Subject: Re: Talking with my C data
Posted by steinhh on Fri, 13 Nov 1998 08:00:00 GMT
View Forum Message <> Reply to Message

In article <364B0D74.A8A16470@astro.wesleyan.edu>
Eric Williams <eric@astro.wesleyan.edu> writes:

> 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.

Elegant way of persuading the community to contribute :-)

> 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?

There probably is a tiny bit of speed difference (with call_external
probably being slower), but it all depends on exactly how you do
it. If you need a lot of parameter (type) checking etc, this is a
lot easier to do in an IDL wrapper around the call_external, rather
than inside a linkimage routine, but must necessarily slow things
down a little. If you need to do one call to grab each frame (at
video rates or higher), this could have implications, but if you
grab a sequence at a time, should be negligible.

> 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.

The crucial thing here is how you control the size of your data
cubes, and exactly how the data is acquired. If the data is coming
in real-time, at a fixed (or externally controlled) frame rate not
controlled by your program (i.e. not polled, but using
interrupts/signals in some way), you *may* come to the final
conclusion that this is one of the things IDL won't do! It depends
on which interrupts/signals are used, and how program flow is
supposed to be controlled.

If you're setting up an experiment that grabs frames for a given

time (fixed buffer size) and *then* writes the whole thing to
file and shuts down, it should be easy to do.

I'd recommend to start with call_external with small IDL wrapper
functions, and put off writing a linkimage/dlm thing until you know
the stuff will work.

The trick is to create the data space in the IDL wrapper and pass
it along to the call_external routine, which moves the data into
that space.

If you do the upgrade to linkimage/dlm routines, you can use
callable IDL routines (your program is linked into IDL anyway, so it
just makes sense that it can call IDL internals) to generate the
space (as a temporary variable, or to import an array into a
temporary variable) at any time you like.

> 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.

I don't know how frequently this is done, but it's definitely a
great feature. IMO, the feature is suffering a bit from being to
well hidden away in the external development guide.

Regards,

Stein Vidar