C++ Club Meeting Notes

Gleb Dolgich

2018-09-27

CppCon 2018 Slides and Materials

GitHub

The C++ Alliance

- Website
  - Marshall Clow joined as a staff engineer
- C++ Slack workspace invitation
- Reddit

CLion 2018.3 EAP

- Blog post
- Remote development

Use the official Boost.Hana with MSVC 2017 Update 8 compiler

- Post

  Today, we're happy to announce that the vcpkg version of Boost.Hana now just points to the official master repo, instead of our fork.

VS2017 15.9: Step Back – Going Back in C++ Time

- Post

  In the most recent, 15.9, update to Visual Studio 2017 Enterprise Edition, we've added "Step Back" for C++ developers targeting Windows 10 Anniversary Update (1607) and later. With this feature, you can now return to a previous state while debugging without having to restart the entire process.

Catch 2.4.0

- Release
  - Added experimental support for data generators
  - Added support for compiling and running Catch without exceptions
- Reddit
  - Why CATCH?

std::optional: How, when, and why

- VCBlog post
  - Reddit

Scott Meyers: The Errata Evaluation Problem

- Post
  - Reddit 1
  - Reddit 2

I no longer plan to update my books to fix technical errors.

Lifetime Profile by Herb Sutter

- Blog post
  - Includes Godbolt links to try it out using an experimental Clang-based implementation
- CppCon 2015 talk at 29:06
- PDF
- Bind Returned/Initialized Objects to the Lifetime of Parameters, by R. Smith and N. Josuttis

  <This paper> shows how to efficiently diagnose many common cases of dangling (use-after-free) in C++ code, using only local analysis to report them as deterministic readable errors at compile time.

C++ Now 2018: Compile Fast, Run Faster, Scale Forever: A look into the **sol** Lua library, by JeanHeyd Meneide

- Video
- sol2 on GitHub (MIT, but will become Apache 2), Docs, Tutorials, Examples
  - Used in: databases (Redis), OS components, games and game engines, HPC, GUI scripting (Waze, OpenMPT), chat servers etc.
  - Written on top of Lua C API
- Soon: paper on `std::optional<T&>` (rebind on assignment)
- Soon: sol3

What can C++ do for embedded systems developers? - Bjarne Stroustrup

- NDC TechTown - August 2018 - Keynote
  - Reddit
- Most people in the audience use C++ alone or with C and other languages for embedded development
- Grace Hopper is the grandmother of COBOL
- Zero overhead doesn't mean zero cost
- Use predictable subset of C++ for small safety-critical systems (no dynamic memory)
- Allocate at startup, use as a pool, don't ever free (memory fragmentation)
- Unpredictable: `new`/`malloc`, `throw`, RTTI, standard containers, `std::function`

- Start of the series
  - Reddit

Range-based for over a struct object

- Post
- magic_get library (GitHub)

```cpp
struct {
    std::uint32_t fw_version = 0;
    std::uint16_t sector_0_version = 0;
    std::string id = "";
    std::array<std::uint8_t, 6> options{};
} data;

boost::pfr::for_each_field(std::forward<decltype(data)>(data), [](auto&& val)
{
    Process(val);
});
```

Non-Virtual Destructors, by Anders Knatten (CppQuiz)

Post

§5.3.5/3 in the C++11 standard:

> *If the static type of the object to be deleted is different from its dynamic type, the static type shall be a base class of the dynamic type of the object to be deleted and the static type shall have a virtual destructor or the behavior is undefined.*

Blog: C++ on a Friday

Quote on OOP

Joe Armstrong, creator of Erlang:

> *You wanted a banana but what you got was a gorilla holding the banana and the entire jungle.*

Twitter: Almost Always Auto