
Subject: Talking with my C data

Posted by [Eric Williams](#) on Thu, 12 Nov 1998 08:00:00 GMT

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

Howdy,

This group has helped so much in the past I hope you all can help me prove to my bosses, once again, that IDL will do most anything.

I am collecting images with a CCD using C code I wrote with some supplied C libraries we purchased with the camera. Right now once the data is readout from the camera it is stored in a data stream in memory and is then put into a file using fwrite. I have a couple of questions:

I would like to write IDL code to manage the taking and displaying of the data. Should I use call_external or linkimage? Is there a speed difference or just a level of complexity difference?

Once the data is read out of the camera into memory can I use IDL to read the data from the memory or should I continue to have C write the data with fwrite and then can I have IDL read that data from the file created with fwrite? In either case I am not sure how to make IDL get the data.

I've got to believe this is done frequently and I will continue to look through the RSI documentation and David Fanning's great book, but at this point I haven't figured it all out.

Thanks for any help.

Eric

Subject: Re: Talking with my C data

Posted by [steinhh](#) on Sun, 15 Nov 1998 08:00:00 GMT

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

In article <MPG.10b5df3ff23b231698971e@news.frii.com>
davidf@dfanning.com (David Fanning) writes:

> I'm not so sure Call_External is any slower, although
> I would certainly defer to Stein Vidar's experience in
> the matter. I think of the difference between Call_External
> and LinkImage as being one of complexity. In fact, I
> like to think of Call_External as being "LinkImage Lite".

I agree that call_image itself is probably identical to a corresponding call to a linkimage'd routine. However,

I seldom have standalone `call_image` statements floating around my "main" IDL program, because they look ugly, and for portability/reuse purposes I find it better to plug those calls into wrapper routines. Hence a tiny overhead.

Just to clarify.

Stein Vidar

Subject: Re: Talking with my C data
Posted by [rmlongfield](#) on Mon, 16 Nov 1998 08:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

In article <364B0D74.A8A16470@astro.wesleyan.edu>, eric@astro.wesleyan.edu wrote:

> Howdy,
>
> I would like to write IDL code to manage the taking and displaying of
> the data. Should I use `call_external` or `linkimage`? Is there a speed
> difference or just a level of complexity difference?
> I've got to believe this is done frequently and I will continue to look
> through the RSI documentation and David Fanning's great book, but at
> this point I haven't figured it all out.
>
> Thanks for any help.
>
> Eric
>

Hi Eric and All, I have been working with `CALL_EXTERNAL` successfully for two or so months now. One thing that should be emphasized, that I missed initially, was that when you re-compile your C-programs, it is necessary to exit IDL. This is real annoying when trying to debug, but seems to speed things up once the program works. Also a debugging help is to have an independent C program for testing the C program, and then write another C program with the proper IDL connections. This way one knows where the problem is.

I have no problems reading data files and displaying them. But I am currently having real problems with a data file that is being written by a C-program called with `CALL_EXTERNAL`. I've got an independent C program which runs fine, but when I run it in IDL, something doesn't work. I haven't written the newsgroup about it because I have trouble describing what is going on, as it sometimes works, and sometimes doesn't. There is no information being exchanged with IDL, so it is not a parameter passing problem. (Please see past newsgroup discussion if you want to hear about

those problems) I have two data files with data. One set has data written as an array of bytes and the other written with an array of integers. The one with the integers does not get corrupted, but the one with the bytes does. It seems like the file is being overwritten somehow but not all the data is being erased when it is opened (fopen()). My independent C program is calling the same write module, so I think that it is not a C error. I'm going to try to write the byte file as integer and see if that does anything. Maybe linkimage is a solution but I'm just not ready for it yet. Maybe someone else has seen this problem. Any other questions, or sample code (not beautiful but functional), I'll gladly post on the newsgroup. Especially the passing of parameters and filenames, I think I can help.

Rose

-----== Posted via Deja News, The Discussion Network ==-----
<http://www.dejanews.com/> Search, Read, Discuss, or Start Your Own
