WG14 N1992
Meeting notes

C Floating Point Study Group Teleconference

2015-11-10
9 AM PST / 12 PM EST

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

New agenda items:
None.

Last meeting action items:
Ian: Talk to Lawrence Crawl regarding proposing this IEEE-754: 2008 binding to C++ as well. - Not done.
Ian: Update and check the items listed and flagged under Feature_List_Part_1. - Not done.
Jim: Send Mike an email regarding what is needed regarding prior art/implementation for Part 2 features in other languages. - Done.
Jim: p18: Line 9: Put in a footnote showing the practical effect of the selection statement restriction. - Done.
Jim: p18: Line 3: a catch action -> catch actions (look into making the change) - Done, but not in the review draft.
Jim: p18: Line 34: a delayed-catch action -> delayed-catch actions (look into making the change) - Done.
Jim: p18: Line 21: without the break -> without the jump - Done.
Jim: p21: Line 41: Reference the section this footnote is in. - Done.
Jim: Contact David Keaton to see if he has any objection to use this new latest part 5 document for the WG14 meeting since we have addressed all the comments (Joseph's). - Done.
Rajan: Present part 5 to WG14. - Done.
Rajan: Put up PDF's of the feature lists. - Done.
David: See what Oracle has done for the feature lists. - Not done.
Rajan, Fred: See if there is a time pressure for new C Standard proposals. - Done (none).
Rajan, Fred: Report on the WG14 meeting with respect to the TS's. - Done.
All: Consider new issues:
p3: Line 24: Is this a constraint violation if it occurs? If so, it cannot be diagnosable by a translator following the translation phases exactly. - Not in constraint section so doesn't need to be diagnosed.
p17: For library state: Unspecified vs indeterminate state. Should not have the standard library states be indeterminate, but can we guarantee that it is in a valid state?
- Leave as unspecified. Can't see a reasonable way to implement otherwise.
Should we have a note regarding suggestion to not have stateful operations inside an ASAP block? No.

New action items:
Rajan: Get Mike in contact with Jens to get access to the C Standard source in the GIT repository.
Jim: p13: line 22: Add in a statement describing the remaining bullets to show they are not a complete list.

Jim: p17: line 27: List the actions in place of "below" or say "the remaining actions in this sub-clause".
All: Consider mixed nested try and delayed try blocks and look for cases not covered by the specification.
All: Focused Review of part 5 once the updated draft comes out.

Next Meeting:
December 8th, 2015, 12:00 EST, 9:00 PDT
Same teleconference number.

Discussion:
IEEE 754 revision:
- No major issues so far.
- Some discussion on the specification of the math functions.
  - Need to keep an eye on this in case it impacts us.

Arith23:
- Special session request from Marius was accepted.
  - 3 part panel: Half hour review of our TS's. One part from industry presenting the state of the art support for IEEE 754 (variable length time). Final part for the update on the 754 update revision (1h).
- Date: June 10th-13th, 2016
- Location: Bay area: Santa Clara.

WG14 meeting:
- #pragma try/catch/delayed outside the { to allow macro replacement of #if 0 for implementations that do not support this.
- Also wanted a 1-1 correspondence between delayed/try and delayed/catch.
- Mike is interested in looking at convert the C standard from TROFF to something else.
- *Rajan: Get Mike in contact with Jens to get access to the C Standard source in the GIT repository.
- Mike: ODT is probably better to convert to since it is XML. Being used for 754 right now.

Part 5: Various emails, documents (cfp5-20151104.pdf)
- Major WG14 requests: Move control flow pragma's outside the blocks, and have the try list match catch's.

- p13: The second bullet pretty much covers everything. The other thing is extensions.
  - Should we say "the code that is written will give the same results in all implementations"?
    - Jim: That is what this is supposed to say.
    - Defining what reproducible means is probably sufficient. No need to give programming notes to show how to do that.
    - Perhaps restrict this to floating point numerical reproducibility?
    - Jim: Branches and everything else are not floating point and restricting it otherwise is not very useful.
    - Note that random number generators, I/O status, etc. all affect reproducibility as well.
  - Should the header in line 13 be "Reproducible code sequences"?
    - Maybe. Will affect what is listed above.
  - The first bullet says it is for the bound floating point operations.
  - The rest of the bullets lists things that might not be obvious or need emphasis. It is not intended to be a complete/exhaustive list.
  - Perhaps have something after the first two bullets to show the rest are not complete.
    - Ex. Saying "For example the following would be required:"

*Jim: p13: line 22: Add in a statement describing the remaining bullets to show they are not a complete list.

p17: Describe the two kinds of pragmas: {delayed}try/catch and break.

Handles the pragma outside the block WG14 request.

What does "below" mean in "actions specified below"? How big is the "below"? Perhaps list the actions?

*Jim: p17: line 27: List the actions in place of "below" or say "the remaining actions in this sub-clause".

p18: Handles the WG14 1-1 correspondence between try/catch request.

Another WG14 question was can you mix/match delayed and non-delayed try/catch?

Current spec doesn't allow a try and a delayed try in the same scope without the catch and delayed catch blocks in between.

For nested, try with a nested delayed try if the same exception occurs in the nested block, the delayed catch handles it due to the rule about same exception delayed try taking over.

If the exception happens during the try block before the delayed try, the jump can still happen any time even in the delayed try block. But since you are changing the environment, it may be handled already by that rule.

Similarly for try inside delayed try.

*All: Consider mixed nested try and delayed try blocks and look for cases not covered by the specification.

p19-20: Is it useful to have the nearly identical code sequences for try and delayed try?

Yes, having something concrete helps.

Another WG14 comment: No global floating point exception handler.

Default exception handling is the 'global' handler.

Are we ready for a focused review?

No major unresolved issues left so yes.

Evaluation format (7): Marius
Optimization controls (8): David
Reproducibility (9): Rajan
Alternate exception handling (10): Fred
Rest of the parts (intro), etc.: Ian

Start after the next draft is out.
Can be done via email. Finish up in January meeting.

Implementation status:

Mike's web page shows a number of 754 decimal implementations. He also has a decnum package that supports everything except the encode/decode functions.

Intel and HP have a implementatiosn for all that as well.

Should we propose to have parts 1-2 be included into the C standard?

The C1X charter lists the prior art requirement. The other parts probably don't have the prior art.

Some parts of part 4 do have prior art so those parts could maybe be proposed.

The optimization controls in part 5 have prior art too (with different syntax/methods).

Regards,

Rajan Bhakta
z/OS XL C/C++ Compiler Technical Architect
ISO C Standards Representative for Canada
C Compiler Development