C++ Club Meeting Notes

Gleb Dolgich

2019-12-05

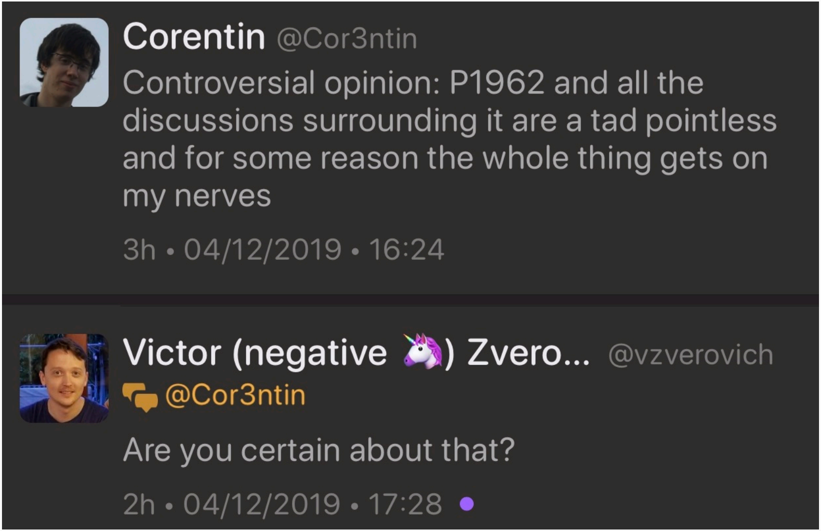
How can you be so certain?

Bjarne Stroustrup

> *We are defining a language for decades of use. A bit of humility is necessary.*

Reddit

How can you be so certain?



Corentin @Cor3ntin

Controversial opinion: P1962 and all the discussions surrounding it are a tad pointless and for some reason the whole thing gets on my nerves

3h • 04/12/2019 • 16:24

Victor (negative 🦄) Zvero... @vzverovich

@Cor3ntin

Are you certain about that?

2h • 04/12/2019 • 17:28 ●

Bjarne Stroustrup on Lex Fridman’s AI Podcast

Podcast page

YouTube

Belfast trip reports

- Bryce Lelbach via Reddit
- Botond Ballo
    - Reddit
    - Papers on GitHub
- Timur Doumler via CppCast
    - Reddit
- Ben Craig via Reddit
- Guy Davidson
    - Reddit

CLion 2019.3

> *A Quality-Targeted Release Focused on Performance and Some Long-Awaited Enhancements*

- Announcement
- Download
- Reddit

MinGW Distro 17.0: GCC 9.2.0 and Boost 1.71.0 for Windows

- Stephan T. Lavavej
  - Reddit

> *I've maintained this distro for over 14 years, and I still don't serve ads, sell anything, or accept donations.*

- GCC 9

Sourcetrail is now free and open-source software

Blog post

GitHub

Reddit

When is it justified to use C++ for a project?

Reddit (1), Reddit (2)

Is it time for a rebased Boost2 that assumes C++20 as its starting point?

Reddit

TL;DR: No.

Eliminating the Static Overhead of Ranges

Colby Pike — Reddit

Without ranges

```cpp
vector<string> child_names;
for (auto& person : all_people) {
    if (person.age < 14) {
        child_names.push_back(person.name);
    }
}
```

With ranges

```cpp
auto children_names =
    all_people
    | filter([](const auto& person) { return person.age < 14; })
    | transform([](const auto& person) { return person.name; })
    | to_vector;
```

The arrow operator (1/2)

StackOverflow:

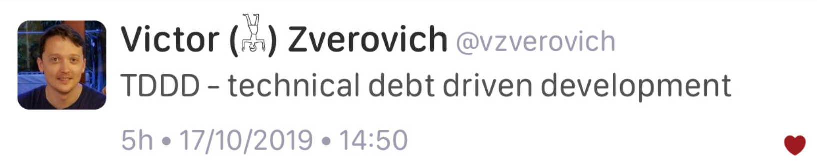
> *The `operator->` has special semantics in the language in that, when overloaded, it reapplies itself to the result. While the rest of the operators are applied only once, `operator->` will be applied by the compiler as many times as needed to get to a raw pointer and once more to access the memory referred by that pointer.*

The arrow operator (2/2)

```cpp
struct A { void foo(); };
struct B { A* operator->(); };
struct C { B operator->(); };
struct D { C operator->(); };
int main() {
    D d;
    d->foo();
}
```

Thanks to Martin Waplington for suggesting this.

Twitter



Victor (🤸) Zverovich @vzverovich

TDDD - technical debt driven development

5h • 17/10/2019 • 14:50

Quote

Ellen Ullman:

> *We build our computer (systems) the way we build our cities: over time, without a plan, on top of ruins.*