Draft Minutes for 21–25 October, 2019
MEETING OF ISO/IEC JTC 1/SC 22/WG 14 AND INCITS PL22.11
WG 14/N 2451

Dates and Times

21 October, 2019	08:30 – 12:00	Lunch	13:30 – 17:00
22 October, 2019	08:30 – 12:00	Lunch	13:30 – 17:00
23 October, 2019	08:30 – 12:00	Lunch	13:30 – 17:00
24 October, 2019	08:30 – 12:00	Lunch	13:30 – 17:00
25 October, 2019	08:30 – 12:00	Lunch	13:30 – 17:00

Meeting Location

Ithaca Marriott Downtown on the Commons
120 South Aurora Street
Ithaca, New York 14850
US

Meeting information

Venue information: N 2327

Local contact information

Aaron Ballman < *aaron@aaronballman.com*>

1. Opening Activities

1.1 Opening Comments (Ballman, Keaton)
Start time 9:24 due to AV problems in meeting room. Aaron introduces us to Ithaca.
1.2 Introduction of Participants/Roll Call

Name	Organization	NB	Comments
Lars Bjonnes	Cisco	USA	
Paul E. McKenney	Facebook	USA	
Maged Michael	Facebook	USA	
Aaron Ballman	GrammaTech	USA	
Rajan Bhakta	IBM	Canada	PL22.11 chair
Melanie Blower	Intel	USA	
David Keaton	Keaton Consulting	USA	WG14 Convener

Clive Pygott	LDRA	USA	
Barry E. Hedquist	Perennial	USA	
Tom Plum	Plum Hall	USA	Phone
Martin Sebor	Red Hat	USA	
David Svoboda	SEI/CERT/CMU	USA	
Fred Tydeman	Tydeman Consulting	USA	PL22.11 Vice chair
Robert Seacord	NCC Group	USA	
Alex Gilding	Perforce Software/Programming Research Ltd	USA	
JeanHeyd Meneide	Self	USA	
Dr. Will Klieber	Self	USA	Phone
Jens Gustedt	INRIA	France	WG14 Fallback Project editor
Peter Sewell	Univ. Cambridge	UK	Memory model study group, phone
Joseph Myers	CodeSourcery/Mentor a Siemens Business	UK	Phone
Niall Douglas	ned Productions Ltd	Ireland	Phone

1.3 Procedures for this Meeting (Keaton)

Straw poll, not formal votes. Representatives to ISO WGs are individual experts, and are expected to vote for the technically best decision. In particular, they cannot be compelled to vote a certain way by a national body.
David Keaton is the meeting Chair.
Melanie Blower is the Recording Secretary.

1.4 JTC 1 Required Reading

1.4.1 ISO Code of Conduct
1.4.2 IEC Code of Conduct
1.4.3 Key points

1.5 Approval of Previous Minutes [N 2376] (PL22.11 motion, WG 14 motion)

Fred sent typographical corrections before the meeting started.

Aaron moved to approve, Fred 2nd. Approved unanimously.

1.6 Review of Action Items and Resolutions

Charles: Ask the C Safety and Security study group what they want to do under the assumption of not being able to use MISRA. -- Done

David K: Look into mechanisms for publishing the MISRA documents under ISO and discuss with Andrew Banks. -- Open

Blaine: Ensure all DR's marked C2X are in the latest working draft (N2346). -- Open

Editor: Move DR501 to C2X. - Done

Editor: Move DR496 to C2X. - Done

Editor: Put N2335 into C2X. - Done

Editor: Put N2267 into the C2X. - Done

Editor: Put N2270 into the C2X. - Done

Editor: Put N2334 into the C2X. - Done

Editor: Uniquely label the bullet points in Annex J. - Done

Editor: Put N2337 into C2X switching the cases for the b/B's. - Done

Editor: Put N2338 into C2X. - Done

C FP: Give 18661 part 4a (not reduction functions) for inclusion into C2X. - Done

Editor: Put N2356 into C2X. - Done

Editor: Put N2319 into C2X. - Done

Editor: Put N2322 into C2X. - Done

Editor: Put N2325 into C2X. - Done

Editor: Put N2326 into C2X with editorial changes on the footnote expected. - Done

Editor: Put N2349 into C2X. - Done

Editor: Put N2353 into C2X. - Done

C FP: Put N2309 into TS 18661-4 and C2X. - Done

Editor: Put N2358 into C2X. - Done

Editor: Put N2350 into C2X. - Done

1.7 Approval of Agenda [N 2437] (PL22.11 motion, WG 14 motion)

Fred moved to approve; Aaron 2nd; no objections – approved.

1.8 Identify National Bodies Sending Experts

Canada, France, UK, USA.

1.9 INCITS Antitrust Guidelines and Patent Policy

1.10 INCITS official designated member/alternate information

2. Reports on Liaison Activities

2.1 SC 22

Straw poll: If WG14 is notified that it is in danger of disbandment, shall the convener request a 4 year revision of the C standard? Result: consensus to approve.

