RPi
and
MicroPython
32-Bit

Using Coss-Compiler on 64-Bit Platform

NORMAN MCENTIRE

Copyright (c) 2023 Servin Corp

References

- https://micropython.org/

Copyright (c) 2023 Servin Corp

Example Project: MicroPython

- Suppose your task is to build MicroPython
- You want to build MicroPython for both your 64-bit Host
  - Previous we built for 64-bit
- And for your 32-bit Target
  - Here we build for 32-bit
  - Cross-Compiler

Copyright (c) 2023 Servin Corp

Cloning the code from GitHub

https://github.com/micropython/micropython

```
$ git clone --depth=1 https://github.com/micropython/micropython
Cloning into 'micropython'...
remote: Enumerating objects: 5612, done.
remote: Counting objects: 100% (5612/5612), done.
remote: Compressing objects: 100% (4561/4561), done.
remote: Total 5612 (delta 1340), reused 3114 (delta 726), pack-reused 0
Receiving objects: 100% (5612/5612), 7.97 MiB | 6.08 MiB/s, done.
Resolving deltas: 100% (1340/1340), done.

$ cd micropython/

$ ls
ACKNOWLEDGEMENTS    CONTRIBUTING.md  examples  LICENSE    ports           README.md  tools
CODECONVENTIONS.md  docs             extmod    logo       py              shared
CODEOFCONDUCT.md    drivers          lib       mpy-cross  pyproject.toml  tests
```

Copyright (c) 2023 Servin Corp

Building MicroPthon for Linux

```
$ cd ports

$  ls
bare-arm  embed  esp8266  minimal  pic16bit  qemu-arm    rp2   stm32   unix         windows
cc3200    esp32  mimxrt   nrf      powerpc   renesas-ra  samd  teensy  webassembly  zephyr

$ cd unix

$ ls
alloc.c          input.h   modmachine.c   mpbtstackport_common.c  mphalport.h     README.md
coverage.c       main.c    modos.c        mpbtstackport.h         mpnimbleport.c  unix_mphal.c
coveragecpp.cpp  Makefile  modsocket.c    mpbtstackport_h4.c      mpnimbleport.h  variants
fatfs_port.c     mbedtls   modtermios.c   mpbtstackport_usb.c     mpthreadport.c
gccollect.c      modffi.c  modtime.c      mpconfigport.h          mpthreadport.h
input.c          modjni.c  mpbthciport.c  mpconfigport.mk         qstrdefsport.h

$ make submodules
. . .

$ make
. . .
```

Copyright (c) 2023 Servin Corp

make submodules

```
$ make submodules
Use make V=1 or set BUILD_VERBOSE in your environment to increase build verbosity.
Package libffi was not found in the pkg-config search path.
Perhaps you should add the directory containing `libffi.pc'
to the PKG_CONFIG_PATH environment variable
Package 'libffi', required by 'virtual:world', not found
Package libffi was not found in the pkg-config search path.
Perhaps you should add the directory containing `libffi.pc'
to the PKG_CONFIG_PATH environment variable
Package 'libffi', required by 'virtual:world', not found
Updating submodules: lib/mbedtls lib/berkeley-db-1.xx lib/micropython-lib
Submodule 'lib/berkeley-db-1.xx' (https://github.com/pfalcon/berkeley-db-1.xx) registered for path '../../lib/
berkeley-db-1.xx'
Submodule 'lib/mbedtls' (https://github.com/ARMmbed/mbedtls.git) registered for path '../../lib/mbedtls'
Submodule 'lib/micropython-lib' (https://github.com/micropython/micropython-lib.git)
 registered for path '../../lib/micropython-lib'
Cloning into '/home/nmcentire/micropython/lib/berkeley-db-1.xx'...
Cloning into '/home/nmcentire/micropython/lib/mbedtls'...
Cloning into '/home/nmcentire/micropython/lib/micropython-lib'...
Submodule path '../../lib/berkeley-db-1.xx': checked out '35aaec4418ad78628a3b935885dd189d41ce779b'
Submodule path '../../lib/mbedtls': checked out '981743de6fcdbe672e482b6fd724d31d0a0d2476'
Submodule path '../../lib/micropython-lib': checked out 'e025c843b60e93689f0f991d753010bb5bd6a722'
```

