The Trials and Tribulations with GLIBC

An Exploration of Cross-Compiling for Embedded Linux

Michael Starch

Agenda

- About the Speaker
- Introduction
  - C, libc, and glibc
  - Cross-Compiling for the Novice
  - The Problem and Our Testbed
- Red Herrings Galore
  - Just Compile
  - Prepackaged Cross-Compiler
  - Sysroot
  - Static Linkage
- What Is Going On?
- A Workable Solution
- Conclusion

About the Speaker

- Embedded Systems Engineer
  - Spacecraft Control Software
  - Works Primarily with C/C++
- Community Manager of F´
  - Designed for cross-platform
  - Works on Inexpensive Hardware
- Long Time SCaLE Enthusiast
  - A/V Team Volunteer

Introduction

C, libc, and glibc

- The C Language
  - Compiled into machine-code
  - Cross-platform and operating system
  - Used in embedded systems
- libc
  - Implementation of C standard functions
  - Built on OS API
  - Supplied by C compiler
- glibc
  - GNU C Compiler (gcc) implementation

https://commons.wikimedia.org/wiki/File:C_Programming_Language.svg

https://commons.wikimedia.org/wiki/File:The_GNU_logo.png

Cross-Compiling for the Novice

https://www.asus.com/us/laptops/for-home/all-series/

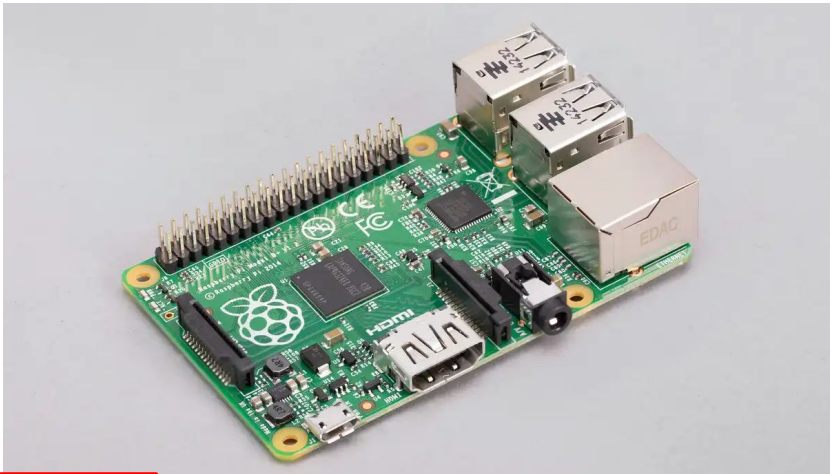
Build Here

https://www.raspberrypi.com/products/raspberry-pi-1-model-b-plus/

Run Here

The Problem and Our Testbed

- Cross-Compiled code can crash
  - Incorrect tools
  - Incorrect libc versions
- Raspberry PI 1B
  - Debian from 2015
  - glibc 2.19



```
./glibc-arm-dynamic: /lib/arm-linux-gnueabihf/libc.so.6: version `GLIBC_2.34' not found (required by ./glibc-arm-dynamic)

Illegal instruction

-bash: ./glibc-dynamic: cannot execute binary file: Exec format error
```

Red Herrings Galore

Just Compile

- What: Run gcc directly
- Why: GCC Builds for Local Host
  - Platforms have specific machine instructions
  - File types and endianness may differ
  - e.g. 32bit vs 64bit computers

- What is it really for?
  - Build for this machine and identical machines
  - Compiling source for local use
  - Building and testing locally

```
pi@raspberrypi:~/bin $ ./glibc-dynamic
-bash: ./glibc-dynamic: cannot execute binary file: Exec format error
pi@raspberrypi:~/bin $ file glibc-dynamic
glibc-dynamic: ELF 64-bit LSB shared object, x86-64, version 1 (SYSV), dynamically linked,
.2, BuildID[sha1]=923d4ad8e81482eb8fac4ad1c1397f2c2a90cd81, for GNU/Linux 3.2.0, not strip
```

