Subject: How to create sockets that don't block the GUI?
Posted by lectus on Fri, 07 Dec 2012 12:38:45 GMT
View Forum Message <> Reply to Message

So far I was able to communicate between sockets, but my problem is that when there's intensive processing the GUI locks and I can't interact with it.

I'd like to have a socket in a while(1) loop while having the GUI fully functional.

Any ideas?

Subject: Re: How to create sockets that don't block the GUI?
Posted by Zbych on Fri, 07 Dec 2012 12:48:31 GMT
View Forum Message <> Reply to Message

Move all communication to second thread.

Subject: Re: How to create sockets that don't block the GUI?
Posted by lectus on Fri, 07 Dec 2012 12:53:24 GMT
View Forum Message <> Reply to Message

That's what I'm doing but it requires a GuiLock to be able to update the GUI inside the thread, then the GUI locks.

If I don't use the GuiLock the application crashes.

Subject: Re: How to create sockets that don't block the GUI?
Posted by Zbych on Fri, 07 Dec 2012 13:06:04 GMT
View Forum Message <> Reply to Message

Try to use PostCallback, to pass information to GUI from communication thread and of course avoid executing callbacks to often (>50Hz).

Subject: Re: How to create sockets that don't block the GUI?
Posted by lectus on Fri, 07 Dec 2012 13:06:43 GMT
View Forum Message <> Reply to Message

I think I got it to work.

I need this line:

GuiLock __;

ONLY right before the GUI gets updated, otherwise my GUI locks the whole time.

Subject: Re: How to create sockets that don't block the GUI?
Posted by nlneilson on Fri, 07 Dec 2012 14:01:20 GMT
View Forum Message <> Reply to Message

Also Zbych is correct that the transfer of data should be limited for transfer per second. Put a Sleep(10); between each transfer if necessary, slower computers take more time.

Remember each transfer is a line ending with \0, I transfer 50 'lines' at a time each second without a problem.
It's the make/break of the connection that takes time as the actual transfer is very fast.

The socket is used to transfer data from a C++ app to a Java GUI app with less that 5% CPU usage.

This may not directly relate to your GUI lock but the concepts may help.

Subject: Re: How to create sockets that don't block the GUI?
Posted by lectus on Fri, 07 Dec 2012 15:00:24 GMT
View Forum Message <> Reply to Message

nlneilson wrote on Fri, 07 December 2012 09:01Also Zbych is correct that the transfer of data should be limited for transfer per second. Put a Sleep(10); between each transfer if necessary, slower computers take more time.

Remember each transfer is a line ending with \0, I transfer 50 'lines' at a time each second without a problem.
It's the make/break of the connection that takes time as the actual transfer is very fast.

The socket is used to transfer data from a C++ app to a Java GUI app with less that 5% CPU usage.

This may not directly relate to your GUI lock but the concepts may help.

Yes, every detail is great help. Thanks.

What do you mean that line ending is \0. Isn't it \n?

Subject: Re: How to create sockets that don't block the GUI?

Posted by nlneilson on Fri, 07 Dec 2012 16:30:10 GMT
View Forum Message <> Reply to Message

Actually what is being sent is a string of char not a line but it is often referred to as "sending a line".
A String has \0 at the end, NUL terminated.
Note in my previous post I have 'lines' as they were separate lines before adding all the char to the buf to be sent/received.

I have a char buf of 3000
I just add the char and only at the end is there \0

Sometime the only thing sent from the client is "+" which is actually '+' \0 or 43 0 or 0x2B 0x0
The server returns the latitude and longitude of the center in the Java app.

For tracking up to 50 objects with lat, lon, alt, ID, etc then all that data for all 50 is added to the char buf then ended with \0.
On the server the \0 means just that, the end.
So if the buf contains 2000 char and then just + \0 is sent the rest of the buf is ignored and not even sent as \0 means THE END.

A better definition would be a C string.
http://stackoverflow.com/questions/10943033/why-are-strings- in-c-usually-terminated-with-0

(Line feed, '\n', 0x0A, 10 in decimal)
So '\n' is just ch = 10;
When that is sent through a socket that is all it is.
How it is dealt with on receiving can be handled.

NULL is same as '\0' which is same as 0x0 or 0

Subject: Re: How to create sockets that don't block the GUI?
Posted by lectus on Fri, 07 Dec 2012 18:28:08 GMT
View Forum Message <> Reply to Message

I'm confused. The documentation says this:

String GetLine(int maxlen = 65536)
Reads single line (ended with '\n', '\r' is ignored). If the whole line cannot be read within timeout or line length is longer than maxlen sets error and returns String::GetVoid().

Subject: Re: How to create sockets that don't block the GUI?
Posted by nlneilson on Sat, 15 Dec 2012 17:40:04 GMT
View Forum Message <> Reply to Message

What is being sent through a socket is a bunch of character ending with \0.

After it is received it can be parsed into separate lines with \n

Do all of the addition of characters before passing the data to the client to be sent and on the server just pass the whole buffer out to be parsed.

