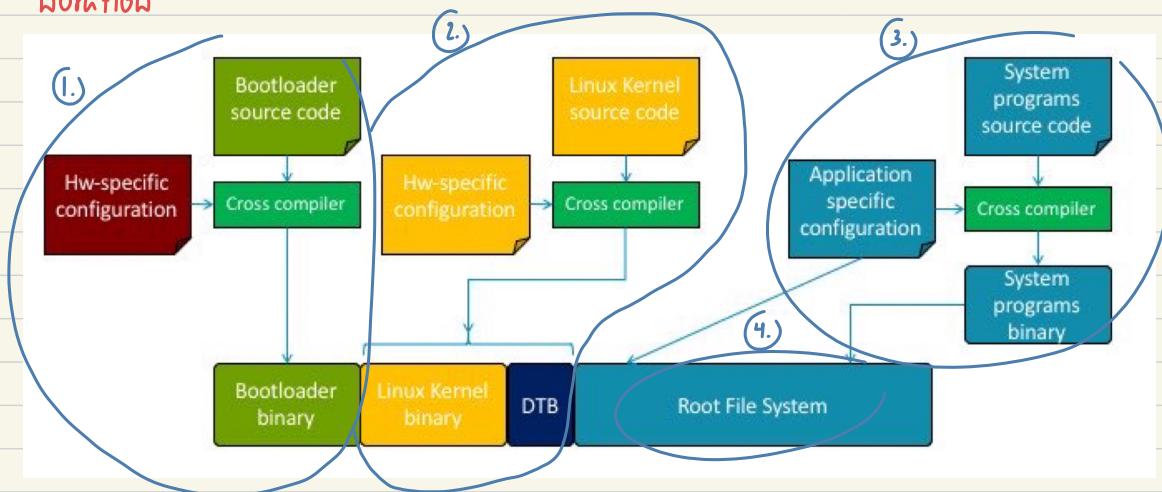


component initialization & runs operate

- **bootloader**
 - **Linux kernel**
 - **device tree blob**
 - **root file system**
- Configured for the embedded system hardware platform
Compiled and linked into an executable format
Deployed into the embedded system persistent storage for booting and operations

Workflow



- (1.)**
 - bootloader source code must be produced
 - must be configured for specific hardware of embedded system
 - cross compiled for the CPU of choice obtaining the executable code
 - must copy to boot device
- (2.)**

Linux kernel binary and DTB must be ready to be copied to bootloader binary
must obtain kernel source code and config for specific hardware of embedded system
- (3.)**
 - system program source code shall be produced
 - configured for specific application
 - cross compiled obtaining the executable binary

4. - root file system shall be prepared

It typically requires

- To create a file and mount it as a volume on the development host
- To format it using any of the file systems Linux supports (e.g., ext3)
- To create the required directory tree
- To populate it with the needed configuration files
- To populate it with the system program binary

- assign copy the embedded system persistent storage

partition on MicroSD device

Bootflash device on embedded system board

Build System

- complex operation
- the tools (aka. build system) available Linux distribution

Build systems takes care of:

- Building the cross compiler for the selected embedded system CPU
- Managing bootloader/kernel/system programs configuration
- Managing bootloader/kernel/system program build
- Preparation of the root file system and boot device image preparation

(General Aspects)

Buildroot

- Focus on simplicity
- Use existing technologies: kconfig, make
- Open community

Yocto

- Provides core recipes and use layers to get support for more packages and more machines
- Custom modifications should stay in a separate layer
- Versatile build system: tries to be as flexible as possible and to handle most use cases
- Open community but governed by the Yocto Project Advisory Board

(Configuration)

Buildroot reuses kconfig from the Linux kernel

- Entire configuration stored in a single .config/.defconfig
- Defines all aspects of the system: architecture, kernel version/config, bootloaders, user-space packages, etc.
- Building the same system for different machines to be handled separately

In Yocto the configuration is separated in multiple parts:

- **Distribution** configuration (general configuration, toolchain selection, etc.)
- **Machine** configuration (defines the hw architecture, hw features, BSP)
- **Image** recipe (what system programs should be installed on the target)
- Local configuration (e.g., how many threads to use when compiling, whether to remove build artifacts, etc.)
- Allows to build the same image for different machines or using different distributions or different images for one machine

(Purpose)

Buildroot is intended for

- Very small root file systems (< 8 MB)
- Simple embedded system (with limited number of system programs)
- Non-dedicated build engineers (e.g., engineers that are not focused only in building embedded Linux)

Yocto is intended for

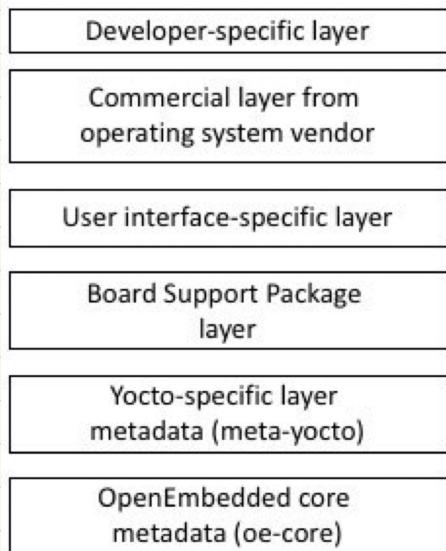
- Large root file systems
- Large embedded systems
- Support for multiple hw configurations
- Dedicated build engineers

