

Zero-sysroot hermetic LLVM cross-compilation using Bazel

By Corentin Kerisit & David Zbarsky

Motivation

- 01** cross-compilation is arcane
- 02** setup steps suck
- 03** hermeticity unlocks new capabilities

Status quo



Existing approaches



Existing approaches

Sysroots, docker, prebuilts env



osxcross

MacOS Cross-Toolchain for Linux and *BSD

 C++ ★ 3.3k  350



dockcross

Cross compiling toolchains in Docker images

CMake ★ 3.6k  422



xx

Dockerfile cross-compilation helpers

 Shell ★ 535  39



cross

“Zero setup” cross compilation and “cross testing” of Rust crates

 Rust ★ 8.0k  442

Existing approaches

Source based

buildroot



Buildroot, making embedded Linux easy. Note that this is not the official repository, but only a mirror. The official Git repository is at <https://gitlab.com/buildroot.org/buildroot/>. Do not open issues or file pull requests here.

 Makefile ★ 3.3k 🍷 2.7k

crosstool-ng

A versatile (cross-)toolchain generator.

 Shell ★ 2.4k 🍷 729

llvm-mingw

An LLVM/Clang/LLD based mingw-w64 toolchain

 C ★ 2.7k 🍷 238

Existing approaches

Zig cc



Existing approaches

Nix



pkgsCross.*

The hermetic approach

Requirements

Cross-compiling
everything from source

The hermetic approach

Requirements

- 01** An LLVM prebuilt toolchain
- 02** The LLVM source code
- 03** All other runtimes' source code
- 04**  Bazel

The hermetic approach

Requirements



Bazel

{ Fast, Correct } — Choose two

The hermetic approach

LLVM Driver

```
OVERVIEW: llvm toolchain driver
```

```
USAGE: llvm [subcommand] [options]
```

```
SUBCOMMANDS:
```

```
objcopy  
strip  
nm  
ar  
ranlib  
lib  
dlltool  
lld  
clang  
clang++  
...more...
```

```
Type "llvm <subcommand> --help" to get more help on a specific subcommand
```

```
OPTIONS:
```

```
--help - Display this message
```

The hermetic approach

FOSDEM 2026

Prebuilt

▼ Assets 8

 llvm-toolchain-minimal-21.1.8-darwin-arm64.tar.zst	sha256:680ac53e1b0e47...		25.6 MB	last week
 llvm-toolchain-minimal-21.1.8-linux-amd64-musl.tar.zst	sha256:69ace4e9c71ce7...		29.6 MB	last week
 llvm-toolchain-minimal-21.1.8-linux-arm64-musl.tar.zst	sha256:8cdb101e505853...		28 MB	last week
 llvm-toolchain-minimal-21.1.8-windows-amd64.tar.zst	sha256:bdbe6a839f05af...		29.3 MB	last week
 llvm-toolchain-minimal-21.1.8-windows-arm64.tar.zst	sha256:f24b0527c1de96...		26.8 MB	last week
 SHA256.txt	sha256:e90e6f19fd73f7...		595 Bytes	last week
 Source code (zip)				last week
 Source code (tar.gz)				last week

The hermetic approach

Runtimes



▼ Assets

 compiler-rt-21.1.8.src.tar.xz	sha256:dd54ae21aee1780fac59445b51...		2.52 MB	Dec 16, 2025
 libcxx-21.1.8.src.tar.xz	sha256:6422a58a5c29b7f4fda224cfdc...		4.35 MB	Dec 16, 2025
 libcxxabi-21.1.8.src.tar.xz	sha256:709c9a63bde1e36a80d8675bec...		571 KB	Dec 16, 2025
 libunwind-21.1.8.src.tar.xz	sha256:03e8adc6c3bdde657dcaedc948...		119 KB	Dec 16, 2025