For small amounts of data it can be added and parsed in the client and server.

The \n or \r are just like any other character as far as what is in the buffer or packet sent. If the actual packet size limit is 65536 then there could be many \n or whatever.
The only character that has real significance is \0 which indicates the END of the packet.

Parse what is in the packet outside the server code unless the amount of data is small.

Subject: Re: How to create sockets that don't block the GUI?
Posted by Alexander_Ag on Wed, 20 Mar 2013 20:28:37 GMT
View Forum Message <> Reply to Message

lectus wrote on Fri, 07 December 2012 16:38So far I was able to communicate between sockets, but my problem is that when there's intensive processing the GUI locks and I can't interact with it.

I'd like to have a socket in a while(1) loop while having the GUI fully functional.

Any ideas?

Very interesting topic - can anyone give a sample code with GUI that use TcpSocket as server, i just begin work around sockets.
For example http://www.ultimatepp.org/reference$SocketServer$en-us.html but with GUI.

Subject: Re: How to create sockets that don't block the GUI?
Posted by dolik.rce on Thu, 21 Mar 2013 05:57:08 GMT
View Forum Message <> Reply to Message

Alexander_Ag wrote on Wed, 20 March 2013 21:28lectus wrote on Fri, 07 December 2012 16:38So far I was able to communicate between sockets, but my problem is that when there's intensive processing the GUI locks and I can't interact with it.

I'd like to have a socket in a while(1) loop while having the GUI fully functional.

Any ideas?

Very interesting topic - can anyone give a sample code with GUI that use TcpSocket as server, i just begin work around sockets.
For example http://www.ultimatepp.org/reference$SocketServer$en-us.html but with GUI.

It works the same way as any other time consuming process. You just have to make sure that you call Ctrl::ProcessEvents() from time to time to update the GUI. In case of sockets you might want to lower the timeout, so the loop executes faster and call ProcessEvents in each iteration.

Alternatively, use two separate threads...

Best regards,
Honza

Subject: Re: How to create sockets that don't block the GUI?
Posted by jlh67 on Mon, 17 Feb 2014 08:51:04 GMT
View Forum Message <> Reply to Message

look my code ...

File Attachments
1) main.cpp, downloaded 436 times
2) TestServerGUI.h, downloaded 655 times
3) TestServerGUI.lay, downloaded 675 times

Subject: Re: How to create sockets that don't block the GUI?
Posted by ManfredHerr on Mon, 17 Feb 2014 16:31:33 GMT
View Forum Message <> Reply to Message

I would like to advise rethinking the concept with regard to the client server model. Normally, a server serves requests from a number of clients and the clients are there for GUI not the server. OK?

Subject: Re: How to create sockets that don't block the GUI?
Posted by jlh67 on Mon, 17 Feb 2014 16:52:23 GMT
View Forum Message <> Reply to Message

ManfredHerr wrote on Mon, 17 February 2014 17:31I would like to advise rethinking the concept with regard to the client server model. Normally, a server serves requests from a number of clients and the clients are there for GUI not the server. OK?

I agree with you.But the TCP protocol can be used to transmit data to a GUI application. In this case, we are not talking client-server model: It is a simple transmission of data via the TCP protocol.

excuse me for my english.

Subject: Re: How to create sockets that don't block the GUI?
Posted by ManfredHerr on Tue, 18 Feb 2014 11:00:13 GMT
View Forum Message <> Reply to Message

transfer? Sockets can be read like a file. If you try to read a huge file in a GUI application then your GUI is inresponsive as long as you read the file. For your intensive data stream the same happens. As long as data is coming in the process is busy; too busy to process User Input. One option is to control the data stream by something like XON/XOFF. So you can handle the User Input with equal or even higher priority than data transfer.

Subject: Re: How to create sockets that don't block the GUI?
Posted by jlh67 on Tue, 18 Feb 2014 20:19:52 GMT
View Forum Message <> Reply to Message

Correct! But if the data stream is low, there is no problem. it's a simple solution without flow control.

Subject: Re: How to create sockets that don't block the GUI?
Posted by ManfredHerr on Tue, 18 Feb 2014 21:52:50 GMT
View Forum Message <> Reply to Message

So, what is the problem then?

Subject: Re: How to create sockets that don't block the GUI?
Posted by jlh67 on Wed, 19 Feb 2014 17:07:20 GMT
View Forum Message <> Reply to Message

ManfredHerr wrote on Tue, 18 February 2014 22:52So, what is the problem then?

There is no problem! I just gave my solution to the question
"How to create sockets that don't block the GUI?"
[message #39480]

Subject: Re: How to create sockets that don't block the GUI?
Posted by MrSarup on Tue, 20 Dec 2016 12:02:42 GMT
View Forum Message <> Reply to Message

Hello!

Well, I am working on a Server<--->Client model. I have created a new package and saved the codes published above in cpp, h and lay files. I used the nightly build available today. It works

fine.

I wish the Ultimate Dev. team could have implemented more examples of TCPServer, in particular on Server/Client with GUI.
