C++ Club Meeting Notes

Gleb Dolgich

2018-10-11

JetBrains C++ team at CppCon 2018: trip report

Post by Anastasia Kazakova and Phil Nash

Deterministic Disappointment, by Niall Douglas (1/4)

- Video
  - Example of a poorly-designed file system API
  - Illustration of how expensive it is to allocate memory
  - History of C++ exceptions
  - Throwing/catching is approximately 300 times slower than other error handling mechanisms
  - Many code bases ban exceptions (high maintenance costs, allocation needed, unpredictable)
  - Explanation of C++ `<system_error>` and how it is used wrong everywhere
    - P1028: SG14 `status_code` and standard error object is better
- Slides
- Niall's proposal N2289/P1095
- Herb Sutter's proposal P0709

Deterministic Disappointment, by Niall Douglas (2/4)

Let's add exceptions to C (and Rust/Go/Python etc.)!

- If a C function is marked with `fails(E)` its calling convention changes to return `union` of return type `T` and error type `E`.
- Discriminant is lightweight and architecture-specific (for example, CPU carry flag)
- Fails-functions must be explicitly called with `catch(...)` or `try(...)`
- `fails_errno` is a boilerplate expansion and solves the global `errno` problem

Deterministic Disappointment, by Niall Douglas (3/4)

C++

- Functions can be marked with:
  - `throws`
  - `throws(E)`
  - `fails(E)`
  - `noexcept`
  - `nothing`

Deterministic Disappointment, by Niall Douglas (4/4)

- One possible implementation of P0709
- Solves a few long-standing problems in C and POSIX
- Enables C code to call C++ code without exception translation wrappers (also Rust, Go, Python etc.)
- C++ can send/get exceptions to/from C

IT Hare posts on std::error exceptions

- 1: The Good
  - “While existing C++ exceptions DO have (about)-zero runtime CPU cost when the exception is not fired, it comes at the cost of the path-when-the-exception-IS-fired being huge, and worse – being next-to-impossible to predict in advance.”
  - “With existing exception model, we cannot see which functions are allowed to throw. This leads us to the situation where we need either to (a) think that everything out there can throw (leading to very inefficient use of our brains to make everything out there exception-safe), or (b) forget about exception safety entirely (which actually happens way too often in real-world projects).”
  - “No single error handling method is good enough for ALL the projects – which in turn leads to creation of C++ dialects, with some of the projects using exceptions, and some others using error codes.”

IT Hare posts on std::error exceptions

- 2: The Discussion
- 3: Unchecked exceptions for C++
  - “There EXIST real-world cases when failing hard is NOT a good option” (Ariane 5)
  - “Fail-Fast-AND-Soft”
  - “Unchecked exceptions MAY be thrown out of nothrow functions without causing trouble” *(Hmmm...)*
  - ““unchecked” std::errors are treated as “something which should never ever happen, but in practice MAY occur as a result of potentially-recoverable bug””

Concepts TS vs. C++20 Concepts

Reddit

> *Why is the short-form of concept constraints isn't included in C++20?*

Exploring C++ types with `puts(__PRETTY_FUNCTION__)`

- Post
  - Reddit
  - boost::core::typeinfo

```
template<class T>
void f() {
    puts(__PRETTY_FUNCTION__); // __FUNCSIG__ in MSVC
}

#define EXPLORE(expr) \
    printf("decltype(" #expr ") is... "); \
    f<decltype(expr)>();
```

Mutexes are passé

- Post
  - Reddit

Reboot Your Dreamliner Every 248 Days To Avoid Integer Overflow

- Article
  - Reddit

This condition is caused by a software counter internal to the Generator Control Units (GCUs) that will overflow after 248 days of continuous power. We are issuing this to prevent loss of all AC electrical power, which could result in loss of control of the airplane.

Twitter

SAFETY FAIL
@Safethink

↑ 3 Quotes

Every workplace needs a Todd.

08/07/2018, 00:08 (Sunday)
Twitter for iPhone

83 Likes | 24 Retweets | Thread >