Copyright (c) 2023 Servin Corp

apt-cache search libffi

```
apt-cache search libffi
libffindex0 - library for simple index/database for huge amounts of small files
libffindex0-dev - library for simple index/database for huge amounts of small files (development)
libgirepository-1.0-1 - Library for handling GObject introspection data (runtime library)
libghc-libffi-dev - A binding to libffi
libghc-libffi-doc - A binding to libffi; documentation
libghc-libffi-prof - A binding to libffi; profiling libraries
libjffi-java - Java Foreign Function Interface
libjffi-jni - Java Foreign Function Interface (JNI library)
libffi-dev - Foreign Function Interface library (development files)
libffi8 - Foreign Function Interface library runtime
libffi-c-perl - C data types for FFI
libffi-checklib-perl - module to check availability of a library for FFI
libffi-platypus-perl - module to create Perl bindings to non-Perl libraries with FFI
libffi-platypus-type-enum-perl - custom platypus type for dealing with C enumerated types
```

Copyright (c) 2023 Servin Corp

apt show libffi-dev

```
$ apt show libffi-dev
Package: libffi-dev
Version: 3.4.4-1
Priority: optional
Section: libdevel
Source: libffi
Maintainer: Debian GCC Maintainers <debian-gcc@lists.debian.org>
Installed-Size: 293 kB
Depends: libffi8 (= 3.4.4-1)
Conflicts: libffi4-dev
Homepage: https://sourceware.org/libffi/
Tag: devel::library, role::devel-lib
Download-Size: 56.0 kB
APT-Sources: http://deb.debian.org/debian bookworm/main arm64 Packages
Description: Foreign Function Interface library (development files)
 This package contains the headers and static library files necessary for
 building programs which use libffi.
 .
 A foreign function interface is the popular name for the interface that
 allows code written in one language to call code written in another
 language.
```

Copyright (c) 2023 Servin Corp

sudo apt install libffi-dev

```
sudo apt install libffi-dev
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  libffi-dev
. . .
```

Copyright (c) 2023 Servin Corp

dpkg -L libffi-dev

NOTE! These include
Files are for aarch64-linux-gnu

```
dpkg -L libffi-dev

/usr
/usr/include
/usr/include/aarch64-linux-gnu
/usr/include/aarch64-linux-gnu/ffi.h
/usr/include/aarch64-linux-gnu/ffitarget.h
/usr/lib
/usr/lib/aarch64-linux-gnu
/usr/lib/aarch64-linux-gnu/libffi.a
/usr/lib/aarch64-linux-gnu/libffi_pic.a
/usr/lib/aarch64-linux-gnu/pkgconfig
/usr/lib/aarch64-linux-gnu/pkgconfig/libffi.pc
/usr/share
. . .
```

Copyright (c) 2023 Servin Corp

Also install the armhf versions

```
$ sudo apt install libffi-dev:armhf
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libffi8:armhf
The following NEW packages will be installed:
  libffi-dev:armhf libffi8:armhf
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 74.5 kB of archives.
After this operation, 353 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Copyright (c) 2023 Servin Corp

dpkg -L libffi-dev

Notice the arm-linux-gneeabuhf
Versions of header files

```
$ dpkg -L libffi-dev:armhf
/.
/usr
/usr/include
/usr/include/arm-linux-gnueabihf
/usr/include/arm-linux-gnueabihf/ffi.h
/usr/include/arm-linux-gnueabihf/ffitarget.h
/usr/lib
/usr/lib/arm-linux-gnueabihf
/usr/lib/arm-linux-gnueabihf/libffi.a
/usr/lib/arm-linux-gnueabihf/libffi_pic.a
/usr/lib/arm-linux-gnueabihf/pkgconfig
/usr/lib/arm-linux-gnueabihf/pkgconfig/libffi.pc
. . .
```

Copyright (c) 2023 Servin Corp

make CROSS_COMIPILE=arm-linux-gnueabihf-

```
$ make CROSS_COMPILE=arm-linux-gnueabihf-
Use make V=1 or set BUILD_VERBOSE in your environment to increase build verbosity.
GEN build-standard/genhdr/qstr.i.last
GEN build-standard/genhdr/qstr.split
. . .
CC ../../shared/readline/readline.c
LINK build-standard/micropython
   text    data     bss    dec    hex   filename
 685353   59144    6976 751473  b7771   build-standard/micropython


file build-standard/micropython
build-standard/micropython: ELF 32-bit LSB pie executable, ARM, EABI5 version 1 (SYSV), dynamically linked,
interpreter /lib/ld-linux-armhf.so.3, BuildID[sha1]=02dccf5a7502a4b33d0827b2914ea13bb955efdf, for GNU/Linux
3.2.0, stripped
```

Copyright (c) 2023 Servin Corp

Running 32-bit micro python

```
$ ./build-standard/micropython
MicroPython 91a3f18 on 2023-10-21; linux [GCC 12.2.0] version
Use Ctrl-D to exit, Ctrl-E for paste mode
>>> print("hello micropython")
hello micropython
>>>
```

Copyright (c) 2023 Servin Corp

Try on 32-Bit RPI system