The hermetic approach

Upstreaming changes

🔗 3 Open ✓ 5 Closed		Author ▾	Label ▾	Projects ▾	Milestones ▾	Reviews ▾	Assignee ▾	Sort ▾
🔗	Support multicall binary for llvm-link	•	bazel					3
#178425 opened 6 hours ago by dzbarsky								
🔗	[llvm-cov] Support multicall binary	✓	bazel					
#178294 opened yesterday by dzbarsky								
🔗	Migrate llvm-profdata to Opt flag parsing	✓	bazel	PGO				11
#177868 opened 3 days ago by dzbarsky								
🔗	[bazel] dsymutil should link against CF framework on darwin	✓	bazel					
#177394 by dzbarsky was merged last week								
🔗	Fix typo in clang/www/builtins.py	✓	clang					2
#174304 by dzbarsky was merged last week								
🔗	[bazel] Improve building on/for Windows	✓	bazel	llvm:support				18
#171761 by dzbarsky was merged last week								
🔗	[bazel] Rewrite overlay handling to starlark	✓	bazel					5
#170000 by dzbarsky was merged on Dec 1, 2025								
🔗	[bazel] Add filegroup for builtin_headers	✓	bazel					7
#67757 by dzbarsky was merged 5 days ago								

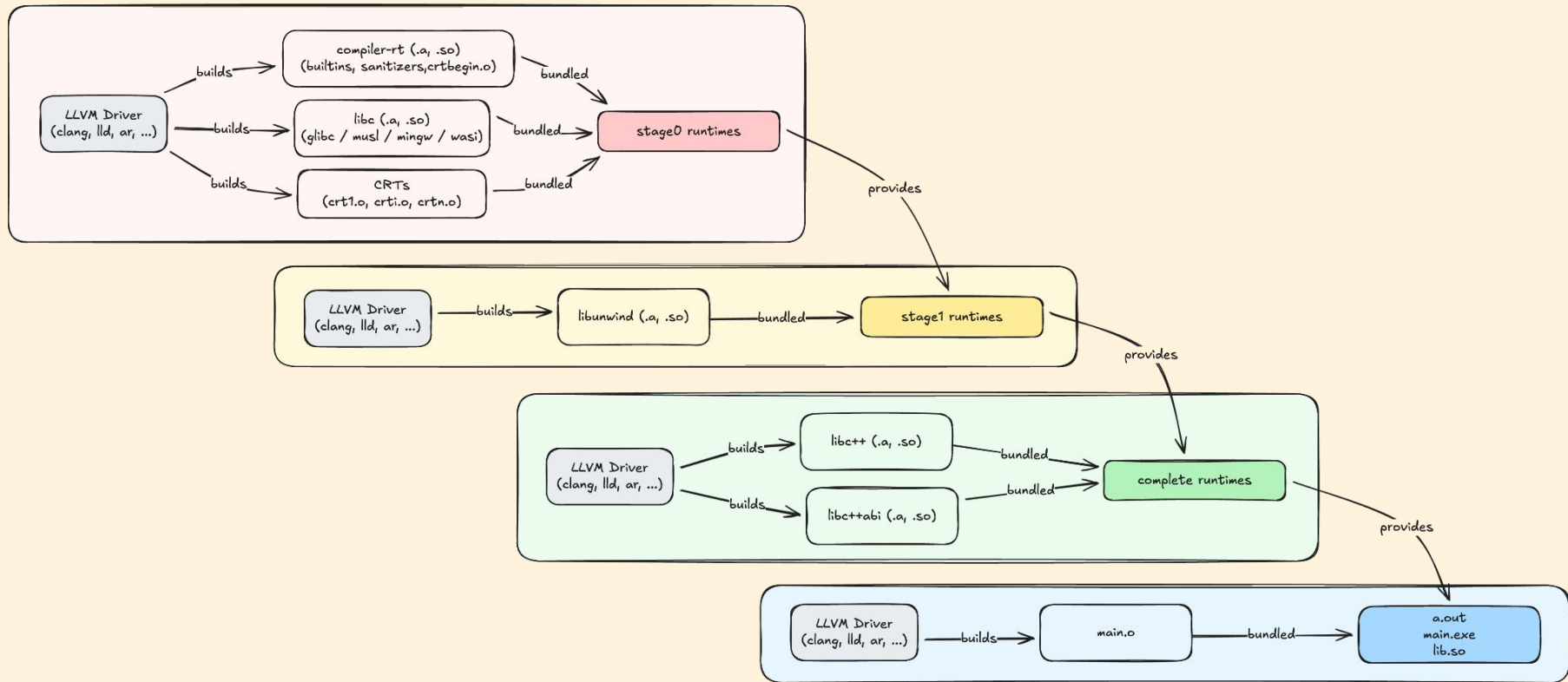
The hermetic approach

Wiring

Prebuilt toolchain
+
Runtime sources
+
BUILD.bazel files
+
Bazel `toolchain()` definitions

The hermetic approach

Bazel transitioned toolchains



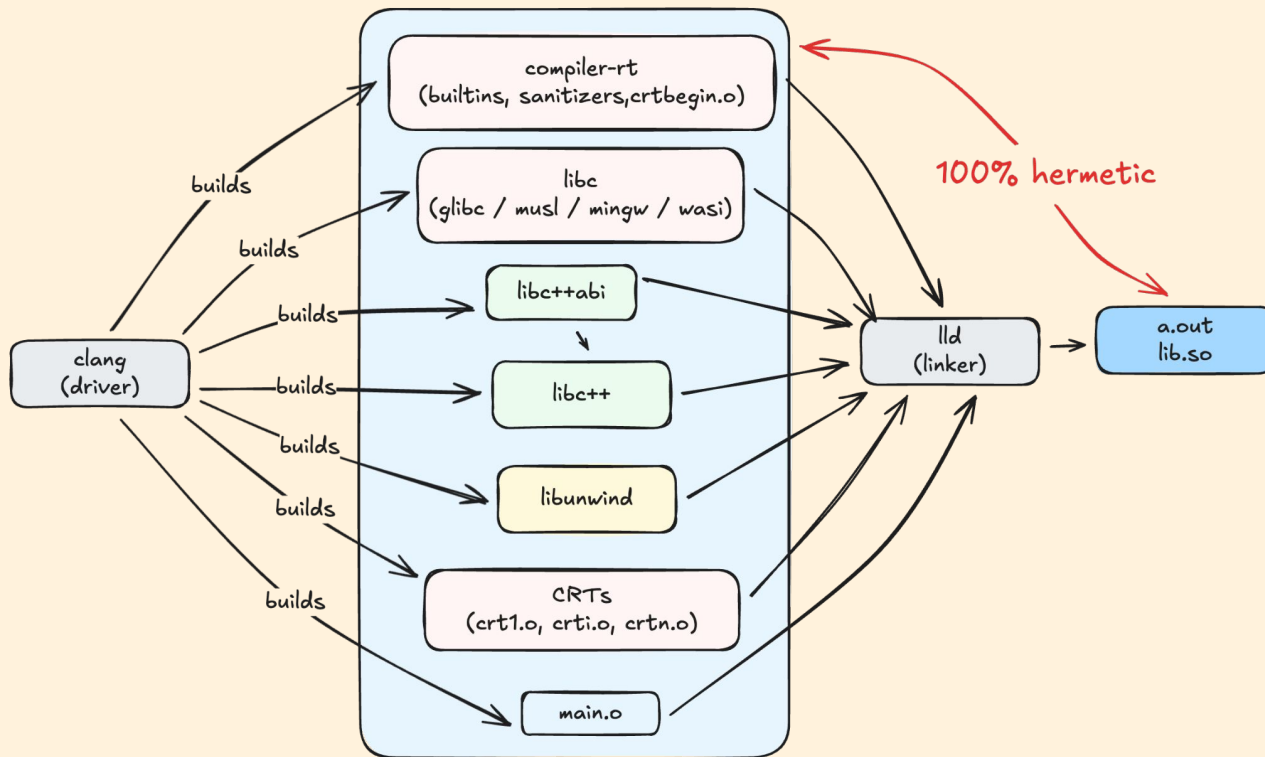
The hermetic approach

Hermetic link graph

-sysroot=/dev/null

The hermetic approach

Hermetic link graph



The hermetic approach

Key consequence

any host  any target

The hermetic approach

Key consequence

Supported platforms

✓ Currently supports cross-compilation between all combinations of the following platforms:

