WG14 N2925
Meeting notes

C Floating Point Study Group Teleconference

2022-01-19
8 AM PST / 11 AM EST / 4 PM UTC

Attendees: Rajan, Jim, Fred, Mike, Ian, David H,

New agenda items (https://wiki.edg.com/pub/CFP/WebHome/CFP_meeting_agenda_20220119-update.pdf):
Mike: Question from Rex J.

Next Meeting(s):
Note: Different time: New time = Regular time slot + 1h.
Thursday, February 10th, 2022, 5PM UTC
ISO Zoom teleconference
Please notify the group if this time slot does not work.

Double postings: See CFP2324,CFP2325
No one other than Fred seeing this. Leave it as is.

C++ Liaison:
Nothing new.

C23 Integration: Review draft: See CFP2320
CFP2319,...,CFP2340 dealing with the draft to validate papers we submitted made it in right.
Fred: Jim's editorial change did not get discussed in a meeting and not in the draft list of papers put in (renaming of the SNAN macros). Not sure if it was done in the draft.
Jim: You sent the summary to him too right?
Fred: Yes, I sent it to the CFP group and he should have gotten it.
Jim: My action item may cover that.
Fred: I noticed for SNAN prefixes, not quiet NANs.
Jim: I can add on SNAN prefixes to my item.

New agenda item: Mike: Rex talking about C# has normative references to C. He asked if he should use 2011 IEC 60559 or not. I suspect not. Any list of changes between 2011 and 2020 for 754?
David H: There were some listed under the background document on my website (ucbtest.com/background) and 754's.
Jim: He does have the 2017 C right?
Mike: Yes, I believe so.

Carry-over action items:
Rajan: For CFP2291, update the color to be more readable and add in wcstod functions updates and refer to overflow/underflow in 7.12.1 if possible. See [Cfp-interest 2341] Update to N2823.
Jim: The first change didn't have a bulleted list. The second does. Should have the same.
Fred: I prefer non-bulleted. Note bullet 3 is not a part of overflow.
^Jim: Also make the last sentence a semicolon continual sentence.

Jim: Since you can't access errno, is it really an issue having errno there?

^Jim: It seems the way it is specified here is x but not y. Seems different. Can we say errno can't be accessed along with the floating point environment.

^Jim: The identifiers for fenv.h and math.h should also be reserved as per stdlib.h in alternative 1.

Mike: Bulleted lists should not be capitalized since they are not sentences.

Fred: Ask C++ what their issues with *_HAS_SUBNORM are and if they are OK with obsoleting it.

Fred: Asked the C/C++ shared reflector and got one comment back, but after my second post, got none.

Fred: During the WG14 meeting, we only discussed removing SUBNORM macros, but nothing about adding the paragraph about flushing. On the agenda for WG14's next meeting.

Fred: Keep this item on the carry over list.

Last meeting action items (done unless specified otherwise, details below):

Jim: Submit [update to “normal”] to WG 14.

See: https://wiki.edg.com/pub/CFP/WebHome/n2881.pdf

Jim: Submit [update to “max exponent macros”] to WG 14.

See: https://wiki.edg.com/pub/CFP/WebHome/n2882.pdf

Jim: [in update to 5.2.4.2.2] copy changes to footnote 22 into set of suggested changes to N2806+N2596.

Jim: [in update to 5.2.4.2.2] change the last item which says "are called zero floating-point numbers" to just "are zeros".

Jim: Submit [update to 5.2.4.2.2] to WG 14, after making [two] changes above.

See: https://wiki.edg.com/pub/CFP/WebHome/n2879.pdf

Jim: [in update to “overflow”] change 7.12.11.5 nextup to match 11.6 nextdown as follows: If x is the positive number (finite or infinite) of maximum magnitude in the type, nextup(x) is x.... and submit to WG 14.

See: https://wiki.edg.com/pub/CFP/WebHome/n2880.pdf

Jim: Track the many changes to infinity and nan macros. - Not Done.

Fred: Add printf rounding issue in [Cfp-interest 2256] to list of issues to be revisited for next version.

All: Review N2726 (Meneide, _Imaginary_I and _Complex_I Qualifiers, revision 0) for next time.

Jim: send note to editor asking for editorial change in 7.12.7.3 “of a floating-point number x” to “of x”. Copy Rajan and Fred.

Anyone interested: Review terminology introduced in 5.2.4.2.2 and report.

See CFP22{58,63,64,71}

New action items:

Rajan: Update CFP2341 to use paragraph form with semicolon separators.

Rajan: CFP2341: Look into seeing if we can say errno and the floating point environment can't be accessed (like with the "as if" rule).

Rajan: Update CFP2341 to say the identifiers for fenv.h and math.h should also be reserved as per stdlib.h in alternative 1.

Jim: Track the many changes to infinity and nan macros. Particularly still need to check prefixes for signaling nan macros. - Carry over.

Fred: Post the list of issues to be revisited for the next version on the CFP wiki.

Fred: Add an issue for a future version to look at the 5.2.4.2.2 terminology and ensure it is correct and consistent.

