Clarifications on null pointers in the library
WG14 N3403

Title:	Clarifications on null pointers in the library
Author, affiliation:	Aaron Ballman, Intel
Author, affiliation:	Nikita Popov, RedHat
Date:	2024-11-25
Proposal category:	Bug fixes
Target audience:	WG14 members, C implementers

Abstract: Clarifies some scenarios based on implementer feedback after the adoption of N3322 (Allow zero-length operations on null pointers) in Minneapolis.

Clarifications on null pointers in the library

Reply-to: Aaron Ballman (aaron@aaronballman.com), Nikita Popov (npopov@redhat.com)
Document No: N3403
Date: 2024-11-25

Summary of Changes

N3403

- Initial version

Introduction and Rationale

In Minneapolis in 2024, WG14 adopted N3322 which allows zero-length operations on null pointers. This paper also updated the standard library to clarify when a null pointer and a zero length are well-defined. As implementers react to the changes in the standard, a few questions have come up that require clarification.

For `strncat`/`wcsncat`, can `s1` be null or is only `s2` allowed to be null? Our perspective is that only `s2` may be null and that `s1` must be nonnull.

For `fwrite`, can you pass a null pointer for the buffer to write if either `size` or `nmemb` is zero? Our perspective is that passing zero for `size` or `nmemb` implies that the buffer does not need to be read, and so you can pass a null pointer in that case.

Proposed Wording

The wording proposed is a diff from the WG14 N3301 working draft of ISO/IEC 9899. Green text is new text, while ~~red~~ text is deleted text.

Modify 7.26.3.2p2:
… If `s1` is a null pointer value or copying takes place between objects that overlap, the behavior is undefined.

Modify 7.31.4.3.2p2:
Drafting note: Unlike with `strncat`, `wcsncat` does not have a prohibition against overlapping objects. That difference is retained here, but that may be an oversight from the original specification.
… A terminating null wide character is always appended to the result.[fnt)] If `s1` is a null pointer value, the behavior is undefined.

Modify 7.23.8.2p3:
Drafting note: `fread` already seems to be covered by: If `size` or `nmemb` is zero, `fread` returns zero and the contents of the array and the state of the stream remain unchanged.
If `size` or `nmemb` is zero, `ptr` may be a null pointer, `fwrite` returns zero, and the state of the stream remains unchanged.

Acknowledgements

I would like to recognize the following people for their help in this work: Jakub Jelinek.