To ↓ / From →	macOS aarch64	Linux aarch64	Linux x86_64	Windows aarch64	Windows x86_64
aarch64-apple-darwin	✓	✓	✓	✓	✓
x86_64-apple-darwin	✓	✓	✓	✓	✓
aarch64-linux-gnu ¹	✓	✓	✓	✓	✓
x86_64-linux-gnu ¹	✓	✓	✓	✓	✓
aarch64-linux-musl	✓	✓	✓	✓	✓
x86_64-linux-musl	✓	✓	✓	✓	✓
aarch64-windows-gnu ²	✓	✓	✓	✓	✓
x86_64-windows-gnu ²	✓	✓	✓	✓	✓
wasm32-unknown-unknown	✓	✓	✓	✓	✓
wasm64-unknown-unknown	✓	✓	✓	✓	✓

+ **many more** in progress

The Hard Parts

FOSDEM 2026

Bazelifying everything

But we need to write
the **BUILD.bazel** files

> Either adapt existing runtimes or provide new ones (wasi, ape, etc.)

The Hard Parts

Bazelifying everything

```

  ▾ glibc
    > csu
    > nptl
    > stdlib
    ≡ 0001-Add-empty-config.h.patch
    ≡ 0002-Add-abi-tag.h.patch
    ≡ 0003-Adding-libc-modules.h.patch
    ♥ BUILD.bazel
    💡 BUILD.tpl
    ≡ COPYING.LIB
    ♥ helpers.bzl
    🗝 LICENSE
    ≡ NOTICE
    ⓘ README.md

```

```

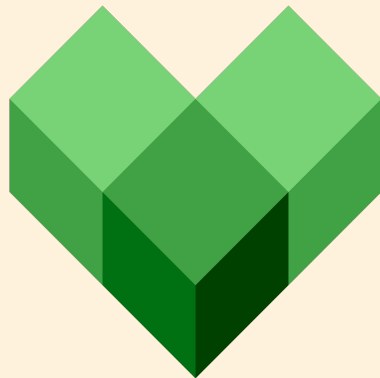
  ▾ llvm-project
    > compiler-rt
    > libc
    > libcxx
    > libcxxabi
    > libunwind
    ♥ BUILD.bazel

```

```

  ▾ musl
    ≡ 1.2.5-CVE-2025-26519-1.patch
    ≡ 1.2.5-CVE-2025-26519-2.patch
    ♥ BUILD.bazel
    💡 BUILD.tpl
    ♥ BUILD.bazel

```



The Hard Parts

FOSDEM 2026

The glibc case



The glibc case

The issue

- 01** Complex makefile and custom scripts
- 02** Requires a gcc cross compiler
- 03** Clang support since 2.43 (Jan 2026)
- 04** Support needed for **many** versions

The glibc case

The solution

Generate a **stub library** and
link against it

aka interface library (.ifso)

The glibc case

Thank you Zig

```

1 .text
2 .balign 8
3 .globl __strtoull_l
4 .type __strtoull_l, %function;
5 .symver __strtoull_l, __strtoull_l@@GLIBC_2.2.5
6 __strtoull_l: .quad 0
7 .balign 8
8 .globl __wcsftime_l
9 .type __wcsftime_l, %function;
10 .symver __wcsftime_l, __wcsftime_l@@GLIBC_2.3
11 __wcsftime_l: .quad 0
12 .balign 8
13 .globl __xmknodat
14 .type __xmknodat, %function;
15 .symver __xmknodat, __xmknodat@@GLIBC_2.4
16 __xmknodat: .quad 0
  