Details: New development: secretary role is now "committee manager". JTC1 may also make this change. In ISO, when a WG finishes its project, it is automatically disbanded. It can be reconstituted for a revision and then it gets a new WG number. JTC1 is thinking of switching to this model. Currently JTC1 continues until they decide to quit. Officially there are no C2X working groups so we may be in jeopardy of disbandment. David plans to ask for a 4-year term to work on C2X when necessary. Commencement of WG will force a release within 4 years. Do revisions of TS's count to extend the service? Yes. Can we get an extension if 4 years isn't enough time? No.

2.2 PL22.11/WG 14

2.2.1 Document system

Action item: Jens will take an Action Item to look at a document control system alternatives, will talk to David and Dan about the possibilities.

We should start the C numbering system. David will work out a manual system for numbering with Dan until we have an automatic system. Jens has a problem with bottleneck about getting number for papers, and we need to have revisions of papers quickly. The benefit to switch to C numbers; ISO is astonished by the number of N documents that our committee produces.

Jens doesn't want to use github to hold our documents because of commercial association. David would like to experiment with document control between now and our next meeting.

2.2.2 Convener's Report and Business Plan [N 2415]

Action item: David: please change the convener's report to explicitly talk about the floating Liaison to C++.

2.3 PL22.16/WG 21

Aaron: WG21 met in Cologne. Ballot resolution, meeting in Belfast in 2 weeks.

2.4 PL22

Chris Tandy is the new chair

2.5 WG 23

Nothing major. Three documents have been approved. Working with ISO editor about rewrites. ISO is tightening the screws on “directives part two”. How a standard is supposed to be written. In the past there have been guidelines and we can get an exception. Now they are making it more strict, if it's not in the directives then it is not allowed. In addition, there are unwritten rules which they refuse to publish and rules change from month to month. Lots of flux in the ISO staff. We are getting a decent relationship with “unspecified behaviors” subgroup in WG21.

2.6 MISRA C

Action Item for David Keaton to find out what happened to N2008.

A liaison report was submitted by Andrew Banks, N2445. Third edition of MISRA-C based on C99 was published in 2019. Working on MISRA for C11 and C17.

In 7.21.2 please clarify: “characters” is ambiguous. Rajan: this is the normal English character, physical source file character. (Larry had called in and made this pronouncement in a previous meeting).

MISRA is having internal discussions about N2258, if you are interested in joining the conversation, Alex Gilding invites you to get in touch with him. Aaron is interested.

Other Liaison Activities

Nick Stoughton is the liaison for POSIX.

Fred: 754 committee is trying to get an ISO standard.

3. Reports from Study Groups

3.1 C Floating Point activity report

Working closely with 754. Creating updated TS for the items that have been voted in. Some items are still in flux, there are discussions on Part 3. The binding is being created for moving IEEE 2019 into TS.

3.2 C Safety and Security Rules Study Group

Clive: report from Charles, may have arrived this morning in the email. Attendance has been very low at the teleconference. David notes that INCITS requires 4 people in a group. Based on feedback from London, how to best proceed given impasse about MISRA-C? [The problem with MISRA-C is due to copyright usage versus ISO.] 5 options considered, and 3 will be worked:

1. Update security rules for C17 and parallelism
2. Cooperative standards
3. Defined specified and unspecified safety behavior e.g. type safety
4. Fix annex K
5. Annotate Annex J with taxonomy. (Rajan)

These are the options that will be worked:

1. Update security rules for C17 and parallelism
   - review of TS17961 for any changes needed to bring it into line with C17
   - consider proposals for new rules, e.g. in the area of parallelism (but not limited to this)
2. Cooperative standards
   - How to handle any incompatibilities. In London it was suggested there was at least one, but this hasn't been pinned down
   - Recognise different usage models and how that impacts the rules to be applied. TS17961 was written with the aim of being applied to existing code, so there were concerns over 'noisy' rules that would generate too many 'false positives'. By contrast, MISRA is written with the expectation that the rules will be applied during development, so the 'noisy' constructs never get written into the code. However, these are not exclusive concerns. It would be useful to consider the development of secure

applications and there are times were a safety system may be looking to adopt existing code - particularly at low SIL (Safety Integrity Level - see ISO 61508).

- A particular concern regarding the adoption of existing code is what to do with libraries. MISRA currently says libraries should be developed to the same standard as the rest of the code, but this is often impractical. There has to be some argument that the conservative use of a well-established library is safer than the production of new code to do the same job. The revision of MISRA C++ is finding this challenging, as we keep wanting to make exceptions for libraries, e.g. 'no unused type definitions or functions in a project, unless introduced by a library header'

3. Defined specified and unspecified safety behavior e.g. type safety
   - The idea for this topic largely came from the suggestions from Martin Sebor, that the compiler could provide specific warnings where unspecified behaviour would make the code unpredictable or unportable - for example, the order of evaluation of subexpressions is UB, but usually it has no impact on the behaviour of the program. The only time it does is if two or more subexpressions access the same variable and at least one of them modifies it. There'd still be arguments about what to do if one of the subexpressions is a function pointer. Does the tool need to track all functions that may be assigned to the pointer, or take the pessimistic assumption that it will access a common variable, or take the optimistic view that it won't?

