Embedded building tools BOF

Thomas Petazzoni - Michael Opdenacker
Bootlin
https://bootlin.com

Rights to copy

Document sources available on:
https://bootlin.com/pub/conferences/2009/elc/

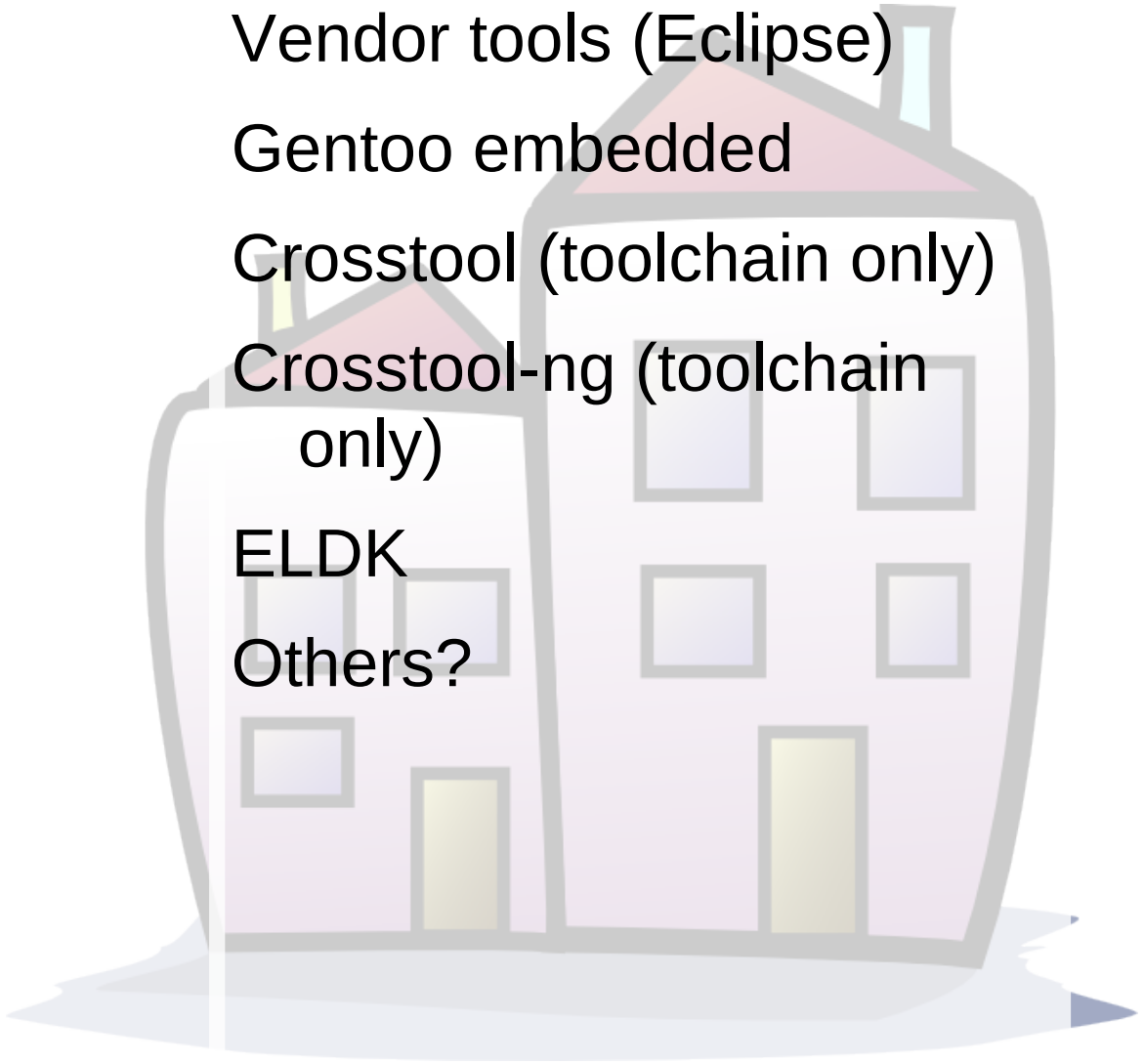
Corrections, suggestions, contributions and translations are welcome!

