Subject: some U++ stuff on reddit/cpp_question
Posted by Xemuth on Thu, 06 Aug 2020 01:55:49 GMT
View Forum Message <> Reply to Message

https://www.reddit.com/r/cpp_questions/comments/gb6h7p/u_vs_
wxwidgets_vs_qt_best_c_gui/?utm_medium=android_app&utm_s ource=share

https://www.reddit.com/r/cpp/comments/gb5shc/u_vs_wxwidgets_
vs_qt_best_c_gui/?utm_medium=android_app&utm_source=shar e

I was spending some time on the subreddit cpp and cpp_question and found this post about U++
It seems Upp is not known

Subject: Re: some U++ stuff on reddit/cpp_question
Posted by mirek on Thu, 06 Aug 2020 08:39:38 GMT
View Forum Message <> Reply to Message

Yes. Unfortunately.

Frankly, I have run out of options how to change that years ago....

Subject: Re: some U++ stuff on reddit/cpp_question
Posted by Xemuth on Thu, 06 Aug 2020 11:25:10 GMT
View Forum Message <> Reply to Message

But what can explain some framework like Qt or WxWidgets is well more known, is QT team or else has made advertisement ?

Both Qt and WxWidgets have their own license, do it make them (from an enterprise point of view) more serious way of working than a BSD framework ?
(if yes then, what about Linux ? Apache ? etc ?)

from my knowledge, beginner developer don't do C++, but everytime I introduce some of them to U++ they get convinced by the power of the tool. the only problem is their lack of skills about C++ and all the complexicity of "pointers" "mutex" "stack/heap memory" . Damn you modern alternatives to C++ !

Subject: Re: some U++ stuff on reddit/cpp_question
Posted by mirek on Thu, 06 Aug 2020 15:32:09 GMT
View Forum Message <> Reply to Message

Xemuth wrote on Thu, 06 August 2020 13:25But what can explain some framework like Qt or WxWidgets is well more known, is QT team or else has made advertisement ?

Qt is developed by big commercial team. Frankly, that creates an image of stability and future.

WxWidgets are somewhat older than U++ and were used for some pretty important apps. Also wxWidgets archicture, if I remember well, is unique in a sense that it is not emulated toolkit (means wxWidgets is using widgets from host platform, unlike U++ or Qt, which has own implementation) - this probably has some appeal for developers too.

Mirek

Subject: Re: some U++ stuff on reddit/cpp_question
Posted by Xemuth on Thu, 06 Aug 2020 15:59:55 GMT
View Forum Message <> Reply to Message

mirek wrote on Thu, 06 August 2020 17:32
if I remember well, is unique in a sense that it is not emulated toolkit (means wxWidgets is using widgets from host platform, unlike U++ or Qt, which has own implementation) - this probably has some appeal for developers too.

Maybe it is a stupid question but in what's point of having widgets from host platform instead of emulated (which allow cross platform) could be attractive for developers ?
(knowing Upp widget can work on different operating system comfort me on the idea of one time I will be able to release my application not only on windows but also on Linux)

Subject: Re: some U++ stuff on reddit/cpp_question
Posted by Klugier on Thu, 06 Aug 2020 18:14:24 GMT
View Forum Message <> Reply to Message

Hello Xemuth,

Quote:
Maybe it is a stupid question but in what's point of having widgets from host platform instead of emulated (which allow cross platform) could be attractive for developers ?
(knowing Upp widget can work on different operating system comfort me on the idea of one time I will be able to release my application not only on windows but also on Linux)

The reason for that is that you have better look and fell in compare to Upp. The tricky part about it is that it is still multiplatform. WxWidgets creates wrapper class for that, so in application code you call wrapper not use the native code. So, your application is potable across operating systems. FileZilla is good example of multi-platform app created with that technology. The same code base for Windows, Linux and macOS. However, I always find WxWidgets application heavy and slow. It runs slows in compare to native or Upp applications.

Also, please notice that QT uses the same approach as Upp. No native widgets (KDE is the

exception) all is draw and the styled depending on the platform.

Quote:
from my knowledge, beginner developer don't do C++, but everytime I introduce some of them to U++ they get convinced by the power of the tool. the only problem is their lack of skills about C++ and all the complexicity of "pointers" "mutex" "stack/heap memory" . Damn you modern alternatives to C++ !

The problem with C++ is that it is 40 years old technology. Within this time humanity advanced in the term of thinking about programming language. The good example of that is Rust programming language. It is similar to C++ in context of performance (no garbage collector), but better designed. In context of memory safety (crashes are common problems with application written in C/C++) and easy to use (header files are outdated). Also it posses package manager, which makes whole platform easy to expand. Please notice that right know Linux kernel is experimenting with that technology and will probably allow to write drivers with it.

