Document Number: P3455R0
Date: 2024-05-22
Authors: Michael Wong
Project: Programming Language C++, SG14 Games Dev/Low Latency/Financial Trading/Banking/Simulation/Embedded
Reply to: Michael Wong <fraggamuffin@gmail.com>

SG14: Low Latency/Games/Embedded/Financial Trading virtual Meeting Minutes 2024/6/12-2024/10/9

Contents

Minutes for 2024/06/12 SG14 Conference Call

On Wed, Jun 12, 2024 at 12:24 AM Michael Wong <fraggamuffin_at_[hidden]> wrote:

> Hi, this SG14 meeting will focus on Games.
>
> We have a few papers on our docket this month.
>
> New Finacne SIG chair
>
> 1. An Allocator-Aware inplace_vector
>
> https://www.open-std.org/jtc1/sc22/wg21/docs/papers/2024/p3160r1.html
>
> 2. P3282 "Static Storage for C++ Concurrent bounded_queue"
>
> https://www.open-std.org/jtc1/sc22/wg21/docs/papers/2024/p3282r0.html
>
> 3. P2966R0 – Making C++ Better for Game Developers: Progress Report
>
> https://www.open-std.org/jtc1/sc22/wg21/docs/papers/2023/p2966r1.pdf
>
>
> 4.https://isocpp.org/files/papers/P3313R0.html

> Michael Wong is inviting you to a scheduled Zoom meeting.

> Other then Patrice's ongoing Games paper, are there ny other discussions?

>

> Topic: SG14 monthly

> Time: 2nd Wednesdays 02:00 PM Eastern Time (US and Canada)

> Every month on the Second Wed,

>

> Join from PC, Mac, Linux, iOS or Android:

> https://iso.zoom.us/j/93151864365?pwd=aDhOcDNWd2NWdTJuT1loeXpKbTcydz09

> Password: 789626

>

> Or iPhone one-tap :

> US: +12532158782,,93151864365# or +13017158592,,93151864365#

> Or Telephone:

> Dial(for higher quality, dial a number based on your current location):

> US: +1 253 215 8782 or +1 301 715 8592 or +1 312 626 6799 or +1

> 346 248 7799 or +1 408 638 0968 or +1 646 876 9923 or +1 669 900 6833

> or 877 853 5247 (Toll Free)

> Meeting ID: 931 5186 4365

> Password: 789626

> International numbers available: https://iso.zoom.us/u/abRrVivZoD

>

> Or Skype for Business (Lync):

> https://iso.zoom.us/skype/93151864365

>

> Agenda:

>

> 1. Opening and introduction

>

> ISO Code of Conduct

> <

>

> https://isotc.iso.org/livelink/livelink?func=ll&objId=20882226&objAction=Open&nexturl=%2Flivelink%2Flivelink%3Ffunc%3Dll%26objId%3D20158641%26objAction%3Dbrowse%26viewType%3D1

> *>*

>

> ISO patent policy.

>

> https://isotc.iso.org/livelink/livelink/fetch/2000/2122/3770791/Common_Policy.htm?nodeid=6344764&vernum=-2

>

> IEC Code of Conduct:

>

> https://www.iec.ch/basecamp/iec-code-conduct-technical-work

>

> WG21 Code of Conduct:

>

>

> https://isocpp.org/std/standing-documents/sd-4-wg21-practices-and-procedures

>

> 1.1 Roll call of participants

>

> Ville, Andrew Kostur, Andrew Drakeford, Brad Larson, Bryan St. Armour,
> Detlef Vollman, Gianluca Delfino, Jonas Persson, Matthew Butler, Pablo
> Halpern, Patrice Roy, Paul Bendixen, Sebastien Starosz, Michael Caise, Guy,
> Arthur, Michael W

>

> 1.2 Adopt agenda
>
> 1.3 Approve minutes from the previous meeting, and approve publishing
> previously approved minutes to ISOCPP.org
>
> 1.4 Action items from previous meetings
>
> 2. Main issues (125 min)
>
> 2.1 General logistics
>
> 2024 planning
> C++23 and C++26 status
> CPPCON SG14
>

+Guy help
+Bryan ST. Amout, our new Finance SIG chair

>
> Future and past meeting plans
>
> * Jan 10, 2024 02:00 PM ET: Games DONE
> * Feb 14, 2024 02:00 PM ET: Embedded DONE
> * Mar 13, 2024 02:00 PM ET: Cancelled due to Tokyo 3-18-23
> * Apr 10, 2024 02:00 PM ET: Finance DONE
> * May 8, 2024 02:00 PM ET: Games DONE
> * June 12, 2024 02:00 PM ET: Embedded; St.louis 6-24-29
> * July 10, 2024 02:00 PM ET: Finance
> * Aug 14, 2024 02:00 PM ET: Games
> * Sep 11, 2024 02:00 PM ET: CPPCON Sept 15-20 so cancelled