3.3 C Memory Object Model Study Group

Went to Cologne meeting, useful. Talked to the UB group (wg21, sg12). They were enthusiastic about one of the options (the more complex option). Jens has the revised storage instance paper for Ithaca. Paul McKenny has another version of the lifetime-end zap paper for Ithaca. Talked about sub-objects and effective types. Presented a paper at C++ Cologne which has not been submitted to WG14.

4. Teleconference Meeting Reports

4.1 Report on any teleconference meetings held

5. Future Meetings

5.1 Future Meeting Schedule

- 30 March - 3 April, 2020 – Freiburg, Germany
- 12-16 October, 2020 – Minneapolis, Minnesota, US (tentative)
- Spring, 2021 – Strasbourg, France (tentative)
- Fall, 2021 – TBD
- 31 January - 4 February, 2022 – Portland, Oregon, US (tentative)

5.2 Future Mailings

- Post-Ithaca – 18 November 2019
- Pre-Freiburg – 2 March 2020

- Post-Freiburg – 27 April 2020
- Pre-Minneapolis – 14 September 2020
- Post-Minneapolis – 9 November 2020

6. Document Review

Monday morning

6.1 Tydeman, Follow-up from last meeting on SD3 13: DR 482: Macro span files: undefined [N 2324]
Straw poll: Should we change the behavior of macro invocations that span files cf: N2324?
3-12-0 (Yes-No-Abstain) Defeated
Action Item: David Keaton to put a note into SD3 about the resolution of this issue.
General discussion: This paper was covered in London. Follow-up directive was not clear. Aaron requests that the paragraph in DR to be rewritten to clarify that committee has decided to not modify behavior.

6.2 Gustedt, Two's complement sign representation for C2x [N 2412]

Straw poll: Shall we add N2412 to C2X in the next WG14 meeting if there are no objections before Feb 2, 2020? 15-0-1 Approved.

Rajan asked a question about "power of two" wording for each bit., 6.2.6.2. David pointed out that the sentence is an ambiguous wording. Martin observes, the changes minimum ptrdiff_t width from 16 to 17. 7.20.3.1. There are 16-bit targets which cannot conform to this.

6.3 Gustedt, intmax_t, a way out v.2 [N 2425]
Straw poll:
Shall the requirements for the types [u]**intmax_t** be relaxed to cover only basic integer types and other semantic integer types as proposed in N2425? 2-4-7 No.
Shall we mark the {str|**wcs**}**to**[ui]max, imaxabs, imaxdiv functions as deprecated as proposed in N2425? 7-3-4 Yes.
Shall we deprecate [u]intmax_t? 5-4-5 "No consensus"
Should the new floating point functions change from intmax_t to long long? 7-3-4 Yes.

Multipurpose: intmax_t establishes maximum size of preprocessor. Provide fallback type for casting when you don't know the size. Type used in function interface. Restrict intmax_t to the types used in the standard. Aaron recommends to deprecate intmax_t, thinks this proposal is too difficult to explain to programmers. David: additional source of misunderstanding, the word 'leastmax'. It's really useful to have this for the preprocessor, to describe the largest value accepted. Rajan: Use intmax_t only for formatted i/o. Martin: don't introduce a new type.
Do we have to retain intmax_t to support existing code? David frequently uses "str-to-intmax_t" conversion function when reading from a file--how do you make a generic solution for this? If intmax_t is deprecated note that this type is used for other facilities like ABS,MAX,MIN functions. A type-generic macro can be offered to the user as a substitute. Currently the imax_abs has undefined behavior for some argument values, but the type-generic macro can

solve the UB. Joseph Myers has requested that there always be available a stable external name for a function corresponding to the type-generic macros.

Monday afternoon

6.4 Editors' discussions

Gustedt, ISO/IEC 9899 editor report June 2019 [N 2387]
Aaron: Subclause to Annex J.6 (reserved identifiers), expect it to be a paper not an editor report. What's the line? David: this is consistent with our previous policy. Larry's position is that Annex J is entirely editorial. Aaron is concerned about the bulk of new material added. There's a new ISO rule against having an Index (ISO wants to be format independent, e.g. hyperlink and page numbers).

Gustedt, ISO/IEC 9899 editor report September 2019 [N 2435]
Gustedt, ISO/IEC 9899 working draft September 2019 [N 2433]
Gustedt, ISO/IEC 9899 working draft September 2019, diffmarks [N 2434]

Gustedt, change bullet points in Annex J to referable labels or numbers, a partial implementation [N 2427]
Amend the text so that it contains tag's for annotating UB and others. C++ is working on something similar. The security study group, taxonomy of Unspecified Behavior (Clive). They have completed about 1/3 and have identified topics. The two groups could work together. Jens needs help on tagging, good naming techniques and also the library needs to be tagged. Alex remarks that C++ document is tagged differently, the C++ is tagged because the document numbering changes frequently. Alex, Clive and Aaron will assist.