Quote:
But what can explain some framework like Qt or WxWidgets is well more known, is QT team or else has made advertisement ?

For the larger compones (corporations) QT is better that WxWidget or Upp. The main reason for that is they have money to pay for professional license. But, it is not the case. The main reason for that is they offer stability. It is develop under QT company, so in compare to Upp which is mainly maintenance by single developer - Mirek. Please imagine what will happen with Upp when Mirek will stop working on that? In case of QT they can replace employees more easily. They also offer commercial support which is nice. If you are for example blocked by issue in the library you can force them to fix that issue. In case of Upp nobody can force Mirek to do anything

The features I like about QT:
- Styling via CSS (Non technical person can modify the look of the app like UX Designer - very important for bigger companies)
- QML - UI separated from the binary. You can modify layouts without recompiling application. Still important for non technical users for modifying the look of application.
- Support for mobile devices (no critical, but nice to have)
- UI can be tested - important in big projects that want to provide high quality for the users. Moreover, the risk is greater when more developers work on project
- Other IDE integration (Visual Studio, CMake, Conan packages etc).

Sincerely,
Klugier

Subject: Re: some U++ stuff on reddit/cpp_question
Posted by Novo on Thu, 06 Aug 2020 19:25:44 GMT
View Forum Message <> Reply to Message

mirek wrote on Thu, 06 August 2020 04:39Yes. Unfortunately.

Frankly, I have run out of options how to change that years ago....
For some reason I never saw your presentation at Meeting CPP

Subject: Re: some U++ stuff on reddit/cpp_question
Posted by Novo on Thu, 06 Aug 2020 19:50:52 GMT
View Forum Message <> Reply to Message

Xemuth wrote on Thu, 06 August 2020 07:25But what can explain some framework like Qt or WxWidgets is well more known, is QT team or else has made advertisement ?

They not just advertise. They advertise extremely aggressively. They attack everybody who can compete with them. I personally was attacked by one of the top QT guys for just saying that we are making a similar product in a different domain (they didn't have similar technology at that time, probably, still do not have it). I'm pretty sure QT is spending money to make U++ invisible.

Subject: Re: some U++ stuff on reddit/cpp_question
Posted by Xemuth on Thu, 06 Aug 2020 21:59:16 GMT
View Forum Message <> Reply to Message

Hello Klugier, thanks for this explanation,

Klugier wrote on Thu, 06 August 2020 20:14
Within this time humanity advanced in the term of thinking about programming language

Indeed but the goal of differents C++ release is not to follow the humain thinking about programming language ?

Klugier wrote on Thu, 06 August 2020 20:14
Please notice that right know Linux kernel is experimenting with that technology and will probably allow to write drivers with it

Do it mean that C++ is condemned to extinguish itself?

Klugier wrote on Thu, 06 August 2020 20:14
For the larger compones (corporations) QT is better that WxWidget or Upp. The main reason for that is they have money to pay for professional license. But, it is not the case. The main reason for that is they offer stability. It is develop under QT company, so in compare to Upp which is mainly maintenance by single developer - Mirek. Please imagine what will happen with Upp when Mirek will stop working on that? In case of QT they can replace employees more easily. They also offer commercial support which is nice. If you are for example blocked by issue in the library you can force them to fix that issue. In case of Upp nobody can force Mirek to do anything

Indeed from a company view the fact Qt is a company wich provide services to client make it a strong choice. However this BSD license + all the source code could allow a company in a long term view to have a solide framework wich allow them to compete (in terme of needed features) with QT without any licenses ? (maybe the cost of learning the entire framework and improving it in their way is to expensive ?)

Klugier wrote on Thu, 06 August 2020 20:14
Please imagine what will happen with Upp when Mirek will stop working on that?

Disaster ! The fact Upp is a BSD could allow some developper to take back is work to improve it ? Maybe it's Too good to be true. At Least the footprint of the framework will remain since the source is accessible

Maybe a way to make Upp more used would be to provide by differents way some trainings about all axis of Upp (indeed it already exist via all example) but something deeper ? maybe video about it or virtual classroom ? Maybe I'm a dreamer and the state of informatique in a company point of view is too stuck/ hard coded to start using new technology

---

Subject: Re: some U++ stuff on reddit/cpp_question
Posted by koldo on Thu, 06 Aug 2020 22:33:35 GMT
View Forum Message <> Reply to Message

IMHO a way to get momentum would be to get paid support/paid developers to advance in some interesting structural areas (not just developing a new widget) like other ide integration, mobiles, full Github integration, etc.
With that push we may make a quantum leap to be in the premier league.

