santiago)
(www: http://asantiago.blogsite.org)

Grup de Recerca Aplicada en Hidrometeorologia (GRAHI)
Universitat Politï¿½cnica de Catalunya
Barcelona - SPAIN
---
http://www.grahi.upc.edu