6.5 Gustedt, Clean up atomics, non-normative changes [N 2389]
General discussion. This adds some generic functions. Atomic_fetch_add
David recommends that we need to coordinate closely with C++ SG1 on these changes. Aaron is available to present the paper to C++. Jens is eager to get it added to the standard. Martin and Joseph: some of the changes are normative and need to be discussed via N paper, e.g. in 6.2.6.1 there is a change from indeterminate to unspecified. It's easier to get changes approved when they are in smaller chunks.

6.6 Gustedt, Remove ATOMIC_VAR_INIT [N 2390]
Result: N2390 Melanie will investigate whether there is language in the standard stating that simple assignment to atomic variables is indeed an atomic operation, and will write a paper if necessary.

Martin and David Keaton: both have an issue with new words regarding "data race" (5.1.2.4). Jens agrees to remove that clause. We could use a new paper addressing these questions: Is it required to initialize with atomic_init or is mere assignment sufficient? Is assignment to an atomic variable always an atomic operation, regardless of whether it has been initialized?

Further discussion later in the week about ATOMIC_VAR_INIT on Wednesday morning. David Keaton refers us to ATOMIC_FLAG_INIT, where the flag is initially in an "indeterminate state". (versus indeterminate value). Atomic flag has 'set state' and 'clear state'. Implementations tend to provide lock free atomic's or to use a separate hash table. David has worked on an implementation where the lock state is adjacent to the value. Rajan couldn't find a statement that assignment to an atomic is itself atomic. Need to do more study on atomic and make sure that the standard states that simple assignment is indeed atomic. Direction for Jens:

1. Delete adding to data race
2. Remove example 2
3. Ensure that simple assignment is indeed an atomic operation. Melanie takes an action item to investigate and write a paper if necessary. Atomic non-normative changes N2389 discussed atomic compound statements.

6.7 Gustedt, Synchronization at thread and execution termination [N 2391]
Action Item: Jens will update wording for N2391, and bring back this week if possible.

6.8 Rytarowski, Add methods for setting and getting the thread name [N 2419]
Result: Straw poll: Are we in favor of something along the lines of N2419? 5-1-8 No.
Action Item: David Keaton will discuss the issues with the author.

Rajan comments: overriding issue, the macro thread_max_namelen; need consistency to use same convention see 6.26. The parameter THR, but THR is not a pointer, needs clarification. Return value prior to calling in thread_getname, but this statement doesn't make sense. Not enough information in the proposal, no discussion of race conditions etc. Jens also has issues and has discussed it on the Reflector. Martin asks if this facility is universally available. Cisco has added thread attributes which allows the names to be changed. David: Changing the name can also be implemented with Thread Specific Storage.

Tuesday morning

6.9 Gustedt, Introduce the term storage instance v2 [N 2388]
Peter noted that the phrase "copied as if by memcpy" doesn't work with the provenance paper. Seacord 'storage instance duration', should the standard introduce a new term?
6.2.6.1 "address space" definition doesn't allow address of code, something needs to be changed here. Jens: add a footnote? DavidS and Martin: Lifetime of storage instance vs. lifetime of object, circular definition. Does a storage instance have a lifetime? "Martin: immediately follows" introducing a new term which isn't used: omit. ^live storage instance^storage instance^. DK suggests "end address" should not be one past the end, and this removes the need for 'immediately follows'. Martin objects to the change ^indeterminate^unspecified^ important to retain original wording.

6.10 Douglas, Memory region stores flush and reloads force [N 2436]
Result:
Straw Poll: Would we like to see normative text for something along the lines of N2436? 8-5-5 "Yes"

General discussion: ‘volatile’ doesn’t work for shared memory objects. C++ is getting rid of volatile: volatile_load and volatile_store.
Martin: request implementation experience e.g. gcc or clang or library.
Rajan: too hardware specific e.g. cache lines. We don’t usually standardize for classes of hardware; perhaps add it to e.g. OpenMP which deals with shared memory.

6.11 McKenney, Lifetime-End Pointer Zap [N 2443]
Result: Straw poll (Preference vote):
In N2443, can you live with Status Quo? 5
In N2443, can you live with Eliminate Lifetime-End Pointer Zap Altogether? 12
In N2443, can you live with Something in between? 9

Prefer to not remove the UB. Martin: Arm implementation invalidates some of the algorithms described in the paper. Niche area of algorithms. Prefer programmer to annotate the code e.g. describe that the program is reading an indeterminate pointer. Niall: Support zapping validity but not zapping value.

