Require `span` & `basic_string_view` to be Trivially Copyable

Document number: P2251R0
Date: 2020-11-09
Project: Programming Language C++, Library Evolution Working Group
Reply-to: Nevin "☺" Liber, nliber@anl.gov

Table of Contents

Introduction

Given its definition, it is strongly implied that `span` & `basic_string_view` are trivially copyable, but that is not yet a requirement.

Motivation and Scope

Both `span` and `basic_string_view` have:

- Defaulted copy constructors
- Defaulted copy assignment operators
- Defaulted destructors
- Exposition-only types consisting of a raw pointer and a `size_t`.
- Many member functions that are `constexpr`, and in C++17 would have required trivial destruction (for `basic_string_view` as `span` was added in C++20).

Because of the above, it is strongly implied that these are trivially copyable types. However, that is not a stated requirement.

Furthermore, both libstdc++ and libc++ implement them as trivially copyable types: https://godbolt.org/z/nWY3dv.

I ran this by LWG and there was support for it with no objections.

Impact on the Standard

This is purely additive to the standard.

Technical Specifications

These changes are relative to C++20:

In [span.overview], add:

`span<ElementType, Extent>` is a trivially copyable type.

`ElementType` is required to be a complete object type that is not an abstract class type.

In [string.view.template.general], add:

The complexity of `basic_string_view` member functions is O(1) unless otherwise specified.

`basic_string_view<charT,traits>` is a trivially copyable type.

Acknowledgements

Thanks to Barry Revzin for pointing out that `basic_string_view` is strongly implied but not strictly required to be trivially copyable. Thanks to Jonathan Coe, Casey Carter for supporting this direction and encouraging me to write this paper, and an additional thanks to Jonathan Wakely for that as well as reviewing the wording.

This was supported by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of two U.S. Department of Energy organizations (Office of Science and the National Nuclear Security Administration) responsible for the planning and preparation of a capable exascale ecosystem, including software, applications, hardware, advanced system engineering, and early testbed platforms, in support of the nation's exascale computing imperative. Additionally, this research used resources of the Argonne Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC02-06CH11357.

References

C++20: Programming Languages – C++, International Standard ISO/IEC 14882, Sixth edition 2020-10

Stack Overflow: Is std::string_view trivially copyable?