

GCC and LLVM collaboration

LPC 2014, Düsseldorf, Germany

Renato Golin
LLVM Tech-Lead
Linaro

Purpose

- What is this all about?
 - Sharing ideas, not code (license issues)
 - Standardising public extensions
 - Creating a common user interface

The Good

- We share standards!
- We share projects!
- We share goals!

Current Status

- What do we share?
 - Standards: C/C++/ELF/Dwarf etc.
 - Assembly language: AT&T/Intel/ARM-UAL
 - Many command line options' syntax/semantics
- Common entry/exit points
 - Libraries (glibc, STL, sanitizers)
 - Binutils (assembler, linker [bfd, gold])
- Design
 - LTO, PGO
 - Operating System logic (driver)

Common Projects

- Binutils
 - GAS is required for many targets, important for many others
 - LD is required for all (for now)
- Glibc/newlib
 - LLVM has no alternative (should it?)
- Sanitizers
 - Same source on both, different integration
 - Both compilers are more than just *users*, they're direct contributors

The Bad

- We could aim towards a common user interface!
- We could document better our extensions!
- We could avoid surprising users with unintended side effects!

Driver - Common user interface

- Build systems can be very complex
 - Command line options vary across compilers (-f*)
 - As well as their semantics, even if similar names (-O*)
 - Target description can have subtle differences
 - As well as binary names (arm-linux-gnueabi-gcc)
- Default behaviour should be common
 - Ex. target attributes, not optimisation levels
- Flag semantics should be common
 - Ex. -foptions, not -debug=whatever

Extensions Standards

- GNU and LLVM extensions can break compatibility
 - If the implementation is only in one compiler
 - If the documentation is not complete, nor accurate
- Inline Assembler
 - Register definitions are undocumented and confusing
 - GCC changes behaviour across versions
- Attributes
 - Position and semantics do vary, not always meaningfully
- Language extensions
 - VLAIIS will never get into LLVM (against the standard)
 - Nested functions either (standardisation of lambdas)

The End

Questions!