> * Oct 9, 2024 02:00 PM ET: Embedded
> * Nov 13, 2024 02:00 PM ET: Cancelled Wroclaw F2F
> * Dec 11, 2024 02:00 PM ET: Finance
>
> 2.2 Paper reviews
> Embedded:
> * P3132 Accept attributes with user-defined prefixes
> * P3134 Attribute [[asserts_rvo]]
> Deterministic Exception for Embedded by James Renwick
>
> https://www.pure.ed.ac.uk/ws/portalfiles/portal/78829292/low_cost_deterministic_C_exceptions_for_embedded_systems.pdf
>
> Freestanding Updates
>
> Games paper review
>
> Arthur's suggestions:
> (1) I put in the Slack channel
> <https://cpplang.slack.com/archives/C3TK2M6HH/p1703947057425609> a while
> ago Clang PR #76596 <https://github.com/llvm/llvm-project/pull/76596>,
> from
> one Max Winkler, apparently in game dev. I don't think the PR stands much
> chance of getting merged into Clang; but it might still be of interest to
> SG14 folks. The issue description is very long and somewhat detailed, and
> then there's more discussion/debate in the comments
> <https://github.com/llvm/llvm-project/pull/76596#issuecomment-1872601156>.
> (I'd actually be interested in talking to Max, but he doesn't publish his
> email address on GitHub and I guess that might be on purpose.)
>

> (2) LEWG will be seeing my P3055 "Relax wording to permit relocation
> optimizations in the STL"
> <https://quuxplusone.github.io/draft/d3055-relocation.html> in a telecon
> on
> February 20th. (Related blog post.
> <https://quuxplusone.github.io/blog/2024/01/02/bsl-vector-erase/>) Might
> be interesting to folks who do EASTL-style containers. I'd be interested in
> early feedback and/or telecon attendance.
>
>
> Discussion on Embedded:
> Paul's suggestions
> The next meeting would then be Embedded and I would be interested in
> knowing if people think a module std.freestanding is worth pursuing.
> In that context I'd like to get some feedback perhaps already for the
> upcoming meeting, if people have started using modules, and if so if it has
> brought the promised expectations or if you are holding back if you see any
> relevance in modules.
>
> Review latest mailings:
> P2532 Removing exception_ptr from the receivers concept
> Based on the last meeting and the discussions here.
> P2544 C++ Exceptions are becoming more and more problematic
> We might want to chime in here.
> /Paul
> P. S. P2327 de-deprecating volatile received a "consensus" straw poll.
>
>
> Discussion on Low Latency/Finance topics
>

> http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2022/p1839r4.pdf
>
>
>
>
>
> Discussion about Games topics:
>
> P2388R1 - Minimum Contract Support: either Ignore or Check_and_abort
> <http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2021/p2388r1.html>
>
>
> 1. An Allocator-Aware inplace_vector

https://www.open-std.org/jtc1/sc22/wg21/docs/papers/2024/p3160r1.html

Pablo presenting.

motivating example

pushing small string back to inplace vector

but if we push large strings, it needs to allocate

have space set aside to allocate content of strings

if inplace _vector were not allocator aware, cant we just pass to the elements? yes but it reduces the level of abstraction, and now you have to know about the elements, and it will be error prone

little runtime costs

want fine grain control for embedded/games/performance

allocators are in std lib and interoperate with all containers except std array

inplace_vector doesnt allocate memory but can contain objects that allocate memory, for itself and its contained elements

GD: LEWG gave tepid guidance so wants to seek stronger guidance from SG14

who wants this
some motivation was not presented in LEWG, like non-PMR example
Yes, we should add that example

PR: space cost is missing, important for embedded? if you dont use the feature, its 0, if you use it, it is the size of the stable allocator

VV: std::allocator is not freestanding? WHat about string in free-standing?
but can use a non-standard allocator because std::allocator is a problem for embedded and there is no fallback
attempt here is to make things consistent, but those facilities in the standard dont help me
nested thing that would require full allocators. That demand isn't there when we look at embedded.
It doesn't matter here because we are talking about PMR allocator. It is stateless,

PB: elaborate on freestandding where you can make your own, not what we are looking in an embedded system, the short one is the array

AO: SF for this, they snap together like legos, but no mention of optional, it is isomorphic, should propose how to solve the problem. They can be disengaged, can hold or not hold an object. When destroyed still needs an allocator (though is it designed to hold optional?) How does that work with inplace_vector (normal vector move assignment allows me to pilfer the RHS) can you show this? move assigning or move constructing element using allocator, if propagation involved, it will just propagate. That would be bad for PMR because some elements will ... No always use destination allocator. Will clarify, but not a problem. You dont have to erase all on the LHS. CAn do an element by element move

DV: many embedded systems would use inplace_vector but not allocator, it also have a lot of use outside the embedded world. U should look for support also outside of this group.

KE: +1 to DV can use in a lot of the places, having anallocator does not make me not want it, as long as default allocator

VV:
AO,the explanation for how a type like this works, and how propagation works in it, is in
https://www.open-std.org/jtc1/sc22/wg21/docs/papers/2024/p2047r7.html
propagation traits assignment ... destruction prior assignment ... PMRs are not the problem here; our allocator API do not support non-eh behaviour? we dont subset the language; embedded to day is different; ok to have that complexity in a separate type: POCA is always false, AO: but it is what LEWG spends time specifying

AO:
I believe it's feasible to make the skeleton of `std::allocator` freestanding, and just omit the `allocate` and `deallocate` methods.
See Ben Craig's recent
https://www.open-std.org/jtc1/sc22/wg21/docs/papers/2023/p2407r5.html
for the standardese.

