Using llvm-libc in LLVM Embedded Toolchain for Arm

Peter Smith

2024/10/20

Building LLVM-libc for an embedded system

- Documentation available in full_cross_build.rst
  - https://github.com/llvm/llvm-project/blob/main/libc/docs/full_cross_build.rst#bootstrap-cross-build
- Part of the runtimes under “libc”.
- Use the full build mode.
- Provide a baremetal config
  - LLVM-libc has configs for Arm and RISCV.
  - Arm config can be used for AArch64.
- Configs define the headers to include and entrypoints (functions) and alters build time configuration of functions:
  - errno
  - printf
  - qsort
- Can be built with one directory per target.

What is available today?

Headers

- assert.h
- ctype.h // ASCII only
- errno.h
- fenv.h
- inttypes.h
- math.h // Mostly complete
- setjmp.h
- stdfix.h
- stdint.h
- stdio.h // No FILE*, no floating point printf
- stdlib.h
- string.h
- time.h
- uchar.h

Overlay package for LLVM embedded toolchain for Arm

- LLVM Embedded Toolchain for Arm
  - Scripts to build a LLVM based toolchain like the GNU Arm Embedded Toolchain.
  - Uses picolibc as the C-library.
- LLVM-libc support can be built as an overlay package
  - Includes all configurations supported by the toolchain.
  - Adds a llvmlibc directory to lib/clang-runtimes to act as sysroot.
  - Adds a config file to find the sysroot.
  - Includes startup code and compiler-rt builtins.
- LLVM-19 binary release has a prebuilt package.
  - Building from source recommended to pick up most recent changes.
- C-only at the moment
  - Work ongoing to build libc++ against llvm-libc.

Startup code

Startup code,

- Default entry is `_start`
- Minimum required is to set up a stack pointer.
- May need a vector table.
- May need to initialize RW and ZI data.
- Any requirements from stdio
  - Arm SemiHosting requires streams to be opened.
- If exit is possible then call main via `_Exit(main())`
  - Implement `__llvm_libc_exit`

```
extern long __stack[];
__attribute__((naked)) void _start(void) {
  __asm__("mov sp, %0" : : "r"(__stack));
  __asm__("b c_startup");
}

__attribute__((used))
static void c_startup(void) {
  _platform_init();
  _Exit(main(0, NULL));
}
```

Support code and linker defined symbols

- To use stdio
  - Implement `__llvm_libc_stdio_cookie` struct.
  - Define `__llvm_libc_stdin_cookie`, `llvm_libc_stdout_cookie`, `llvm_stderr_cookie.`
  - Define `__llvm_libc_stdio_read(void *cookie, char *buf, size_t);`
  - Define `__llvm_libc_stdio_write(void *cookie, const char *buf, size_t size);`
- To use malloc
  - Heap delimited by symbols `[_end, __llvm_libc_heap_limit)`
- To use math.h
  - Define `int *__llvm_libc_errno()` to provide address of your `errno` .

Using llvm-libc

```
clang \
    --config=llvmlibc.cfg \
    --target=armv6m-none-eabi \
    -march=armv6m \
    -mfpu=none \
    -nostartfiles \
    -lsemihost \
    -T microbit-llvmlibc.ld \
    -o hello.elf \
    crt0llvmlibc.c hello.c vector.c
```

```
llvm-objcopy \
    -O ihex  hello.elf hello.hex

qemu-system-arm -M microbit \
    -semihosting \
    -nographic \
    -device loader,file=hello.hex
hello world!
```

Pigweed SDK

- Developer preview of Pigweed SDK available that uses llvm-libc
  - https://opensource.googleblog.com/2024/08/introducing-pigweed-sdk.html
  - Supports the Raspberry Pi Pico RP2350 and RP2040.
  - Toolchain uses llvm-libc for armv6-m, armv8-m and riscv32.
- Development is via Bazel
  - Toolchain configured automatically when running the sense tutorial.
- C-library functionality roughly the same as in the LLVM Embedded Toolchain for Arm
  - Libc++ is available.

References

- LLVM libc documentation
  - https://github.com/llvm/llvm-project/tree/main/libc/docs
- LLVM Embedded Toolchain for Arm
  - https://github.com/ARM-software/LLVM-embedded-toolchain-for-Arm
- LLVM-libc overlay package instructions
  - https://github.com/ARM-software/LLVM-embedded-toolchain-for-Arm/blob/main/docs/llvmlibc.md
- LLVM-libc CMake invocations
  - https://github.com/ARM-software/LLVM-embedded-toolchain-for-Arm/blob/main/CMakeLists.txt
  - Search for function add_llvmlibc
- Semihosting startup code
  - https://github.com/ARM-software/LLVM-embedded-toolchain-for-Arm/tree/main/llvmlibc-support
- Pigweed Sense tutorial
  - https://pigweed.dev/docs/showcases/sense/tutorial/
- Pigweed SDK
  - https://opensource.googleblog.com/2024/08/introducing-pigweed-sdk.html
  - https://www.raspberrypi.com/news/google-pigweed-comes-to-our-new-rp2350/

Thank You
Danke
Gracias
Grazie
谢谢
ありがとう
Asante
Merci
감사합니다
धन्यवाद
Kiitos
شكرًا
ধন্যবাদ
תודה
ధన్యవాదములు