6.12 Follow-up from last meeting on C Memory Object Model Study Group discussions
Result:
Straw Poll: Should the committee accept N2311 PVNI-ae-udi model subject to satisfactory feedback from compiler implementors. 13-0-4

> The Memory Object Model Study Group requested time to follow up on the previous meeting's discussions. For reference, the papers from the previous meeting are shown here:
> Sewell, Exploring C Semantics and Pointer Provenance [N 2311]
> Also discussed and voted on in the UB group at C++ meeting, where the 4th option was preferred. David suggested a TS could be used as an intermediary mechanism, Aaron thinks TS is often not implemented and thinks an implementation is essential before rolling it into the standard. Martin quoted on the gcc-list where Richard Beiner was deeply concerned about obstacles to optimization. Rajan was unable to run the Cerberus test cases on IBM because it didn’t pass their security restrictions.
> TS would need to be above the existing standard. Timetable: INCITS gives 3 years to finish. Provide normative text for next C meeting, in one year is earliest time to do DTS ballot, after that is publication, 1.5 .. 3 years. Next C standard expected ’22 – ’23. Suggest to push implementation in C++. Next C++ standard in ’23. Rajan thinks the testsuite should be added to the TS, perhaps as an Appendix. Make sure TS adheres to Directives part 2. Aaron requests a statement of acceptance criteria before it gets merged into the standard.
>
> Sewell, Moving to a provenance-aware memory model for C: proposal for C2x [N 2362]
> Sewell, C provenance semantics: examples [N 2363]
> Sewell, C provenance semantics: detailed semantics [N 2364]
>
> Straw poll: Should an explicit cast to an int pointer, should that be enough to justify linear traversal? 6-4-4

“representation pointer arithmetic” == convert to char*, do pointer arithmetic, convert back
poll: Move among members of struct with representation-pointer arithmetic? 8-3-6
poll: Move among members of struct using other pointer arithmetic? 2-10-3
poll: After writing a struct to a malloc’d region, can the first member be accessed via pointer to individual member type? 12-0-4
poll: Same question but non-first member? 8-0-9
poll: Q93 – 8-4-5

Tuesday afternoon

6.13 Follow-up from last meeting on C Memory Object Model Study Group discussions (continued)

Wednesday morning

6.14 Tydeman, SD3#1 (DR 440): FP types being 60559 [N 2379]
Result: Straw poll: Should N2379 be added to C2x? 7-2-6 Yes.

General discussion. Why was this not in FP TS? To avoid bias to IEEE. This adds a requirement for those systems not supporting IEEE to provide these 2 macros. Rajan argued against it, would prefer it to be in annex F.

6.15 Tydeman, printf of NaN() [N 2380]
Result: Straw poll: Is the committee in favor of something along the lines of N2380? 9-2-4 Yes.

(Fred brought new wording and is getting an N-number.)

Straw poll: Does the committee want to add N2380 with new wording to C2X? 8-2-2 Yes.

Security problem since need to know the size of output buffer: provide maximum length. Discussion about previous alternate proposal from Martin. Joseph suggests the limit needs to be larger than 32, Joseph agrees 64 is sufficient. Question about the macro name, max is usually at the end.

6.16 Thomas, C2X proposal - F.8 update [N 2384]
Result: Straw poll: Should N2384 be added to C2X? 10-0-5 Yes.

Make explicit what is the default exception handling, etc.

6.17 Thomas, C2X proposal - why no wide string strfrom functions [N 2400]
Result:
Straw poll: Does the committee want something along the lines of N2400 to be added as footnote? 15-0-1
Action Item: CFP will reword this as a recipe.

Discussion: Jens prefers this to be a footnote, and that is OK with author.

6.18 Thomas, TS 18661-4a for C2X [N 2401]
Informative discussion: Rebased to C17 and removing part B that hasn't yet been approved. Some of the functions here use intmax_t, so OK to use intmax_t and change later if the committee chooses. Is freestanding required? No. Barry disagreed with phrasing, but this sentence will not go into standard.

6.19 Thomas, TS 18661-3 as annex for C2X [N 2405]
Informative: rebasing to C17.

Wednesday afternoon

6.20 Tydeman, SNAN: initialization and unary + [N 2406]
Result:
Straw poll: Should N2406 be added to C2X? 10-0-4 Yes.

Resolve ambiguity. Fred: mere clarification. Rajan: this wasn't specified. Aaron: clang will fix it.

6.21 Thomas, Proposal for C2X - TS 18661-5abc supplementary attributes [N 2407]
Aaron: What happens to pragma's when functions are inlined? Answer: quality of implementation. Fred: even when I request the compilers to not modify code, it is modified. Jens favors to wait until there is an implementation.
(see results of straw poll at 6.23)

6.22 Thomas, Proposal for C2X - floating-point negation and conversion [N 2416]
Result: Does the committee want N2416 to be added to C2X? 11-0-2

6.23 Thomas, TS 18661-5abc for C2X - slides [N 2421]
Result:
Straw poll: Does the committee want TS 18661-5a to C2X 5-3-5 Not approved
Straw poll: Does the committee want TS 18661-5b to C2X 4-3-6 Not approved

General discussion. Aaron dislikes pragmas and requests reformulating without using the word attribute, likewise thinks that the word "law" is problematic for 'iso', but Rajan mentions that WG14 specified pragma's previously. Jens is uncomfortable with complexity and requests the existence of an implementation. Rajan refutes the 'law' comment. Aaron prefers attributes over pragma's due to clear apertainment.

6.24 Thomas, C2X proposal - footnote about why logp1 [N 2424]
Straw poll: Does the committee want N2424 to be added to C2X? 9-1-2 Yes.