Subject: Re: some U++ stuff on reddit/cpp_question
Posted by pvictor on Fri, 07 Aug 2020 07:05:20 GMT
View Forum Message <> Reply to Message

As an occasional U++ user, I think that it is a great cross platform RAD tool and only lacks two things:

- a good book (especially for beginners in C++).
There are many subtle places in advanced programs which are unclear and having a book, describing these technics would greatly help.

- a comprehensive help system.
As for now, there are many useful classes and functions in U++, that are not even mentioned in

help and can be found only accidentally. For example: ImagePainter, BufferPainter, FileSelNative, PaintInfo, msecs(), usecs(), Random().

I understand that writing help is a boring work, but it could increase the popularity of U++.

Best regards,
Victor

Subject: Re: some U++ stuff on reddit/cpp_question
Posted by mirek on Fri, 07 Aug 2020 07:17:39 GMT
View Forum Message <> Reply to Message

Quote:
The problem with C++ is that it is 40 years old technology. Within this time humanity advanced in the term of thinking about programming language. The good example of that is Rust programming language. It is similar to C++ in context of performance (no garbage collector), but better designed. In context of memory safety (crashes are common problems with application written in C/C++) and easy to use (header files are outdated). Also it posses package manager, which makes whole platform easy to expand. Please notice that right know Linux kernel is experimenting with that technology and will probably allow to write drivers with it.

To be fair, C++ was continually evolving over those 40 years. C++ today is not the same as C++ in 1985... I guess the problem is with C++ is that it is very successful (large portion of software in use today is written in C++) while hard to master, so that creates optics of "old quirky technology".

Subject: Re: some U++ stuff on reddit/cpp_question
Posted by mirek on Fri, 07 Aug 2020 07:26:15 GMT
View Forum Message <> Reply to Message

pvictor wrote on Fri, 07 August 2020 09:05As an occasional U++ user, I think that it is a great cross platform RAD tool and only lacks two things:

- a good book (especially for beginners in C++).
There are many subtle places in advanced programs which are unclear and having a book, describing these technics would greatly help.

- a comprehensive help system.
As for now, there are many useful classes and functions in U++, that are not even mentioned in help and can be found only accidentally. For example: ImagePainter, BufferPainter, FileSelNative, PaintInfo, msecs(), usecs(), Random().

I understand that writing help is a boring work, but it could increase the popularity of U++.

Best regards,
Victor

I agree. But one thing to add is that all these things should be 'outward'. So far I am in endless cycle of improving U++, adding more docs, help etc, but problem is that all that happens in our repo and nobody know. I am sorry but somewhat I lack the energy to make things more public...

I still believe that the best chance is to carefully disseminate articles in places that (still) allow it, like CodeProject. If I would be able to write more than single article a year, things would probably be better

---

Subject: Re: some U++ stuff on reddit/cpp_question
Posted by koldo on Sun, 23 Aug 2020 19:24:51 GMT
View Forum Message <> Reply to Message

koldo wrote on Fri, 07 August 2020 00:33IMHO a way to get momentum would be to get paid support/paid developers to advance in some interesting structural areas (not just developing a new widget) like other ide integration, mobiles, full Github integration, etc.
With that push we may make a quantum leap to be in the premier league.
For example, you can compare us with Rust. They have big sponsors including tenths of full time employees. That is a huge advantage.
The bad side of this is when the sponsor cuts suddenly the support.

Subject: Re: some U++ stuff on reddit/cpp_question
Posted by koldo on Mon, 24 Aug 2020 06:11:41 GMT
View Forum Message <> Reply to Message

You can imagine, for example, ten full-time expert programmers working under the direction of Mirek.
Just for 6 months, they could get a lot of things.

Subject: Re: some U++ stuff on reddit/cpp_question
Posted by mirek on Mon, 24 Aug 2020 08:08:46 GMT
View Forum Message <> Reply to Message

koldo wrote on Mon, 24 August 2020 08:11You can imagine, for example, ten full-time expert programmers working under the direction of Mirek.
Just for 6 months, they could get a lot of things.

Yeah, we can all dream....

Subject: Re: some U++ stuff on reddit/cpp_question
Posted by mirek on Tue, 01 Sep 2020 10:35:03 GMT
View Forum Message <> Reply to Message

mirek wrote on Fri, 07 August 2020 09:26
I still believe that the best chance is to carefully disseminate articles in places that (still) allow it, like CodeProject. If I would be able to write more than single article a year, things would probably be better

At least I am trying: https://www.codeproject.com/Articles/5268022/Getting-Started-with-Uplusplus-2020-1

Now if every U++ dev published just single article per year, we would be much better known now... (I have already fullfilled my quota this year with 2 articles

Publish or perish...

