

Re: [SLE] Developing a Real Time Data System

Source: <http://linux.derkeiler.com/Mailing-Lists/SuSE/2005-11/3081.html>

From: John Perry (*j.e.perry_at_cox.net*)

Date: 11/23/05

Date: Wed, 23 Nov 2005 16:59:12 -0500

JUAN ERNESTO FLORES BELTRAN wrote:

> *John wrote:*

>

>> *Just html?*

>

>

> *not only html, i have worked under other languages too*

OK, this sounds less discouraging...

>

>> *... Other, non-real time considerations sometimes make it better to*

>> *use the desktop and server programs (Win and Linux), but this is*

>> *separate from real-time considerations.*

>

>

> *Could explaine about this desktops and server programs more detailed..??*

The typical business computer complex has a set of computers called "file servers", whose function is to keep files and databases, and supply them to other computers for use by users or applications. Individuals have on their desks computers they use for their own computing purposes, called "desktop computers". These are all connected by a local area network (lan), which carries the information from one computer to another. The desktops typically use information from the servers, so it's all in one place and can be maintained and backed up easily.

Scientific and academic organizations frequently need very high computing power, so have a set of supercomputers frequently called "compute servers". Scientists and students use their desktop computers to gain access to these compute servers. These organizations will also have file servers in their network.

File servers are usually based on linux or special server editions of Windows. Desktops are usually based on Windows. Linux and Macintosh have a small but growing presence in desktop computer operating systems.

>
>> *With all this said, many people often miscall "interactive",
>> "event-driven", and "quick-response" programming real-time, so we'd
>> have to know what you're trying to do.*

>
>

For genuine real-time applications, there are several specially designed operating systems that provide the guaranteed response to events that Windows and linux can't ordinarily provide. Examples would be QNX, OS-9, and LYNXOS. There are also bare-bones programs called "executives" that provide real-time functions, but no file systems or, usually, networking. Examples would be VXWorks, RTEMS, or UC/OS II.

> *have you worked under labview?? (sorry this is not free software, but
> i must mention it), i would like to create an application which allows
> me to get data from a Data Acquisition Card. Actually I am thinking of
> working with a National Instruments Acquisition data card, this card
> is used for applications on which is required to measure any process,
> from voltage to temperature to an electric engine speed.*

Don't worry about empty-headed ideologs. Are you talking about using Labview, or using the National Instruments card under other programs? If you're using Labview and a National Instruments board, I'm very surprised that you want to do any programming at all, since NI normally provides their hardware fully interfaced to their software (that's not always true for brand-new stuff, as I found out in my last data acquisition project, but they respond pretty well with bug fixes).

If you're using an NI card outside the Labview environment, then yes, you're on your own, and will need a lot more information about your data acquisition project, the card's interface specs, and your computing environment.

>
> *This card just get an analog input that can vary from -10 volts to -10
> volts (no matter the process to be tested it is required to transform
> it into a ± 10 v signal and connect it to the card) then internally the
> analog signal is transformed to a digital signal and sent to the
> computer using a PCI, USB or PCMCIA bus.*

>
> *For the USB (low cost option) bus the specifications as follows:*

>
> *Bus Interface: USB 2.0 full-speed (12 Mb/s)
> Analog Input Sample rate: 10 kS/s
>
> This is just a basic description of course...such card can handle up to
> 16 analog inputs*

And this sounds as though you're just starting to design your system. If you're going to use Labview, you need to talk to a Labview applications engineer for help in setting up your system; then simply use Labview,

SuSE: Re: [SLE] Developing a Real Time Data System

realizing that although it performs well, it is not real-time (that is, there are no guaranteed response times) unless you spend a good deal of money for the special Labview real-time system.

If you're going to roll your own system software, you'd do better to get another data acquisition card, which will be much less expensive and usually easier to interface. Much of NI's stuff is premium quality and performance, but others have premium stuff, too, with a less premium price :-).

>
>
>> *With all this said, many people often miscall "interactive",
>> "event-driven", and "quick-response" programming real-time, so we'd
>> have to know what you're trying to do.*
>>
>> *And, of course, procedural programming, real-time or not, is
>> completely different from text markup programming, so you have a lot
>> to learn before you go very far, no matter what you're really doing :-).*
>
>
> *I know this requires lot of reading and learning, that's why I am
> asking for help at this forum :) any manual, tutorial or link to
> introduce me to the real data programming...I would appreciate your
> suggestions*

comp.realtime

comp.arch.embedded

Application notes from NI

Application notes from QNX

<http://hardware.ittoolbox.com/topics/t.asp?t=371&p=371&h1=371>

All have appropriate information available, and comp.realtime at least has frequently posted faqs. Either of these is more appropriate than SLE, which is concerned with desktop and server environments.

This is not to say that you can't build a good data acquisition system with suse. It will perform much better and more reliably than Windows, but you'll need much more support than you can get here. Indeed, you'll need more than you'll likely get on c.r or c.a.e, if you don't have a good knowledge base to build upon. If this is a student or hobby project, I'd say go ahead full steam -- if you stick it out, you'll build the necessary knowledge base to understand what people are saying when they try to help. If someone is depending upon this project for something important, find someone knowledgeable you can talk to frequently for help. If it's needed soon, get professional help :-).

>
>> *If your html programming included javascript and/or php, you have a
>> vague idea what the difference is.*
>

SuSE: Re: [SLE] Developing a Real Time Data System

- >
- > *Thinking of python to develop the GUI interface, there is a powerful*
- > *library called Matplotlib which allows graphics interface familiar to*
- > *matlab and code seems simple, on the other hand could i handle the*
- > *data by coding C ??*

The data acquisition part is harder, but much less complicated, than data storage and display. But you can use any of the above, and many other languages besides, to do the job. If you have to. If you really have to do your own card driver, you'll need C and/or assembly language.

If on the other hand, you're looking to get a data acquisition/control job done, why not just use Labview or one of its lookalikes? They've already done the dirty detail work, and you just need to build your application.

John Perry

--

Check the headers for your unsubscription address
For additional commands send e-mail to suse-linux-e-help@suse.com
Also check the archives at <http://lists.suse.com>
Please read the FAQs: suse-linux-e-faq@suse.com