Thursday morning

6.25 Ballman, What we think we reserve [N 2409]

We could prefix all externals with stdc_ ; we could split up the standard library e.g. the way that libm is split off. The Microsoft library uses underscore followed by lower case letter. Possibly introduce namespace, C now has the :: token, but it would break language interoperability and necessitate name mangling. Jens: possible solution is structured naming. Aaron will work on another version of the paper.

6.26 Gustedt, Contain the floating point naming explosion [N 2426]
General discussion. Mixed reception, much skepticism about the proposal. Aaron: The Austin group, posix, reserved the _t name so we need to discuss the issue with them.

6.27 Gustedt, Revise spelling of keywords and make them feature tests [N 2392]
Result:
Straw poll: Is the committee in agreement with the general direction of N2392? 13-1-1 Yes.
Straw poll: Is the committee in agreement with the new keywords listed in N2392 to be required to be macros? 4-8-3 No.

General discussion. Is there a need to mandate that e.g. static_assert is a macro? So the user can use it as a feature test.

Revised version, Friday morning, N2392-revised. Exempt false and true from preprocessor “decay to 0 when identifier not found”. False and true as unsigned for preprocessor vs Boolean. -true will be -1 or a very large number. State that they are type bool for language context, and signed type for preprocessor. C++ has true and false as macro definitions.

6.28 Gustedt, Make false and true first-class language features [N 2393]
Result:
Straw poll: Is the committee in favor of changing the keywords false and true to be of type bool. 15-0-0 Yes.
Straw poll: Is the committee in favor of keeping the recommended practice as described in N2393 4-6-5 No.
Straw poll: Is the committee in favor of keeping the 6.11.5 “predefined constants” future language direction as described in N2393 Vote: 2-7-4 No.

N2393-revised, Jens has direction for how to go on these papers, will submit another paper.

General discussion. Conflicting with user code that defines their own false and true. It has been obsolete for 20 years. Fred has added this new usage to his test suite and it did not cause any problems. Aaron disagrees with the “recommended practice” and “future directions” section. Joseph recommends that the true and false constants be marked as ‘u’ [unsigned].

6.29 Gustedt, Introduce the nullptr constant [N 2394]
Result:
Straw poll: Is WG14 in favor of introducing the nullptr feature into C2X? 15-1-1 Yes.
Straw poll: Is WG14 in favor of adopting N2394 into C2X? 2-8-6 No.
Straw poll: Is WG14 in favor of a transitional implementation of nullptr along the lines of N2394? 4-7-5 No.

General discussion. Some details are unspecified, need to be filled in, e.g. is nullptr a macro and allowing NULL to be #define to 0? Why is the type of the constant unspecified versus making it a pointer type? Add nullptr_t? Wording in sentence "either of the second or third…". Any incompatibilities with C++ will cause implementation difficulty. Prefer to not deprecate NULL due to existing code. Suggest to eliminate undefined behavior 6.3.2.3.3

6.30 Gustedt, Remove support for function definitions with identifier lists [N 2432]
Result:
Straw ballot: Is WG14 in favor of adding N2432 into C2X? 11-3-2 Yes.

General discussion: Aaron: the feature has been deprecated for 20 years, we should remove it. Rajan and Melanie noted that customer code uses this feature, some customers may not be able to move to the new standard because of this change to the language will invalidate their programs.

Thursday afternoon

6.31 Gustedt, Function failure annotation [N 2429]
Result:
Straw ballot: Is WG14 in favor of a mechanism along the lines described in N2429? 8-2-8 "Yes"

General discussion. This is not for C2X. Concern about function pointer types. Grammatical ambiguity concerning colon character, please clarify in the next edition of the paper. Is this a pure library feature? No, compiler needs to support. There is prototype implementation experience, need a shipping implementation, is there ABI impact? Consider proposing a TS. Are the attributes ignorable and the program gives the same behavior if attribute is ignored? Liaison statement from C++: consider using discriminated union for return versus different versions. Concern with incompatible syntax with C++ from the Cologne meeting.

6.32 Gustedt, Modernize time.h functions v.2x [N 2417]
Result:
Straw poll: Does WG14 want to adopt the POSIX functions asctime_r, ctime_r, gmtime_r, localtime_r into C2X as in N2417 without the static modifier? 12-1-4 Yes.

General discussion. Since C++ does not support [static 1] it would create difficulty. Is it an ABI breaking change for C++ interoperability? Problematic for MISRA – they have banned it. Have you checked any non-POSIX systems to see if implementation is possible? Implementation is optional.

6.33 Svoboda, Towards Integer Safety [N 2428]
Result:
Straw poll: Does WG14 want something along the line of N2428 core proposal? 13-0-2 Yes.
Straw poll: Does WG14 want something along the lines of N2428 supplemental proposal? 10-1-5 Yes.

Unsigned integer wrapping as an error condition. What about short, char, extended? Fred suggests adding a width parameter so you don't need to have multiple versions. New types are added but it doesn't explain where/how the types are declared. David likes the composability. Naming: convention for atomics differ from the convention used here—recommend that API follow the atomic naming scheme. E.g. "atomic_int_" vs. "atomic_i_".

