

## alignas

## Chapter 2 Conditionally Safe Features

specifying such substantially stricter alignment on consecutive data members necessarily increases the size of the object; see *Potential Pitfalls — Overlooking alternative approaches to avoid false sharing* on page 178.

### Potential Pitfalls

#### Underspecifying alignment is not universally reported

The Standard is clear when it comes to underspecifying alignment<sup>7</sup>:

The combined effect of all *alignment-specifiers* in a declaration shall not specify an alignment that is less strict than the alignment that would be required for the entity being declared if all *alignment-specifiers* were omitted (including those in other declarations).

The compiler is required to honor the specified value if it is a **fundamental alignment**,<sup>8</sup> so imagining how underspecifying alignment would lead to anything other than an ill-formed program is difficult:

```
alignas(4) void* p;           // Error, alignas(4) is below minimum, 8.

struct alignas(2) S { int x; }; // Error, alignas(2) is below minimum, 4.

struct alignas(2) T { };
struct alignas(1) U { T e; }; // Error, alignas(1) is below minimum, 2.
```

Each of the three errors above are reported by Clang. MSVC and ICC issue a warning, whereas GCC provides no diagnostic at all, even in the most pedantic warning mode. Thus, one could write a program, involving statements like those above, that happens to work on one platform, e.g., GCC, but fails to compile on another, e.g., Clang.<sup>9</sup>

#### Incompatibly specifying alignment is IFNDR

It is permissible to forward declare a UDT without an **alignas** specifier so long as all defining declarations of the type either have no **alignas** specifier or have the same one. Similarly, if any forward declaration of a UDT has an **alignas** specifier, then all defining declarations of the type must have the same specifier, and that specifier must be *equivalent to*; not necessarily the same as; that in the forward declaration:

<sup>7</sup>iso11a, section 7.6.2, “Alignment Specifier,” paragraph 5, pp. 179

<sup>8</sup>“If the constant expression evaluates to a fundamental alignment, the alignment requirement of the declared entity shall be the specified fundamental alignment”: iso11a, section 7.6.2, “Alignment Specifier,” paragraph 2, item 2, p. 178.

<sup>9</sup>Underspecifying alignment is not reported at all by GCC 10.2 (c. 2020), using the `-std=c++11 -Wall -Wextra -Wpedantic` flags. This behavior is reported as a compiler defect; see **wakely15**. With the same set of options, Clang 10.1 (c. 2020) produces a compilation failure. ICC 2021.1.2 (c. 2020) and MSVC 19.29 (c. 2021) will produce a warning and ignore any alignment less than the minimum one.