The Yocto

- open - source project hosted by linux Foundation

Collaboration of multiple projects that make up the "Yocto Project"

- Bitbake: build tool
- OpenEmbedded core: software framework used for creating Linux distributions
- Poky: a reference distribution of the Yocto Project, containing the OpenEmbedded Build System (BitBake and OpenEmbedded Core) and a set of metadata to start building custom embedded Linux systems
- Application Development Toolkit: provides application developer a way to write sw running on the custom-built embedded Linux system without the need for knowing build systems



Yocto Build system

- the multiple layer is the container in the building block no system
- layer will contain components source code
- contain their metadata -- recipes
 - define how to build }
 - binary output -- packages

(configuration file)

meta/conf/bitbake.conf – default configuration

build/conf/bblayers.conf – layers to be used during build process

*/conf/layers.conf – layer configuration

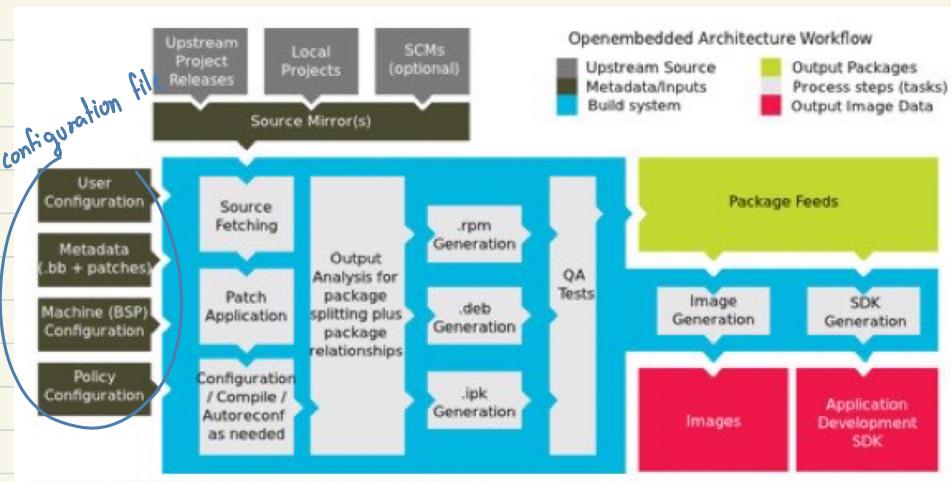
build/conf/local.conf – user-define configuration

meta-yocto/conf/distro/poky.conf – distribution policy

meta-yocto-bsp/conf/machine/board-name.conf – configuration of the board support package

meta/conf/machine/include/tune-CPU-name.inc – CPU-specific configuration

Build System Workflow



: Distribution policy

Defines distribution policies that affect the way individual recipes are built

- May set alternative preferred versions of recipes
- May enable/disable features
- May configure specific package rules
- May adjust image deployment settings

Enabled via the DISTRO setting

Four predefined settings:

- poky-bleeding: enables bleeding edge packages
- poky: core distribution definition, defines the base
- poky-lsb: enable items required for LSB support
- poky-tiny: construct a smaller than normal system

Quiz

1. "Which of these components are required for an embedded Linux system?"

- Bootloader
- Linux kernel
- Device tree blob
- Root file system
- All of the above

2. "Which of the following are the typical workflow for an embedded system?"

- Root file system is prepared > Bootloader source code is procured > Linux Kernel source code is procured
> System programs source code is procured
- Bootloader source code is procured > Root file system is prepared > Linux Kernel source code is procured
> System programs source code is procured
- Bootloader source code is procured > Linux Kernel source code is procured > System programs source code is procured > Root file system is prepared
- Bootloader source code is procured > Linux Kernel source code is procured > Root file system is prepared
> System programs source code is procured

3. "What is the purpose of a build system?"

- The automation of configuring and compiling multiple sources of code to build a system or piece of software
- To build the hardware for an embedded device
- A guide for a developer to build an embedded Linux system

4. "What is the "Yocto Project"?"

- A collaboration of multiple projects related to the building and configuration of custom embedded Linux distributions.
- A custom embedded Linux distribution
- A toolchain for cross-compiling source code
- A library for building Linux systems

5. "What is the "build/conf/bblayers.conf" file used for?"

- A script that generates a custom Linux image
- Overriding the default configuration and defining what to build
- Determining which BitBake layers to be used during build the process
- Defining the distribution policy of the distro

6. "What is the "build/conf/local.conf" file used for?"

- Overriding the default configuration and defining what to build
- Determining which BitBake layers to be used during the build process
- A script that generates a custom Linux image
- Configuration of the board support package