Project: Programming Language C++, Evolution Working Group
Document number: P0257R0
Date: 2016-02-11
Reply-to: Neil MacIntosh neilmac@microsoft.com

A byte type for increased type safety

Contents

Introduction

This paper suggests a simple distinct standard library type `std::byte` specifically for representing a byte of storage. A simple modification to the current language rules around type aliasing is needed for this type to fully serve its purpose.

Motivation and Scope

Many programs require byte-oriented access to memory. Today, such programs must use either the `char`, `signed char`, or `unsigned char` types for this purpose. However, these types perform a "triple duty". Not only are they used for byte addressing, but also as arithmetic types, and as character types. This multiplicity of roles opens the door for programmer error – such as accidentally performing arithmetic on memory that should be treated as a byte value – and confusion for both programmers and tools.

Having a distinct *byte* type would improve type-safety, by distinguishing byte-oriented access to memory from accessing memory as a character or integral value. It improves readability. Having the type would also make the intent of code clearer to readers (as well as tooling for understanding and transforming programs).

However, adding a new fundamental type to the language is unnecessary because it can be expressed in library almost entirely. It also creates substantial risk of breaking legacy code through adding a new keyword. This paper proposes instead adding a `byte` type as a definition in the standard library. This approach is both backward-compatible and forward looking.

Such a type could be defined in the standard library as simply as:

```
enum class byte : unsigned char {};
```

A definition like this has the advantage of not supporting arithmetic operators, not being a character type, but being readily convertible to and from its underlying type with an explicit cast operation.

This proposal suggests clarifying the existing paragraph 3.10/10 of the standard text to allow use of pointers of type `std::byte*` to access object representation of an object. That can be achieved in at least ways: (a) allow any scoped enumeration with a character type as its underlying type to alias any other storage; or (b) specifically singling out `std::byte` as a permitted type for object representation access.

Impact on the Standard

It is proposed to modify the wording of the standard to make such a library definition of *byte* useful.

One possibly approach (with relevant proposed wording offered below) would be to amend 3.10/10.8 to include scoped enumeration types whose underlying type is unsigned char and who have no enumerators declared.

Proposed Wording Changes

The following proposed changes are relative to N4567 [1]. They would consist of one addition, highlighted here in green.

3.10 Lvalues and rvalues [basic.lval]

10 If a program attempts to access the stored value of an object through a glvalue of other than one of the following types the behavior is undefined:

—(10.1) the dynamic type of the object,

—(10.2) a cv-qualified version of the dynamic type of the object,

—(10.3) a type similar (as defined in 4.4) to the dynamic type of the object,

—(10.4) a type that is the signed or unsigned type corresponding to the dynamic type of the object,

—(10.5) a type that is the signed or unsigned type corresponding to a cv-qualified version of the dynamic type of the object,

—(10.6) an aggregate or union type that includes one of the aforementioned types among its elements or nonstatic data members (including, recursively, an element or non-static data member of a subaggregate or contained union),

—(10.7) a type that is a (possibly cv-qualified) base class type of the dynamic type of the object,

—(10.8) a `char` or `unsigned char` type, or a scoped enumeration type that has an underlying type of `char` or `unsigned char` and that has no enumerators defined.

Or alternatively:

—(10.8) a `char` or `unsigned char` type, or `std::byte`.

Acknowledgements

Gabriel Dos Reis originally suggested the definition of byte as a library type by using a scoped enumeration and also provided valuable review of this proposal.

References

[1] Richard Smith, "Working Draft: Standard For Programming Language C++", N4567, 2015, [Online], Available: http://open-std.org/JTC1/SC22/WG21/docs/papers/2015/n4567.pdf