```
scp ./build-standard/micropython metaembedded@192.168.4.34:./Downloads/.
metaembedded@192.168.4.34's password:
micropython
```

```
ssh metaembedded@192.168.4.34
metaembedded@192.168.4.34's password:
Linux raspberrypi 6.1.21-v7l+ #1642 SMP Mon Apr  3 17:22:30 BST 2023 armv7l

$ arch
armv7l

$ ldd Downloads/micropython
Downloads/micropython: /lib/arm-linux-gnueabihf/libc.so.6: version `GLIBC_2.33' not found (required by Downloads/micropython)
Downloads/micropython: /lib/arm-linux-gnueabihf/libc.so.6: version `GLIBC_2.34' not found (required by Downloads/micropython)
  linux-vdso.so.1 (0xbeeec000)
  /usr/lib/arm-linux-gnueabihf/libarmmem-${PLATFORM}.so => /usr/lib/arm-linux-gnueabihf/libarmmem-v7l.so (0xb6e55000)
  libm.so.6 => /lib/arm-linux-gnueabihf/libm.so.6 (0xb6dd1000)
  libffi.so.8 => not found
  libc.so.6 => /lib/arm-linux-gnueabihf/libc.so.6 (0xb6c7d000)
  /lib/ld-linux-armhf.so.3 (0xb6ed2000)
```

Issue #2

Issue #1

Copyright (c) 2023 Servin Corp

Solving issue #1: Missing libffi.so.8 not found

```
apt-cache search libffi
libalt-alien-ffi-system-perl - simplified alternative to Alien::FFI that uses system libffi
libffi-checklib-perl - module to check availability of a library for FFI
libffi-dev - Foreign Function Interface library (development files)
libffi-platypus-perl - module to create Perl bindings to non-Perl libraries with FFI
libffi6 - Foreign Function Interface library runtime
libffi6-dbg - Foreign Function Interface library runtime (debug symbols)
libffi6-dev - Foreign Function Interface library (development files)
libffi7 - Foreign Function Interface library runtime
libffindex0 - library for simple index/database for huge amounts of small files
libffindex0-dev - library for simple index/database for huge amounts of small files (development)
libjffi-java - Java Foreign Function Interface
libjffi-jni - Java Foreign Function Interface (JNI library)

sudo apt install libffi-dev
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
libffi-dev is already the newest version (3.3-6).
The following packages were automatically installed and are no longer required:
  libirs-export161 libisccfg-export163 policycoreutils selinux-utils
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 88 not upgraded.
```

Copyright (c) 2023 Servin Corp

Version Issue!

```
find /usr/lib | grep libffi
/usr/lib/arm-linux-gnueabihf/libffi.so.7
/usr/lib/arm-linux-gnueabihf/libffi_pic.a
/usr/lib/arm-linux-gnueabihf/pkgconfig/libffi.pc
/usr/lib/arm-linux-gnueabihf/libffi.so
/usr/lib/arm-linux-gnueabihf/libffi.a
/usr/lib/arm-linux-gnueabihf/libffi.so.7.1.0
```

```
ldd ./Downloads/micropython
./Downloads/micropython: /lib/arm-linux-gnueabihf/libc.so.6: version `GLIBC_2.33' not found (required by ./Downloads/micropython)
./Downloads/micropython: /lib/arm-linux-gnueabihf/libc.so.6: version `GLIBC_2.34' not found (required by ./Downloads/micropython)
  linux-vdso.so.1 (0xbeee7000)
  /usr/lib/arm-linux-gnueabihf/libarmmem-${PLATFORM}.so => /usr/lib/arm-linux-gnueabihf/libarmmem-v7l.so (0xb6eec000)
  libm.so.6 => /lib/arm-linux-gnueabihf/libm.so.6 (0xb6e68000)
  libffi.so.8 => not found
  libc.so.6 => /lib/arm-linux-gnueabihf/libc.so.6 (0xb6d14000)
  /lib/ld-linux-armhf.so.3 (0xb6f69000)
```

Copyright (c) 2023 Servin Corp

Version Issues Are Common!

- Possible Solutions
  - Upgrade to latest version
    Did not work here!
  - Create Symbolic Link
    - Risky because interfaces can change
  - Copy needed files from other system

```
sudo apt install --only-upgrade libffi-dev
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
libffi-dev is already the newest version (3.3-6).
The following packages were automatically installed and are no longer required:
  libirs-export161 libisccfg-export163 policycoreutils selinux-utils
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 88 not upgraded.
```