6.34 Meneide, [[nodiscard("should have a reason")]] [N 2430]
Result:
Straw poll: Does WG14 want something along the lines of N2430 in C2X? 16-0-0

Minor concern: gcc provides users the ability to create their own diagnostic e.g. providing format string with arguments. If the standard specifies this it could step on the gcc facility. Suggest to use gcc::nodiscard to enable the gcc mechanism. The "recommended practice" section should be updated.

N2430-revised, no vote. Jean will submit with a new number.

6.35 Meneide, Restartable and Non-Restartable Functions for Efficient Character Conversions V2 [N 2440]
Result:
Straw poll: Does WG14 want something along the line of N2440 in a future revision of the C standard? 14-0-2 Yes.
Straw poll: Does WG14 want size functions as described in N2440 in a future revision of the C standard? 13-0-3 Yes.

General discussion, question about function naming. Note: add "restrict" qualifier.

6.36 Stoughton, Realloc with size 0 ambiguity [N 2438]
Action item: Robert Seacord will write a new paper on realloc with size 0.
General discussion: Not happy with the proposed wording. Origin of the issue: WG14 can support 0 sized alloc but not required to support 0. For portability, don't request size 0.

Friday morning

6.37 Ballman, Allowing unnamed parameters in a function definition [N 2381]
Result:
Straw poll: Does WG14 approve of adding something along the lines of N2381 into C2X? 14-0-2 Yes.
General discussion, C++ supports this.

6.38 Ballman, The fallthrough attribute - updates N2268 [N 2408]
Result:
Straw poll: Does WG14 approve of adding N2408 into C2X? 15-0-1 Yes.

6.39 Ballman, The noreturn attribute [N 2410]

Result:
Straw poll: Does WG14 approve of something along the lines of adding [[noreturn]] as in N2410 into C2X? 13-0-0 Yes.
Straw poll: Does WG14 agree that the standard library be decorated with [[noreturn]] instead of _Noreturn? 13-0-2 Yes.
Straw poll: Does WG14 agree to deprecate _Noreturn and the <stdnoreturn.h> header file and its contents? 12-0-1 Yes.

Discussion:
Is “noreturn” part of the function type specifier? No, same status as “inline”.
Suggestion: can use the syntactic form [[__noreturn__]] to guard against macro expansion.

6.40 Ballman, Querying attribute support - updates N2333 [N 2411]
Discussion only.

6.41 Ballman, Adding the u8 character prefix - updates N2198 [N 2418]
Result:
Straw poll: Does WG14 agree that N2418 should be added to C2X? 12-0-0 Yes.

General discussion. Include recommended practice for diagnostic warning?

6.42 Ballman, Unclear type relationship between a format specifier and its argument [N 2420]
Clarification request concerning “correct type”. General discussion. Joseph: comments made to reflector: defer to va-arg/stdarg specification. Jens: Doesn’t help for all cases e.g. scanf where arguments are pointers. Aaron will send draft to Joseph before posting to reflector. Aaron finds it difficult to understand the process for filing a defect report, writing a paper is heavy overhead.

Friday afternoon

Reserved for remainder of agenda

7. Clarification Requests

7.1 Discussion on the Clarification Request Process
Result: Blaine is resigning as editor of clarification requests. Need a lighter weight process for handling CR’s. Please think about it. Possibility: We could use a bug tracking system for the backend.

Action item: Clive will investigate supplying a bug tracking system.

All of the clarification requests in C standard and TS 19661 are now closed.

All clarification requests have been processed. The lists below are provided for review.

7.2 IS 9899:2011/9899:2018 Clarification Requests [N 2396]

7.3 TS 18661 Clarification Requests [N 2397]

8. Other Business

9. Resolutions and Decisions reached

9.1 Review of Decisions Reached

If WG14 is notified that it is in danger of disbandment, shall the convener request a 4 year revision of the C standard? Result: Yes.

Should we change the behavior of macro invocations that span files cf: N2324? Result: 3-12-0 (Yes-No-Abstain) No.

Shall we add N2412 to C2X in the next WG14 meeting if there are no objections before Feb 2, 2020? Result: 15-0-1 Yes.

Shall the requirements for the types [u]intmax_t be relaxed to cover only basic integer types and other semantic integer types as proposed in N2425? Result: 2-4-7 No.

Are we in favor of something along the lines of N2419? Result: 5-1-8 No.

Would we like to see normative text for something along the lines of N2436? 8-5-5 Yes.

(Preference vote): In N2443, can you live with Eliminate Lifetime-End Pointer Zap Altogether? Preferred (garnered 12 votes)

Should the committee accept N2311 PVNI-ae-udi model subject to satisfactory feedback from compiler implementors. 13-0-4 Yes.

Should N2379 be added to C2x? Result: 7-2-6 Yes.

Is the committee in favor of something along the lines of N2380? Result: 9-2-4 Yes.