Latest update: Jul 25, 2018

Which tools to build your system?

- OpenEmbedded
- Buildroot
- Scratchbox
- PTXdist
- LTIB
- Home made tools
- Firmware Linux

- Vendor tools (Eclipse)
- Gentoo embedded
- Crosstool (toolchain only)
- Crosstool-ng (toolchain only)
- ELDK
- Others?



Buildroot

Pros

Supports uClibc

Simple design (kernel config interface, set of Makefiles)

Efficient

Reproducibility of the build process

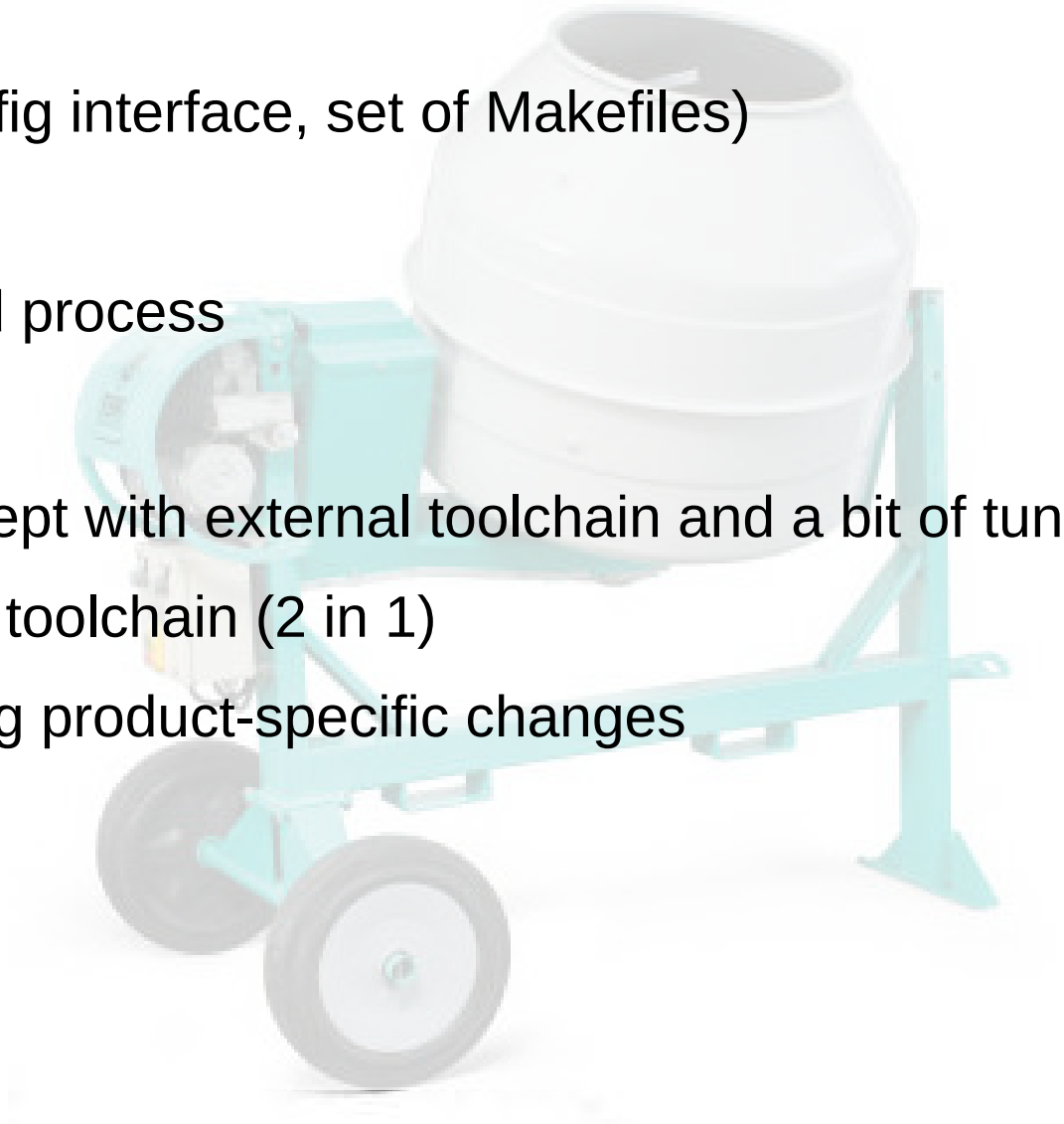
Cons

Doesn't support glibc, except with external toolchain and a bit of tuning

Takes care of building the toolchain (2 in 1)