Distribution or Prepackaged Cross-Compiler

- What: Install and Run Distribution's Cross-Compiler
  - apt install gcc-arm-linux-gnueabihf
  - arm-linux-gnueabihf-gcc
- Why: glibc Is Built-In to GCC
  - Machine instructions correct
  - libc Version is incorrect
  - Version correctness is by luck
- What is it really for?
  - Building for the distribution running on another architecture
  - Alternate platforms for packages (e.g. DEB)
- Extends to downloadable cross-compilers from ARM

```
pi@raspberrypi:~/bin $ ./glibc-ubuntu-dynamic
./glibc-ubuntu-dynamic: /lib/arm-linux-gnueabihf/libc.so.6: version `GLIBC_2.34' not found (required by ./glibc-ubuntu-dynamic)
pi@raspberrypi:~/bin $ file ./glibc-ubuntu-dynamic
./glibc-ubuntu-dynamic: ELF 32-bit LSB shared object, ARM, EABI5 version 1 (SYSV), dynamically linked, interpreter /lib/ld-linux
-armhf.so.3, BuildID[sha1]=4c834a8d3b23382efc41a0f8adc3e3fb847730b, for GNU/Linux 3.2.0, not stripped
pi@raspberrypi:~/bin $ ./glibc-arm-dynamic
./glibc-arm-dynamic: /lib/arm-linux-gnueabihf/libc.so.6: version `GLIBC_2.34' not found (required by ./glibc-arm-dynamic)
pi@raspberrypi:~/bin $ file ./glibc-arm-dynamic
./glibc-arm-dynamic: ELF 32-bit LSB executable, ARM, EABI5 version 1 (SYSV), dynamically linked, interpreter /lib/ld-linux-armhf
.so.3, for GNU/Linux 3.2.0, not stripped
```

Sysroot

- What: Use the --sysroot flag to specify the libraries
- Why: libc not overridden
  - Libc is still supplied by compiler
  - Not all compilers allow sysroot
  - May not work altogether

- What is it really for?
  - Used when building using specific cross-compilers
  - May be used to add in custom user libraries
- Note: may not always work

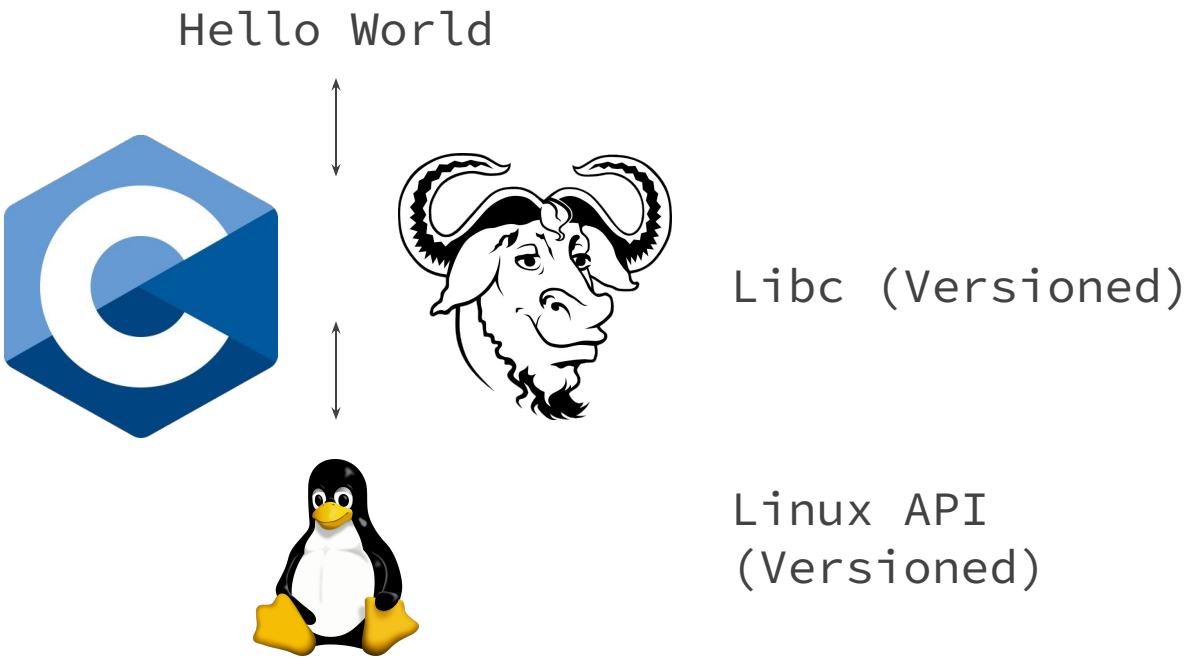
```
pi@raspberrypi:~ $ ./glibc-sysroot-dynamic
./glibc-sysroot-dynamic: /lib/arm-linux-gnueabihf/libc.so.6: version `GLIBC_2.34' not found (r
pi@raspberrypi:~ $ file glibc-sysroot-dynamic
glibc-sysroot-dynamic: ELF 32-bit LSB shared object, ARM, EABI5 version 1 (SYSV), dynamically
BuildID[sha1]=dab4945b161302c007802d217616d453d21e8a24, for GNU/Linux 3.2.0, not stripped
```

