C++ Club Meeting Notes

Gleb Dolgich

2019-11-28

Podcast page

YouTube

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

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

- Announcement
- Download
- Reddit

- 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

Reddit (1), Reddit (2)

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

Reddit

TL;DR: No.

Corentin Jabot

P0443R11 The Unified Executors Proposal

Eliminating the Static Overhead of Ranges

Colby Pike — Reddit

Without ranges

```
1 vector<string> child_names;
2 for (auto& person : all_people) {
3     if (person.age < 14) {
4         child_names.push_back(person.name);
5     }
6 }
```

With ranges

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

- Wikipedia
- We don't need no stinking expression templates by Andy G
  - Reddit

Change standard containers' `size()` method to return signed integer?

Reddit

Is requiring lambdas to explicitly list what they capture a good coding standard?

Reddit

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.*

```
1 struct A { void foo(); };
2 struct B { A* operator->(); };
3 struct C { B operator->(); };
4 struct D { C operator->(); };
5 int main() {
6     D d;
7     d->foo();
8 }
```

Thanks to Martin Waplington for suggesting this.

P1930R0

Reddit

Robert Ramey:

> *The value of a paper like this would be to narrow the scope or domain of a problem to something that would be useful component in solving bigger problems. This paper does the opposite — expanding the domain to encompass the whole world of physics.*

C++ `std::string_view` for better performance: An example use case

Article

Reddit

Arthur O'Dwyer: std::string_view is a borrow type

> *Borrow types are essentially "borrowed" references to existing objects. They lack ownership; they are short-lived; they generally can do without an assignment operator. They generally appear only in function parameter lists; because they lack ownership semantics, they generally cannot be stored in data structures or returned safely from functions.*

cppreference: std::basic_string_view (C++17)

Reddit

JF Bastien @jfbastien

Today's episode of "something I didn't know about C and C++":

```
int size() {
  struct empty {};
  return sizeof(struct empty);
}
```

This code returns 0 in C and 1 in C++, because empty structs have different size in both languages. 🥺

💬 thread

20w • 21/02/2019 • 20:33

Arthur O'Dwyer

> *Here's how to build a "Hello world" program using*
> *Clang's implementation of C++2a Modules, as it cur-*
> *rently stands as of November 2019.*

Reddit

Vinnie Falcou:

> *A survey of existing JSON libraries shows impressive diversity and features. However, no library is known to meet all of the design goals mentioned here. In particular, we know of no library that supports incremental parsing and serialization, and also supports custom allocators robustly.*

Reddit — GitHub (C++11, Boost License) — Docs — Benchmarks

> *This is currently NOT an official Boost library.*

http://www.lurklurk.org/linkers/linkers.html

```
1 g++ -o test1 test1a.o test1b.o
2 test1a.o(.text+0x18): In function `main':
3 : undefined reference to `findmax(int, int)'
4 collect2: ld returned 1 exit status
```

> *If your reaction to this is 'almost certainly missing*
> *extern "C"' then you probably already know everything*
> *in this article.*

Victor (🦀) Zverovich @vzverovich
TDDD – technical debt driven development
5h • 17/10/2019 • 14:50

Ellen Ullman:

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