Subject: Re: [call_external] how to use it ?
Posted by Dominik[1] on Wed, 12 Dec 2001 17:12:18 GMT
View Forum Message <> Reply to Message

> 2nd: in my own example it seems that I have to use long
> interger in idle to be able to use (int) in the C code.
thats right, I used normal int in IDL and shorts in C (but running on a Windows system)

>
> By the way, does any one knows about calling an idl
> procedure on unix command line as any other unix exe ?

I think you have to create a *.sav-file and then you can call it (but not sure about it.

>
> I mean gently passing the args via command line ?
>
> Thanks
> --
> richard Hitier

Subject: Re: [call_external] how to use it ?
Posted by Joel Gales on Wed, 12 Dec 2001 19:38:20 GMT
View Forum Message <> Reply to Message

Richard Hitier wrote:

> Hi there,
>
> I've been trying to use this fonctionality for hours now,
> but... couldn't succeed :(
>
> 1st: the manual example doesn't work for me
> the array pointer doesn't point to the good values.
> (getting very high values ...)
>
> 2nd: in my own example it seems that I have to use long
> interger in idle to be able to use (int) in the C code.
>
> I'm trying to modify an array throug C routine, but
> couldn't:
> working with the array pointer doesn't seem to alter the
> idl array at all.

>
> I'm working on a Solaris station,
> using gcc compiler.
>
> There are the files:
>
> ----------test.c----------
> int function(int argc, void *argv[])
> {
> int s=0,*ip;
> int i;
> for (i=0, ip=(int*)argv[0]; i<*(int*)argv[1]; i++, ip++)
> {
> s+=*ip;
> printf("i:%d ip:%f \n",i,*ip);
> }
> return(s);
> }
> ----------
>
> ----------test.pro----------
> pro none, zize
> tab = indgen(zize)
> machin=10
> print, "d'abord"
> print, tab
> i=call_external('none.so', 'function', tab, n_elements(tab))
> print, "ensuite", i
> print, tab
> end
> ----------
>
> ----------Makefile----------
> .SUFFIXES:
> .SUFFIXES: .c .so
>
> .c.so:
> gcc -g -fPIC $< -shared -o $@
>
> ----------
>
> By the way, does any one knows about calling an idl
> procedure on unix command line as any other unix exe ?
>
> I mean gently passing the args via command line ?
>
> Thanks
> --

> richard Hitier

You need to declare *ip as short int in the C program because the
indgen() function
in IDL corresponds to 16 bit integers in C. Also to display the integer
array values,
you need "%d" rather than "%f" in the printf statement.

I also added a statement "*ip= -i" which sets the array values to their
negative on
return to IDL.

```
-----------------------------------------------------
int function(int argc, void *argv[])
{
      int s=0;
      short int *ip;  /* indgen() in IDL corresponds to 16bit integer
in C */
      int i;
      for (i=0, ip=(short int*)argv[0]; i<*(int*)argv[1]; i++, ip++)
      {
          s+=*ip;
          printf("i:%d ip:%d  \n",i,*ip);
    *ip = -i;
      }
      return(s);
}
-----------------------------------------------------
```

Joel Gales
SIMBIOS GSFC/NASA Code 970.2
Phone: (301) 286-1403
FAX: (301) 286-0268

Subject: Re: [call_external] how to use it ?
Posted by Richard Hitier on Thu, 13 Dec 2001 10:41:25 GMT
View Forum Message <> Reply to Message

Thank you for the answers,

at least I could get some nice results,

but it appears that I have to quit IDL interpreter as often
as possible to have my C routine changes understood.

Does any one knows something about this ?

This is a bit uncomfortable, and I still can't understand
why, but anyway I now know how to use call_ext.

Thanks again :)

--
richard Hitier

Subject: Re: [call_external] how to use it ?
Posted by Craig Markwardt on Thu, 13 Dec 2001 18:52:59 GMT
View Forum Message <> Reply to Message

Richard Hitier <hitier@cnrs-orleans.fr> writes:

> Thank you for the answers,
>
> at least I could get some nice results,
>
> but it appears that I have to quit IDL interpreter as often
> as possible to have my C routine changes understood.
>
> Does any one knows something about this ?
>
> This is a bit uncomfortable, and I still can't understand
> why, but anyway I now know how to use call_ext.

I think the problem is that you haven't unloaded the object file.
Once it's loaded into memory, it doesn't matter what you do to the
file on disk. .full_reset_session will unload everything, so you can
start fresh.

Craig

--
---------- ----------
Craig B. Markwardt, Ph.D. EMAIL: craigmnet@cow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
---------- ----------

Subject: Re: [call_external] how to use it ?
Posted by Stein Vidar Hagfors H[1] on Fri, 14 Dec 2001 19:53:26 GMT
View Forum Message <> Reply to Message

Craig Markwardt <craigmnet@cow.physics.wisc.edu> writes:

> Richard Hitier <hitier@cnrs-orleans.fr> writes:
>
>
>> Thank you for the answers,
>>
>> at least I could get some nice results,
>>
>> but it appears that I have to quit IDL interpreter as often
>> as possible to have my C routine changes understood.
>>
>> Does any one knows something about this ?
>>
>> This is a bit uncomfortable, and I still can't understand
>> why, but anyway I now know how to use call_ext.
>
> I think the problem is that you haven't unloaded the object file.
> Once it's loaded into memory, it doesn't matter what you do to the
> file on disk. .full_reset_session will unload everything, so you can
> start fresh.

The trouble is, you then start *really* fresh.. as in not having executed the startup file, etc.. So, if most of your overhead in "power cycling" IDL is in your IDL startup file, setting paths, looking up stuff, creating any special system variables etc etc, you won't gain all that much.... unless you use "state caching"... Let me dig a little bit here...ouch, cannot find those anymore.

The idea is to put in your IDL startup file statements which will detect the presence of a cache file (named after that particular machine only, of course, if you're in a networking environment), and skip the normal startup, doing only a "restore,'thismachine-save.dat"' command that sets everything back to normal.

If the relevant cache file is *not* present, then you do the usual stuff, and then make the savefile with
save,/all,filename='thismachine-save.dat'

Simple and effective..

--

---------- ----------
Stein Vidar Hagfors Haugan
ESA SOHO SOC/European Space Agency Science Operations Coordinator for SOHO

NASA Goddard Space Flight Center, Email: shaugan@esa.nascom.nasa.gov
Mail Code 682.3, Bld. 26, Room G-1, Tel.: 1-301-286-9028/240-354-6066

Greenbelt, Maryland 20771, USA. Fax: 1-301-286-0264
