

Izmir Institute of Technology
Computer Engineering Department
CENG513 Final Exam Spring 2024
Question 2

Student Name: Gökay Gülsoy Student No: 270201072

June 11, 2024



Question 2

I have chosen the github issue with name "Incorrect code generation when initializing a data member from a consteval function call". Issue is related to unexpected return value from a function due to some problems in code generation [1]. Code snippet for the issue is given as follows:

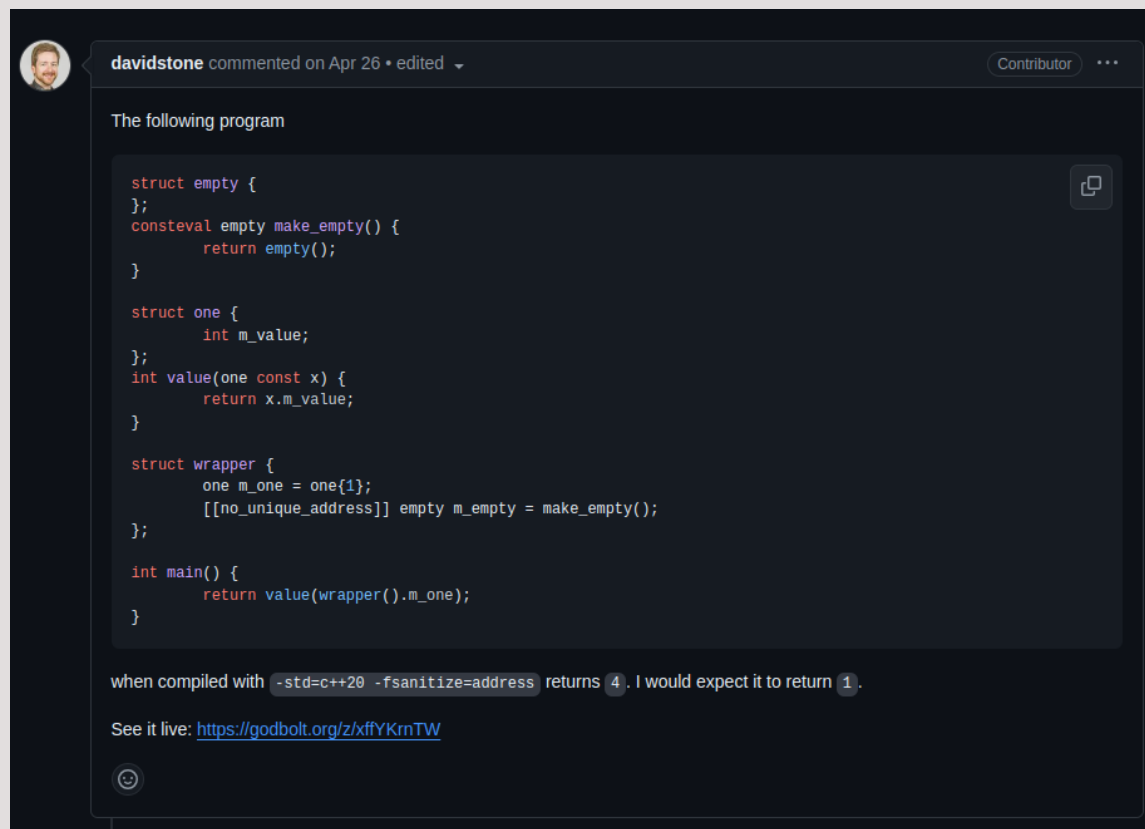


Figure 1: Incorrect code generation when initializing a data member from a consteval function call

The problem indicated in the issue #90130 may be related to use of `[[no_unique_address]]` attribute which allows compiler to optimize the memory layout of structs by allowing members that don't require unique address (like empty structs) to potentially occupy the same address as another [2]. C++ standard allows compilers a leeway in terms of memory layout optimizations, especially with empty members in structs. The combination of those optimizations and specific behaviors of memory sanitizer might lead to undefined behavior [3], where "m_one" is incorrectly initialized or its memory is incorrectly shared or overwritten. My suggestions to fix the problem are, removing the `[[no_unique_address]]` attribute and recompile to see if the output changes because if we force each data member to occupy unique address in struct, corresponding

generated assembly code may not interfere with different data member's addresses. Compiling the code without the sanitizer to check whether it's inference might be causing the unexpected behavior by removing command line option `-fsanitize=address`. Compiling with different optimization levels (`-O0`, `-O1`, `-O2`, `-O3`) to see how they affect the output.

References

- [1] *Llvm 19.0 documentation*, <https://llvm.org/docs/>, The LLVM Compiler Infrastructure, 2020.
- [2] ISO/IEC JTC1/SC22/WG21, *ISO International Standard ISO/IEC 14882:2020 - Programming Language C++*. Geneva, Switzerland: International Organization for Standardization, 2020.
- [3] Google, *Addresssanitizer - a fast memory error detector*, Online, <https://github.com/google/sanitizers/wiki/AddressSanitizer>, 2021.