```

```

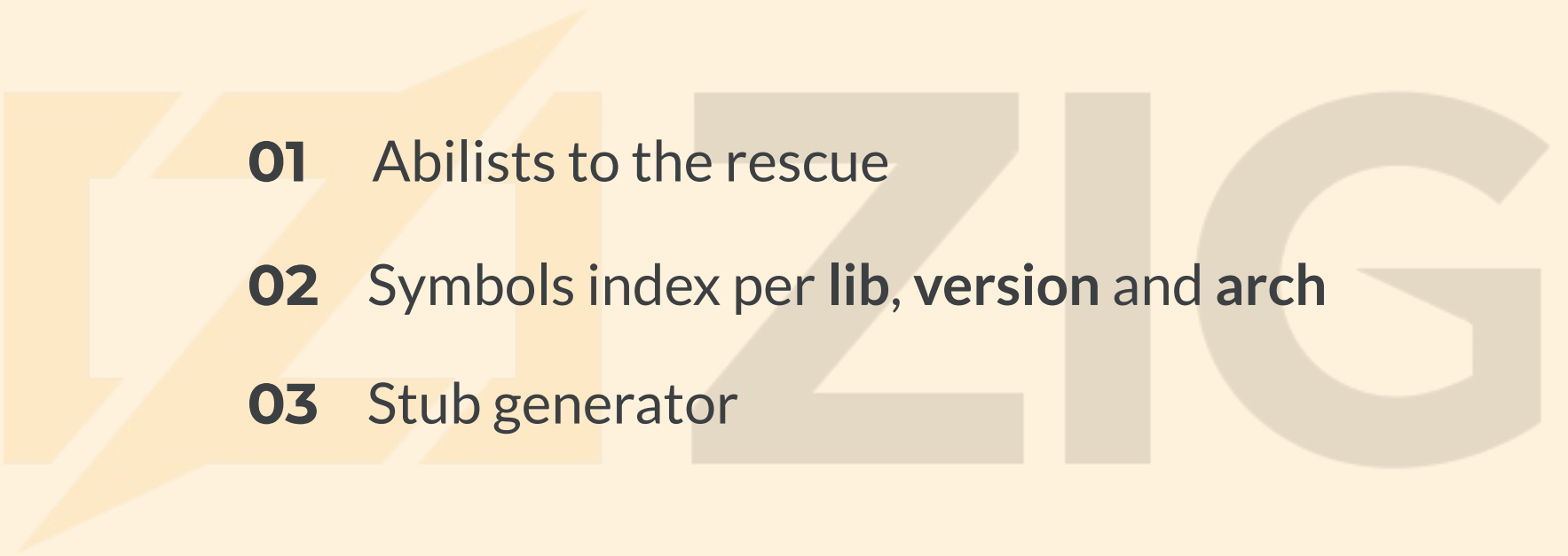
$ clang -nostdlib -shared c.s -o libc.so
$ nm -D libc.so
0000000000001b9f0 T __strtoull_l@@GLIBC_2.2.5
0000000000001c780 T wcsftime_l@@GLIBC_2.3
0000000000001ba00 T __xmknodat@@GLIBC_2.4
  
```

```

$ clang -L. -lc main.c
$ readelf -d a.out
Dynamic section at offset 0xaeac0 contains 28 entries:
  Tag                               Type                               Name/Value
  0x0000000000000001 (NEEDED)         Shared library: [libc.so.6]
  0x000000000000001e (FLAGS)           BIND_NOW
  0x0000000006ffffff (FLAGS_1)        NOW PIE
  
```

The glibc case

Thank you Zig

- 
- 01** Abilists to the rescue
 - 02** Symbols index per **lib**, **version** and **arch**
 - 03** Stub generator



\$ demo

- 01** Built from LLVM sources
- 02** Support fully static or dynamic linking
- 03** Composable with the rest of the build.
Build a UBSAN'ed LLVM?

