WG 14, N2867

Checked N-Bit Integers?

David Svoboda
svoboda@cert.org

Date: 2021-11-28

The Problem

The June 2021 WG14 meeting saw the acceptance of two documents:

- N2709 Ballman, Adding a Fundamental Type for N-bit integers (Proposed Wording Alternative One)
- N2683 Svoboda, Towards Integer Safety (Core Proposal, along with N2776)

While both document authors were aware of each other's work, that awareness did not manifest itself in the normative text of either document. Furthermore, N2683 references all integer types in this normative text:

> 7.24.1 Checked Integer Operations
>
> Synopsis
>
> 1

```
#include <stdckdint.h>
bool ckd_add(type1 *result, type2 a, type3 b);
bool ckd_sub(type1 *result, type2 a, type3 b);
bool ckd_mul(type1 *result, type2 a, type3 b);
```

> Description

…

> 3 Both `type2` and `type3` shall be any integer type other than plain `char`, `bool`, or an enumeration type, and they need not be the same. `*result` shall be a modifiable lvalue of any integer type other than plain `char`, `bool`, or an enumeration type.

Paragraph 3 specifies "any integer type". Before the June meeting, this implicitly did not include the N-bit integer types proposed by N2709. However, after the June meeting, the acceptance of both proposals raises a question that WG14 had not resolved: Could the macros be used with the N-bit integer types or not?

While it would be nice if the integer safety macros could apply to all of the N-bit integer types, the lack of implementation experience might be cause for concern. The committee could always choose to remain silent on this question, and hope that platform developers resolve the issue themselves. However, if the committee, out of caution, decides that these accepted proposals should have no interaction with each other, this proposal provides normative text:

Change s 7.24.1, p3 thusly:

3 Both `type2` and `type3` shall be any integer type other than plain `char`, `bool`, a bit-precise integer type, or an enumeration type, and they need not be the same. `*result` shall be a modifiable lvalue of any integer type other than plain `char`, `bool`, a bit-precise integer type, or an enumeration type.

Acknowledgements

This proposal was inspired by a post on the reflector: [SC22WG14.20725] by Joseph Myers. He noted that N2792 (the supplemental integer safety proposal, not currently accepted into C23) had not taken bit-precise integer types into account. Upon further discussion, we realized that the core proposal had also not done so [SC22WG14.20754]. While a proposal that supports checked bit-precise integers is theoretically possible, it would require more work, perhaps implementation experience, and simply forbidding interaction between the two proposals was much easier ☺

Thanks go to Joseph Myers and Aaron Ballman for reviewing this proposal.