Subject: "Share data with external processes"
Posted by Ken Stone on Thu, 16 Jan 1997 08:00:00 GMT
View Forum Message <> Reply to Message

Hello,

Does anyone know of a fast way to transfer a large array (say fltarr(2000)) from another process? My first thought was to write the data to a file, and then read it back into IDL. But time is of a critical nature in this case. Then I found CALL_EXTERNAL, but it seems to only return a scalar (Is that true?). LINKIMAGE now looks like the way to go. Has anyone had any luck using LINKIMAGE to pass around arrays like this? What _is_ the best way to "share data with external processes"? (I put that in quotes because it's in the RSI advertisement for IDL. :-))

Thanks,

Ken

Subject: Re: "Share data with external processes"
Posted by davidf on Thu, 16 Jan 1997 08:00:00 GMT
View Forum Message <> Reply to Message

Ken Stone <k.a.stone@larc.nasa.gov> writes:

> Does anyone know of a fast way to transfer a large array (say
> fltarr(2000)) from another process? My first thought was to write the
> data to a file, and then read it back into IDL. But time is of
> a critical nature in this case. Then I found CALL_EXTERNAL, but it
> seems to only return a scalar (Is that true?).

Well, this is *literally* true, but it's not what you mean. CALL_EXTERNAL accepts parameters just like any other IDL procedure or function and these parameters can be passed by reference or by value to the C program. Your large array will be one of these parameters. You can think of it as sort of an output parameter to a normal IDL procedure.

The only real caveat to using CALL_EXTERNAL to pass data is that the variable must be created and sized on the IDL side. You can fill it on the C side, but don't change its data type or size. Your code will look something like this:

```
dataArray = FLTARR(2000)
junk = CALL_ETERNAL('fill_it.so', 'fillarray', dataArray)
PLOT, dataArray
```

The first two parameters to CALL_EXTERNAL are the name of the
shared object library you have to create and the entry point name
into that library (i.e, the function in the library you are calling).

The only difficult part of using CALL_EXTERNAL (and it is not
always difficult) is getting the shared object module made
correctly. This is compiler and operating system specific. RSI
provides lots of example Make files to help with this, but if
you have a wierd compiler you are sometimes on your own.

> LINKIMAGE now looks
> like the way to go. Has anyone had any luck using LINKIMAGE to pass
> around arrays like this? What _is_ the best way to "share data with
> external processes"? (I put that in quotes because it's in the RSI
> advertisement for IDL. :-))

LINKIMAGE is harder to use than CALL_EXTERNAL and requires that
you know significantly more about how IDL works internally. I would
say you pretty much have to be an expert C programmer to work with
LINKIMAGE, while even someone like me can usually get CALL_EXTERNAL
to work. ;-)

Good luck!

David

David Fanning, Ph.D.
Fanning Software Consulting
2642 Bradbury Court, Fort Collins, CO 80521
Phone: 970-221-0438 Fax: 970-221-4762
E-Mail: davidf@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com

Subject: Re: "Share data with external processes"
Posted by steinhh on Thu, 16 Jan 1997 08:00:00 GMT
View Forum Message <> Reply to Message

In article <32DE6333.63EA@larc.nasa.gov>, Ken Stone <k.a.stone@larc.nasa.gov> writes:
|> Hello,
|>
|> Does anyone know of a fast way to transfer a large array (say
|> fltarr(2000)) from another process? My first thought was to write the
|> data to a file, and then read it back into IDL. But time is of
|> a critical nature in this case. Then I found CALL_EXTERNAL, but it

|> seems to only return a scalar (Is that true?). LINKIMAGE now looks
|> like the way to go. Has anyone had any luck using LINKIMAGE to pass
|> around arrays like this? What _is_ the best way to "share data with
|> external processes"? (I put that in quotes because it's in the RSI
|> advertisement for IDL. :-))
|>
|> Thanks,
|>
|> Ken
|>

As long as you allocate the arrays to store the result before the call,
you can use call_external to transfer large amounts of data
back and forth as well, i.e.:

```
transfer = fltarr(2000)
dummy = call_external(....,...,transfer)
plot,transfer
```

You should perhaps take a look at the IDL RPC (remote procedure call)
library as well - one advantage is that you can actually run your
external routines on a different machine, so you could in effect have
parallel processing as well! I've used this once to make IDL (run as
an RPC server process) visualize data while calculations were made by
a program on another machine.

I believe you can have IDL running as either a master or slave process,
though I've never tried the former.

Stein Vidar

Subject: Re: "Share data with external processes"
Posted by djackson on Mon, 20 Jan 1997 08:00:00 GMT
View Forum Message <> Reply to Message

> Ken Stone <k.a.stone@larc.nasa.gov> writes:
>
>>   Does anyone know of a fast way to transfer a large array (say
>> fltarr(2000)) from another process? My first thought was to write the
>> data to a file, and then read it back into IDL. But time is of
>> a critical nature in this case. Then I found CALL_EXTERNAL, but it
>> seems to only return a scalar (Is that true?).

If you're doing this on a Unix system, I have some experience I might
share. In one of my programs, I use 'spawn' to start the other process
with the 'unit=' option (and '/noshell' might be useful, too) which then
works as a bidirectional pipe. You can write to it and read from it, just

as if you were at the command line.

I really can't comment on whether this would be faster than other methods,
or if it even could work in your situation, but I thought I'd drop it into
the hat!

Cheers,
-Dick

Dick Jackson djackson@ibd.nrc.ca Institute for Biodiagnostics
Opinions are mine alone. National Research Council Canada, Winnipeg
"And I told him my dream was to live for all time
In some perfect refrain, like the man who wrote 'Danny Boy'."
- Joe Jackson, from the album _Night_Music_, 1994.
