

# What's up on Haiku?

*R1/beta2, packaging, porting and contributing.*

François Revol  
[revol@free.fr](mailto:revol@free.fr)



# Haiku?

- Free Software Operating System
- Inspired by the BeOS
- Our own kernel
- Our on GUI
- Started 2001
- R1/beta1 2018



# Official platforms

- x86\_gcc2h (32bit) : Recommended
  - GCC2h : GCC2.95 hybrid + GCC3 er, 7, er, 8
  - Binary compatible with BeOS R5
- x86 (32bit)
  - GCC8 only, no BeOS R5 compat
- x86\_64
  - Almost official, good support



# Exotic platforms

- PowerPC
  - Still not finished
- ARM still in the works
  - Lot of demand (Raspberry Pi...)
- M68k barely started
- SPARC64 started by PulkoMandy
- RISC-V just started



# Nightly Builds

- Haiku & HaikuPorts repositories
  - buildbots
- **Online** updates
- Transactional updates
  - You can boot previous packages states



# R1/beta2

- Real Soon Now™
- Infrastructure makes it easier now
- R1/beta1 DVDs are still available
  - And you can [upgrade](#) after the install.



# We started optimizing

- Boot with less than 256MB of RAM
  - A long time ago it used to be 40MB...
  - But Installer needs more
- Various improvements
  - Thread synchronization
  - Memory allocation
  - We tried *rpmalloc*... back to *Hoard2* for now



# And cleaning up

- Switch to musl libc for math on new platforms





# Security

- Area cloning limitations
- **SMAP + SMEP**
- Lot of fixes thanks to **PVS-Studio**
- Still a lot to do
  - (we're cloning an OS which ran all apps as root )



# More drivers!

- xHCI (USB3) stabilized
- UEFI support integrated
- NVMe SSD support
- PS/2 fixes (trackpad...)
- Oh, and we still support !



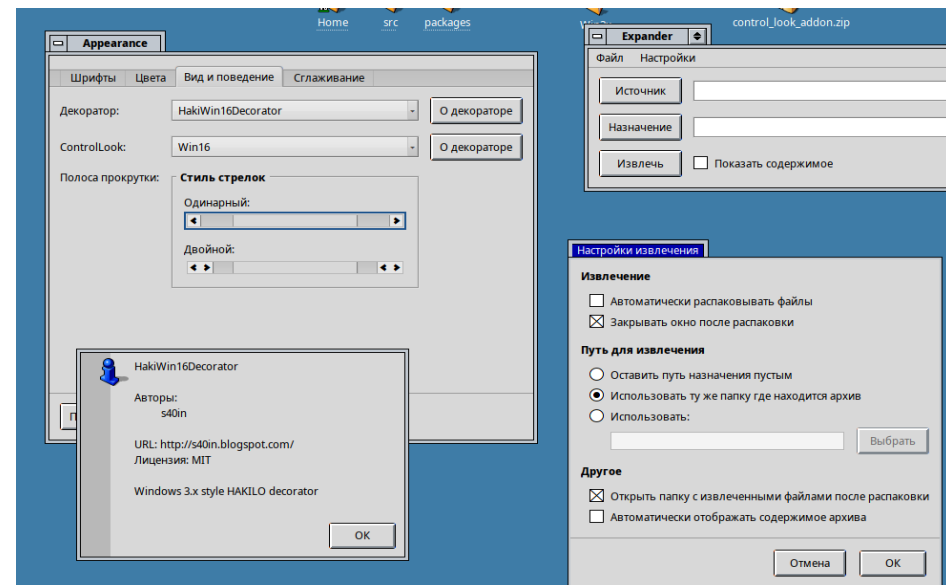
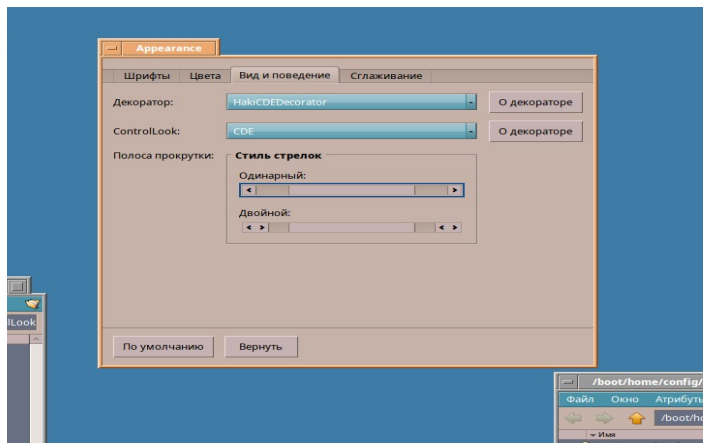
# HDPI

- Better adaptation to very high resolutions
- GUI layout scales with the system font size
- But, we already had vector icons
  - Custom HVIF format allows storing them in inodes



# Theming

- BControlLook class can be replaced
  - HakiControlLook...



