

Hello -

who is experienced in running IDL with C shared objects?

I'm trying to make an IDL GUI for a rather complex C batch program. The effects are rather strange: sometimes it works perfectly, but mostly it doesn't: suddenly on the C-side strange and wrong number appear in calculations leading to crashes. It seems like something is initialized or dereferenced in a wrong way. However the behaviour is deterministic: a small change in the IDL program like declaring a new variable anywhere causes the crash, after undoing the change it works well again. Maybe a memory conflict ?

The C-program alone in batch mode runs reliably. It does a lot of mallocs but never frees any memory (because it is batch). IDL communicates via the CALL_EXTERNAL function. I'm rather sure that I have checked the variables on both sides of the fence are really of same type. (However I'm a poor C-programmer...) I'm using IDL 5.3 on a Linux system and the GNU C-compiler.

Now the questions:

- Is it possible that IDL overwrites or frees memory allocated by the C shared object ? Is there a general way to prevent it from doing so ?
- What is the effect of the /CDECL keyword to CALL_EXTERNAL ? I tried with and without but no success.
- Is it possible that the C program "forgets" something between the IDL CALL_EXTERNALs ?

(important:)

- Is there an opinion whether this problem can be solved in principle and within finite time ?!

(has nothing to do with the problem but I would like to know:)

- How can I return an array via CALL_EXTERNAL or have I always to loop over calls returning scalars ? The EZCA library (channel access to EPICS control system) manages to return arrays, but I couldn't figure out how.

Thanks for any help.

Andreas

--

Dr. Andreas Streun
Paul Scherrer Institute, SLS-BD, WSLA/106
CH-5232 Villigen PSI, Switzerland
Tel/Fax: ++41 (56) 310 3688/4528