```
$ arch
aarch64

$ find /usr/lib | grep libffi
/usr/lib/arm-linux-gnueabihf/libffi.so
/usr/lib/arm-linux-gnueabihf/libffi.a
/usr/lib/arm-linux-gnueabihf/pkgconfig/libffi.pc
/usr/lib/arm-linux-gnueabihf/libffi_pic.a
/usr/lib/arm-linux-gnueabihf/libffi.so.8.1.2
/usr/lib/arm-linux-gnueabihf/libffi.so.8
/usr/lib/aarch64-linux-gnu/libffi.so
/usr/lib/aarch64-linux-gnu/libffi.a
/usr/lib/aarch64-linux-gnu/pkgconfig/libffi.pc
/usr/lib/aarch64-linux-gnu/libffi_pic.a
/usr/lib/aarch64-linux-gnu/libffi.so.8.1.2
/usr/lib/aarch64-linux-gnu/libffi.so.8
```

Copyright (c) 2023 Servin Corp

Copy needed version of libffi

Issue #1 Solved!

```
$ scp dev@192.168.4.73:/usr/lib/arm-linux-gnueabihf/libffi.so.8 ~/Downloads
. . .


$ sudo mv Downloads/libffi.so.8 /usr/lib/arm-linux-gnueabihf/.

$ ldd Downloads/micropython
Downloads/micropython: /lib/arm-linux-gnueabihf/libc.so.6: version `GLIBC_2.33' not found (required by Download
micropython)
Downloads/micropython: /lib/arm-linux-gnueabihf/libc.so.6: version `GLIBC_2.34' not found (required by Download
micropython)
  linux-vdso.so.1 (0xbed92000)
  /usr/lib/arm-linux-gnueabihf/libarmmem-${PLATFORM}.so => /usr/lib/arm-linux-gnueabihf/libarmmem-v7l.so
(0xb6ea9000)
  libm.so.6 => /lib/arm-linux-gnueabihf/libm.so.6 (0xb6e25000)
  libffi.so.8 => /lib/arm-linux-gnueabihf/libffi.so.8 (0xb6e04000)
  libc.so.6 => /lib/arm-linux-gnueabihf/libc.so.6 (0xb6cb0000)
  /lib/ld-linux-armhf.so.3 (0xb6f26000)
  libgcc_s.so.1 => /lib/arm-linux-gnueabihf/libgcc_s.so.1 (0xb6c83000)
```

Issue #1 Solved!

Copyright (c) 2023 Servin Corp

Now to focus on Issue #2

GLIBC_2.33 / GLIBC_2.34 Not Found

```
$./Downloads/micropython
./Downloads/micropython: /lib/arm-linux-gnueabihf/libc.so.6: version `GLIBC_2.33' not found
(required by ./Downloads/micropython)
./Downloads/micropython: /lib/arm-linux-gnueabihf/libc.so.6: version `GLIBC_2.34' not found
(required by ./Downloads/micropython)
```

Copyright (c) 2023 Servin Corp

Target System

```
$ readelf -a Downloads/micropython | grep libc
 0x00000001 (NEEDED)                     Shared library: [libc.so.6]
00066044  00001116 R_ARM_JUMP_SLOT   00000000   __libc_current_si[...]@GLIBC_2.4
000661cc  00009516 R_ARM_JUMP_SLOT   00000000   __libc_start_main@GLIBC_2.34


$ ldd --version
ldd (Debian GLIBC 2.31-13+rpt2+rpi1+deb11u5) 2.31
Copyright (C) 2020 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
Written by Roland McGrath and Ulrich Drepper.


$ getconf GNU_LIBC_VERSION
glibc 2.31
```

Backwards compatibility
Is fully supported with
Versioned symbols —- but
What we have her is building with NEW VERSION
And trying to run on older version!

Host System

```
ldd --version
ldd (Debian GLIBC 2.36-9+rpt2+deb12u3) 2.36
Copyright (C) 2022 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
Written by Roland McGrath and Ulrich Drepper.