# Packaging, before

- BeOS
  - ZIP files, unzip anywhere
  - Or PKG files with click-o-drome
  - No dependency management
- Haiku
  - `installoptionalpackage` (mostly `wget+unzip` \$1)



# Packaging, now

- /system/packages/
  - \*.hpkg
  - administrative/
    - active-packages
    - state\_`\${date}`/
    - writeable-files/
- Installing Haiku = copy 200 files
- Updating = download 15 files



# packagefs

- Mounted on `/system/` and `~/config/`
- Union of active packages contents
- Shine-through directories for manual install
  - `.../non-packaged/` (`=/usr/local`)



# HaikuPorter

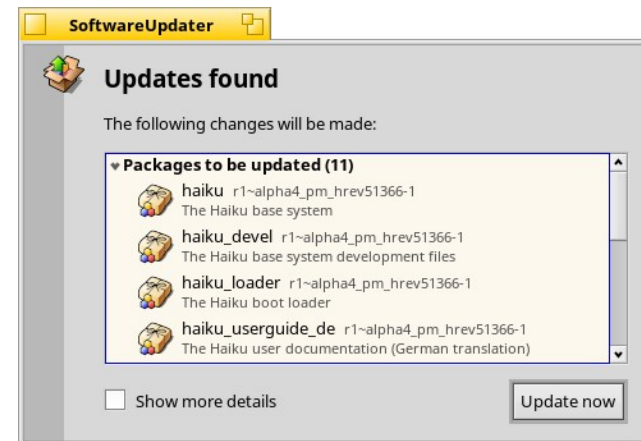
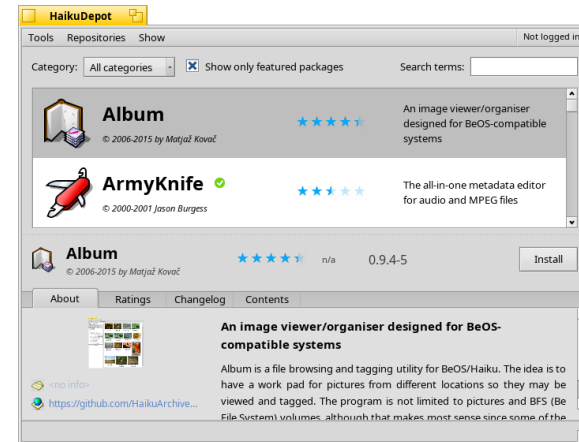
- Python
- Package names from Gentoo
- Builds recipes in a chroot with deps
- HaikuPorts on GitHub has them





# Interface

- GUI
  - HaikuDepot
  - SoftwareUpdater
- pkgman



# Challenges

- Since R1/beta1: Packaging
  - Almost reproducible build
  - But requires strict dependencies
- Haiku needs Haiku to build
- Bootstrap builds = easy to break
  - HaikuPorts-cross not well maintained...
- Config files & list-of-installed-by-{pip,...}



# The good

- Updates are sooooo fast
- `catattr SYS:PACKAGE /foo/bar`
  - Tells you which package has it
  - You know it didn't change
- You can blacklist files in packages
- You can boot previous states if it breaks



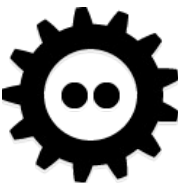
# Problems

- Stage1 loader locates haiku\_loader.hpkg
  - If this one breaks...
  - Happened last week on nightly
- Failed updates must be redownloaded
- Uses more memory & CPU
  - We started benchmarking
  - We used to cache both compressed & uncompressed content



# Numbers

- Published packages (x86\_gcc2)
  - 4298 on HaikuPorts
  - 1511 in the 3 other repos (Clasqm, BeSly, FatElk)
- On my dev VM
  - `$ ls /system/packages/*.hpkg | wc -l`  
689



# So you want to port your app?



# POSIX...ish

- We never claimed full POSIX compatibility
- But many Linux folks use syscalls as if...
  - And forget to check in configure.ac
  - And then complain when we add our calls with proper checks
- “You don’t have inotify|whatever”
  - BeOS did it 20 years ago, differently
  - Inotify isn’t POSIX either
- It’s usually ok to have fallbacks



# Modularize

- If your code is clean you shouldn't need a lot of `#ifdefery`
- Subclass implementation per platform
- Don't insist on extra stuff
  - Added value vs dependency hell
- It's usually ok to have fallbacks
  - Did I say that already?