Static Linkage

- What: Build **All Code** Into the Executable
  - Avoids shared libraries
  - Should remove all version dependencies
- Why: glibc Builds on Linux API
  - Version cascades to Linux API
- What is it really for?
  - Self-contained libraries
  - User supplied code
  - Vendor supplied code
  - **NOT** libc

```
pi@raspberrypi:~/bin $ ./glibc-ubuntu-static
Segmentation fault
pi@raspberrypi:~/bin $ file glibc-ubuntu-static
glibc-ubuntu-static: ELF 32-bit LSB executable, ARM, EABI5 version 1 (GNU/Linux), statically l
c4bb422220029bf, for GNU/Linux 3.2.0, not stripped
pi@raspberrypi:~/bin $ ./glibc-arm-static
Illegal instruction
pi@raspberrypi:~/bin $ file ./glibc-arm-static
./glibc-arm-static: ELF 32-bit LSB executable, ARM, EABI5 version 1 (SYSV), statically linked,
pi@raspberrypi:~/bin $
```

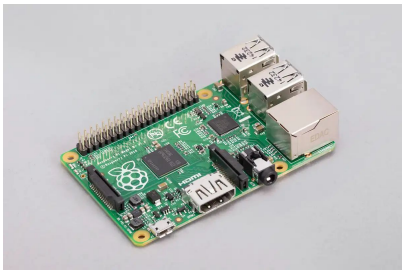
What Is Going On?



Attribution: lewing@isc.tamu.edu Larry Ewing and The GIMP Simon Budig

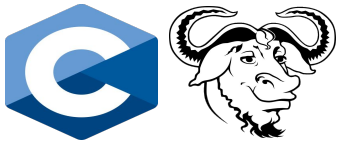
What Is Going On?

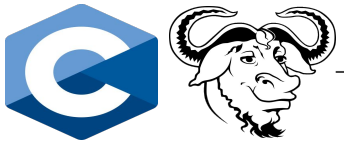
Hello World

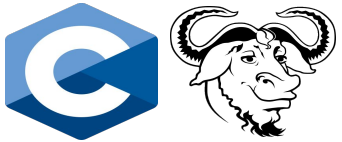


Host OS ⟷ Compiler

Target







A Workable Solution

Build A Cross-Compiler: Crosstool-NG

- What: Build your own cross-compiler!
  - Select C Library and version
  - Select Kernel version
- Why: Creates a dedicated tool
  - Specific to your platform
  - Builds correct version of libc into your toolchain

- What is it really for?
  - Cross-Compiling!
  - No need to build a Kernel!

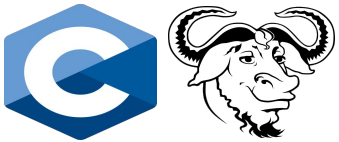
```
    Source of linux (Released tarball)  --->
    Version of linux (4.1.49)  --->

    C library (glibc)  --->
    *** Options for glibc ***
    Show glibc versions from (GNU)  --->
    Source of glibc (Released tarball)  --->
    Version of glibc (2.19)  --->
```

```
pi@raspberrypi:~ $ ./glibc-ng-dynamic
Hello World
pi@raspberrypi:~ $ file ./glibc-ng-dynamic
./glibc-ng-dynamic: ELF 32-bit LSB executable, ARM, EABI5 version 1 (SYSV), dynamically linked,
```

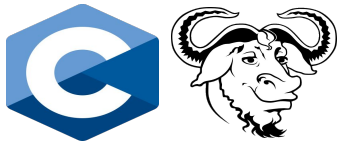
What Is Going On?