MC: we dont have EH, this allows me to pay for an expense, the container is super useful in a place that dont have containers
and having inplace_vector vocabulary type is always important

OK to not solve some of the stickier problems.

Get this useful type, then add something else later

PH: IF inplace vector gets in without allocator, then we can do a separate type

Poll: If the embeded issues can be solved, would you be OK with adding the allocator template parameter to inplace vector

6/8/1/3/0

33/44/6/17/0%

18 voted

In favor

2. P3282 "Static Storage for C++ Concurrent bounded_queue"

https://www.open-std.org/jtc1/sc22/wg21/docs/papers/2024/p3282r0.html

bounded queues constructor only has the number elements and then allocator, when u have allocator it will use new for it, and it only allocates in its constructor, it never allocate later

added required_size and alignment

now you can define type alias

giving ptr to constructor

do you like this? its not part of the type whether the object owns the storage or not, object has to remember it internally

KE: unsafe? like span. Preferred API for constructing a bounded queue? I can live with span

AO: looks unsafe, ordered length, ptr, at least switch, seems like 2 features, one change memory footprint to say if I own my memory, static members require alignment. What is alignment of list node? what about the rest? this looks like a major feature trying to get out?
A queue is not a container, its more like a future
in embeded world there is no new, so size must be known at compile time (same with lists, deque) but these allocate all the time at each push whereas this only allocates at construction.

VV: seems to be trying to convey require size. Why 8? its my static storage because I dont know the implementation of bounded queue works.

aiming for SG1 as well.
Who is in favor of this interface
1/2/5/5/2
7/13/33/33/13%
Not in favor

3. P2966R0 – Making C++ Better for Game Developers: Progress Report
https://www.open-std.org/jtc1/sc22/wg21/docs/papers/2023/p2966r1.pdf
Patrice presenting
a spreadsheet of the subpapers will add to Google sheet
so others can add as authors specific papers:
https://docs.google.com/spreadsheets/d/1uCP49gOKZvheLGDZnbliCg50nHMEaxmjqWVLoc-Gz_A/edit?gid=0#gid=0

4. https://isocpp.org/files/papers/P3313R0.html

Khalil presenting

writing my own EH for arm

Function EH rank:

1. no index entry: leaf function
2. inlined index data: inlined into index directly
3. table unwinding instructions: additional data outside of index
4. GCC LSDA: generic data structure to manage try/catch blocks

GCC uses large data structures for fns with noexcept

include a single except function will poison

GUidance on how to save space

deduce noexcept in functions

PR: conditional noexcept: code gen improvement through tooling.

MC: LTO used? no cant get it to work, mess up my build

MW: what about other platforms? No change expected.

VV: enabling LTO in package build may have caused/or not caused more problems

> 2.2.1 any other proposal for reviews?
>
>
>
>
>
>
> SG14/SG19 features/issues/defects:
>
>
>

>
https://docs.google.com/spreadsheets/d/1JnUJBO72QVURttkKr7gn0_WjP--P0vAne8JBfzbRiy0/edit#gid=0

>

> 2.3 Domain-specific discussions

>

> 2.3.1 SIG chairs

>

> - Embedded Programming chairs: Ben Craig, Wouter van Ooijen and Odin

> Holmes, John McFarlane

>

> - Financial/Trading chairs: Robin Rowe, Staffan TjernstrÃm

> Carl Cooke, Neal Horlock,

> - Games chairs: Rene Riviera, Guy Davidson and Paul Hampson, Patrice Roy

>

> - Linear Algebra chairs: Bob Steagall, Mark Hoemmen, Guy Davidson

>

> 2.4 Other Papers and proposals

>

> 2.5 Future F2F meetings:

>

> 2.6 future C++ Standard meetings:

> https://isocpp.org/std/meetings-and-participation/upcoming-meetings

>

> -

>

> 3. Any other business

> Reflector

> https://lists.isocpp.org/mailman/listinfo.cgi/sg14

> As well as look through papers marked "SG14" in recent standards committee

> paper mailings:

> http://open-std.org/jtc1/sc22/wg21/docs/papers/2015/
> http://open-std.org/jtc1/sc22/wg21/docs/papers/2016/
>
> Code and proposal Staging area
> https://github.com/WG21-SG14/SG14
> 4. Review
>
> 4.1 Review and approve resolutions and issues [e.g., changes to SG's
> working draft]
>
> 4.2 Review action items (5 min)
>
> 5. Closing process
>
> 5.1 Establish next agenda
>
> 5.2 Future meeting
>
>
> * Jan 10, 2024 02:00 PM ET: Games DONE
> * Feb 14, 2024 02:00 PM ET: Embedded DONE
> * Mar 13, 2024 02:00 PM ET: Cancelled due to Tokyo 3-18-23
> * Apr 10, 2024 02:00 PM ET: Finance DONE
> * May 8, 2024 02:00 PM ET: Games DONE
> * June 12, 2024 02:00 PM ET: Embedded; St.louis 6-24-29
> * July 10, 2024 02:00 PM ET: Finance
> * Aug 14, 2024 02:00 PM ET: Games
> * Sep 11, 2024 02:00 PM ET: CPPCON Sept 15-20 so cancelled
> * Oct 9, 2024 02:00 PM ET: Embedded
> * Nov 13, 2024 02:00 PM ET: Cancelled Wroclaw F2F

