MOVIAL

IDEAS IN MOTION

Cross-Compiling Debian from Scratch

DebConf 5
17 July 2005
Otaniemi, Espoo, Finland

Timo Savola <tsavola@movial.fi>
Movial Corporation

Embedded development concerns

- Slow devices
  - No fast, binary-compatible alternatives
- Little memory
- No FPU

- Cross-compilation
  - *Cross-configuring* is tricky
  - Most build systems cause pain
- Custom toolchain
  - Soft-float

- See also Wookey's *Embedded Debian* presentation:
  `http://www.aleph1.co.uk/talks/emdebian/Debconf2005.pdf`

Scratchbox toolkit

- Automated solution to cross-development problems
- Self-contained environment
  - “Empty” sandbox
  - Provides GNU toolchain and common utilities
  - Debian devkit provides tools for building and packaging
- Emulates (the relevant aspects of) a target system
  - Build systems think they are compiling natively
  - `dpkg` manages packages for the target architecture
- Technical overview of Scratchbox 1.0:
  `http://scratchbox.org/~tsavola/scratchbox-fosdem2005.pdf`

Cross-compiling Debian

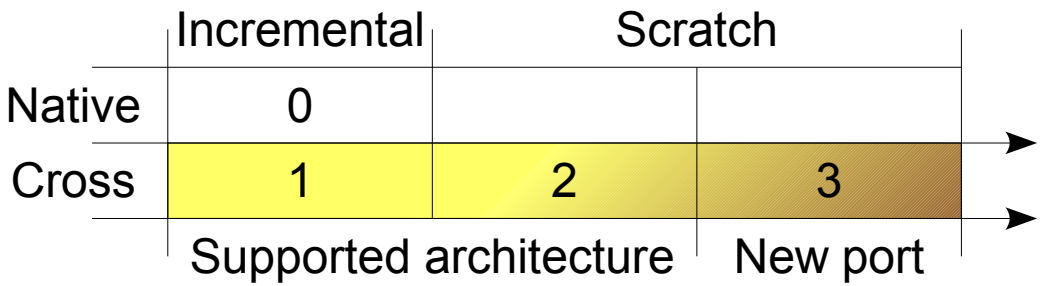
- *Crocodile* project started in 2003
- Build a custom Debian system for ARM
- Use Scratchbox 0.9.7

- Initialise target filesystem with standard Debian packages
  - Libraries
  - Additional build tools
- Produce filesystem images (“rootstraps”) for Scratchbox
- Custom packages built incrementally

- → Toolchain must be compatible with Debian's
- → Build-dependencies needed on the target filesystem
- → Clean rebuilds are not possible/straightforward

Cross-compiling Debian from scratch

- Current objective of the *Crocodile* project
- Build Debian packages onto an empty target filesystem
- Use Scratchbox 1.0



0) The Debian approach
1) Old Crocodile approach
2) Possible to build a truly custom system
3) Straightforward to bootstrap new architecture ports

Minimal Debian system

- 22 *essential* binary packages (Sarge, ARM architecture)
  - 47 with dependencies
- 29 source packages
  - ~240 with build dependencies
  - Circular dependencies

Building the minimal Debian system

- Scratchbox provides build tools
  - Modified `dpkg-checkbuilddeps` recognises them
- Cross-toolchain includes `glibc`, `libstdc++` and `libgcc`
  - Built for the target architecture
  - Built with Debian patches
  - Create packages that are compatible with official Debian

→ Only ~100 source packages needed

→ No circular build dependencies

- Do it yourself:
  `http://scratchbox.org/~tsavola/crocodile`

Patches to the minimal Debian system

- Tests fail for several packages
  - Added support for `DEB_BUILD_OPTIONS=nocheck`
  - Tests can be run later when the *essential* system is available
- Perl
  - Disabled `debian/checkperl`
  - Fixed a bug in `makedepend.SH`
- Not building `libdb4.2-java` for now
  - `gcj` support needs to be added to the toolchain
  - More tools need to be added to Scratchbox

Ideas for Debian Policy

- Building from scratch is a good thing
  - C, C++ library migration
  - New architectures
  - Embedded (and other custom) distributions
  - Improved QA?
- Small things can make a difference
  - `DEB_BUILD_OPTIONS=nocheck`
  - `DEB_BUILD_OPTIONS=nodoc` in addition to indep targets?
- Stricter requirements for *essential* packages
  - Don't build language bindings in the same source package (or make it optional)
  - Depend explicitly on other *essential* packages?

Future of Crocodile

- More flexible build scripts
- Build a larger Debian system from scratch
- Build the minimal Debian for a new architecture
  - Proof of concept
  - CRIS
- Add more tools to Scratchbox
  - Java?

Scratchbox 2

- Modular, configurable
- Support any cross-toolchain
- Install build tools from standard Debian binary packages
- Multiarch could replace sandbox?

Questions?

- scratchbox-users@lists.scratchbox.org
- #scratchbox @ irc.freenode.net