Host OS

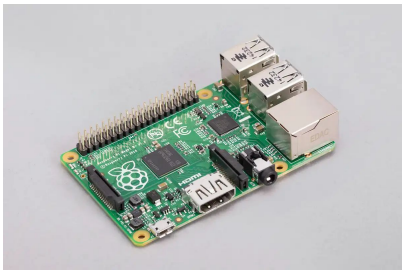


Hello World

Compiler



Target



Conclusion

- glibc Version Not Found, Segfault, and Illegal Instructions caused by Cross-Compiler Toolchains

- Many Obvious Solutions Are Error-Prone

- Users should be prepared to roll their own compiler

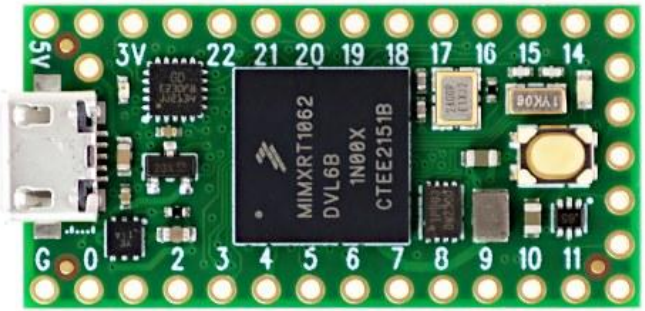
Questions

Alternate Solutions

Vendor Supplied Toolchain

- Compile with a toolchain supplied by a Vendor

- Why is it not ideal?
  - Not available for every embedded system
  - Often not OpenSource
- What is it really for?
  - Use on vendor supplied Linux distributions



https://www.pjrc.com/store/teensy40.html

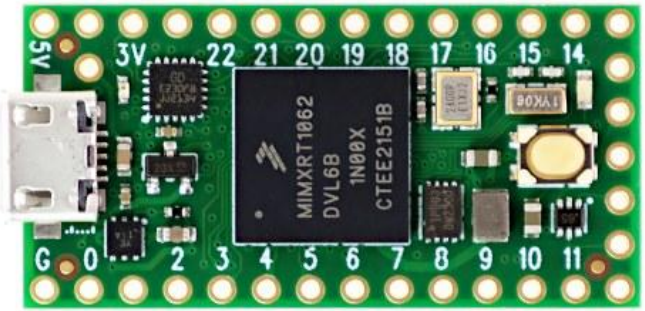
Alternate libc: newlibc

- Newlib C is an alternate implementation of LibC

- Not built on the glibc implementation

- Why is it not ideal?
  - Not intended for use on Linux

- What is it really for?
  - Use on Baremetal systems



https://www.pjrc.com/store/teensy40.html

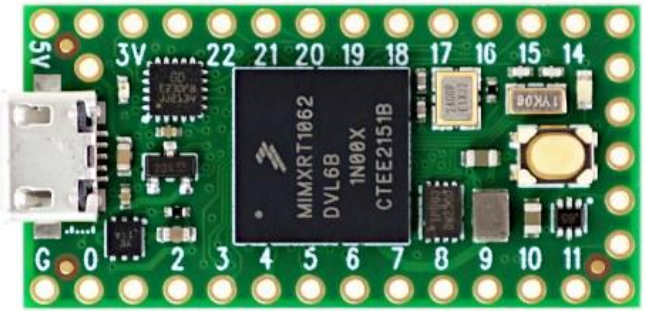
Build The Kernel: Yocto

- By building the Linux Kernel you get a compiler for free!

- glibc is built-in!

- Why is it not ideal?
  - Requires a lot of work

- What is it really for?
  - Complete control



https://www.pjrc.com/store/teensy40.html