> * Dec 11, 2024 02:00 PM ET: Finance

>

Minutes for 2024/7/10 SG14 Conference Call

Bryan chairing

On Tue, Jul 9, 2024 at 6:16 PM Michael Wong <fraggamuffin_at_[hidden]> wrote:

> Hi, this SG14 meeting will focus on Finance.
>
> We can also continue the discussion on P2966. Are there any other papers?
>
>
> P2966R0 – Making C++ Better for Game Developers: Progress Report
> https://www.open-std.org/jtc1/sc22/wg21/docs/papers/2023/p2966r1.pdf
>
> Michael Wong is inviting you to a scheduled Zoom meeting.
> Other then Patrice's ongoing Games paper, are there ny other discussions?
>
> Topic: SG14 monthly
> Time: 2nd Wednesdays 02:00 PM Eastern Time (US and Canada)
> Every month on the Second Wed,
>
> Join from PC, Mac, Linux, iOS or Android:
> https://iso.zoom.us/j/93151864365?pwd=aDhOcDNWd2NWdTJuT1loeXpKbTcydz09
> Password: 789626
>
> Or iPhone one-tap :
> US: +12532158782,,93151864365# or +13017158592,,93151864365#
> Or Telephone:
> Dial(for higher quality, dial a number based on your current location):
> US: +1 253 215 8782 or +1 301 715 8592 or +1 312 626 6799 or +1
> 346 248 7799 or +1 408 638 0968 or +1 646 876 9923 or +1 669 900 6833

> or 877 853 5247 (Toll Free)

> Meeting ID: 931 5186 4365

> Password: 789626

> International numbers available: https://iso.zoom.us/u/abRrVivZoD

>

> Or Skype for Business (Lync):

> https://iso.zoom.us/skype/93151864365

>

> Agenda:

>

> 1. Opening and introduction

>

> ISO Code of Conduct

> <

>

> https://isotc.iso.org/livelink/livelink?func=ll&objId=20882226&objAction=Open&nexturl=%2Flivelink%2Flivelink%3Ffunc%3Dll%26objId%3D20158641%26objAction%3Dbrowse%26viewType%3D1

> *>*

>

> ISO patent policy.

>

> https://isotc.iso.org/livelink/livelink/fetch/2000/2122/3770791/Common_Policy.htm?nodeid=6344764&vernum=-2

>

> IEC Code of Conduct:

>

> https://www.iec.ch/basecamp/iec-code-conduct-technical-work

>

> WG21 Code of Conduct:

>

>

> https://isocpp.org/std/standing-documents/sd-4-wg21-practices-and-procedures

>

> 1.1 Roll call of participants

>

> Vryan, Kris Jusiak, Matthew Butler, Michael Levine, Paradox84, Nurullah
CITIL), Stefano Fiorentino, Bryan St. AMour, Michael Wong

> 1.2 Adopt agenda

>

> 1.3 Approve minutes from the previous meeting, and approve publishing

> previously approved minutes to ISOCPP.org

>

> 1.4 Action items from previous meetings

>

> 2. Main issues (125 min)

>

> 2.1 General logistics

>

> 2024 planning

> C++23 and C++26 status

> CPPCON SG14/19 Sept 18 Wednesday 2 pm Mountain time

> https://isocpp.org/files/papers/P3313R0.html

> Khalil presenting

> writing my own EH for arm

>

>

> Future and past meeting plans
>
> * Jan 10, 2024 02:00 PM ET: Games DONE
> * Feb 14, 2024 02:00 PM ET: Embedded DONE
> * Mar 13, 2024 02:00 PM ET: Cancelled due to Tokyo 3-18-23
> * Apr 10, 2024 02:00 PM ET: Finance DONE
> * May 8, 2024 02:00 PM ET: Games DONE
> * June 12, 2024 02:00 PM ET: Embedded; St.louis 6-24-29 DONE
> * July 10, 2024 02:00 PM ET: Finance
> * Aug 14, 2024 02:00 PM ET: Games
> * Sep 11, 2024 02:00 PM ET: CPPCON Sept 15-20 so cancelled
> * Oct 9, 2024 02:00 PM ET: Embedded
> * Nov 13, 2024 02:00 PM ET: Cancelled Wroclaw F2F
> * Dec 11, 2024 02:00 PM ET: Finance
>
> 2.2 Paper reviews
>
> Embedded:
>
> * P3132 Accept attributes with user-defined prefixes
> * P3134 Attribute [[asserts_rvo]]
>
> Deterministic Exception for Embedded by James Renwick
>
> https://www.pure.ed.ac.uk/ws/portalfiles/portal/78829292/low_cost_deterministic_C_exceptions_for_embedded_systems.pdf
>
> Freestanding Updates
>
> Games paper review
>
> Arthur's suggestions:
>
> (1) I put in the Slack channel

