14 September 2017

Boost 1.65.1 released

Release notes

LLVM 5 released

- Announcement
- (Why the big jump from 4.0.1? With the new versioning scheme, the major version is incremented for each major release: http://blog.llvm.org/2016/12/llvms-new-versioning-scheme.html)
- Release notes

Clang 5 released

- Release notes
- Support for the C++17 standard has been completed.
- C++ coroutines TS implemented
- New UBSan diagnostics

Two-phase name lookup support comes to MSVC

Blog post

```cpp
#include <cstdio>

void func(void*) { std::puts("void*"); }

template<typename T>
void g(T x) { func(0); }

void func(int) { std::puts("int"); }

int main()
{
    g(3.14);
}
```

TDD in C++

- Article by Alex Ott
- Another article
- Reddit thread
- Google Test, now with Google Mock
- Book: Modern C++ Programming with Test-Driven Development (Amazon, Review)

C++Now 2017: Kris Jusiak “Towards Painless Testing”

YouTube

Hell is a multi-threaded C++ program

Mark Bessey

- POSIX Threads, Mach Threads, Windows Threads, Java Threads, and C# Threads all work very much the same.
- The POSIX threading model is just about the simplest possible implementation of multi-threading you could have: all of your threads share the same address space.
- With Pthreads, *it's too easy to make something that almost works*.
- The other major model for multi-threading is known as message-passing multiprocessing: your threads don't share any state by default.
- Two popular variants of the message-passing model are "Communicating Sequential Processes" and the "Actor model".

Another thread on . . . threads

Mark Bessey

- Do consider whether you need to use threads at all
- Don't use threads to avoid blocking on I/O
- Do know what each thread in your program is for
  - Don't spawn threads in response to external events
- Don't reinvent the wheel
  - Do stay on the well-trodden path
- Do consider developing a strategy for detecting and/or avoiding deadlocks
- Do consider a message-passing design
- Don't hold a lock or semaphore any longer than actually necessary
- Do use multiple threads to get better performance on multi-processor systems

CppCast: Volta and CUDA C++

Podcast

Olivier Giroux has worked on eight GPU and four SM architecture generations released by NVIDIA. Lately, he works to clarify the forms and semantics of valid GPU programs, present and future. He was the programming model lead for the new NVIDIA Volta architecture. He is a member of WG21, the ISO C++ committee, and is a passionate contributor to C++'s forward progress guarantees and memory model.

See-phit: A template engine that uses compile-time HTML parsing

- GitHub
- C++14 (uses `constexpr` functions)
- Fails to compile if HTML is malformed
- The maximum number of nodes and attributes per parse is hardcoded to 1024
- LGPL 3.0 (how does that even work with a compile-time library?)

Seven Ineffective Coding Habits of Many Programmers

Kevlin Henney, ITT 2016

Video

- Content-to-noise ratio of the code
- Refactoring-safe code formatting
- Naming things
- Word cloud for your code

Zach Laine: Pragmatic Type Erasure

```
1  struct anything {
2    template<typename T> anything(T t);
3    template<typename T> anything& operator =(T t);
4    int value() const {return handle_->value();}
5
6    struct handle_base {
7      virtual ~handle_base() {}
8      virtual handle_base* clone() const = 0;
9      virtual int value() const = 0;
10   }
11
12   template<typename T>
13   struct handle: public handle_base {
14     handle(T value);
15     virtual handle_base* clone() const;
16     virtual int value() const {return value.value();}
17     T value;
18   }
19
20   std::unique_ptr<handle_base> handle_;
```

Cheinan Marks: Practical Type Erasure

- Type erasure is the glue between generic (front end) and OO (back end) code
- Used in `std::shared_ptr`, `std::function`
- Type-safe configuration system using `boost::any`
- Type erasure article by Thomas Becker, Artima

CppCon 2014: Jens Weller “0xBADC0DE”

Video

- Antipatterns
- Bad working environments
- Non-developer managers
- Absence of tests

A hamster wheel can look like a career ladder from the inside.

Post on Medium

YouTube

- Type erasure
- Code examples

Rant

- A great presentation ruined by terrible sound, editing and postprocessing.
- Laptop mic ▯ tinny inaudible sound
- Editor's "comments"
- Lack of sync between the video and the slides
- 15 FPS postal stamp-size video
- Futile attempts by the editor to move the video window around to avoid obscuring the slides
- Non-working mouse (with audible clicks, no less)