

# Forum:[Hack Forums](#)

Thread: [Rust-Lang and C\C++](#).

**N00b\_n00b** - June 5, 2020, 1:31 pm

Hello,

Is it true that the Rust-Lang is a replacement for the C\C++ and the projects that wrote in the C\C++ porting to the Rust-Lang?

If it is truth, then can it mean C\C++ moving to the history and no one use them anymore?

Thank you.

**saltycracker** - June 5, 2020, 11:51 pm

Rust is new and creating a big stir in the system programming arena but remember Rust is trotting out ideas that have not been tried in the main stream 'system' programming field.

Will Rust eventually overtake C++? Its hard to say but the ideas presented in Rust might outlive both Rust and C++ in the mainstream system programming field.

People have to understand that Rust is new and presenting some radical ideas that have to be field tested and programmer tested.

Some things to consider when considering Rust.

1. Rust's OOP model is so minimalistic that some claim that Rust doesn't support OOP.
2. Rust supports pattern matching and type deconstruction.
3. Rust adopts let binding.
4. Rust has/borrows many concepts from the ML family of languages(Standard ML, OCaml, Haskell).
5. Rust has a borrow checker that will make you pull your hair out in fits of rage.
6. Rust's documentation is lacking.
7. Rust is primarily an expression language and this will confuse you in the most unexpected ways.

**Goodies** - June 7, 2020, 2:58 am

Well, yes, it is far, far safer for novice programmers and allows you to develop safe and virtually bug-free applications so long as you abide by the normal rules of rust (i.e. don't abuse `unsafe{}` keyword). Even multithreading safety is inherent to the language, so it's a great start for programmers who want to begin develop complex low-level applications swiftly and safely.

That said, hacky, niche solutions to all kinds of problems will always be required and nothing beats the power and simple sophistication of C. I'd like to see it slowly become overlapped with C++ users even though I do like C++.

**saltycracker** - June 7, 2020, 4:27 am

(Jun 7, 2020 02:58 AM)Goodies Wrote: [\[CITE\]](#) Well, yes, it is far, far safer for novice programmers and allows you to develop safe and virtually bug-free applications so long as you abide by the normal rules of rust (i.e. don't abuse `unsafe{}` keyword). Even multithreading safety is inherent to the language, so it's a great start for programmers who want to begin develop complex low-level applications swiftly and safely.

That said, hacky, niche solutions to all kinds of problems will always be required and nothing beats

the power and simple sophistication of C. I'd like to see it slowly become overlapped with C++ users even though I do like C++.

Before people go running to Rust...

I've talked to seasoned system level programmers(20+ years experience, some 30+ years experience) who had to be babysat through their first real Rust programming effort. Its not an easy language to write code in and its made worse by the lack of good documentation. A system level programming had this one piece of good advice -> If the are interested in Rust then get this book and read: Programming Rust fast safe systems development by O'Reilly and camp out at this site:

<https://users.rust-lang.org/>

**N00b\_n00b** - June 7, 2020, 1:49 pm

(Jun 7, 2020 04:27 AM)saltycracker Wrote: [\[CITE\]](#) Before people go running to Rust...

I've talked to seasoned system level programmers(20+ years experience, some 30+ years experience) who had to be babysat through their first real Rust programming effort. Its not an easy language to write code in and its made worse by the lack of good documentation. A system level programming had this one piece of good advice -> If the are interested in Rust then get this book and read: Programming Rust fast safe systems development by O'Reilly and camp out at this site:

<https://users.rust-lang.org/>

I guess a Rust-Lang problem is that it is hard.

In your opinion, the large projects that written in C\C++ will rewrite in the Rust-Lang?

**Sajuuk** - June 7, 2020, 11:47 pm

Although in the last few years I've heard a lot about Rust becoming this replacement-esque solution for C++, I don't see it happening anytime soon. It does seem to provide great safeguards against bugged low level code, which neither C nor C++ really have right out of the box. Granted, I haven't used it much, but even after giving it a few go's I am still sticking with C and C++. I guess namely because I'm super familiar with them, but also (in the same light) I don't much care for change. They do what I need them to do, and if it ain't broke don't fix it. While Rust does share numerous, really quite a few syntax similarities with both C and C++, the benefits of organized, safety-concerned code generation is almost never one of my main concerns when developing any type of application. Perhaps that is bad practice in some sense, but I have yet to experience some kind of catastrophic event, something getting away from me due to a lack of attentiveness on my part.

Meh, I won't say I don't see it happening. Those features it has over while also providing much of the same from C++... yeah, I can see the potential. But Rust is still in its early stages while C++ is well-rooted with an enormous audience, both professional and hobbyist. Not to mention whether or not the safety-focused nature of why Rust came about in the first place really will outshine C++ by itself. Rust code runs faster, more efficiently, and safer. That "faster code", by itself, to me is what Rust holds high over both C and C++. Put that as your bait, cast out, and you've got all the optimization-crazed lunatics on the line.