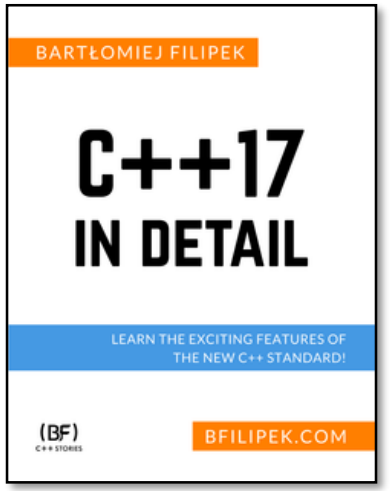
EMPTY BASE CLASS OPTIMISATION, [[NO_UNIQUE_ADDRESS]]

And Other C++20 Attributes…

About me

- See my coding blog at: www.bfilipek.com
  - Slowly moving to cppstories.com
- ~13y coding experience
- Microsoft MVP, since 2018
- C++ ISO Member
- @Xara.com
  - Mostly text related features for advanced document editors
- Somehow addicted to C++ ☺



C++17 In Detail

Xara Cloud Demo

Microsoft® MVP Most Valuable Professional

cpp-polska.pl
BLOG PROGRAMISTYCZNY

The plan

- Unique_ptr and a custom deleter
- Digging into the STL implementation
- Empty Base Class Optimisation
- [[no_unique_address]]
- Other C++20 Attributes and Features

Custom deleter for unique_ptr

- Background, the article from 2016:
  - https://www.bfilipek.com/2016/04/custom-deleters-for-c-smart-pointers.html#custom-deleter-for-uniqueptr
  - Why unique_ptr is just 8 bytes (one pointer) with default deleters? Or stateless deleters?
- The working example with a custom deleter:
  - http://coliru.stacked-crooked.com/a/e2638649daffd406

Digging into the STL implementation

- Let’s try to understand the internal implementation of unique_ptr
  - We can go to:
  - https://github.com/microsoft/STL
  - https://github.com/microsoft/STL/blob/master/stl/inc/memory#L2435
- Unique_ptr seems to hold the pointer and the deleter in something called compressed pair
  - It has two specialisation: the first one if both types are non empty and the second one if the first type is empty
    - In that case it just declares one member field and derive from the empty type
    - By inheritance we get access to all of the type member functions

Empty Base Class Optimisation

- When a class is empty it costs at least 1 byte on stack, but if you inherit from such a class then in costs you nothing
  - https://en.cppreference.com/w/cpp/language/ebo
- See an example:
  - http://coliru.stacked-crooked.com/a/affe60d81ac52163
- Some example:
  - https://github.com/microsoft/STL/blob/master/stl/inc/xmemory#L1319
  - GCC tuple: https://github.com/gcc-mirror/gcc/blob/master/libstdc%2B%2B-v3/include/bits/unique_ptr.h#L201

[[no_unique_address]]

- Let’s rewrite the compressed pair into something really simple thanks to the new attribute from C++20
  - http://coliru.stacked-crooked.com/a/7ccae3a3168e73b9

Attributes in C++17

[[noreturn]]	indicates that the function does not return
[[carries_dependency]]	indicates that dependency chain in release-consume std::memory_order propagates in and out of the function
[[deprecated]] [[deprecated("reason")]]	indicates that the use of the name or entity declared with this attribute is allowed, but discouraged for some reason
[[fallthrough]]	indicates that the fall through from the previous case label is intentional and should not be diagnosed by a compiler that warns on fall-through
[[nodiscard]]	encourages the compiler to issue a warning if the return value is discarded
[[maybe_unused]]	suppresses compiler warnings on unused entities, if any

https://www.bfilipek.com/2017/07/cpp17-in-details-attributes.html
https://www.bfilipek.com/2017/11/nodiscard.html

Attributes in C++20

[[nodiscard("reason")]]	encourages the compiler to issue a warning if the return value is discarded
[[likely]] [[unlikely]]	indicates that the compiler should optimize for the case where a path of execution through a statement is more or less likely than any other path of execution
[[no_unique_address]]	indicates that a non-static data member need not have an address distinct from all other non-static data members of its class

- Apply [[nodiscard]] to the standard library - P0600
- [[nodiscard]] for constructors - P1771

Summary

- Empty Base Class Optimisation relies on the fact that if you inherit from a empty class then you don’t need more memory, but you get access to member functions.
  - The technique is however a bit complicated
  - Thanks to the new C++20 attribute – [[no_unique_address]] the code can be much simpler
- Since C++11 the Standard has been taking vendor specific annotation syntax into a common form of [[attrib_name]].
- C++20 adds new attributes: [[no_unique_address]], [[likely]] and [[unlikely]]

C++ Lambda Story

Free coupon code for Cracow C++ User Group
https://leanpub.com/cpplambda/c/cppcracow1bsgfa