Subject: Re: Multitasking??
Posted by JD Smith on Mon, 22 Nov 2004 16:55:53 GMT
View Forum Message <> Reply to Message

On Sun, 21 Nov 2004 08:53:48 -0800, Paul wrote:

> Ken Mankoff <mankoff@yahoo.com> wrote in message
news:<Pine.OSX.4.61.0411191245570.26987@gouda.local>...
>> On Fri, 19 Nov 2004, Henry Roe wrote:
>>> Just an idle under-caffeinated thought: Is there a simple way for
>>> two IDL processes to communicate? (besides writing to a disk on
>>> file) If so, then the two GUI's could actually be separate IDL
>>> processes.
>>
>> You can use the SOCKET command for communication w/o files.
>>
>> -k.
> In addition, you can use IDL's shared memory (shmmap) for this. I
> have done this for camera control where one process presents a GUI and
> the 2nd process handles the setting of a buffer, acquiring data and
> rapid display. This way, you don't loose focus on your main GUI. I
> suspect your subGUI could be running in process two in a similar way
> my 'slave' process was running (no GUI).
>
> Basically, what you do is set up a structure variable that contains
> data fields that need to be shared between processes. I then created
> a main procedure that simply spawned mainGUI to run under IDL's
> runtime engine (idlrt). From mainGUI, I had a button that started the
> data acquistion. In this button's event handler, I spawned off the
> 2nd process also under the idlrt engine. At this point you now have
> three processes going - main, mainGUI and in your case, subGUI.
>
> There is a very simple example on RSI's user contrib site that I put
> together to prototype the final solution described. See:
> http://www.rsinc.com/codebank/search.asp?FID=259
>
> If you choose to try the above and end up having questions, do not
> hesitate to contact us. Thanks and good luck.

Is there an advantage to using a second IDL process for the data
acquisition? I would think that since you've probably had to write
the camera (or whatever I/O) interface code in C, you might as well
have a data capture tool that runs stand-alone, and can communicate
with an IDL GUI process via shared memory (e.g. an SHMVAR variable).
A socket could also be used, but shared memory is probably the fastest
way to pull any amount of data in.

JD