Fred: Put the NAN not raising exceptions footnote idea into the list of items for a future revision of C.

Upcoming WG14 meetings:

January 31-February 4, 2022, Portland, Oregon, US (Tentative)

Mailing deadline: December 31, 2021
July 11-15, 2022, Strasbourg, France (Tentative)
Mailing deadline: June 10, 2022

Action item resolutions:

Jim: Submit [update to “normal”] to WG 14:
See https://wiki.edg.com/pub/CFP/WebHome/n2881.pdf
Done.

Jim: Submit [update to “max exponent macros”] to WG 14:
See https://wiki.edg.com/pub/CFP/WebHome/n2882.pdf
Done.

Jim: [in update to 5.2.4.2.2] copy changes to footnote 22 into set of suggested changes to N2806+N2596.
Submitted.

Jim: [in update to 5.2.4.2.2] change the last item which says "are called zero floating-point numbers" to just "are zeros".
Submitted.

Jim: Submit [update to 5.2.4.2.2] to WG 14, after making [two] changes above:
See https://wiki.edg.com/pub/CFP/WebHome/n2879.pdf
Submitted.

Jim: [in update to “overflow”] change 7.12.11.5 nextup to match 11.6 nextdown as follows: If x is the positive number (finite or infinite) of maximum magnitude in the type, nextup(x) is x....
and submit to WG 14: See https://wiki.edg.com/pub/CFP/WebHome/n2880.pdf
Jim: Rajan, are you OK with these 4 papers?
Rajan: Yes, but will contact you directly if I do have issues or questions.

Jim: Track the many changes to infinity and nan macros.
^Jim: Still to do. Particularly still need to check prefixes for signaling nan macros. Keep as carry over item.

Fred: Add printf rounding issue in [Cfp-interest 2256] to list of issues to be revisited for next version.
Done.
Jim: Can you post the list on the wiki?
^Fred: Post the list of issues to be revisited for the next version on the CFP wiki.

All: Review N2726 (Meneide, _Imaginary_I and _Complex_I Qualifiers, revision 0) for next time.
Jim: It is still usable for static initialization and this change doesn't affect that.

Jim: send note to editor asking for editorial change in 7.12.7.3 “of a floating-point number x” to “of x”. Copy Rajan and Fred.
Jim: This was already in a proposal so we can close this.

Anyone interested: Review terminology introduced in 5.2.4.2.2 and report: See CFP22{58,63,64,71}
Fred: Is this the large floating point values being nearest to infinity?
Jim: I don't think so. The key issue for me was the use of the term "floating point number" or "floating point value". Those terms appear throughout the standard so we have to be careful if we change them.
^Fred: Add an issue for a future version to look at the 5.2.4.2.2 terminology and ensure it is

correct and consistent.

Other Issues:

Infinity and exceptions: See CFP23{07,17,18,...,30} infinity and exceptions

Jim: The conclusion was INFINITY and NAN macros were not prohibited from raising exceptions.

Fred: The macros are only required if they are supported right?

Jim: Yes, but they don't have to follow Annex F.

Fred: NAN is defined to be a quiet NaN so it can't have exceptions.

Jim: There is a statement what quiet NaN's don't have exceptions I believe.

Fred: 5.2.4.2.2 says a QNaN propagates through *almost* every operation. I believe it is *every* operation not *almost* every.

Jim: My response is in CFP2335. Your examples there can raise exceptions.

Fred: There is no definition in the C standard that says what an arithmetic operator is. Some believe it is only +,-,*,/.

Fred: Evaluating NaN alone without any operations should not have exceptions since it is defined as being quiet.

Rajan: Microsoft and IBM z do have exceptions.

Fred: Can we have a footnote or something saying a quiet NaN has no exceptions. So "evaluating the macro NAN has no exceptions" or something like that. We can put it into the updates to INFINITY and NANs or just leave it for a new standard.

Rajan: Not opposed to this, but should be a new proposal.

Jim: Some simple uncontestable clarifications can be brought forward at some point to WG14.

^Fred: Put the NAN not raising exceptions footnote idea into the list of items for a future revision of C.

Correctly rounded math functions: See CFP2308

Anyone interested in this can engage Paul.

Flushed subnormals: See CFP2314-CFP2316

Fred: I do intend to present words to say it is implementation defined what happens when flushing subnormals.

Fred: Send the WG14 paper number for the flushed subnormals to the CFP list.

Paper is N2797.

“almost” and Quiet NaN: See CFP2332,CFP2335

Fred: Do we need to define arithmetic operation?

WG14 question.

Fred: C standard never defines integer arithmetic, or binary arithmetic so it is implicitly undefined. I don't have the access to the vocabulary document so can't check if it is the normal English term.

Jim: Before going to WG14, you should have a definition to give. 754 did it for classifications of operations I believe.

nan("11111111111111111111...."): See CFP23{33,...,2339}

Fred: It sets errno to ERANGE but does not raise any FP exceptions. Vincent looked at glibc and the nan function calls strtoul which gets the ERANGE.

Jim: The ERANGE is wrong. It doesn't fit the definition of a range error.

Others:

None.