Adding GNU/Hurd support to GNU Guix

Porting GNU Guix to a new platform

Manolis Ragkousis
manolis837@gmail.com

FOSDEM
30 January 2016

What is distro bootstrapping?

Simply speaking

- Bootstrapping refers to the process of getting the distribution built “from nothing”.

But how does the first package gets built?

How is it usually done?

- Distros, like Debian or Arch Linux, use whatever is already present in the system.
- There’s no clear notion of “bootstrap binaries”.
- It's hard to track down the origins of a port (and reproduce it).

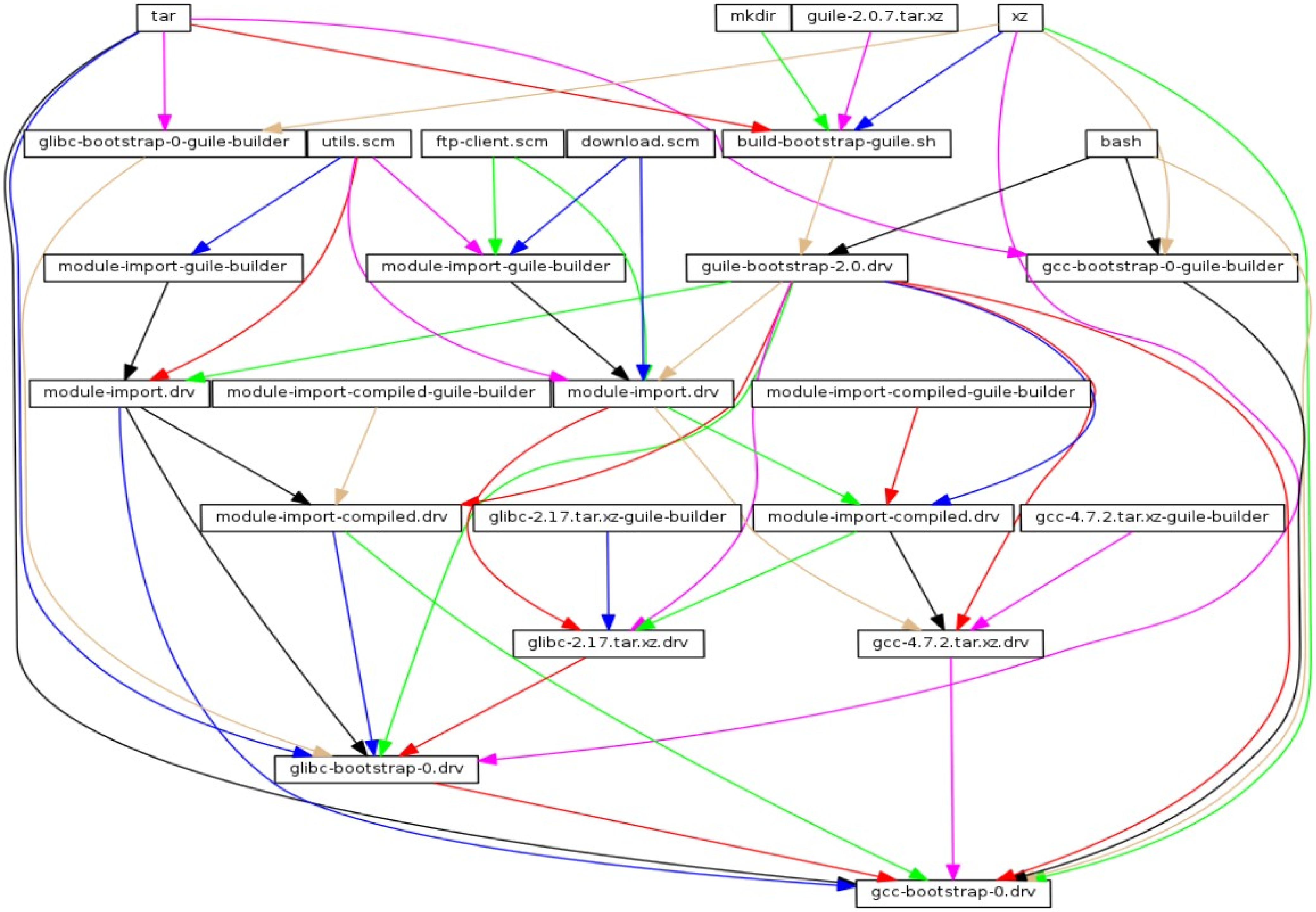
Guix

Bootstrapping Guix

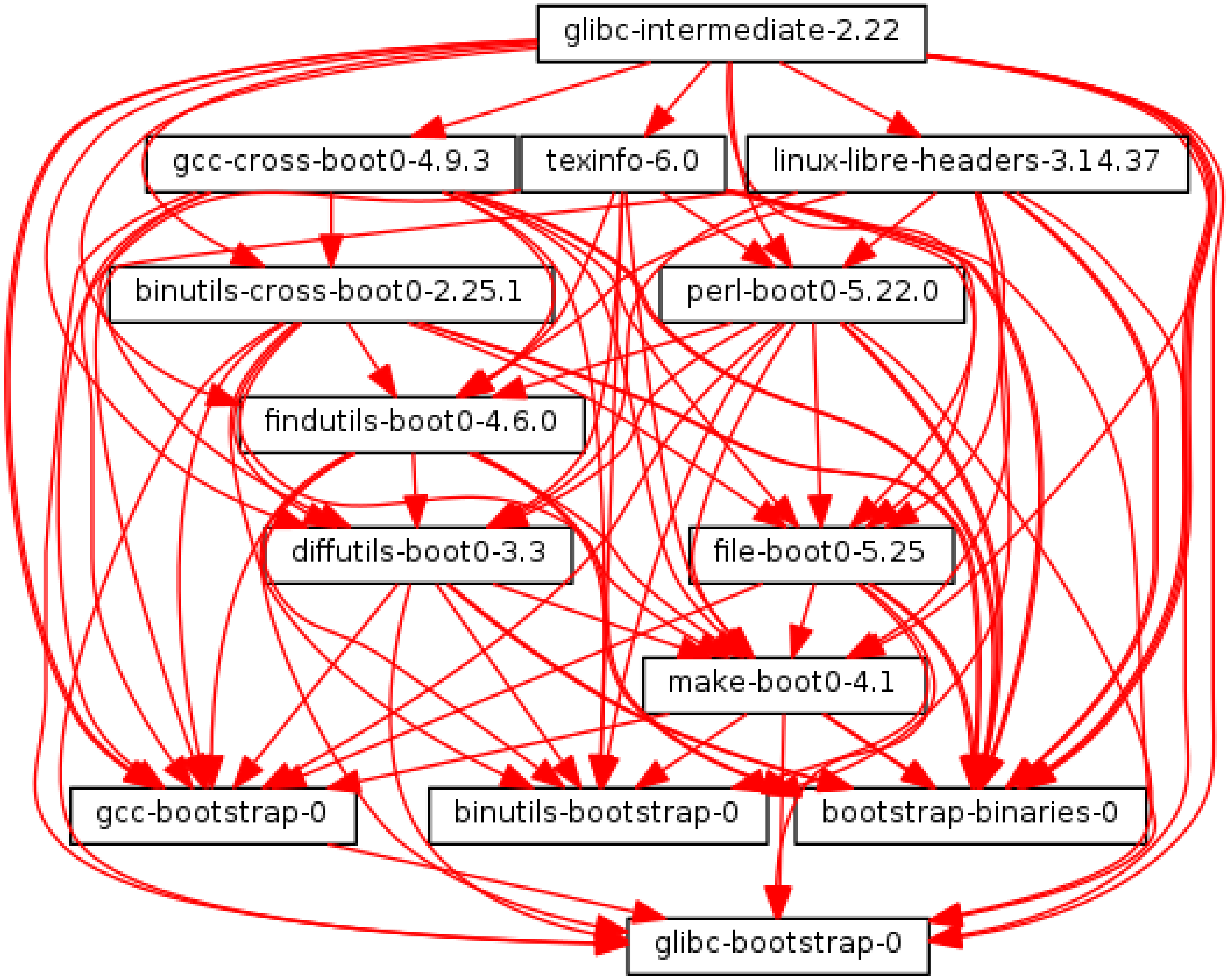
Guix relies on a clearly defined set of pre-built binaries

- Guile
- GCC
- Binutils
- Libc
- Bootstrap binaries (Bourne shell, Coreutils, Awk, Finutils, sed and grep)

Bootstrapping is complete when we have a full tool chain that does not depend on the pre-built bootstrap tools


tar
mkdir
guile-2.0.7.tar.xz
xz
glibc-bootstrap-0-guile-builder
utils.scm
ftp-client.scm
download.scm
build-bootstrap-guile.sh
bash
module-import-guile-builder
module-import-guile-builder
guile-bootstrap-2.0.drv
gcc-bootstrap-0-guile-builder
module-import.drv
module-import-compiled-guile-builder
module-import.drv
module-import-compiled-guile-builder
module-import-compiled.drv
glibc-2.17.tar.xz-guile-builder
module-import-compiled.drv
gcc-4.7.2.tar.xz-guile-builder
glibc-2.17.tar.xz.drv
gcc-4.7.2.tar.xz.drv
glibc-bootstrap-0.drv
gcc-bootstrap-0.drv

Bootstrapping the toolchain



Porting to a new Platform

1) Cross-build the bootstrap binaries:

```
guix build –-target=i585-pc-gnu bootstrap-binaries
```

2) “Inject” the bootstrap binaries in the package DAG for that platform.

