

C++ Club Meeting Notes

Gleb Dolgich

2018-01-18

Spectre mitigations in MSVC

- ▶ [Article](#)
- ▶ [Reddit thread](#)

Podcast

Tweet by STL



Stephan T. Lavavej

@StephanTLavavej

↑ 6 Replies, 1 Quote



It turned out that all of 2017 was speculatively executed based on an incorrect branch prediction for 2016. However, the cache effects cannot be undone.

04/01/2018, 05:09 (Thursday)
Tweetbot for iOS

Retweeted by [@robwiring](#)
04/01/2018, 16:54

240 Likes

71 Retweets

Thread >

Free books from O'Reilly

- ▶ Website
- ▶ Practical C++ Metaprogramming, by Edouard Alligand and Joel Falcou
- ▶ C++ Today, by Jon Kalb and Gašper Ažman

Website

A curated list of awesome C/C++ frameworks, libraries, resources, and shiny things.

Article

Exceptions vs expected: Let's find a compromise

Post – [Reddit thread](#)

- ▶ Exceptions: Writing exception safe code is hard
- ▶ Exceptions are not easily composable: There is only one current exception, you can't have multiple ones. This was a problem, for example, for the C++17 [parallel algorithms](#). What if an exception is thrown in multiple of the worker threads? How to report all of them back to the caller? The implementation gave up on solving that problem and just decided to terminate the program if any exception is thrown.
- ▶ As with most things, the disadvantages are opposites: “exceptions are too implicit!” — “ADTs are too explicit!”

Exceptions vs expected: Let's find a compromise (cont.)

You're writing a library and have a function that may fail — how do you report the error? If you want to write a truly flexible API, you have to do both: exceptions and error return codes. Because sometimes the error is “exceptional” and sometimes it isn't.

Swift chose to use exceptions. (*Swift doesn't use exceptions — GD*)

defer allows guaranteed cleanup without the boilerplate of RAII” (*defer is equivalent to finally, and what boilerplate is that? — GD*)

Exceptions vs expected: Let's find a compromise (cont.)

Rust: ADT for error handling. `Result<T,E>`.

```
1 result = foo();
2 if (!result)
3     return result.error();
4 // do something with result.value()
5
6 // old way
7 result = try!(foo());
8
9 // new built-in language feature
10 result = foo()?;
```

Introduction to proposed `std::expected` - Niall Douglas - Meeting C++ 2017

YouTube

How to handle errors in constructors without exceptions

Post

If you do not have exceptions, reporting errors from a constructor is impossible without sacrificing guarantees. Where possible, simply use an alternative and non-recoverable way of error reporting.

If that's not applicable, provide a static function as the only way to create the object. It does not return an object directly, but an optional type. Carefully craft the implementation, so that the actual private constructor will only be called, when no operation can fail. Then every object will be valid, just like it were the case when using exceptions.

Post

- ▶ C++ Core Guidelines: Error Handling
- ▶ Error handling links and resources

YouTube

Reddit: What are the weakest points of C++ in your opinion?

Thread

- ▶ Setting up a build environment, adding third party libraries, like pulling teeth. #
 - ▶ [Conan V1.0.0 beta](#), [build2](#), [cppget](#), [cget](#) for CMake; Modules won't help here.
- ▶ Unicode support #
- ▶ Rust and D got mentioned
- ▶ Template metaprogramming is ridiculously verbose and complicated (but Concepts will help) #
- ▶ “The biggest strength and weakness of C++ is its legacy” #
- ▶ Header files #

Reddit: What are the weakest points of C++ in your opinion? (cont.)

- ▶ “Every time there is a proposal accepted in a given standard, it is getting fixes and amendments for the next ten years” #
- ▶ “Complexity. It could be reasoned that C++ is the most complex programming language ever created.”
- ▶ String manipulation #
- ▶ The preprocessor #
- ▶ Lack of a consistent stable ABI #
- ▶ Lack of reflection #
- ▶ Compile times
- ▶ Irregular void

YouTube

- ▶ Know the language (all of it)
- ▶ Know C++ idioms
- ▶ Follow C++ Core Guidelines
- ▶ Write simple code

Reddit

```
1 void spanish_inquisition() noexcept
2 {
3     throw std::unexpected;
4 }
```

Tweet



Adam of Earth

@AdamOfEarth

↑ 19 Quotes



Me: "Well, at least my computer doesn't judge me."

Computer:

12/12/2017, 15:28 (Tuesday)

Twitter Lite

Retweeted by @dwarfland

14/12/2017, 12:36

3171 Likes

1093 Retweets

Thread >



```
//-----  
//  
//   This code was  
//   generated by a tool.  
//
```

Kevlin Henney:

A common fallacy is to assume authors of incomprehensible code will be able to express themselves clearly in comments.

Steve C. McConnell:

The problem with quick and dirty is that the dirty remains long after the quick has been forgotten.