Bootstrap Debian for M32R

g新部　裕

NIIBE Yutaka <gniibe@fsij.org>

The Free Software Initiative of Japan

Before the talk...

- snapshot.debian.net
- ...

Agenda

- Who am I?
- Motivation
- History
- GNU/Linux on M32R
- How to Bootstrap Debian
- Current Status
- Future Work

Who am I?

- Hacker around GNU, Linux, ...
- Founder of GNU/Linux on SuperH Project (1999)
- Chairman of Free Software Initiative of Japan (2002)
- Researcher at: National Institute of AIST, Japan
- Who hacks for embedded kernel technology:
  - Cache optimization for SH-4
  - Mutual exclusion technique (gUSA)
  - MMU usage improvement (exec-bit handling for M32R)
- Debian developer (soon)

Motivation

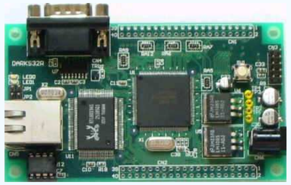
> With a free operating system, we could again have a community
> of cooperating hackers — and invite anyone to join.
> — Richard M. Stallman
>
> —
>
> in “The GNU Operating System and
> the Free Software Movement”

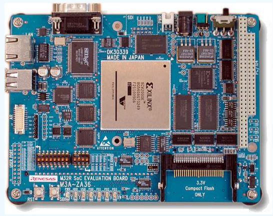
History

- 1999 Linux port for SuperH (by me)
- 2000 GNU/Linux on SuperH
- ... packages for SuperH: not so good
- 2002 Linux port for M32R (by Takata)
- 2003-2005 Debian GNU/Linux for M32R
  - 2004 M32R port merged into Linux 2.6
  - 2005 Debian M32R!

GNU/Linux on M32R

- M32R
  - Renesas Technology Corp.
  - Embedded market
- Wide-range support
  - No-MMU
  - Single processor
  - Multi-processor
- Work
  - Linux 2.6.x
  - GCC, Binutils, GDB
  - GNU C library
  - Debian





The Porting Work

- Porting
  - Need to face with all Debian packages
- Aspect of Quality Assurance
  - Cross build-ability

Bootstrap Debian GNU/Linux

- Goal

  Porting Debian, every packages should be able to be built cleanly *by* Debian with *no* "black magic" manual intervention

- Debian *BY* Debian

How to Bootstrap Debian (1/2)

- Problem:
  - Given a Debian distribution for an architecture
  - Provide a Debian distribution for another architecture
- So far, people did “ad hoc” solution (mostly by hand)
- Clean Solution:
  - Everything by Debian packages

How to Bootstrap Debian (2/2)

- Perfect:
  - Cross build all the packages

    “Perfect is the enemy of good enough”
    — Soviet Admiral Sergei Georgievich Gorshkov
- Good enough:
  - Cross build some important packages:
    - debootstrap + build-essential
    - ... so that ...
    - we can setup (native) autobuilder by cross build
- But watch out!
  - Minimalist approach sometimes doesn’t work in real world.
  - Cross build marginal packages too, in practice.

Bootstrap Debian! (1/8)

- GCC, Binutils port
- Support of m32r-unknown-linux-gnu
  - config.guess
  - config.sub
  - ...
- GDB port

Bootstrap Debian! (2/8)

- Kernel port
  - Headers (string manipulations, bit handling)
  - entry.S (exception, interrupt handling)
  - io, setup
  - Page fault handling (MMU support)
  - Drivers
  - ...

Bootstrap Debian! (3/8)

- GNU C library port
  - String manipulations, bit handling
  - System call interface
  - Thread implementation
  - Dynamic loader
  - ...

Bootstrap Debian! (4/8)

- Cross compile environment
  - binutils
  - GCC
  - linux-kernel-headers
  - file
  - dpkg
  - dpkg-cross
  - glibc

Bootstrap Debian! (5/8)

- Cross compile essential libraries
  - zlib
  - ncurses
  - slang
  - db4.2
  - gdbm
  - pam
  - ...

Bootstrap Debian! (6/8)

- Cross compile essential packages
- Packages needed by debootstrap
  - tar
  - grep
  - sysvinit
  - ...
  - perl
  - ...

Bootstrap Debian! (7/8)

- Cross compile build-essential packages
- Packages for autobuilder
  - GCC (native)
  - dpkg
  - ...

Bootstrap Debian! (8/8)

- Native build by autobuilder
  - Native build machines needed

Current Status

- Done: 7/8
- Scope: 134 packages for cross-build
  - 72 packages: No problem
  - More than 70 patches sent to Debian BTS
  - About 40 have been accepted
  - Debian development process ... takes time
- We can do ’apt-get’ for M32R now

Experiences

- GNU autotools (autoconf, automake, libtools)
  - configure –build, –host
  - cross libtool usage
  - `AC_TRY_RUN` of autoconf
- dpkg-cross
- Various issues
  - make check, make test (only runs native machine)
  - CC/CXX hardcoded as gcc or g++
  - manual generation (help2man) which should done by maintainer
- automake does not support cross build well (yet)
- perl build (which assumes native build: miniperl generates code)

Future Work

- More cross build support
  - Perl modules
  - Python and its modules
  - Ruby and its modules
  - ...
- `automake` for cross build
  - `CC_FOR_BUILD`
- dpkg & dpkg-cross integration
- lintian support for cross build
- Cross autobuilder

Summary

- The Hacker Attitude #2:

  “No problem should ever have to be solved twice.”
- But it has been done *ten* times, no more for M32R
- Formalize the process and reuse the Software

Acknowledgement

- Thanks LinuxTag for giving me opportunity
- Thanks Martin Schulze for Debian Day
- Thanks Alexander Schmehl for this arrangement

Question?