> <https://cpplang.slack.com/archives/C3TK2M6HH/p1703947057425609> a while
> ago Clang PR #76596 <https://github.com/llvm/llvm-project/pull/76596>,
> from
> one Max Winkler, apparently in game dev. I don't think the PR stands much
> chance of getting merged into Clang; but it might still be of interest to
> SG14 folks. The issue description is very long and somewhat detailed, and
> then there's more discussion/debate in the comments
> <https://github.com/llvm/llvm-project/pull/76596#issuecomment-1872601156>.
> (I'd actually be interested in talking to Max, but he doesn't publish his
> email address on GitHub and I guess that might be on purpose.)
>
> (2) LEWG will be seeing my P3055 "Relax wording to permit relocation
> optimizations in the STL"
> <https://quuxplusone.github.io/draft/d3055-relocation.html> in a telecon
> on
> February 20th. (Related blog post.
> <https://quuxplusone.github.io/blog/2024/01/02/bsl-vector-erase/>) Might
> be interesting to folks who do EASTL-style containers. I'd be interested in
> early feedback and/or telecon attendance.
>
>
> Discussion on Embedded:
> Paul's suggestions
> The next meeting would then be Embedded and I would be interested in
> knowing if people think a module std.freestanding is worth pursuing.
> In that context I'd like to get some feedback perhaps already for the
> upcoming meeting, if people have started using modules, and if so if it has
> brought the promised expectations or if you are holding back if you see any
> relevance in modules.
>

> Review latest mailings:
>
> P2532 Removing exception_ptr from the receivers concept
>
> Based on the last meeting and the discussions here.
>
> P2544 C++ Exceptions are becoming more and more problematic
>
> We might want to chime in here.
>
> /Paul
>
> P. S. P2327 de-deprecating volatile received a "consensus" straw poll.
>
>
>
> Discussion on Low Latency/Finance topics
>
> http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2022/p1839r4.pdf

We discussed this document at length to gather future interest and co-authors.

>
>
>
> https://docs.google.com/document/d/1DmHQKgigCe1Ei6DYtztzUvfZ6oJdphiPbZRDpyT185U/edit
>
>
>
> Discussion about Games topics:
>
> P2388R1 - Minimum Contract Support: either Ignore or Check_and_abort
>
> <http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2021/p2388r1.html>
>
>
>

>

>

> 2.2.1 any other proposal for reviews?

>

>

>

> SG14/SG19 features/issues/defects:

>

>

> https://docs.google.com/spreadsheets/d/1JnUJBO72QVURttkKr7gn0_WjP--P0vAne8JBfzbRiy0/edit#gid=0

>

> 2.3 Domain-specific discussions

>

> 2.3.1 SIG chairs

>

> - Embedded Programming chairs: Ben Craig, Wouter van Ooijen and Odin

> Holmes, John McFarlane

>

> - Financial/Trading chairs: Robin Rowe, Staffan TjernstrÃm

> Carl Cooke, Neal Horlock,

> - Games chairs: Rene Riviera, Guy Davidson and Paul Hampson, Patrice Roy

>

> - Linear Algebra chairs: Bob Steagall, Mark Hoemmen, Guy Davidson

>

> 2.4 Other Papers and proposals

>

> 2.5 Future F2F meetings:

>

> 2.6 future C++ Standard meetings:

> https://isocpp.org/std/meetings-and-participation/upcoming-meetings
>
> -
>
> 3. Any other business

> Reflector

> https://lists.isocpp.org/mailman/listinfo.cgi/sg14

> As well as look through papers marked "SG14" in recent standards committee
> paper mailings:

> http://open-std.org/jtc1/sc22/wg21/docs/papers/2015/

> http://open-std.org/jtc1/sc22/wg21/docs/papers/2016/
>
> Code and proposal Staging area

> https://github.com/WG21-SG14/SG14

> 4. Review
>
> 4.1 Review and approve resolutions and issues [e.g., changes to SG's
> working draft]
>
> 4.2 Review action items (5 min)
>
> 5. Closing process
>
> 5.1 Establish next agenda
>
> 5.2 Future meeting
>
>
> * Jan 10, 2024 02:00 PM ET: Games DONE
> * Feb 14, 2024 02:00 PM ET: Embedded DONE

> * Mar 13, 2024 02:00 PM ET: Cancelled due to Tokyo 3-18-23

> * Apr 10, 2024 02:00 PM ET: Finance DONE

> * May 8, 2024 02:00 PM ET: Games DONE

> * June 12, 2024 02:00 PM ET: Embedded; St.louis 6-24-29 DONE

> * July 10, 2024 02:00 PM ET: Finance

> * Aug 14, 2024 02:00 PM ET: Games

> * Sep 11, 2024 02:00 PM ET: CPPCON Sept 15-20 so cancelled

> * Oct 9, 2024 02:00 PM ET: Embedded

> * Nov 13, 2024 02:00 PM ET: Cancelled Wroclaw F2F

> * Dec 11, 2024 02:00 PM ET: Finance

Minutes for 2024/10/9 SG14 Conference Call

Meeting Notes

On Wed, Oct 9, 2024 at 2:04 PM Michael Wong <fraggamuffin_at_[hidden]> wrote:

> Thanks Khalil, please join now.
>
> On Wed, Oct 9, 2024 at 12:17 PM Khalil Estell via SG14 <
> sg14_at_[hidden]> wrote:
>
>> I'll be there. I can go over my paper.
>>
>> On Wed, Oct 9, 2024, 09:16 Patrice Roy via SG14 <sg14_at_[hidden]>
>> wrote:
>>
>>> I won't be there, sadly, as I have a meeting at work at the same time
>>> and my presence there is required.
>>>
>>> Some updates, though:
>>>
>>> The paper on [[invalidate_dereferencing]] that was presented at CppCon
>>> should be in the next mailing (it's pretty much done).
>>>
>>> There might be another one on a trait that has been requested by some
>>> game developers that contacted me last month, and that I discussed a bit
>>> with the ever wonderful Mark Hoemmen at CppCon. The idea is that
>>> std::has_unique_object_representation<T> currently does not work with a
>>> type T that has floating point data members (due in part to the potential
>>> presence of NaN values), but there are use-cases for such a trait when user
>>> code knows that the floating point numbers therein will not be NaN, or
>>> (maybe) that the only NaN values that could occur would be some canonical

