Subject: Future of C++
Posted by Xemuth on Mon, 09 Dec 2019 11:11:57 GMT
View Forum Message <> Reply to Message

Hello community,

I'm currently ending my studies and looking for a job in C++ in France and after doing 5 or 6 job's interview in several big french developpement company, it appear they don't look for any C++ programmer. Indeed they all ask me if I know some Web languages (JS PHP JAVA EE) or Python.
Even if I know C++ have good placement in the industry due to is speed.
It seems like the IT market don't need C++ anymore and tend to replace it with python or FullStack JS.

Have you the same feeling about it ?
Moreover what you think about python ? can it replace C++ in future ?

I will take the exemple of Discord : it is an application available as WEB service or applicatif.
the application is made with "electron" a powerfull tool allowing the conversion of JS to .exe by wrapping a chronium.
it work well, it's smooth and fast to use. But the application is 300 MO where in C++ it would be less than 10MO (in my opinion)

After all, since our systems are today overSized (memory, size, CPU and GPU power) compared to 20 years ago.
Do you think that the differences in size and performance between C++ and some higher-level

power) that they are direct competitors to C++ in the heavy application area?

Note that I only speak about C++ in public domain.

to replace it based on my jobs interviews)

Thanks in advance for all of your responses.

Xemuth

Subject: Re: Future of C++
Posted by zsolt on Mon, 09 Dec 2019 18:49:30 GMT
View Forum Message <> Reply to Message

Here, in Hungary, I can see the same tendency. Web development world is very loud because they are very hungry for coders and it is hard for them to find good people.
I think, the best programmers want interesting work, not web development with some scripting.
If you are confident in your knowledge as a C++ programmer, you will find interesting work in

embedded development, backend development (e.g. web services) or developing desktop apps.

BTW, my opinion is, that web development world is changing. They are tending to use well working and simple to use web services and developing the fancy designed pages on top of it with no real programmers.

Subject: Re: Future of C++
Posted by Novo on Mon, 09 Dec 2019 19:54:16 GMT
View Forum Message <> Reply to Message

C++ is on the rise.
Just try to watch all presentation from CppCon to understand what people are doing with C++.

Subject: Re: Future of C++
Posted by koldo on Tue, 10 Dec 2019 07:14:09 GMT
View Forum Message <> Reply to Message

Yes, I have seen that tendency too about Python.

I have heard that a kind of "mantra" is spread stating that it is more difficult, verbose and error prone to code in C++ than in Python, so Python programmers are simply more productive for a company.

I am tired of explaining that that image belongs to C 30 years ago, and using the wrong C++ libraries.

I myself witnessed a comparative test doing some simple scientific applications (for example, parsing a 4 Gb text file, extracting some summary data, and filling an Excel spreadsheet) and U++ tied or smashed Matlab, Fortran and Python in all ranks as source size, software speed and size.

Despite the facts, many dept. heads and managers follow the fashion and impose Python in their departments. In addition new graduates from STEM careers come with Python background, while at the same time they have been told wearily that C++ is arcane, impossible for them, just for freaky hackers.

(Added: On the other hand, good friends with top scientific level, flip out on what some of my programs do. They think that I am an amazing programmer, and really they are stronger programmers than me, but they have that negative C++ image burned on their brains)

Subject: Re: Future of C++
Posted by mr_ped on Tue, 10 Dec 2019 09:24:17 GMT
View Forum Message <> Reply to Message

For example discord and slack. Slack since complete rewrite and re-architecture is now "acceptable" (before that it was "abysmal"), discord is similar level, although it has more major weird features, like eating almost 100% CPU when my notebook does lose Internet connectivity...

Both apps I have to use daily due to my work and hobby groups, and I'm not happy with their performance, nor the UI/UX is that-much-better that it would make me accept their horrible performance.

And the most funny thing I find about it, lately I have seen some projects written in modern way, that are so convoluted complex mess, that just adding single screen to app takes editing 14 files, and if you forget about any, the app does crash instead of build error... while when you look at the source, it's all "clean code" and other good practices, but somehow over-done IMO. Sometimes I feel like having one single ugly spaghetti procedure with 400 lines can be actually easier to read+maintain+modify than 40 small 10-line files of interfaces/objects/dependency-injection/other-magic which somehow work together as long as you write the correct mapping in another three xml files with navigation and entry points definitions...

Subject: Re: Future of C++
Posted by Didier on Tue, 10 Dec 2019 19:31:27 GMT
View Forum Message <> Reply to Message

Well C++ isn't the most popular at the moment but it has some strong arguments for it's future

quick execution good ressource management (use only what you need !) no licencing : with java, things are very different now... java might die soon ... code confidentiality: you're code secrets are not not given away when you deliver to you're client (never true with interpreted languages) most/all low level codes (like OS, drivers,) are writen in C/C++ a lot of librairies are now available the old argument saying :'it's not cross-OS' is not true anymore => Upp, Qt, ... very good tool base available now : editors, compilers, debuggers, static analysers, profilers, ... long-life time of the code : more recent languages move fast ... and are quickly replaced by some newer tech or no more compatible with you're code :(Most of the hight performance SW are written in C/C++ allmost all good librairies propose a C++ API (even the most recent ones)

The main "drawback" about C++ is that it requires higer starting skills than other more permissive languages ... but the result is only seen latter in the projet phase (to slow, to much ram used, ...) So if you only have java coders on you're hand (since learning C++ is a litle harder because it requires more)... you might go java on you're project (which might not be the best way to go)

If you want to develop web pages, it's not the best way to go
If you want to work close to HW : it's the way to go
For the rest, I think Upp demonstrates that it can do the job great :)