hackable:1

GNOME Mobile distribution for Hackable Devices

FSOSHRUDCON '09

Pierre Pronchery

bearstech

Who's talking

- Working for Bearstech
- Embedded software development
- Hackable Devices R&D
- Small Free Software services company
- Cooperative of hackers
- Distributed over France and Germany
- Meet us at La Cantine in Paris!

Genesis of hackable:1, part one

Back in July 2008...

- ASU was not out yet
- SHR would mix 2007.2 and ASU FTW
- Android was just hype
- Debian just arrived

Genesis of hackable:1, part two

Our point of view:

- Complete 2007.2 which looked closer to anything functional
- Benefit from Debian: packages, know-how...
- Already familiar with Gtk+ and GNOME

More context: hackable:Devices

- Hardware with open specifications
- The obvious example: Openmoko Freerunner :)
- More devices:
  - BUGlabs
  - Elphel cameras
  - Linksys WRT54G...
- But more on that later!

Our first releases

- Archives of a customized Debian system
  - Four are out so far (the last being based on lenny instead of sid)
- Pre-installed and configured for embedded software development:
  - For the Openmoko only
  - Fits only on MicroSD
  - Native compilation (with distcc support)

From now on

- Generate images automatically
- Support more hardware
- Fulfill more purposes
- Ease embedded development and quick prototyping

Project with two sides

- Packaging software
  - Original Debian packages
  - External repositories (Om2007.2)
  - Internal repository
- Building images
  - From package binaries
  - For given hardware platforms and purpose combinations

But how does this work?

Packaging software

- Packages database in *trunk/packages/packages.txt*
- Some helper scripts provided for maintenance
- Binaries are uploaded twice a day by the build bot, currently:
  - Amd64 (native)
  - I386 (native)
  - ARM (cross-compiled)

Package database

```
tar  pkg-config          0.23  2
  http://pkgconfig.freedesktop.org/releases/pkg-
  config-0.23.tar.gz

svn  libmokoui2          0.3   1
  openmoko/libraries/libmokoui2      4878

tar  evolution-data-server-dbus    2.20.0   2
  ftp://ftp.gnome.org/pub/gnome/sources/evolution-data-
  server-dbus/2.20/evolution-data-server-
  dbus-2.20.0.tar.gz

src  wifig             0.1   1
  applications/wifig           640

meta hackable1           0.1   1
        581
```

Where the software comes from

Different sources for packages:

- **svn**: external SVN repository (Om2007.2)
- **src**: internal SVN repository (hackable:1 specific, backports...)
- **tar**: formal releases of given projects (evolution-data-server-dbus)
- **meta**: just meta-packages

Building images

- Needed a replacement for debootstrap
- Came up with *trunk/build/build.sh*
- Highly portable:
  - Less than 1000 lines of POSIX shell script
  - Works from *BSD too
- Generates images in 20 minutes
- Some issues: dependencies, footprint...
- Whitepaper in progress

How to build images

```
Usage: build.sh [option=value...] target...
Targets:
  archive       Create an archive of an image contents
  clean         Remove temporary data
  config        Configure installed packages
  help          Display this help screen
  image         Create a native image file
  list          List the available profiles
Options:
  VENDOR        (Openmoko)
  MODEL         (Freerunner)
  PURPOSE       (user)
```

The build process

- Desired profile sourced from *trunk/build/profiles* (packages list, platform name...)
- Dependencies are resolved and cached
- Packages are downloaded and extracted
- Packages database filled in on the fly
- Configuration sourced from *trunk/build/packages*
- Data purged as necessary (locales...)
- The final image is generated (jffs2...)

Supported platforms

Sadly, too few currently:

```
$ ./build.sh list
Listing available profiles:
Openmoko-Freerunner-cross
Openmoko-Freerunner-developer
Openmoko-Freerunner-user
Openmoko-Neo1973-developer
Openmoko-Neo1973-user
PC-Freerunner-user
PC-Neo1973-user
```

A particular case: cross-compiler

- The build process is able to generate to complete cross-compiling environment:
  - Choice of the native platform
  - Installs the relevant cross-compiler from the Emdebian project
  - Choice of the target platform
- Runs out of the box:
  - As a chroot environment on Linux boxes
  - As a virtual machine from other OSes

The hackable:Devices initiative

- Will be announced more formally within the next three months, but:
  - A website, www.hackable-devices.com
  - Gather makers, developers, users and consumers of hackable devices
  - Monthly fee for hobbyists for flat rate assistance: hacker-for-rent, spare parts...
  - Provide a shop and shipping backend for hackcenters, hobbyist shops...
  - Quick prototyping for businesses

The website

- As soon as possible:
  - Document devices and hacks
  - Vote on hardware and content
- Mid-term:
  - Provide direct links to buy hardware
- Long term:
  - Offer a shop backend and assistance
  - Provide the hackable:Devices community brand

The subscription

- Revive the hobbyist spirit
- Have regular shops host workshops
- Void the concept of warranty:
  - Repair or get your hardware repaired at the shop whatever happened
  - Get spare parts easily
- Involve hackcenters and help them to be started and grow

The business model

For us it's about:

- Gaining experience on embedded development
- Lower the bar for hardware and software access
- Be able to deliver quick and functional prototypes

Current and future focuses

Besides making it work, two R&D projects:

- Multi-homing: *(ongoing)*
  - Provide seamless data+voice communication roaming between GSM, Wifi, WiMax...
- Secure phone: *(pending)*
  - Encrypt voice communications from caller to callee
  - Provide transparent, confidential access to resources at home or work

And of course...

- We're looking for more hackable devices!
- We're working on a hackable Digital Picture Frame:
  - Cheap
  - ubiquitous

Final goal

At times where:

- Internet is being filtered or censored,
- Physical limitations are enforced in software (DRM),
- Technology is abused to confuse users while it may actually get simpler,

We want to put the user back in control.

Wanna join?

- In our opinion much easier to get started than OpenEmbedded, Scratchbox...
- Open development with decisions made
- We hang out on `#hackable1` on Freenode
- Official website on http://www.hackable1.org/
- Trac and wiki on http://trac.hackable1.org/
- Improvized workshop? Come and see me!