Subject: Simple IPC using Sockets
Posted by dolik.rce on Sun, 14 Feb 2010 23:36:40 GMT
View Forum Message <> Reply to Message

Hello All,

I would like to share one very small class I wrote. It is simple object providing mostly automatic and very easy to set up interprocess communication. Example of usage is attached as well.

Basic goals: Simple to use
No specialized server app required
Everything should be as automatic as possible
Programmer can simply define communication protocol
Communication is done via Sockets, no shared memory (related to the next point)
It is not meant to be robust, but to allow easy and rapid development for small scale apps.
Implemented features: Everything mentioned in goals (I hope)
Protocol is specified using Callbacks, so rules can have different forms (e.g: single function with switch for handling messages OR several functions, each is defining one rule (used in example app) OR even pipeline style processing, where each function modifies the output of previous function and then passes it to next)
One instance of application performs the tasks of server in separate thread. This role is automatically picked up by another instance when the previous server is closed.
Optionally, programmer can start/stop server thread.

Warning: This package was not extensively tested yet, you might experience all sort of bad things from loss of information to deadlocks. It did not happen to me, but it doesn't mean it can't happen There is a high probability that I will develop this code further in future, since I wan't to use it in one of my projects.

About the example app: It's a console app that does nothing else then manages a list of strings (Vector<String> internally) from a simple shell. You can display it's content (command "show"), add ("add something") and delete items ("del number"). To exit, type "quit". Nothing much, heh? The interesting part is that if you open few more instances of this application, they will all behave as if they worked with single variable (but they of course do not, they just keep their own variable in sync). And this magic is done in ~50 lines of code, out of which is ~4O lines are definitions of protocol.

If you think this is interesting/useful please let me know. Suggestions for further development are welcomed too

Best regards,
Honza

File Attachments

1) IPC.zip, downloaded 708 times

Subject: Re: Simple IPC using Sockets
Posted by tojocky on Mon, 15 Feb 2010 13:43:17 GMT
View Forum Message <> Reply to Message

Nice,

I will test!
dolik.rce wrote on Mon, 15 February 2010 01:36Hello All,

I would like to share one very small class I wrote. It is simple object providing mostly automatic and very easy to set up interprocess communication. Example of usage is attached as well.

Basic goals: Simple to use
 No specialized server app required
 Everything should be as automatic as possible
 Programmer can simply define communication protocol
 Communication is done via Sockets, no shared memory (related to the next point)
 It is not meant to be robust, but to allow easy and rapid development for small scale apps.
Implemented features: Everything mentioned in goals (I hope)
 Protocol is specified using Callbacks, so rules can have different forms (e.g: single function with switch for handling messages OR several functions, each is defining one rule (used in example app) OR even pipeline style processing, where each function modifies the output of previous function and then passes it to next)
 One instance of application performs the tasks of server in separate thread. This role is automatically picked up by another instance when the previous server is closed.
 Optionally, programmer can start/stop server thread.

Warning: This package was not extensively tested yet, you might experience all sort of bad things from loss of information to deadlocks. It did not happen to me, but it doesn't mean it can't happen There is a high probability that I will develop this code further in future, since I wan't to use it in one of my projects.

About the example app: It's a console app that does nothing else then manages a list of strings (Vector<String> internally) from a simple shell. You can display it's content (command "show"), add ("add something") and delete items ("del number"). To exit, type "quit". Nothing much, heh? The interesting part is that if you open few more instances of this application, they will all behave as if they worked with single variable (but they of course do not, they just keep their own variable in sync). And this magic is done in ~50 lines of code, out of which is ~4O lines are definitions of protocol.

If you think this is interesting/useful please let me know. Suggestions for further development are welcomed too

Best regards,
Honza

Subject: Re: Simple IPC using Sockets
Posted by dolik.rce on Tue, 23 Feb 2010 11:57:31 GMT
View Forum Message <> Reply to Message

Hello all!

Thanks Ion, let me know how did it work!

Here is new version. Some little bugs were fixed and new class, IPC2, was added. This new class extends the possibilities of IPC with broadcast and client-to-client messages. The examle application was extended a bit to show new capabilities. Also user documentation is enclosed and up-to-date.

The communication is implemented using polling. Related to this, I should mention that in console mode, the package uses bazaar/Timer (thanks kohait00!) and with GUI flag it uses functions from CtrlCore.

And of course, I'm still opened to suggestions...

Best regards,
Honza

File Attachments

1) IPC.zip, downloaded 677 times

Subject: Re: Simple IPC using Sockets
Posted by tojocky on Tue, 23 Feb 2010 16:39:24 GMT
View Forum Message <> Reply to Message

Honza,

Preview version worked only on win32. on linux did not work. I had not sufficient time for complete test and sugestion.

One good practice is google Chrome:
 http://dev.chromium.org/developers/design-documents/inter-pr ocess-communication

I will test the latest version.

Regards,
Ion Lupascu (tojocky)

dolik.rce wrote on Tue, 23 February 2010 13:57Hello all!

Thanks Ion, let me know how did it work!

Here is new version. Some little bugs were fixed and new class, IPC2, was added. This new class

extends the possibilities of IPC with broadcast and client-to-client messages. The examle application was extended a bit to show new capabilities. Also user documentation is enclosed and up-to-date.

The communication is implemented using polling. Related to this, I should mention that in console mode, the package uses bazaar/Timer (thanks kohait00!) and with GUI flag it uses functions from CtrlCore.

And of course, I'm still opened to suggestions...

Best regards,
Honza

Subject: Re: Simple IPC using Sockets
Posted by dolik.rce on Tue, 23 Feb 2010 16:54:22 GMT
View Forum Message <> Reply to Message

tojocky wrote on Tue, 23 February 2010 17:39Honza,

Preview version worked only on win32. on linux did not work.
Funny, I never tested on windows and yet it works (only) there I will try on some other machines, to see if I can reproduce your problem. On ubuntu 9.10 and Arch Linux, it works fine.

Anyway, thank you for your interest!

Honza

Subject: Re: Simple IPC using Sockets
Posted by dolik.rce on Sat, 27 Feb 2010 12:54:40 GMT
View Forum Message <> Reply to Message

I just found it hangs on exiting when run on multicore processor So long for a simple class... I'll try to fix it of course, but it may take a lot of time as I'm not really skilled in MT.

Honza

Subject: Re: Simple IPC using Sockets
Posted by tojocky on Sat, 26 Feb 2011 10:24:03 GMT
View Forum Message <> Reply to Message

Hello Honza,

How about to improve a little your IPC?

I propose to add possibility to integrate named pipes. In the local computer and speed LAN the communication is more faster.

I will try to change your code and propose a variant with named pipes. Sure if you allow to change your code.

Some utils links:
http://msdn.microsoft.com/en-us/library/aa178138(v=sql.80).a spx
http://www.chromium.org/developers/design-documents/inter-pr ocess-communication

---

Subject: Re: Simple IPC using Sockets
Posted by dolik.rce on Sat, 26 Feb 2011 10:56:02 GMT
View Forum Message <> Reply to Message

Hi Ion,

Well, why not I know that the sockets are not ideal and actually I was thinking about rewriting this as well. My idea was to use DBus, as it is more or less standard solution these days and can be used to interact with other applications as well... Or maybe the DBus could be completely separate package. Also, if I am not mistaken, the most common DBus server implementation uses named pipes for comunication as well.

But back to the topic: If you wish so and have the time, you can modify this package as you want. I didn't have time to develop this for very long time and didn't even need to use it, so I'll be more than happy if someone improves it

Honza
