LLVM/Clang and ARM 32-bit

Stefan Agner

About me

- Open Source Software enthusiast
- Work for Toradex AG - ARM based embedded modules
  - Board Support Package work: U-Boot/Linux/OpenEmbedded customization
  - Upstreaming drivers/driver enhancements/device-trees/fixes
- LLVM/Clang hacking as a Hobby
  - https://blog.printk.io/

Why?

- Prints *really* useful warnings (which often give hints to actual bugs):

```
drivers/gpu/drm/tegra/dc.c:408:18: warning: variable
    'tegra124_primary_formats' is not needed and will not be emitted
      [-Wunneeded-internal-declaration]
static const u32 tegra124_primary_formats[] = {
                 ^
drivers/gpu/drm/tegra/dc.c:835:18: warning: variable
    'tegra124_overlay_formats' is not needed and will not be emitted
      [-Wunneeded-internal-declaration]
static const u32 tegra124_overlay_formats[] = {
                 ^
2 warnings generated.
```

My history LLVM/Clang Linux

- 2012-2014: LLVMLinux (Linux Foundation project)
  - Behan Webster, Jan-Simon Möller, Mark Charlebois
- 2015-: Compiled kernels/rebased the patches
  - https://blog.printk.io/2015/03/cross-compile-linux-for-arm-using-llvm-clang-on-arch-linux/
- 2017: Google Android team pushing upstream 2017
  - Matthias Kaehlcke, Nick Desaulniers, Miguel Ojeda, Sedat Dilek
  - v4.4/v4.9 https://lkml.org/lkml/2017/8/22/912
  - v4.14/state https://lkml.org/lkml/2017/11/22/943
- 2017: Pushed fixes for build errors/warnings
  - E.g. build error for ARM in MPI
  - Lots of warnings: e.g. implicit conversion from enumeration
- 2018: Initial complete defconfig support for ARM 32-bit
  - https://lkml.org/lkml/2018/3/20/837

Cross compiling with LLVM/Clang

- LLVM/Clang can cross compile by default (multiple backends)

  ```
  llc --version
  ```

- Currently, only compiling is done by LLVM/Clang
  - GNU cross-compiler toolchain with assembler/linker required

- Environment setup as usual (cross compiler in PATH!)

  ```
  export ARCH=arm
  export CROSS_COMPILE=arm-linux-gnueabihf-
  ```

Cross compiling with LLVM/Clang

- Compile using

```
make CC=clang HOSTCC=clang multi_v7_defconfig
make CC=clang HOSTCC=clang nconfig
make CC=clang HOSTCC=clang -j24
```

- v4.18 added compiler flag checks in Kbuild
  => CC and HOSTCC are required at config time!

- If using distro LLVM/Clang, add symlink in cross compiler bin dir [<4.20]

```
cd ~/gcc-linaro-7.3.1-2018.05-x86_64_arm-linux-gnueabihf/bin
ln -s /usr/bin/clang clang
```

Upstream ARM 32-bit support

- Clang as C-Compiler (CC/HOSTCC)

- v4.18 multi_v7_defconfig-CONFIG_EFI
  - Patch queued for v4.20: https://lkml.org/lkml/2018/8/9/658

- v4.19 multi_v7_defconfig compiles
  - Clang 7.0+ for ARMv7
  - Clang 8.0+ for ARMv6

- v5.5 multi_v7_defconfig and multi_v7_defconfig still compiles fine

- CONFIG_FUNCTION_TRACER needs Clang 10.0+

Known issues/Future work

- Warning when building Thumb2 kernel:

  ```
  warning: inline asm clobber list contains reserved registers: R7
  [-Winline-asm]
  ```

  - https://github.com/ClangBuiltLinux/linux/issues/701

- Assembler Warning in Clang 10+

  ```
      Warning: index register overlaps transfer register
  ```

  - Clang issue: https://reviews.llvm.org/D70072
  - https://github.com/ClangBuiltLinux/linux/issues/838

Known issues/Future work AS=clang

- LLVM/Clang assembler refuses to assemble Co-processor instructions with argument p10

  ```
  arch/arm/vfp/vfpmodule.c:528:3: error: invalid operand for instruction
  […]
          asm("mcr p10, 7, %0, " vfpreg(_vfp_) ", cr0, 0 \
              ^
  ```

  - Some kernel code is multi-arch (pre-v7). Trying to convert this into a warning in Clang
  - https://github.com/ClangBuiltLinux/linux/issues/306
- LLVM/Clang assembler does not support .if conditions in assembly

  ```
  arch/arm/mm/proc-v7.S:169:107: error: expected absolute expression
  […] .if . - 9997b == 2 ; nop ; .endif ; .if . - 9997b != 4 […]
      ^
  ```

  - Should be fixed in LLVM/Clang 11? (by Jian Cai)
  - https://github.com/ClangBuiltLinux/linux/issues/742

Known issues/Future work AS=clang

- Relocation issue (also in many machine specific assembly code: Tegra, OMAP2, Zync...)

```
arch/arm/kernel/entry-common.S:-133:2: error: unsupported relocation on
symbol
adr tbl, sys_call_table @ load syscall table pointer
```

- Architecture selection?

```
arch/arm/kernel/entry-armv.S:527:4: error: invalid instruction, any one
of the following would fix this:
2: ldrht r5, [r4]
   ^
```

- OMAP (adrl)

```
arch/arm/mach-omap2/sleep34xx.S:75:2: error: invalid instruction, did
you mean: adr?
```

Known issues/Future work AS=clang

- Support Thumb2 mode with Clang as assembler
  - Use Clang's `-mimplicit-it=always` (in `CFLAGS_ISA`)
- More (rather trivial) assembler errors like:

```
<instantiation>:11:8: error: invalid instruction
 9998: pldw.w [r1]
       ^
```

- However, doesn't boot currently :-(
  - Currently crashes in early initialization code (Qemu and real-world)

Thank you!

Stefan Agner
stefan@agner.ch

Debugging Techniques

- Compile single threaded & verbose

  ```
  make CC=clang HOSTCC=clang -j1 V=1
  ```

- Invoke the compiler manually verbose/or through CC

  ```
  clang … -v
  make CC=”clang -v” HOSTCC=clang -j1 V=1
  ```

- To retain intermediate files use `-save-temps`
- Debug compiler flag detection
  - Edit scripts/Kbuild.include to echo command before execute (cc-option)
  - Better alternative?