

Re: porting curses driven to gui application

Source: <http://unix.derkeiler.com/Newsgroups/comp.unix.programmer/2004-09/0690.html>

From: Theo Harlauch (t.harlauch_at_freenet.de)

Date: 09/12/04

Date: Sun, 12 Sep 2004 18:27:52 +0200

Jens.Toerring@physik.fu-berlin.de wrote:

> *It looks a bit as if you're mixing up a few things here. Having a
> GUI or not has nothing at all to do with network related stuff or
> remote database connections. It seems as if you want to do a lot
> more than just replace the curses interface by a GUI.*
>
> *When you write that the users log in via ssh it indicates that they
> are not sitting at a terminal connected to the machine where the
> application is running but connect from external machines and work
> on the machine they log in from. As long as the other machine a
> user is coming from has a X server running (and ssh is configured
> to allow X forwarding) you don't need to change anything about your
> application beside replacing the curses interface by the GUI –
> everything else works as before, the GUI can be easily shown on
> the external users screen via X – the users still log in via ssh
> and start the application by typing its name at the prompt, only
> that they now get a GUI to interact with. Your only headache will
> probably be the time it takes to display and update the GUI on the
> external machines, but that won't be that problematic as long as
> everything runs on the same LAN or over some other fast network
> connection.*

In fact, most of the users are running PuTTY on their windows 2000 clients, some of them are using xterm on linux.

> *Things get quite a bit more complicated if the external machines
> don't have a X server (e.g. because they are just running Windows
> without much more installed than ssh). Then you will have to
> resort to some kind of client/server architecture and you will
> have to decide what of the functionality goes into the client
> and what into the server,*

Well, what about installing a local X server on the windows clients like Cygwin/X [1]? I haven't seen Cygwin/X in action yet, though. So there might be some pitfalls too since it is just a port of the X Window System.

> *[Using http server as client-server architecture]*

I don't like web applications. I think, html is not suitable for my kind of application. Many things had to be solve with javascript and this is not the kind of "programming language" I want to fiddle around.

- > *Next solution would be to have a program running on the clients*
- > *machine that shows the GUI and connects to a server process*
- > *running on the machine where the database is also running. You*
- > *have to develop some kind of protocol how data to be put into*
- > *the database or retrieved from it are to be exchanged between the*
- > *server and the client on the user side. Part of the original*
- > *functionality (mainly the user interaction) now is on the client*
- > *side while on the server the interaction with the database is*
- > *done. What exactly of the functionality you put into the client*
- > *and what in the server is up to you. It could go from just sending*
- > *some information about what the user just entered into the GUI to*
- > *the server and reacting to what the server then says to do (so most*
- > *of the work is done by the server side) to sending complete database*
- > *commands to the server, which it just passes on to the database, and*
- > *having the results of queries returned to the client (thus having*
- > *nearly all of the functionality on the client side). Still, no*
- > *external connections to the database are necessary.*

This looks like a possible solution. Are there any standards floating around for writing such kind of protocols? I guess we are talking about sockets and TCP here...

Thanks for your time, it cleared some things up on my side!