# Example: qtkeychain

- PR just got merged

- Commit 1

```
- #if defined(Q_OS_UNIX) && !defined(Q_OS_DARWIN) && !defined(Q_OS_ANDROID)  
+ #if defined(KEYCHAIN_DBUS)
```

- Commit 2

- CmakeList.txt: changed 17 lines
- keychain\_haiku.cpp

- That's it.



# DON'T:

- `return -EF00;`
  - We have them negative already
  - For BeOS compat, & POSIX couldn't **make their mind** anyway back then
  - Best is to `#define myerr(e)` depending on `$OS`

## Issue 6

The following new requirements on POSIX implementations derive from alignment with the Single

- The majority of the error conditions previously marked as extensions are now mandatory, e:

Values for `errno` are **now required to be distinct positive values** rather than non-zero values. This c



# Build tools

- CMake, meson...
- Autotools
  - don't forget to AC\_CHECK\_LIBS, AX\_PTHREAD...
    - POSIX doesn't tell which lib has what anyway



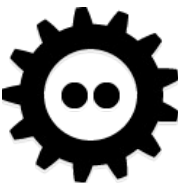
# Compiler

- Official target `x86_gcc2`
- Hybrid: `gcc2 + gcc8`
  - Like Linux' `/lib + /lib64`
- GCC2 (so C89) required for system parts
- Otherwise it's ok with latest



# X11

- Well, cleanup first?
- Conky: I started to **cleanup X11 everywhere**
- Wayland users will thank you



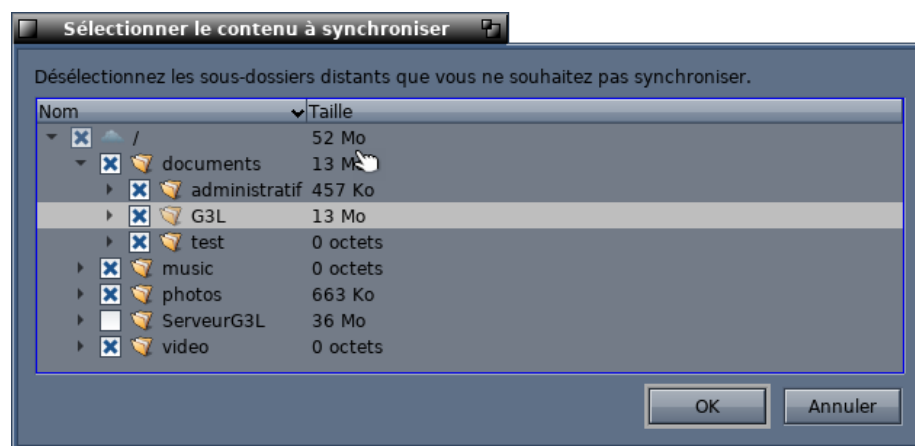
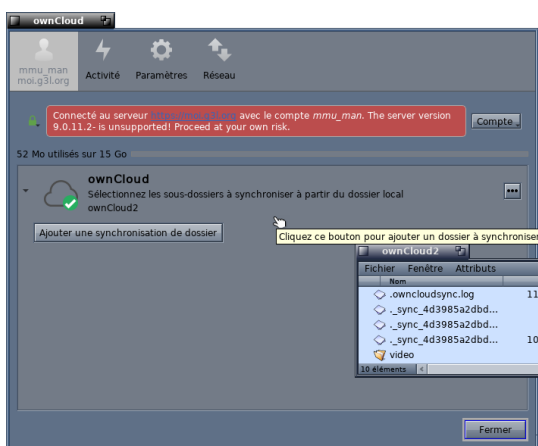
# Qt[4,5]

- Keep Qt5X11Extras optional, Wayland users will love you too
- Qt is quite well done
- Haiku plugin gives native look (& feel mostly)
- OpenGL not working yet in Qt



# Example: {own,next}cloud client

- OwnCloud
  - QtWebEngine optional
- Port already running
- folderwatcher\_haiku
- NextCloud
  - QtWebEngine req'd
    - it's a Chromium fork
- Would take years to port



# Fix your toolkit

- LWN: [The Platform Problem](#)
- You are entitled to sending patches instead of working around missing features





# Contributing

- Port apps
- Write code
- Translations
  - System
  - Apps
- Design
  - Icons...
- Advocacy...



# GSoC + GCI + Outreachy

- More write support on btrfs
- WebKit updates...
- Potential new contributors



# I want to help! Where do I start?

- [HaikuPorts wiki](#)
- [www.haiku-os.org/ .../getting-started](http://www.haiku-os.org/.../getting-started)
- [dev. .../EasyTasks](#)
- [dev. .../SubmittingPatches !](#)
  
- [Donate](#) to Haiku, Inc.
  
- Questions?