Mirek

Subject: Re: some U++ stuff on reddit/cpp_question
Posted by Oblivion on Tue, 01 Sep 2020 11:46:48 GMT
View Forum Message <> Reply to Message

Quote:
At least I am trying: https://www.codeproject.com/Articles/5268022/Getting-Started-with-Uplusplus-2020-1

Nice.

How about if I write a two-part tutorial on Core/SSH package.

The first part would use the reference/SshBasics example to demonstrate the classes and usage (it already covers the most aspects of SSH package, including MT)

The second part would use my SftpBrowser example to demonstrate the simplicity of creating SSH-based GUI apps using U++.

I am confident the I can get the first part done before the new year. But would you kindly review the articles before I submit them?

Best regards,
Oblivion

Subject: Re: some U++ stuff on reddit/cpp_question
Posted by mirek on Sat, 05 Sep 2020 08:31:44 GMT
View Forum Message <> Reply to Message

mirek wrote on Tue, 01 September 2020 12:35mirek wrote on Fri, 07 August 2020 09:26
I still believe that the best chance is to carefully disseminate articles in places that (still) allow it, like CodeProject. If I would be able to write more than single article a year, things would probably be better

At least I am trying: https://www.codeproject.com/Articles/5268022/Getting-Started-with-Uplusplus-2020-1

Now if every U++ dev published just single article per year, we would be much better known now... (I have already fullfilled my quota this year with 2 articles

https://sourceforge.net/projects/upp/files/upp/stats/timelin e

Subject: Re: some U++ stuff on reddit/cpp_question
Posted by Xemuth on Sat, 05 Sep 2020 13:37:01 GMT
View Forum Message <> Reply to Message

mirek wrote on Sat, 05 September 2020 10:31mirek wrote on Tue, 01 September 2020 12:35mirek wrote on Fri, 07 August 2020 09:26
I still believe that the best chance is to carefully disseminate articles in places that (still) allow it, like CodeProject. If I would be able to write more than single article a year, things would probably be better

At least I am trying: https://www.codeproject.com/Articles/5268022/Getting-Started-with-Uplusplus-2020-1

Now if every U++ dev published just single article per year, we would be much better known now... (I have already fullfilled my quota this year with 2 articles

https://sourceforge.net/projects/upp/files/upp/stats/timelin e

This spike of download come from the CodeProject article ?!

Subject: Re: some U++ stuff on reddit/cpp_question
Posted by mirek on Sat, 05 Sep 2020 14:19:09 GMT
View Forum Message <> Reply to Message

Xemuth wrote on Sat, 05 September 2020 15:37mirek wrote on Sat, 05 September 2020 10:31mirek wrote on Tue, 01 September 2020 12:35mirek wrote on Fri, 07 August 2020 09:26
I still believe that the best chance is to carefully disseminate articles in places that (still) allow it, like CodeProject. If I would be able to write more than single article a year, things would probably be better

At least I am trying: https://www.codeproject.com/Articles/5268022/Getting-Started-with-Uplusplus-2020-1

Now if every U++ dev published just single article per year, we would be much better known now... (I have already fullfilled my quota this year with 2 articles

https://sourceforge.net/projects/upp/files/upp/stats/timelin e

This spike of download come from the CodeProject article ?!

Probably. It is hard to say... But there was an article and there is the spike...

Mirek

Subject: Re: some U++ stuff on reddit/cpp_question
Posted by jjacksonRIAB on Wed, 23 Sep 2020 05:39:31 GMT
View Forum Message <> Reply to Message

Yeah, I've noticed on those forums that you hardly ever hear about U++ and though you occasionally get a kind comment mostly they're ignored. Maybe part of the solution is to actually create and contribute to forums in these places. So called "killer apps" would also be helpful. A major part of the reason I created the discord bot library was to spur interest in U++ even though I was not a huge discord user (I don't use it at all anymore due to privacy concerns) but when I posted about it on a subreddit I thought was dedicated to bot creation, they said it was "cool" but deleted the thread and told me it was "advertising". Strange use of a word for something given away for free.

I'm not a marketing guru, so I don't know how to generate buzz around U++. Lately I had this idea of creating some sort of linux distro that used only U++ technologies based on a U++ window manager. The idea still sounds cool to me and could perhaps spur additional development of apps, but it would require significant work to get such a thing off the ground.