>>> NaN with a fixed bit pattern. If I manage to clean it up in time for the

>>> next mailing, I'll add this one too so feel free to react (either on this

>>> list or directly through email) after you've read it. The tentative name

>>> would be std::is_always_exhaustive<T> (taking the name from the work done

>>> by the fine mdspan people if I understand correctly); we'll adjust if need

>>> be.

>>>

>>> Thanks!

>>>

>>>

>>>

>>>

>>> Le mer. 9 oct. 2024 à 11:07, Guy Davidson via SG14 <

>>> sg14_at_[hidden]> a écrit :

>>>

>>>> I am happy to talk about P3375 reproducible floating-point arithmetic

>>>> but I have already had plenty of SG14/19 time. I've just submitted R1 to

>>>> the papers system, but I would much rather defer to other papers.

>>>>

>>>> On Wed, 9 Oct 2024 at 15:52, Bryan St. Amour via SG14 <

>>>> sg14_at_[hidden]> wrote:

>>>>

>>>>> I may be a bit late, but I should be able to join today's call.

>>>>>

>>>>> On Tue, Oct 08, 2024 at 04:07:20PM -0400, Michael Wong via SG14 wrote:

>>>>> > Hi, this SG14 meeting will focus on Embedded.

>>>>> > We can also continue the discussion on P2966. Are there any other

>>>>> papers?

>>>>> >

>>>>> >

>>>>> > P2966R0 – Making C++ Better for Game Developers: Progress Report
>>>>> > https://www.open-std.org/jtc1/sc22/wg21/docs/papers/2023/p2966r1.pdf
>>>>> >
>>>>> > Michael Wong is inviting you to a scheduled Zoom meeting.
>>>>> > Other then Patrice's ongoing Games paper, are there ny other
>>>>> discussions?
>>>>> >
>>>>> > Topic: SG14 monthly
>>>>> > Time: 2nd Wednesdays 02:00 PM Eastern Time (US and Canada)
>>>>> > Every month on the Second Wed,
>>>>> >
>>>>> > Join from PC, Mac, Linux, iOS or Android:
>>>>> >
>>>>> https://iso.zoom.us/j/93151864365?pwd=aDhOcDNWd2NWdTJuT1IoeXpKbTcydz09
>>>>> > Password: 789626
>>>>> >
>>>>> > Or iPhone one-tap :
>>>>> > US: +12532158782,,93151864365# or +13017158592,,93151864365#
>>>>> > Or Telephone:
>>>>> > Dial(for higher quality, dial a number based on your current
>>>>> location):
>>>>> > US: +1 253 215 8782 or +1 301 715 8592 or +1 312 626 6799 or
>>>>> +1
>>>>> > 346 248 7799 or +1 408 638 0968 or +1 646 876 9923 or +1 669 900 6833
>>>>> > or 877 853 5247 (Toll Free)
>>>>> > Meeting ID: 931 5186 4365
>>>>> > Password: 789626
>>>>> > International numbers available:
>>>>> https://iso.zoom.us/u/abRrVivZoD
>>>>> >

>>>>> > Or Skype for Business (Lync):

>>>>> > https://iso.zoom.us/skype/93151864365

>>>>> >

>>>>> > Agenda:

>>>>> >

>>>>> > 1. Opening and introduction

>>>>> >

>>>>> > ISO Code of Conduct

>>>>> > <

>>>>> >

>>>>> https://isotc.iso.org/livelink/livelink?func=ll&objId=20882226&objAction=Open&nexturl=%2Flivelink%2Flivelink%3Ffunc%3Dll%26objId%3D20158641%26objAction%3Dbrowse%26viewType%3D1

>>>>> > *>*

>>>>> >

>>>>> > ISO patent policy.

>>>>> >

>>>>> https://isotc.iso.org/livelink/livelink/fetch/2000/2122/3770791/Common_Policy.htm?nodeid=6344764&vernum=-2

>>>>> >

>>>>> > IEC Code of Conduct:

>>>>> >

>>>>> > https://www.iec.ch/basecamp/iec-code-conduct-technical-work

>>>>> >

>>>>> > WG21 Code of Conduct:

>>>>> >

>>>>> >

>>>>> https://isocpp.org/std/standing-documents/sd-4-wg21-practices-and-procedures

>>>>> >

>>>>> > 1.1 Roll call of participants

>>>>>

>>>> Guy, Paul, Andre, Bjarne, Brad, Gianluca, Jonas, Kris, Matthew,

> Rodrigo, Bryan, Gianluca, Jake, Stefano, Khalil, Michael

>

> >

>>>>> > 1.2 Adopt agenda

>>>>> >

>>>>> > 1.3 Approve minutes from the previous meeting, and approve publishing

>>>>> > previously approved minutes to ISOCPP.org

>>>>> >

>>>>> > 1.4 Action items from previous meetings

>>>>> >

>>>>> > 2. Main issues (125 min)