$ getconf GNU_LIBC_VERSION
glibc 2.36
```

Copyright (c) 2023 Servin Corp

An aside on Symbol Versioning

Copyright (c) 2023 Servin Corp

glib has versioned symbols

Use for backwards compatibility (old code runs on newer systems)

```
$ getconf -a | grep LIBC
GNU_LIBC_VERSION                    glibc 2.31

$ readelf --version-info /lib/arm-linux-gnueabihf/libc.so.6 | head

Version symbols section '.gnu.version' contains 2346 entries:
 Addr: 0x0000000000013188  Offset: 0x013188  Link: 4 (.dynsym)
  000:   0 (*local*)       0 (*local*)       0 (*local*)       0 (*local*)
  004:  1b (GLIBC_PRIVATE)  1b (GLIBC_PRIVATE)  1c (GLIBC_2.4)    1b (GLIBC_PRIVATE)
  008:  1b (GLIBC_PRIVATE)  1b (GLIBC_PRIVATE)  1b (GLIBC_PRIVATE)  1b (GLIBC_PRIVATE)
  00c:  1c (GLIBC_2.4)     0 (*local*)       0 (*local*)      1b (GLIBC_PRIVATE)
  010:   2 (GLIBC_2.4)     2 (GLIBC_2.4)     2h(GLIBC_2.4)     2h(GLIBC_2.4)
  014:   2 (GLIBC_2.4)     2 (GLIBC_2.4)     2h(GLIBC_2.4)     2 (GLIBC_2.4)
  018:   2 (GLIBC_2.4)     2 (GLIBC_2.4)     2 (GLIBC_2.4)     2 (GLIBC_2.4)

$ readelf --version-info /lib/arm-linux-gnueabihf/libc.so.6 | grep GLIBC_2.30
  2ac:   2 (GLIBC_2.4)     2 (GLIBC_2.4)     2 (GLIBC_2.4)    19 (GLIBC_2.30)
  520:   2 (GLIBC_2.4)     2 (GLIBC_2.4)     2 (GLIBC_2.4)    19 (GLIBC_2.30)
  52c:   b (GLIBC_2.13)    2 (GLIBC_2.4)    19 (GLIBC_2.30)    2 (GLIBC_2.4)
  5bc:   2 (GLIBC_2.4)    19 (GLIBC_2.30)    2 (GLIBC_2.4)     2 (GLIBC_2.4)
  83c:   2 (GLIBC_2.4)     2 (GLIBC_2.4)     2 (GLIBC_2.4)    19 (GLIBC_2.30)
  0x0350: Rev: 1  Flags: none  Index: 25  Cnt: 2  Name: GLIBC_2.30
  0x0390: Parent 1: GLIBC_2.30
```

Copyright (c) 2023 Servin Corp

Example - No Symbol Versioning

```
cat mylibv1.c
#include <stdio.h>

void hello() {
  puts("hello: v1");
}
```

```
$ gcc -shared -o libmylib.so mylibv1.c

$ readelf -a libmylib.so | grep -E 'hello' -B 3
     5: 00000000     0 FUNC    GLOBAL DEFAULT  UND puts@GLIBC_2.4 (2)
     6: 00000000     0 NOTYPE  WEAK   DEFAULT  UND __gmon_start__
     7: 00000000     0 NOTYPE  WEAK   DEFAULT  UND _ITM_registerTMC[...]
     8: 00000430    28 FUNC    GLOBAL DEFAULT   11 hello
--
    71: 000002e8     0 NOTYPE  LOCAL  DEFAULT   10 $a
    72: 00000000     0 FUNC    WEAK   DEFAULT  UND __cxa_finalize@G[...]
    73: 00000000     0 NOTYPE  WEAK   DEFAULT  UND _ITM_deregisterT[...]
    74: 00000430    28 FUNC    GLOBAL DEFAULT   11 hello
```

Copyright (c) 2023 Servin Corp

Example - Symbol Versioning

```
$ cat mylibv1.version
MYLIB_1.0 {
  global: hello;
};
```

```
gcc -shared -Wl,--version-script,mylibv1.version -o libmylib.so mylibv1.c

$ readelf -a libmylib.so | grep -E 'hello' -B 3
     5: 00000000     0 FUNC    GLOBAL DEFAULT  UND puts@GLIBC_2.4 (3)
     6: 00000000     0 NOTYPE  WEAK   DEFAULT  UND __gmon_start__
     7: 00000000     0 NOTYPE  WEAK   DEFAULT  UND _ITM_registerTMC[...]
     8: 00000000     0 OBJECT  GLOBAL DEFAULT  ABS MYLIB_1.0
     9: 00000494    28 FUNC    GLOBAL DEFAULT   12 hello@@MYLIB_1.0
```

Copyright (c) 2023 Servin Corp

How to Have Two Versions of glibc on same machine?

- Having two versions of glibc on the same system can be complex and risky!
  - You could even render your system un-bootable!
    - Don’t do it! :)
- Any yet, in Embedded Linux development, you may indeed need to have multiple glibc versions!
  - For development, for testing, for migration, etc.
- What to do?
  - Linux Containers! (Topic for another lesson!)
  - Each container can have it’s own version of glibc!

Copyright (c) 2023 Servin Corp