Foundation REPORT

BY DEB GOODKIN

Development Project Update:

Toolchain Modernization

Removal of GCC 4.2.1 marks significant milestone in FreeBSD's move to an integrated, modern, permissively licensed tool chain.

As announced, on February 29, 2020 we removed GNU Compiler Collection (GCC) version 4.2.1 from the FreeBSD base system. Although GCC has not been used by default for anything in FreeBSD-current / svn head for some time before this commit, this milestone marks the completion of a major journey to modernize the FreeBSD toolchain.

The purpose of this short article is to provide detail about the motivation, scope, and timeline for this initiative. None of this is new, per se. Nevertheless, we felt it was important to pull all the relevant bits that are currently spread across mailing lists and man pages into one canonical piece for easy reference.

Motivation

License needs clearly played a role in motivating the Toolchain Project. Most readers will know that GCC adopted the strong copy-left GPLv3 license in 2007, presenting obvious incompatibilities with the ethos of this community.

A desire to provide a performant, modern toolchain played an important role as well. This is important to support research and development efforts and to give the project more freedom to drive the tool chain's roadmap. Many of the tools in the Toolchain project (see full list in the Scope section below) represent technical improvements. In most cases, they are relatively new projects without much cruft in their code bases, e.g. they only support operating systems, executable formats, and architectures people actually use today.

Take for example the new compiler, Clang/LLVM, which brings advantages like:

- Vastly improved warnings and error messages due to parser design compared to the pre-GPLv3 GCC version 4.2.1 that we were using. Contemporary GCC versions do not have these problems
- Compilation that is fast enough to use for syntax highlighting
- LLVM is always a cross compiler
- Ability to JIT code
- Blocks support

Extensible and modular design allowing developers and researchers the ability to integrate, reuse, or replace individual parts of LLVM

One industry observer goes so far as to credit the recent blossoming of new development languages like Rust, Kotlin, and Swift in part on LLVM. We are gratified that our design decision to get on board early with LLVM has positioned FreeBSD to be one of (if not *the*) best supported open source operating systems when using LLVM.

Development Project Update:

Toolchain Modernization

Scope

In addition to Clang/LLVM, other tools in the Toolchain project are:

- The ELF Tool Chain Project – A project to create BSD-licensed replacements for GNU binutils, and that also adds some other tools
- libc++ – A BSD-licensed C++ standard library
- libcxxrt – A BSD implementation of the Itanium C++ ABI used by FreeBSD. This provides the low-level parts of the C++ implementation: exception handling, RTTI, etc.

This project also adds support for external toolchains, which provides significant benefits to embedded developers who often need to run vendor-provided toolchains. An explicit goal of external toolchain support is to support any modern compiler with a gcc compatible driver. Another goal is that using external toolchains should be easy.

Timeline

These efforts have been underway for some time, and so the February 29 announcement really marks the culmination of what has been years of effort by countless contributors.

Thanks to much hard work, FreeBSD has used Clang by default for x86 and LE arm targets since version 10.0, released in January 2014, and lld by default since 12.0.

Here's the complete timeline:

- May 2010 BSD Toolchain Summit and BSDCan and demonstration of ClangBSD
- Apr 2011, Build Clang by default on x86 and powerpc (Clang available, but FreeBSD still built with GCC)
- January 2012, FreeBSD 9 released with Clang available
- Feb 2012, WITH_CLANG_IS_CC option
- Nov 2012, Clang default for i386 and amd64
- Feb 2013, Build clang for LE arm
- Mar 2013, Clang default for arm
- Sep 2013, Don't build GCC when Clang is default
- Jan 2014, FreeBSD 10 released with Clang and without GCC on x86
- Jan 2015, Use a set of ELF Tool Chain tools by default
- Oct 2016, FreeBSD 11 released with ELF Tool Chain tools
- Nov 2016, Support for building and linking via LLVM IR
- May 2018, Enable lld as the system linker by default on i386
- Dec 2018, FreeBSD 12 released with lld as the linker on x86 and Arm
- Jan 2020, Clang and lld as the default toolchain for RISC-V
- Feb 2020, Retired GCC 4.2.1. All supported architectures either use in-tree Clang, or rely on external toolchain (i.e., a contemporary GCC version from ports)

Closing Thoughts

Clearly, license concerns were a big part of the initial motivation to replace GCC. But the FreeBSD Toolchain project has been and is about much more. Brooks Davis said it well:

"Comprehensive LLVM support in FreeBSD is enormously advantageous in the research world where LLVM is the research compiler of choice. In the DARPA SSITH program, three of five teams are using FreeBSD specifically due to our LLVM support. Further, our collaboration with Arm to bring CHERI to ARMv8-A in the form of the Morello prototype would have been much harder if not impossible without a permissively licensed operating system and toolchain."

Development Project Update:

Toolchain Modernization

Finally, we would like to give a shout out to a few of the contributors without whose leadership and efforts, this major milestone would not have been possible.

FreeBSD Toolchain Hall of Fame

- Brooks Davis
- Roman Divacky
- Dimitry Andric
- Ed Maste
- David Chisnall
- Joseph Koshy
- Kai Wang

Honorable Mention

- Warner Losh

DEB GOODKIN is the Executive Director of the FreeBSD Foundation. She's thrilled to be in her 15th year at the Foundation and is proud of her hardworking and dedicated team. She spent over 20 years in the data storage industry in engineering development, technical sales, and technical marketing. When not working, you'll find her on her road or mountain bike, running, hiking with her dogs, skiing the slopes of Colorado, or reading a good book.