>>>>> >

>>>>> > 2.1 General logistics

>>>>> >

>>>>> > 2024 planning

>>>>> > C++23 and C++26 status

>>>>> > CPPCON SG14/19

>>>>> >

>>>>> > Future and past meeting plans

>>>>> >

>>>>> > * Jan 10, 2024 02:00 PM ET: Games DONE

>>>>> > * Feb 14, 2024 02:00 PM ET: Embedded DONE

>>>>> > * Mar 13, 2024 02:00 PM ET: Cancelled due to Tokyo 3-18-23

>>>>> > * Apr 10, 2024 02:00 PM ET: Finance DONE

>>>>> > * May 8, 2024 02:00 PM ET: Games DONE

>>>>> > * June 12, 2024 02:00 PM ET: Embedded; St.louis 6-24-29 DONE

>>>>> > * July 10, 2024 02:00 PM ET: Finance DONE

>>>>> > * Aug 14, 2024 02:00 PM ET: Games DONE

>>>>> > * Sep 11, 2024 02:00 PM ET: CPPCON Sept 15-20 so cancelled DONE

>>>>> > * Oct 9, 2024 02:00 PM ET: Embedded

>>>>> > * Nov 13, 2024 02:00 PM ET: Cancelled Wroclaw F2F

>>>>> > * Dec 11, 2024 02:00 PM ET: Finance

>>>>> >

>>>>> > 2.2 Paper reviews

>>>>> > Embedded:

>>>>> > * P3132 Accept attributes with user-defined prefixes

>>>>> > * P3134 Attribute [[asserts_rvo]]

>>>>> > Deterministic Exception for Embedded by James Renwick

>>>>> >

>>>>> https://www.pure.ed.ac.uk/ws/portalfiles/portal/78829292/low_cost_deterministic_C_exceptions_for_embedded_systems.pdf

>>>>> >

>>>>

>>>> https://www.open-std.org/jtc1/sc22/wg21/docs/papers/2024/p3313r0.html

Khalil's talk on noexcept and its impact on exception handling is highlighted, with suggestions for a shorter, more accessible version.

The discussion focused on the suboptimality of current strategies for managing exception metadata in C++, particularly with GCC and clang. Key points included the inefficiency of noexcept functions, such as unnecessary metadata and increased code size. Proposed solutions included grouping noexcept functions together to reduce memory usage and using link-time analysis to identify implicit noexcept functions. The conversation also highlighted the challenges of dynamic linking and the potential for future tools to optimize exception handling without relying on noexcept annotations. The meeting emphasized the need for better compiler optimizations and guidance for developers.

Action Items

- [ ] Optimize code size by grouping "no except" functions together.
- [ ] Investigate link time analysis to determine functions that are effectively "no except" without having to label them as such.
- [ ] Provide recommendations for optimizing code generation for "no except" functions.

>>>>> > Freestanding Updates
>>>>> >
>>>>> > Games paper review
>>>>> >
>>>>> > Arthur's suggestions:
>>>>> > (1) I put in the Slack channel
>>>>> > <https://cpplang.slack.com/archives/C3TK2M6HH/p1703947057425609> a
>>>>> while
>>>>> > ago Clang PR #76596 <https://github.com/llvm/llvm-project/pull/76596>,
>>>>> from
>>>>> > one Max Winkler, apparently in game dev. I don't think the PR stands
>>>>> much
>>>>> > chance of getting merged into Clang; but it might still be of
>>>>> interest to
>>>>> > SG14 folks. The issue description is very long and somewhat
>>>>> detailed, and
>>>>> > then there's more discussion/debate in the comments
>>>>> > <
>>>>> https://github.com/llvm/llvm-project/pull/76596#issuecomment-1872601156

>>>>> >.

>>>>> > (I'd actually be interested in talking to Max, but he doesn't

>>>>> publish his

>>>>> > email address on GitHub and I guess that might be on purpose.)

>>>>> >

>>>>> > (2) LEWG will be seeing my P3055 "Relax wording to permit relocation

>>>>> > optimizations in the STL"

>>>>> > <https://quuxplusone.github.io/draft/d3055-relocation.html> in a

>>>>> telecon on

>>>>> > February 20th. (Related blog post.

>>>>> > <https://quuxplusone.github.io/blog/2024/01/02/bsl-vector-erase/>)

>>>>> Might

>>>>> > be interesting to folks who do EASTL-style containers. I'd be

>>>>> interested in

>>>>> > early feedback and/or telecon attendance.

>>>>> >

>>>>> >

>>>>> > Discussion on Embedded:

>>>>> > Paul's suggestions

>>>>> > The next meeting would then be Embedded and I would be interested in

>>>>> > knowing if people think a module std.freestanding is worth pursuing.

>>>>> > In that context I'd like to get some feedback perhaps already for the

>>>>> > upcoming meeting, if people have started using modules, and if so if

>>>>> it has

>>>>> > brought the promised expectations or if you are holding back if you

>>>>> see any

>>>>> > relevance in modules.

>>>>> >

>>>>> > Review latest mailings:

>>>>> > P2532 Removing exception_ptr from the receivers concept

>>>>> > Based on the last meeting and the discussions here.

