Proposal for C23
WG14 N 3107

Title:	Clarify `strtoul`'s behavior on strings representing negative numbers
Author, affiliation:	Robert C. Seacord, Woven Planet
Date:	2023-2-14
Proposal category:	Defect
Target audience:	Implementers
Abstract:	Clarify `strtoul` and related functions behavior on strings representing negative numbers
Prior art:	C

Clarify `strtoul`'s behavior on strings representing negative numbers

Reply-to: Robert C. Seacord (rcseacord@gmail.com)
Document No: **N 3107**
Reference Document: **N 3019**
Date: 2023-9-2

Change Log

2023-2-14:

- Initial version

1.0 Introduction and Rationale

This paper is in response to MB/NC comment GB-232 [n3019]. AG Reference Bug 700 [AG700] describes the issue where `strtol` cannot return `LONG_MIN` with two's complement `long`.

The description of `strtol`, `strtoll`, `strtoul`, and `strtoull` states:

Subclause 7.24.1.7, "The strtol, strtoll, strtoul, and strtoull functions" paragraph 5 states:

If the subject sequence begins with a minus sign, the value resulting from the conversion is negated (in the return type).

If the subject sequence begins with a minus sign, the value resulting from the conversion is negated; for functions whose return type is an unsigned integer type this negation is performed in the return type.

The parenthetical phrase "(in the return type)" was added in C99 in response to DR #006

This clarified the behavior of `strtoul` but broke `strtol`, because with two's complement `signed long`, it is not possible to produce the value `LONG_MIN` by negating a positive value "in the return type".

A similar problem is also present in the `wcsto*` functions.

2.0 Proposed Solution

The following is our Normative reference for vocabulary:

ISO/IEC 2382:2015, Information technology — Vocabulary. Available from the ISO online browsing

platform at http://www.iso.org/obp.

ISO/IEC 2382:2015 provides the following definition:

2124091

negation

NOT operation

Boolean complementation

inversion

monadic Boolean operation whose result has the Boolean value opposite to that of the operand

Note 1 to entry: negation; NOT operation; Boolean complementation; inversion: terms and definition standardized by ISO/IEC [ISO 2382-2:1976].

Note 2 to entry: 02.05.17 (2382)

[SOURCE:ISO-2382-2 * 1976 * * *]

There are two similar operations in C, both defined in Subclause 6.5.3.3, "Unary arithmetic operators" that might be described by the term "negation".

The first is logical negation operator !. The result of the logical negation operator ! is 0 if the value of its operand compares unequal to 0, 1 if the value of its operand compares equal to 0. The result has type int.

The second is the unary – operator. The unary minus operator operates on arithmetic types. in Subclause 6.5.3.3, paragraph 5 states "The result of the unary –operator is the negative of its (promoted) operand."

For `strtol` and related functions seems sensible to be consistent with the wording for the unary – operator as this is the actual operation being performed and not logical negation.

3.0 Wording

Replace the following sentence from Subclause 7.24.1.7, "The strtol, strtoll, strtoul, and strtoull functions" paragraph 5:

If the subject sequence begins with a minus sign, the value resulting from the conversion is negated (in the return type).

with:

If the subject sequence begins with a minus sign, the resulting value is the negative of the converted value; for functions whose return type is an unsigned integer type this action is performed in the return type.

Replace the following sentence from Subclause 7.31.4.1.4, "The wcstol, wcstoll, wcstoul, and wcstoull functions" paragraph 5:

If the subject sequence begins with a minus sign, the value resulting from the conversion is negated (in the return type).

with:

If the subject sequence begins with a minus sign, the resulting value is the negative of the converted value; for functions whose return type is an unsigned integer type this action is performed in the return type.

4.0 Acknowledgements

I would like to recognize the following people for their help with this work: Jonathan Wakely, Aaron Ballman, and Rajan Bhakta.

5.0 References

[AG700] AG Reference Bug 700

[DR #006] Defect Report #006

[n3019] Keaton, David. CD1 9899 ballot comments with progress from first week of ballot resolution.