Subject: timers without widgets
Posted by Benjamin Hornberger on Fri, 22 Apr 2005 16:11:24 GMT
View Forum Message <> Reply to Message

Hi all,

does anyone know if it is possible to have timed events without using widgets? For some instrument control, we would like object methods "triggered" at specified time intervals. Currently, we use an invisible widget, but we would like to run this remotely on a Linux machine without X connection, and even in the background (start with "at" in Linux, then log out). I know about the X virtual frame buffer as discussed a while ago on this group, but I was wondering if there is an alternative.

Thanks,
Benjamin

Subject: Re: timers without widgets
Posted by Rick Towler on Fri, 22 Apr 2005 23:08:25 GMT
View Forum Message <> Reply to Message

Benjamin Hornberger wrote:
> Rick Towler wrote:
>
>>
>> I understand that 95% of events are generated by a GUI but it would be
>> nice to have some sort of mechanism to handle the other 5%. Timers
>> and socket I/O come to mind.
>
>
> Socket I/O is exactly our problem. We use a timer to continuously
> (actually frequently rather than continuously) read from a socket to
> implement a listening socket. But there are many problems with that,
> like IDL not reacting while it tries to read (looks like crashed).

Does SOCKET not provide for non-blocking reads? I haven't really used it... That's too bad. I should add that to my list of feature requests for SOCKET.

> I guess IDL is just made for data analysis and not instrument control
> which involves interaction with device drivers and the OS on a deeper
> level :-(.

Yeah, I guess so. But that is too bad.

I had a similar project and had to use MATLAB. IDL's SOCKET doesn't

support UDP and even if it did the blocking would have killed me. I used a free TCP/UDP library for MATLAB which doesn't provide for callbacks so you have to poll, but it doesn't block and MATLAB's timer is better.

-Rick

Subject: Re: timers without widgets
Posted by Y.T. on Sat, 23 Apr 2005 02:25:18 GMT
View Forum Message <> Reply to Message

Uh -- in my version of IDL, at least, the help for "socket" explicitly states that "socket" opens a client-side socket. Opening server-side sockets is quite a different beast.

That said, there's fine bits of software out there to listen to ports (I'd write it in TCL if I had to) but I have to wonder whether you may be going through too many layers here -- if you are trying to do instrument control, then what is your hardware interface? IEEE4888? What kinds hardware do you have in your computer to facilitate the communication? What kinds of drivers do you use? Can those drivers truly, honestly deliver their data to ports and only to ports? Can they truly not write, say, into a named pipe (from whence you could read it easily from IDL)?

Just tossing out some ideas here...

cordially

Y.T.

--
Remove YourClothes before you email me.

Subject: Re: timers without widgets
Posted by Craig Markwardt on Sat, 23 Apr 2005 15:03:27 GMT
View Forum Message <> Reply to Message

Benjamin Hornberger <benjamin.hornberger@stonybrook.edu> writes:

> Rick Towler wrote:

>> I understand that 95% of events are generated by a GUI but it would
>> be nice to have some sort of mechanism to handle the other 5%.
>> Timers and socket I/O come to mind.
>
> Socket I/O is exactly our problem. We use a timer to continuously
> (actually frequently rather than continuously) read from a socket to
> implement a listening socket. But there are many problems with that,
> like IDL not reacting while it tries to read (looks like crashed).

Can't you set READ_TIMEOUT and WRITE_TIMEOUT to some very small
number? (perhaps even zero?)

It's a pity that IDL doesn't have a select() function for I/O
multiplexing, but using a small timeout value should at least allow
you to poll for data.

Craig

>
> I guess IDL is just made for data analysis and not instrument control
> which involves interaction with device drivers and the OS on a deeper
> level :-(.
>
> Benjamin
>

--
-- --------------
Craig B. Markwardt, Ph.D. EMAIL: craigmnet@REMOVEcow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
-- --------------

Subject: Re: timers without widgets
Posted by Benjamin Hornberger on Sat, 23 Apr 2005 16:39:29 GMT
View Forum Message <> Reply to Message

Craig Markwardt wrote:
> Benjamin Hornberger <benjamin.hornberger@stonybrook.edu> writes:
>> Socket I/O is exactly our problem. We use a timer to continuously
>> (actually frequently rather than continuously) read from a socket to
>> implement a listening socket. But there are many problems with that,
>> like IDL not reacting while it tries to read (looks like crashed).
>
>
> Can't you set READ_TIMEOUT and WRITE_TIMEOUT to some very small
> number? (perhaps even zero?)

>

We are doing that, but still it doesn't run very smoothly. One problem is that we have to read every 50 ms since that's the frequency how fast the data can come in. In the meantime, we would like to be able to click buttons in the GUI (which also receives the data), but that's hardly possible.

I have to say that all that, in particular the other side of the socket connection (C++) is mainly written by somebody else, so I don't have a full understanding. I just started the discussion because I was curious if we could have timers without widgets.

Benjamin

Subject: Re: timers without widgets
Posted by Benjamin Hornberger on Sat, 23 Apr 2005 16:49:47 GMT
View Forum Message <> Reply to Message

Y.T. wrote:
> Uh -- in my version of IDL, at least, the help for "socket" explicitly
> states that "socket" opens a client-side socket. Opening server-side
> sockets is quite a different beast.

That's the point. We would actually need a server-side socket.

>
> That said, there's fine bits of software out there to listen to ports
> (I'd write it in TCL if I had to) but I have to wonder whether you may
> be going through too many layers here -- if you are trying to do
> instrument control, then what is your hardware interface?

The instrument is actually directly controlled by C++ software (GPIB interface and PCI boards with drivers provided by the manufacturer). The user interface, however, is written in IDL because we want to include a lot of data processing and visualization features. We communicate via sockets because we would like to be able to run the user interface on a different machine (even remotely). I realize that if we gave up that idea, there would probably be better ways to talk between IDL and C++.

As I wrote in reply to Craig, I don't write the socket stuff myself (I am writing the GUI), so I don't know all the details (my C++ knowledge is also quite limited).

Anyway, it basically works. Could be smoother though.

Thanks for your suggestions,

Benjamin