Google

Clang built kernels

Greg Hackmann
Nick Desaulniers

Why build the kernel with Clang?

- Consistent toolchain for kernel and userspace
- Better static analysis + dynamic analysis (sanitizers)
- Additional compiler warning flag coverage
- More tools planned in the future (control-flow analysis, LTO, PGO)
- Shake out undefined behaviors
- Improve both code bases

State of upstream

https://lkml.org/lkml/2017/8/22/912

- X86_64 and ARM64 buildable and runnable
- One out-of-tree patch for each architecture

- Revert "x86/uaccess: Add stack frame output operand in get_user() inline asm"
  - Discussion at https://lkml.org/lkml/2017/7/28/775

- arm64: use -mno-implicit-float instead of -mgeneral-regs-only
  - LLVM bug: https://bugs.llvm.org/show_bug.cgi?id=30792

State of upstream

- One Android-specific Kbuild change

- ANDROID: Kbuild, LLVMLinux: allow overriding clang target triple
  - Can tell clang to build for [arch]-linux-gnu triple, even with AOSP binutils
  - Not relevant unless building with AOSP prebuilt toolchain

Reference patchstack

- Patchstacks for 4.4 and 4.9 LTS available from Chromium sandbox
- See LKML announcement for details

- Also experimentally available from AOSP (pre-applied on top of normal Android patchstack)
- https://android.googlesource.com/kernel/common
- Branch names: experimental/android-{4.4,4.9}-llvm

Try it out on your kernel!

```
git merge aosp/experimental/android-4.9-llvm


export ARCH=[arch]
export CROSS_COMPILE=[arch]-linux-android-
export CLANG_TRIPLE=[arch]-linux-gnu-
make CC=clang [HOSTCC=clang]
```

Compiler bugs found

- llc lowers intrinsics incorrectly with the presence of regparm attribute
  https://bugs.llvm.org/show_bug.cgi?id=3997
- AArch64 error: invalid operand in inline asm: 'prfm pstl1keep, ${0:a}
  https://bugs.llvm.org/show_bug.cgi?id=19962
- Invalid LDR instruction with XZR
  https://bugs.llvm.org/show_bug.cgi?id=33134
- [AArch64] -mgeneral-regs-only inconsistent with gcc
  https://bugs.llvm.org/show_bug.cgi?id=30792
  (still open)

Common compilation issues

Clang/LLVM does not support:

1. VLAs
2. Inline functions (GNU C extension)
3. Assembling the kernel (patch in upstream disables LLVM’s assembler, deferring to binutils’)

What’s next?

- Automated testing for both LLVM and kernel
- Remaining architectures supported by Android