>>>>> > P2544 C++ Exceptions are becoming more and more problematic

>>>>> > We might want to chime in here.

>>>>> > /Paul

>>>>> > P. S. P2327 de-deprecating volatile received a "consensus" straw

>>>>> poll.

>>>>> >

>>>>> >

>>>>> > Discussion on Low Latency/Finance topics

>>>>> >

>>>>> > http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2022/p1839r4.pdf

>>>>> >

>>>>> >

>>>>> >

>>>>> >

>>>>> >

>>>>> > Discussion about Games topics:

>>>>> >

>>>>> > P2388R1 - Minimum Contract Support: either Ignore or Check_and_abort

>>>>> > <

>>>>> http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2021/p2388r1.html>

>>>>> >

>>>>> >

>>>>> >

>>>>> >

>>>>> >

>>>>> > 2.2.1 any other proposal for reviews?

>>>>> >

>>>>> >

>>>>> >

>>>>> > SG14/SG19 features/issues/defects:

>>>>> >

>>>>> >

>>>>> https://docs.google.com/spreadsheets/d/1JnUJBO72QVURttkKr7gn0_WjP--P0vAne8JBfzbRiy0/edit#gid=0

>>>>> >

>>>>> > 2.3 Domain-specific discussions

>>>>> >

>>>>> > 2.3.1 SIG chairs

>>>>> >

>>>>> > - Embedded Programming chairs: Ben Craig, Wouter van Ooijen and

>>>>> Odin

>>>>> > Holmes, John McFarlane

>>>>> >

>>>>> > - Financial/Trading chairs: Robin Rowe, Staffan TjernstrÃm

>>>>> > Carl Cooke, Neal Horlock,

>>>>> > - Games chairs: Rene Riviera, Guy Davidson and Paul Hampson,

>>>>> Patrice Roy

>>>>> >

>>>>> > - Linear Algebra chairs: Bob Steagall, Mark Hoemmen, Guy Davidson

>>>>> >

>>>>> > 2.4 Other Papers and proposals

>>>>> >

>>>>> > 2.5 Future F2F meetings:

>>>>> >

>>>>> > 2.6 future C++ Standard meetings:

>>>>> > https://isocpp.org/std/meetings-and-participation/upcoming-meetings

>>>>> >

>>>>> > -

>>>>> >

>>>>> > 3. Any other business

>>>>> > Reflector

>>>>> > https://lists.isocpp.org/mailman/listinfo.cgi/sg14

>>>>> > As well as look through papers marked "SG14" in recent standards

>>>>> committee

>>>>> > paper mailings:

>>>>> > http://open-std.org/jtc1/sc22/wg21/docs/papers/2015/

>>>>> > http://open-std.org/jtc1/sc22/wg21/docs/papers/2016/

>>>>> >

>>>>> > Code and proposal Staging area

>>>>> > https://github.com/WG21-SG14/SG14

>>>>> > 4. Review

>>>>> >

>>>>> > 4.1 Review and approve resolutions and issues [e.g., changes to SG's

>>>>> > working draft]

>>>>> >

>>>>> > 4.2 Review action items (5 min)

>>>>> >

>>>>> > 5. Closing process

>>>>> >

>>>>> > 5.1 Establish next agenda

>>>>> >

>>>>> > 5.2 Future meeting

>>>>> >

>>>>> >

>>>>> > * Jan 10, 2024 02:00 PM ET: Games DONE

>>>>> > * Feb 14, 2024 02:00 PM ET: Embedded DONE

>>>>> > * Mar 13, 2024 02:00 PM ET: Cancelled due to Tokyo 3-18-23

>>>>> > * Apr 10, 2024 02:00 PM ET: Finance DONE

>>>>> > * May 8, 2024 02:00 PM ET: Games DONE

>>>>> > * June 12, 2024 02:00 PM ET: Embedded; St.louis 6-24-29 DONE

>>>>> > * July 10, 2024 02:00 PM ET: Finance

>>>>> > * Aug 14, 2024 02:00 PM ET: Games

>>>>> > * Sep 11, 2024 02:00 PM ET: CPPCON Sept 15-20 so cancelled

>>>>> > * Oct 9, 2024 02:00 PM ET: Embedded

>>>>> > * Nov 13, 2024 02:00 PM ET: Cancelled Wroclaw F2F

>>>>> > * Dec 11, 2024 02:00 PM ET: Finance

>>>>>

>>>>> > ____

>>>>> > SG14 mailing list

>>>>> > SG14_at_[hidden]

>>>>> > https://lists.isocpp.org/mailman/listinfo.cgi/sg14

>>>>>

>>>>> ____

>>>>> SG14 mailing list

>>>>> SG14_at_[hidden]

>>>>> https://lists.isocpp.org/mailman/listinfo.cgi/sg14

>>>>>

>>>> ____

>>>> SG14 mailing list

>>>> SG14_at_[hidden]

>>>> https://lists.isocpp.org/mailman/listinfo.cgi/sg14

>>>>

>>> ____

>>> SG14 mailing list

>>> SG14_at_[hidden]

>>> https://lists.isocpp.org/mailman/listinfo.cgi/sg14

>>>

>> ____

>> SG14 mailing list

>> SG14_at_[hidden]

>> https://lists.isocpp.org/mailman/listinfo.cgi/sg14

>>