Windows

FOSDEM 2026



- 01** Build mingw CRTs
- 02** Build mingwex and other extra C libs
- 03** Build import libs from DLL .defs
- 04** Assemble umbrella import libs from MRI recipes



\$ demo

The perks

Bazel enabled

- 01** Zero-setup remote builds
- 02** Zero-setup C cross compilation and cross link to other languages (Rust, Go, ...)

Thank you BuildBuddy





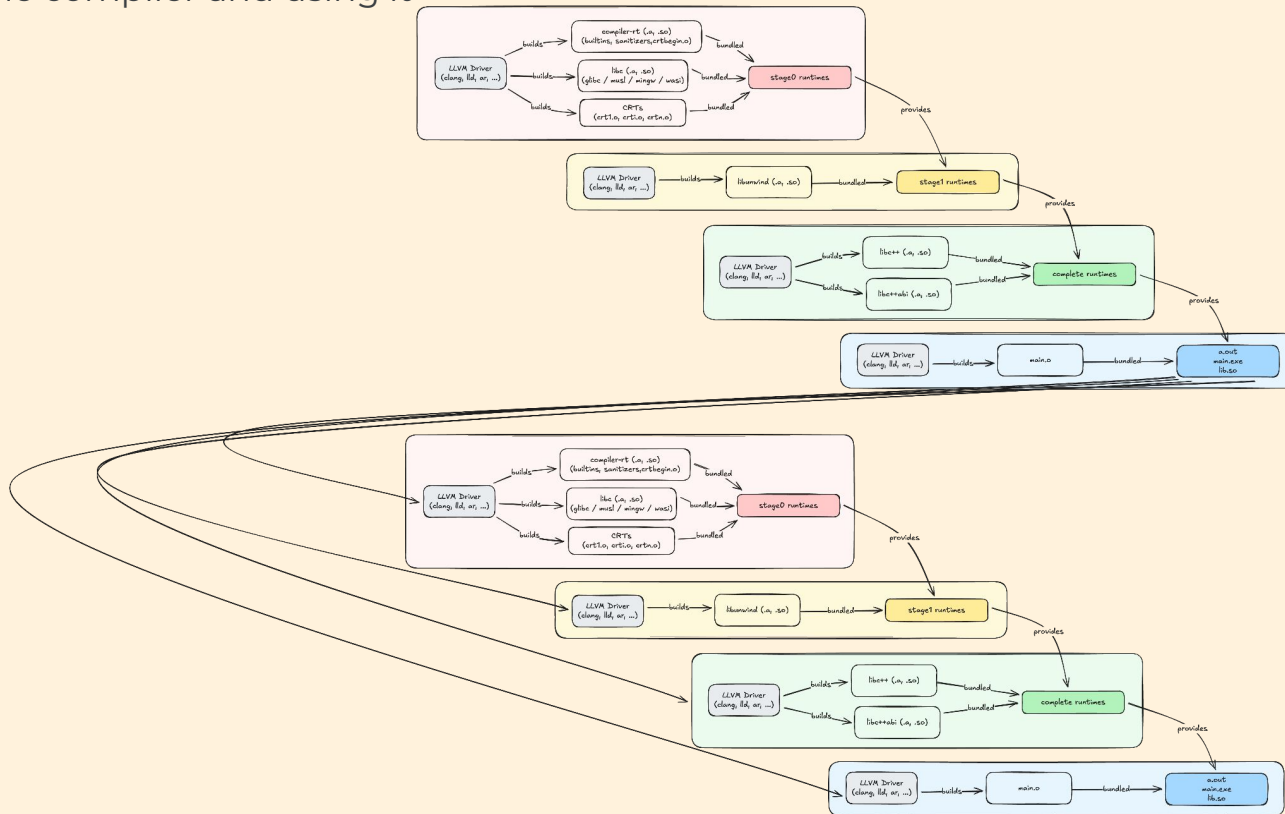
\$ demo

Bootstrapping



Bootstrapping

Compiling the compiler and using it



Bootstrapping

Compiling the compiler and using it

In a single build.

