Filippo Cremonese

fcremo@rev.ng

16/09/2021

- Filippo Cremonese (@fcremo)
- I work at rev.ng
- Expertise in compilers, emulation, binary analysis and translation
- We're building our binary analysis framework and decompiler
  - Architecture-agnostic
  - Based on QEMU + LLVM
  - Apply for the beta at `https://rev.ng`

- LLVM at rev.ng
- Orchestra
- Cross compiling LLVM for Windows
- Our LLVM patches

We are both LLVM distributors and developers.

We ship multiple LLVM components:

- clang-release: stock compiler we use as toolchain
- llvm: our LLVM fork
- llvm-documentation: doxygen generated Zeal/Dash docs

- self-contained portable install root
- binaries work across distros
- easy setup of cross compilation toolchains
- CI friendly

- easy, quick setup
- not tied to a specific distro
- uniform environment
  - modern toolchains
  - same tooling for all developers
- ability to reproduce bugs consistently
- build from source only what you need

We wanted to solve everyone's problems

We built Orchestra for that!

- Fetches, configures and builds all the components we need (~150)
- Movable install root thanks to `RPATH` magic
- Portable build artifacts
  - by linking against a legacy `glibc`
- CI-built binary archives
- multiple build flavors
  - e.g. debug+O0, debug+O2, release, +ASAN
- YAML configuration syntax, ytt templating

Steps:

- (prerequisite) bootstrap Mingw-w64 cross toolchain
  - GCC + libstdc++
  - configured with pthread support
- use x-toolchain to build LLVM
- use x-toolchain to build `libc++` and `libc++abi`

The process was almost straightforward.

Some CMake patching was still required:

- solve missing llvm Support library to link targets
- add custom toolchain file to allow building host tools
  - `llvm-tblgen` and `clang-tblgen`

Using `lld` was crucial.

- `ld.bfd` spins for a couple minutes, prints an error, then gets stuck for another couple minutes (!) before exiting
- `clangAST.dll` fails linking because it exports $> 2^{16}$ symbols

Iterative debugging led us to the following ~~incantations~~ flags

- `CXXFLAGS=-U_LIBCPP_BUILDING_LIBRARY -D_LIBCPP_BUILDING_LIBRARY=`
  `-U_LIBCXXABI_DISABLE_VISIBILITY_ANNOTATIONS`
- `LIBCXXABI_LIBCXX_INCLUDES=.../libcxx/include`
  - why is this needed in monorepo builds?
- `LIBCXXABI_ENABLE_NEW_DELETE_DEFINITIONS=ON`
  - has to be ON in libc++ XOR libc++abi (?)
- `LIBCXXABI_HAS_CXA_THREAD_ATEXIT_IMPL=ON`
- `LIBCXXABI_HAS_WIN32_THREAD_API=ON`
  - can't this be autodetected?

Iterative debugging led us to the following incantations flags

- `CXXFLAGS=-D_LIBCXXABI_BUILDING_LIBRARY -Wno-unused-command-line-argument -Wl,-start-group`
- `LIBCXX_ENABLE_FILESYSTEM=OFF` (unsupported on Windows)
- `-LIBCXX_CXX_ABI_LIBRARY_PATH="$BUILD_DIR/libcxxabi/lib"` and `LIBCXX_ENABLE_STATIC_ABI_LIBRARY=TRUE` and `LIBCXX_CXX_ABI=libcxxabi`
  - would be nice to have a configuration to build them together automatically.
- `LIBCXX_HAS_WIN32_THREAD_API=ON` and `-DLIBCXX_CXX_ABI_INCLUDE_PATHS=...` as before

See branches `feature/windows` on `revng/llvm-project` and `revng/orchestra` on GitHub

- We maintain an LLVM fork tailored to our needs.
- We try not to diverge heavily from upstream.
- We rebase on each release (but not on trunk).
- We do backport some improvements on a per-need basis.

Compiling LLVM and clang with C++20 required some fixes

- Compatibility between StringRef and UTF8 string literals
- Ambiguous comparison operators (mainly ==, !=)

Misc patches

- Zero-copy filters for GraphTraits
- SROA: more aggressive load speculation across multiple `PHI`/`SelectInst`
- LazyValueInfo: better handle masks and `builtin_ctlz`

We're interested in upstreaming some of them if there's interest.

- NDEBUG inconsistencies
  - LLVM could be built with NDEBUG defined
  - while other stuff using its headers could be built without
  - Some structs/classes in public headers change layout
  - Solution: replace NDEBUG defines in headers with a fixed value
  - a similar issue emerges with `#if __address_sanitizer__`
- A test fails if you are root because it bypasses permission checks
  - Known issue? https://bugs.gentoo.org/775050

- `llvm-config --version` shows the git remote (possibly with password if HTTP)
  - tokens might end up in CI logs
  - Solution: we patched CMake to ignore if the source is a git repo
- Not possible to load multiple LLVM versions in the same process
  - CLI argument parsing machinery has global mutable state
  - mesa requires LLVM, rev.ng too
  - Solution: link mesa against our LLVM
  - Future solution: separate the decompiler engine from the UI process

Thanks!

Questions are welcome on

https://github.com/ClangBuiltLinux/llvm-distributors-conf-2021/issues/6

Get in touch at:

{info,fcremo}@rev.ng