Does the committee want to add N2380 with new wording to C2X? Result: 8-2-2 Yes. (Fred brought new wording and is getting an N-number.)

Should N2384 be added to C2X? Result: 10-0-5 Yes.

Does the committee want something along the lines of N2400 to be added as footnote? Result: 15-0-1 Yes.

Should N2406 be added to C2X? Result: 10-0-4 Yes.

Does the committee want TS 18661-5a to C2X? Result: 5-3-5 No.

Does the committee want TS 18661-5b to C2X? Result: 4-3-6 No.

Does the committee want N2424 to be added to C2X? Result: 9-1-2 Yes.

Is the committee in agreement with the general direction of N2392? Result: 13-1-1 Yes

Is the committee in agreement with the new keywords listed in N2392 to be required to be macros? Result: 4-8-3 No. Clear direction to not mandate macros.

Is the committee in favor of changing the keywords false and true to be of type bool as described in N2393. Result: 15-0-0 Yes.

Is WG14 in favor of introducing the nullptr feature into C2X? Result: 15-1-1 Yes

Is WG14 in favor of adopting N2394 into C2X? Result: 2-8-6 No.

Is WG14 in favor of a transitional implementation of nullptr along the lines of N2394? Result: 4-7-5 No, we do not want the transitional approach.

Is WG14 in favor of adding N2432 into C2X? Result: 11-3-2 Yes.

Is WG14 in favor of a mechanism along the lines described in N2429? Result: 8-2-8 Yes.

Does WG14 want to adopt the POSIX functions asctime_r, ctime_r, gmtime_r, localtime_r into C2X as in N2417 without the static modifier? Result: 12-1-4 Yes.

Does WG14 want something along the line of N2428 core proposal? Result: 13-0-2 Yes

Does WG14 want something along the lines of N2428 supplemental proposal? Result: 10-1-5 Yes

Does WG14 want something along the lines of N2430 in C2X? Result: 16-0-0 Yes.

Does WG14 want something along the line of N2440 in a future revision of the C standard? Result: 14-0-2 Yes

Does WG14 want size functions as described in N2440 in a future revision of the C standard? Result: 13-0-3 Yes

Does WG14 approve of adding something along the lines of N2381 into C2X? Result: 14-0-2 Yes

Does WG14 approve of adding N2408 into C2X? Result: 15-0-1 Yes

Does WG14 approve of something along the lines of adding [[noreturn]] as in N2410 into C2X? Result: 13-0-0 Yes.

Does WG14 agree that N2418 should be added to C2X? Result: 12-0-0 Yes.

9.2 Review of Action Items

Jens will investigate document control system alternatives, and will talk to David and Dan about the possibilities.

David: please change the convener's report to explicitly talk about the floating Liaison to C++.

Action Item for David Keaton to find out what happened to N2008.

David Keaton to put a note into SD3 about the resolution of N2324 macro invocations that span files.

cf N2390 Melanie will investigate whether there is language in the standard stating that simple assignment to atomic variables is indeed an atomic operation, and write a paper if necessary.

Jens will update wording for N2391.

David Keaton will discuss N2419 with the author.

CFP will reword N2400 as a recipe.

Robert Seacord will write a new paper for N2438 on realloc with size 0.

Clive will investigate supplying a bug tracking system.

10. PL22.11 Business

10.1 Identification of PL22.11 Voting Members
Cisco: Lars Bjonnes
GrammaTech: Aaron Ballman
IBM: Rajan Bhakta
Intel: Melanie Blower
Keaton Consulting: David Keaton
LDRA: Clive Pygott
Perennial: Barry E. Hedquist
Plum Hall: Tom Plum
Red Hat: Martin Sebor
SEI/CERT/CMU: David Svoboda
Tydeman Consulting: Fred Tydeman

10.1.1 Members Attaining initial Voting Rights at this Meeting
Facebook: Paul E. McKenney
Facebook: Maged Michael

10.1.2 Members who regained voting rights None

10.2 PL22.11 Voting Members in Jeopardy
10.2.1 Members in jeopardy due to failure to vote on Letter Ballots None

10.2.2 Members in jeopardy due to failure to attend Meetings None

10.2.2.1 Members (in jeopardy) who retained voting rights by attending this meeting None

10.2.2.2 Members (in jeopardy) who lost voting rights for failure to attend this meeting None

10.3 PL22.11 Non-voting Members
10.3.1 Prospective PL22.11 Members Attending their First Meeting
NCC Group: Robert Seacord
Perforce Software/Programming Research Ltd: Alex Gilding

10.3.2 Advisory members who are attending this meeting None

==

Other people attending

USA
Self: JeanHeyd Meneide
Self: Dr. Will Klieber

Canada
IBM: Rajan Bhakta

France

INRIA: Jens Gustedt

UK
CodeSourcery / Mentor, a Siemens Business: Joseph Myers
Univ. Cambridge: Peter Sewell

Ireland
ned Productions Ltd: Niall Douglas

11. Thanks to Host

12. Adjournment (PL22.11 motion)

Aaron moved to adjourn, Fred 2nd, approved unanimously. At 10:01am