

Progress report in Pen programming language

July 17th, 2022

[@raviqqe](#)

Agenda

- Progress report
 - Reference counting optimization
 - C calling convention (no progress)
- Next plans

Progress report

Reference counting optimization

- Fibonacci number is 25% faster than the previous version!
 - Now, it's 2 times slower than Rust... (with floating-pointer numbers)
- Static pointer tagging was preventing direct calls of global functions.
 - Now, Pen uses *static* counts for statically allocated memory blocks.
 - Pointer tagging and un-tagging were not removed by LLVM's optimization somehow even with correct alignment of global variables.
 - Clang does it but at the level of CIL?
 - Removal of pointer tags also led to simpler static check of memory blocks.

C calling convention (no progress)

- The Zig's developer says:
https://twitter.com/andy_kelley/status/1527743699836280833
- LLVM's C calling convention is not the one used by C.
 - Linux on x86 uses the System V ABI.
 - LLVM handles only part of it.
 - Register spilling
 - Struct decomposition
- MLIR has it as C wrapper emission for the LLVM dialect.
 - Not released yet. Maybe in LLVM 15?

C calling convention (no progress, continued)

- C backend again for F--?
 - F-- is Pen's lower-level IR.
 - Currently, C backend doesn't support guaranteed tail call optimization.
 - Clang's `musttail` needs function signature matching.
 - C's standard proposal also follows the same design for portability.

Next plans

- Standard library improvements
 - More functionality
 - More completeness
- Application development?
 - Web application?
 - WebGL??
 - Currently, tail calls in WASM is only supported by Chrome (and Node.js.)
- Efficient C calling convention in FFI [#444](#)
 - Compiling to MLIR? Wait for LLVM 15?

Summary

- Progress
 - Reference counting optimization
- Next plans
 - Standard library improvements
 - Application development?