Subject: Re: Calling IDL from Fortran called by IDL
Posted by Craig Markwardt on Wed, 28 Nov 2001 02:38:29 GMT
View Forum Message <> Reply to Message

"Mark Rivers" <rivers@cars.uchicago.edu> writes:

> Kevin A. Park <kpark@prism-cs.com> wrote in message
> news:3c02c150$0$35565$272ea4a1@news.execpc.com...
>> Hi,
>>
>> I have an IDL GUI which sits on top of a calculation engine which
>> is written in Fortran 90. Currently IDL accesses the calculation engine
>> by calling C wrapper functions via CALL_EXTERNAL. These C wrappers then
>> call Fortran routines. The system runs both on Solaris and Windows
>> platforms. I am currently using IDL 5.4, but will upgrade to IDL 5.5
>> soon.
>>
>> Some of the calculations in the Fortran take a long time, so what I
>> would like to do is have IDL create a progress bar which can be updated
>> from the Fortran. Having waded through the IDL External Development
>> Guide, I have a few questions.
>
...
> - The external C routine returns to IDL immediately, leaving the new
> thread running to do the time-intensive work. The new thread
> writes progress information to the IDL variable, which the
> IDL code is peridically checking and using to update a progress
> meter, display new results or whatever.
...

Oooh, but be careful, because if you allow your IDL variable to be
released, say by reassigning it, then your external C routine will
probably end up overwriting some part of memory you didn't want it to.

I would say that the easy compromise is to declare a system variable,
because I believe those are fixed in memory (ie, can't be
unallocated). System variables can be declared via IDL commands, or
within your DLM.

Also, one always needs to worry that things like malloc() and printf()
may not be threadsafe on one's platform. If your work thread and the
IDL thread clash, then *pow*, that will hurt.

Craig

--
---------- ----------
Craig B. Markwardt, Ph.D. EMAIL: craigmnet@cow.physics.wisc.edu