

C++ Club Meeting Notes

Gleb Dolgich

2020-01-23

- [Mailing 2020-01](#)
- [Reddit](#)

Selected papers

- [P2005R0 2D Graphics: A Brief Review](#)
 - [Reddit thread](#)
- [P1371R2 Pattern Matching](#)
- [P1881R1 Epochs: a backward-compatible language evolution mechanism](#)
- [P1950R0 indirect_value: A Free-Store-Allocated Value Type For C++](#)
- [P2027 Moved-from objects need not be valid](#)
- [P2067R0 Allowing trailing commas in ctor-initializer](#)

Selected papers

- P2011R0 A pipeline-rewrite operator: $x \mid > f(y) \Rightarrow f(x, y)$
- P2076R0 Previous disagreements on Contracts
- P2087R0 Reflection Naming: fix `reflexpr` (*Naming is hard. But come on, seriously?*)
- P2043R0 Don't `constexpr` All The Things
- P2062R0 The Circle Meta-model

A new decade, a new tool: libman

- Colby Pike (vector-of-bool)
- Reddit
- GitHub
- Specification

libman is a new level of indirection between package management and build systems.

dds is Drop-Dead Simple build and package manager.

- CppCon 2019: Robert Schumacher “How to Herd 1,000 Libraries”

- Lucid C++ index
- Reddit

A hidden gem: `inner_product`

- Article

A hidden gem: inner_product

4



Conor Hoekstra @code_report

@cjdb_ns & @TartanLlama

This makes me so incredibly happy! I literally just yesterday googled, C++17 / C++20 zip to see if they had anything, because I wrote some code in both C++ and [Python](#) and Python was so much more beautiful.

```
int solve(int h, vector<int> w, vector<int> l) {  
    int p = 0;  
    for (int i = 0; i < w.size(); ++i)  
        p = max(p, w[i] - l[i] / 4);  
    return max(0, p - h);  
}  
  
def solve(h, w, l):  
    p = max(a - b//4 for a, b in zip(w, l))  
    return max(0, p - h)
```

29w • 03/12/2018 • 17:47



Conor Hoekstra @code_report

@cjdb_ns & @TartanLlama

Also, I just discovered `std::inner_product` – a beautiful temporary solution to a lack of `zip`.
[#cpp](#) [#inner_product](#)

```
int solve(int h, vector<int> w, vector<int> l) {  
    return max(0, inner_product(begin(w), end(w), begin(l), 0,  
        [](auto a, auto b) { return max(a, b); },  
        [](auto a, auto b) { return a - b / 4; }) - h);  
}
```

27w • 16/12/2018 • 09:30



- [GitHub: C++11, MIT](#)
- [Reddit](#)

Argumentum is a C++17 library for writing command-line program interfaces, inspired by Python argparse

- [GitHub: C++17, MPL](#)
- [Reddit](#)

- Raymond Chen: How can I handle both structured exceptions and C++ exceptions potentially coming from the same source?
 - [Reddit](#)
- Raymond Chen: Can I throw a C++ exception from a structured exception?

- Website

Move semantics, introduced with C++11, has become a hallmark of modern C++ programming. However, it also complicates the language in many ways. Even after several years of support of move semantics experienced programmers struggle with all the details of move semantics. And style guides still don't recommend the right consequences for programming even of trivial classes.

- Reddit

Happy there is a book. Not happy this requires a book.

[C++ coroutines] Initial implementation pushed to GCC master

- Message
- Reddit

*This is not enabled by default (even for `-std=c++2a`),
it needs **`-fcoroutines`***

Concepts pushed to Clang master



Saar Raz

@saarraz1



[#clang](#) [#concepts](#) [#trunk](#).

```
03:09:53 projects > cd /Users/saarraz/llvm-project & clang git push trunk master
Enumerating objects: 48, done.
Counting objects: 100% (48/48), done.
Delta compression using up to 8 threads
Compressing objects: 100% (24/24), done.
Writing objects: 100% (25/25), 4.10 KiB | 2.05 MiB/s, done.
Total 25 (delta 23), reused 1 (delta 1)
remote: Resolving deltas: 100% (23/23), completed with 23 local objects
To github.com:llvm/llvm-project.git
a156da5fb36..b933d37cd37 master -> master
```

2:31 AM · Jan 22, 2020 · [Twitter for Android](#)

- [Reddit](#)

“Making new friends” idiom by Dan Saks

Wikibooks

The goal is to simplify creation of friend functions for a class template.

```
1 template<typename T>
2 class Foo {
3     T value;
4 public:
5     Foo(const T& t) { value = t; }
6     friend ostream& operator <<(ostream& os, const Foo<T>& b)
7     {
8         return os << b.value;
9     }
10 };
```



Victor (negative 🦄) Zvero... @vzverovich
SFINAE is not an error.

23h • 18/11/2019 • 17:41