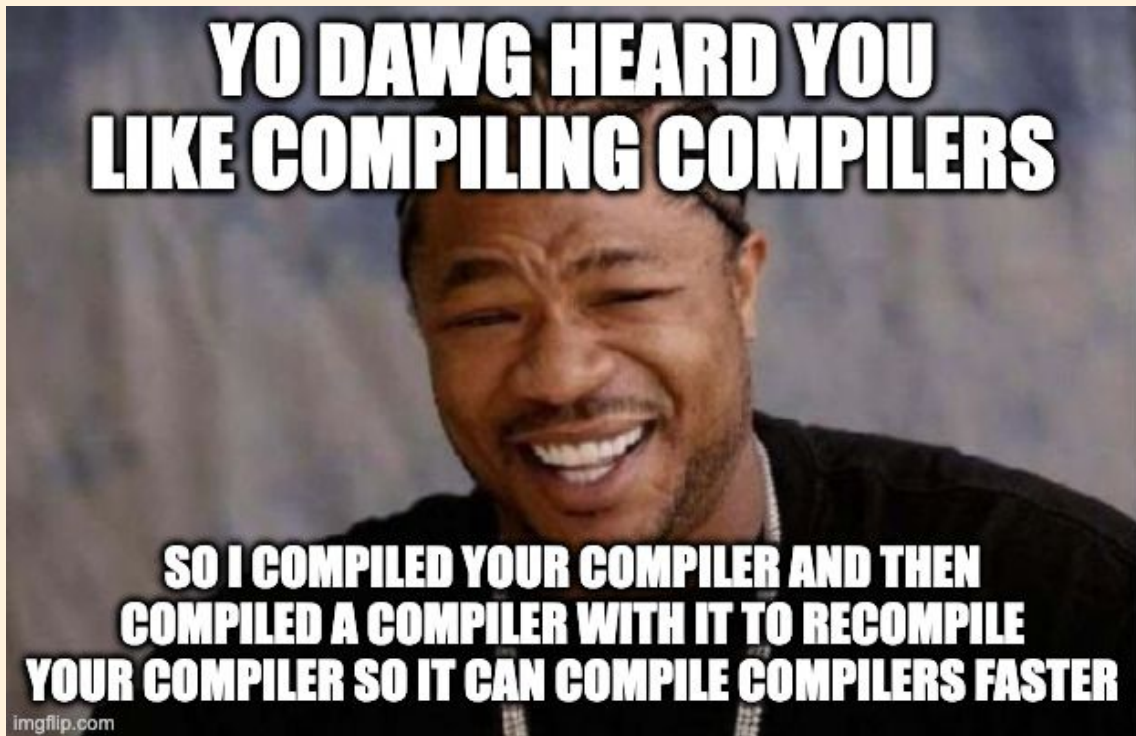


\$ demo

What's next?