No clean way of separating product-specific changes

No packages



Scratchbox

Pros

Transparent cross-compilation

Transparent execution

Supports both uClibc and glibc

Cons

No infrastructure for build reproducibility

Complex to add new host tools (need to compile them)

Only uses its own toolchains (can compile your own, but complicated to integrate in Scratchbox)

No recipes, no tool patches shared with the community.

OpenEmbedded

Pros

Clean separation of the build tool and the recipes

Ability to generate a root filesystem and packages

Clean separation of product-specific changes

Widely used in the community

Can generate packages

Cons

No stable releases

Steep learning curve

Very slow to run (for what reason ?)

Too generic. Huge boot times

Packages mandatory

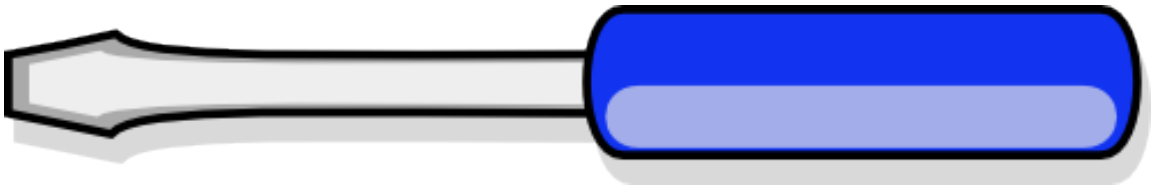
Firmware Linux

By Rob Landley

Not using cross-compiling,
but only native compiling thanks to Qemu

Similar approach than Scratchbox, but less tricks.

How mature is it ?



Home made tools

Pros

Meets your product needs

Cons

No free updates to mainstream software changes

High maintenance cost. Sometimes difficult to extend.
Only one person understands its design.

Lot of legacy cruft in it.

Don't always meet future product needs.

LTIB

Pros

Clean separation of build system and packages

Supports both uClibc and glibc

Easy to extend to support new boards

Accepts standard toolchains

Cons

Only used on Freescale boards?

Size of community?

The other ones ?

- PTXdist
  - Very similar to Buildroot
  - Relies on a separate tool for building the toolchain, but seems limited to glibc
- Emdebian
- Vendor tools
  - Great features
  - But difficult to evaluate without a subscription.

