Build Distribution for
Maintaining the Famous GCC 4.7

Oliver Reiche <oliver.reiche@huawei.com>
Intelligent Cloud Technologies Lab, Huawei Munich Research Center
www.huaweicloud.com

FOSDEM 2024

February 4, 2024
Agenda

Famous GCC 4.7

“Build Distribution”

Patching GCC 4.7.4

Bootstrap Process

Wrap-up
Famous GCC 4.7
Famous GCC 4.7

The **Bootstrappable Builds** movement strives for
- building all software from source
- with only a minimal *binary seed*

But how to build a C++ compiler without a C++ compiler?

**Why is GCC 4.7 famous?**

It plays a key role for bootstrappable builds:
- last GCC that can be build with only a C compiler
- stepping stone for C++ and beyond (Rust, Java, etc.)

**Software preservation**

GCC 4.7 is from 2012–2014
- build on modern systems (glibc, musl)
- build with modern compilers
- build with modern C11 standard
- build reproducibly

O. Reiche | FOSDEM’24 | 2024-02-04
“Build Distribution”
“Build Distribution”

Project: Bootstrappable Toolchain

Using the open source build system JustBuild

- bootstraps latest compilers (GCC13/Clang17)
- bootstraps latest build tools (make/cmake/python)
- requires only a reduced binary seed
  - Coreutils
  - POSIX shell
  - C compiler (e.g., TinyCC)
- all toolchains are built from source
- uses existing build descriptions (make/cmake)
- on-demand tool bootstrap (minimal Linux distribution)

⇝ aka the “Build Distribution” for toolchains
Patching GCC 4.7.4
Patching GCC 4.7.4

Maintenance patches and backports

- building with C11 [Mike Frysinger]
- musl support in config.sub [4.8.0]
- general musl support [6.1.0]
- musl linker support [6.1.0]
- libstdc++ support for musl [6.1.0]
- use new type name context_t [8.1.0]
- musl detection in config.guess [9.1.0, 11.1.0, 11.1.0]
Patching GCC 4.7.4

Maintenance patches and backports

- building with C11 [Mike Frysinger]
- musl support in config.sub [4.8.0]
- general musl support [6.1.0]
- musl linker support [6.1.0]
- libstdc++ support for musl [6.1.0]
- use new type name context_t [8.1.0]
- musl detection in config.guess [9.1.0,11.1.0,11.1.0]
Patching Reproducibility of GCC 4.7.4

**JustBuild** builds in isolation (i.e., using temp directory, possibly in the user’s home).

**Problem:** `cc1/cc1plus` contain checksums, computed from:

1. the path of the linker that was used
2. the relevant object files

... both contain the build directory.

**Compute Reproducible Checksums!** independently of the build directory:

- strip linker path
- strip debug symbols from objects

Note that all patches are applied automatically as part of the **bootstrap process**.
Bootstrap Process
Bootstrap Stage-0: GCC-4.7.4

Stage-0

- toolchain stage-0
- bootstrap busybox
  - bootstrap make
  - bootstrap ar
    - + ar
  - + make
  - + grep, find, sed, patch,...
- binutils
  - + as, ld, ranlib, gcc, g++,...
- gcc-4.7.4

Run bootstrap and install with JustBuild:

```
just-mr --main stage-0/gcc install toolchain -o /opt/gcc-4.7.4
```
Wrap-up
Wrap-up

Works on any x86_64 Linux system
Tested on Arch Linux, Debian, Fedora, and Void Linux, but also very different systems:

- NixOS only GCC/Clang + glibc
- prologic/ulinux TinyCC + musl libc
  (≈6 MB on DockerHub)

Use as explicit dependency
Make the toolchain a committed project dependency

- easier project setup
- git bisect support
- “predict” binary hashes!
  (see our demo application)

Install JustBuild and try it yourself:

1. git clone https://github.com/just-buildsystem/bootstrappable-toolchain.git
2. cd bootstrappable-toolchain
3. just-mr --main gcc-13.2.0-musl install -D\"ARCH\":"x86_64\" -o /opt/gcc
4. # grab some coffee and wait...
Thank you.

Special thanks to the community!

- Rich Felker https://github.com/richfelker/musl-cross-make
- https://bootstrappable.org
- https://reproducible-builds.org
- https://bootstrapping.miraheze.org/wiki/C_compilers