FOSDEM 2026

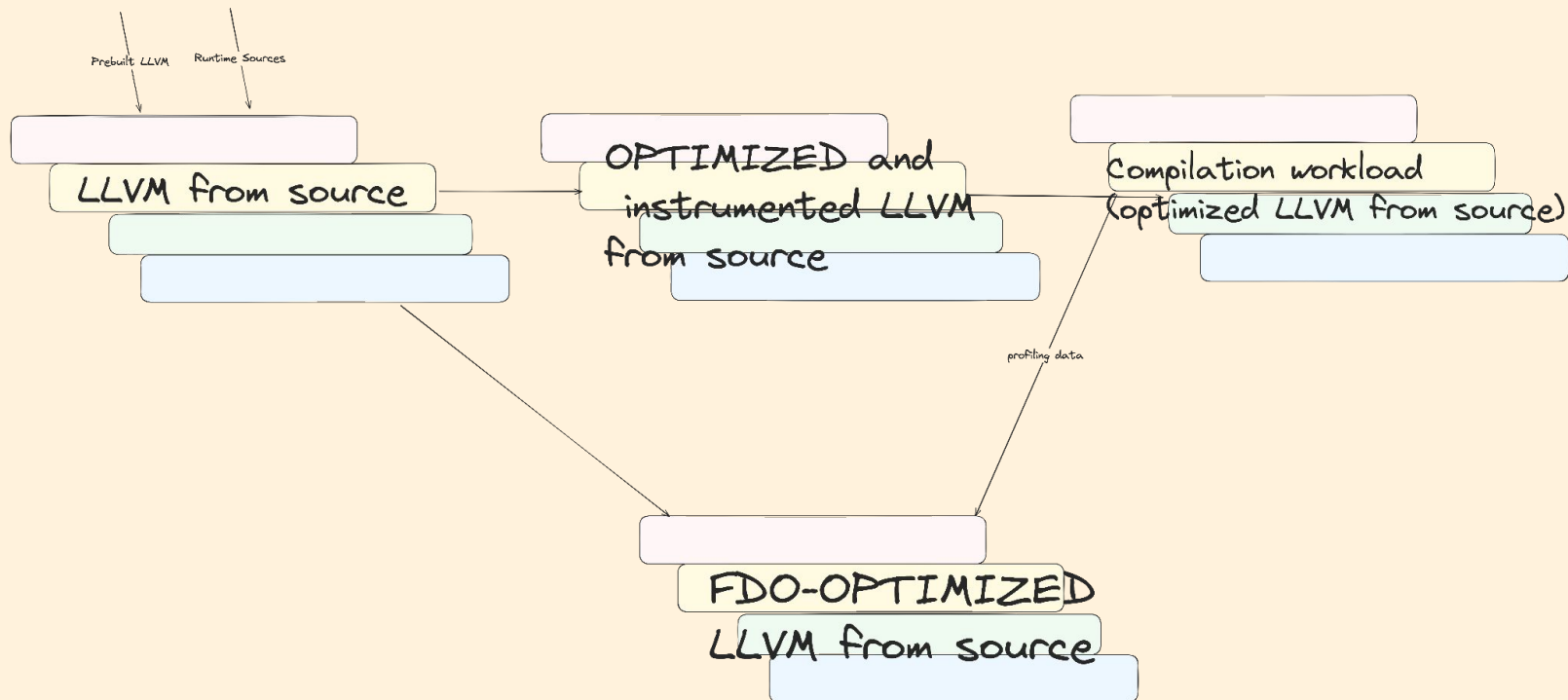
FDO



What's next?

FOSDEM 2026

FDO



What's next?

Exhaustive targets and runtimes support

- armbe
- thumb
- spirv
- risc

- wasi
- cygwin
- ape
- ???

Your favorite **target**?
Send us a PR!

Your favorite **runtime**?
Send us a PR!

What's next?

Cosmopolitan libc/APE – build once, run everywhere



cosmopolitan

build-once run-anywhere c library



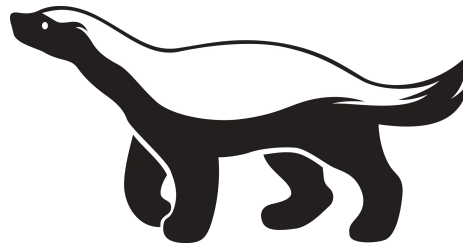
C



20.5k



733



What's next?

Make this useful outside of Bazel



```
bazel build //:crossenv --platforms=//:linux_arm64_musl  
make CC="./bazel-bin/crossenv/clang-wrapper"
```

What's next?

Make this useful outside of Bazel



```
# bazel-crossenv wraps bazel invocation  
make CC="./bazel-crossenv --target armeb-linux-musleabi"
```

Thank you

Feedback and contributions welcomed



toolchains_llvm_bootstrapped

A zero sysroot, fully hermetic C/C++ cross compilation toolchain for Bazel based on LLVM.



Starlark



80



10