Tools compared

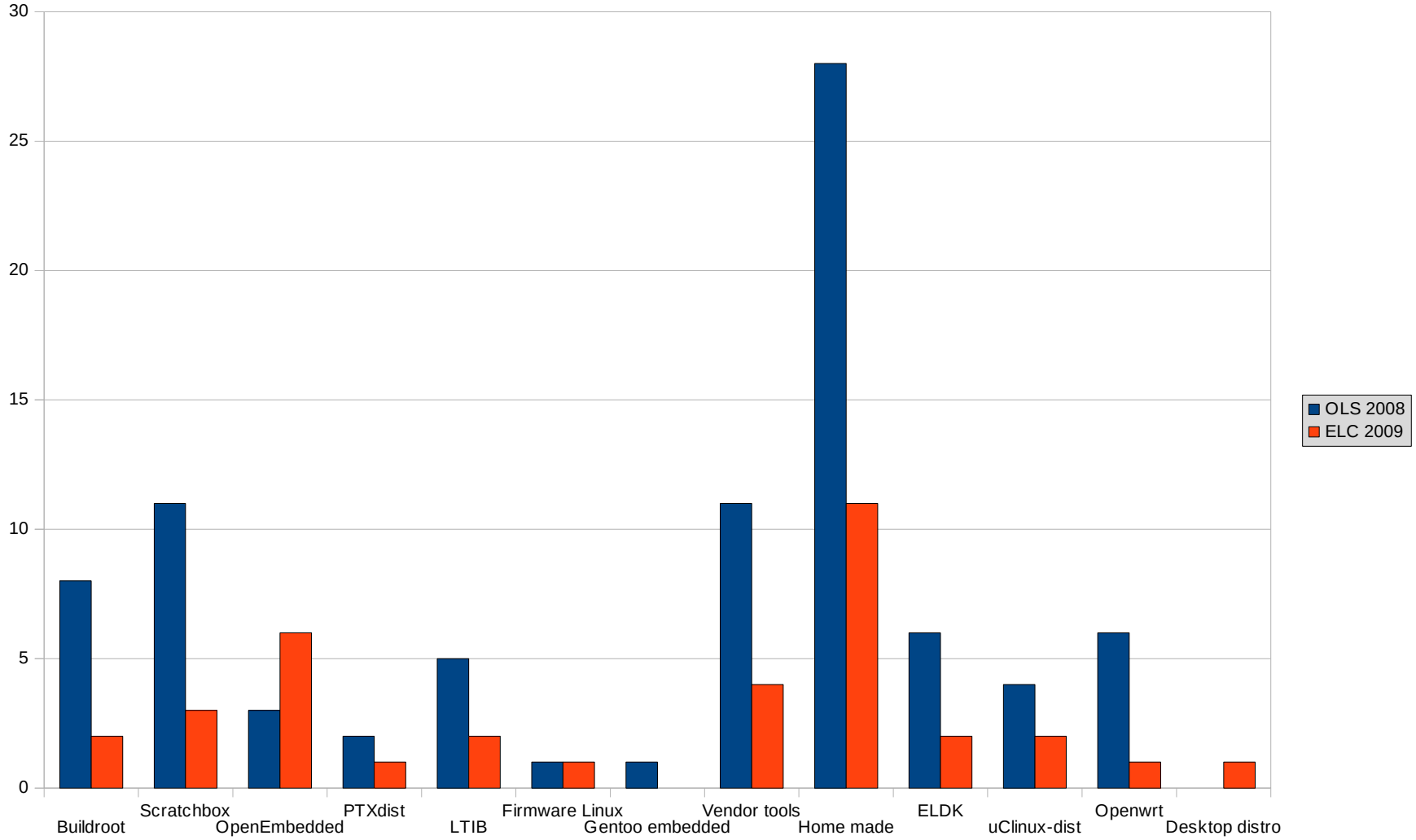
	License	Small systems	glibc (G) uClibc (U)	Reproducibility / Leverage	Popularity	Actively maintained
Buildroot	Free	Yes	G	Good	Good	Yes
Scratchbox	Free	Yes	G + U	Poor	Low	Yes
OpenEmbedded	Free	No	G + U	Good	Very good	Definitely
LTIB	Free	Yes	G + U	Good	Low	Yes
PTXdist	Free	Yes	G	Good	Low	Yes
Gentoo embedded	Free	No?	G (U?)	?	Low?	Yes
Firmware Linux	Free	Yes	?	Poor	Low	Rob never sleeps
Vendor tools	Closed	?	G + U	Good	N/A	Yes
Home made	Closed	?	?	?	N/A	?

Building the toolchain

- Buildroot
  - Mixed with root filesystem construction, not really nice
  - Only uClibc supported
- Crosstool
  - Not really nice configuration through shell scripts
  - Only glibc supported
- Crosstool-ng
  - Much better configuration interface
  - Supports both uClibc and glibc
  - Hasn't attracted a lot of community attention (yet ?)

Tool survey

Number of users per tool in the BOF



Questions

- Why is the community so fragmented ?
  - Because it is not possible to create a universal tool that would match the needs of everybody ?
  - Is it a problem ?
  - NIH syndrome ?
- What are the missing features ?
- What are your complaints about the existing tools ?
  - Complexity ?
  - Lack of flexibility ?
  - Don't see the need ?