An OS to build, deploy and securely manage billions of devices

Latest News:

Apache Mynewt 1.10.0, Apache NimBLE 1.5.0 (/download) released (May 6, 2022)

Porting Mynewt to a new CPU Architecture

A new CPU architecture typically requires the following:

- A new compiler
- New architecture-specific code for the OS

- Helper libraries to help others porting to the same architecture

These are discussed below:

Create A New Compiler

NOTE: Newt does not automatically install the compilers require to build all platforms. Its up to the user using their local machines package manager to install the compilers. The step described here just registers the compiler with newt.

Create a new directory (named after the compiler you are adding). Copy the `pkg.yml` file from another compiler.

Edit the `pkg.yml` file and change the configuration attributes to match your compiler. Most are self-explanatory paths to different compiler and linker tools. There are a few configuration attributes worth noting.

Configuration Attributes	Description
pkg.keywords	Specific keywords to help others search for this using newt
compiler.flags.default	default compiler flags for this architecture
compiler.flags.optimized	additional flags when the newt tool builds an optimized image
compiler.flags.debug	additional flags when the newt tool builds a debug image

Implement Architecture-specific OS code

There are several architecture-specific code functions that are required when implementing a new architecture. You can find examples in the `sim` architecture within Mynewt.

When porting to a new CPU architecture, use the existing architectures as samples when writing your implementation.

Please contact the Mynewt development list for help and advice portingto new MCU.

Previous: Porting Mynewt to a new MCU (port_mcu.html) Next: Baselibc (../../modules/baselibc.html)