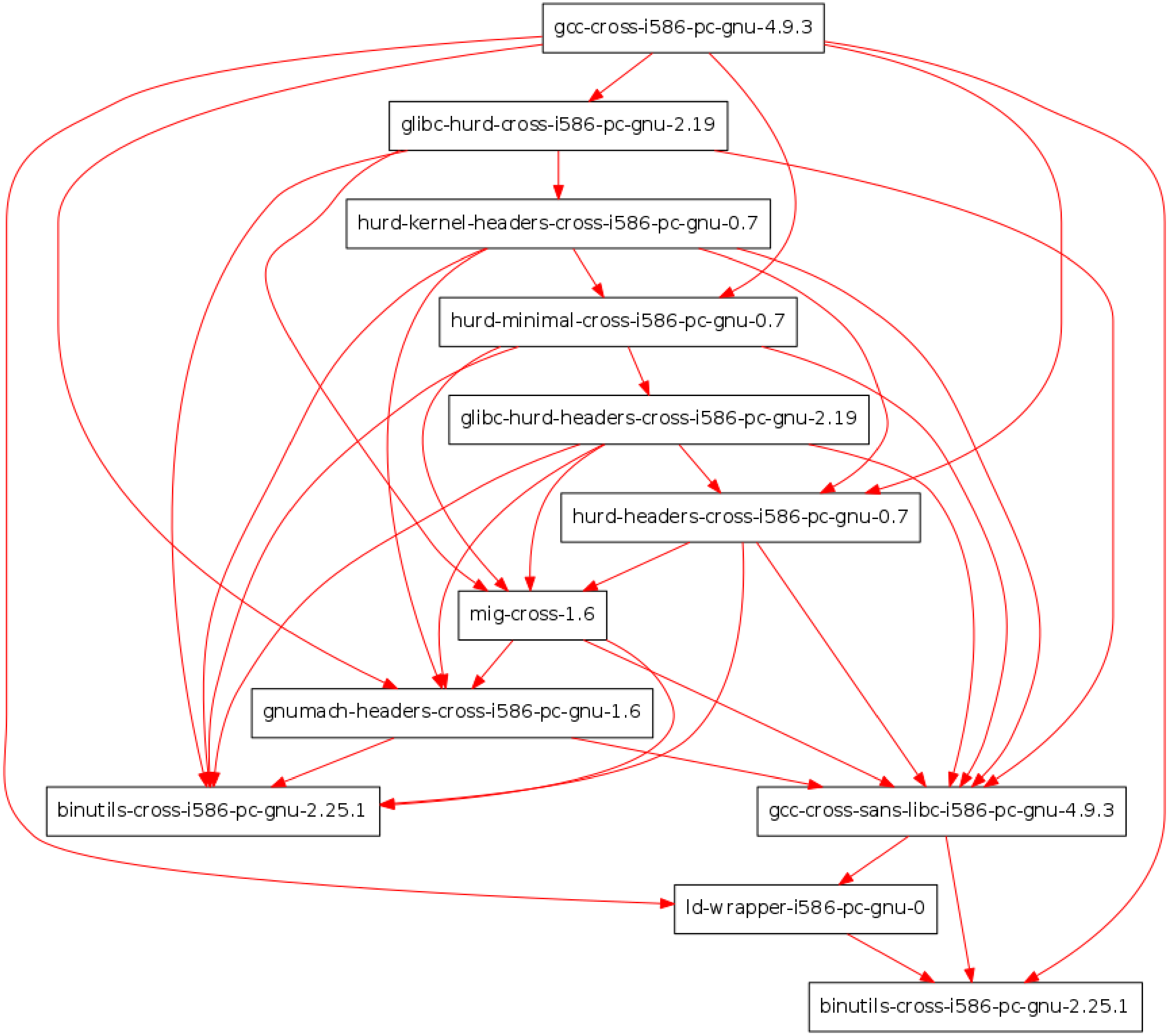
   (the `(gnu packages bootstrap)` module)

3) Build Guix on a running OS of that platform and be done!!

Well it wasn't so simple..

What's special about the Hurd

- The Hurd is a collection of servers that run on top of the Mach microkernel.
- These servers implement features that are normally implemented by the kernel.
- Through glibc the same standard interfaces known from other UNIX-like systems are provided, so usually, compiling higher level programs is essentially transparent.


gcc-cross-i586-pc-gnu-4.9.3
glibc-hurd-cross-i586-pc-gnu-2.19
hurd-kernel-headers-cross-i586-pc-gnu-0.7
hurd-minimal-cross-i586-pc-gnu-0.7
glibc-hurd-headers-cross-i586-pc-gnu-2.19
hurd-headers-cross-i586-pc-gnu-0.7
mig-cross-1.6
gnumach-headers-cross-i586-pc-gnu-1.6
binutils-cross-i586-pc-gnu-2.25.1
gcc-cross-sans-libc-i586-pc-gnu-4.9.3
ld-wrapper-i586-pc-gnu-0
binutils-cross-i586-pc-gnu-2.25.1

Building the bootstrap binaries.

Produce the system specific static binaries.

1) Build static bash, coreutils, xz, tar, etc. (static-binaries)

2) Build binutils

3) Build glibc

4) Build gcc

5) Build guile

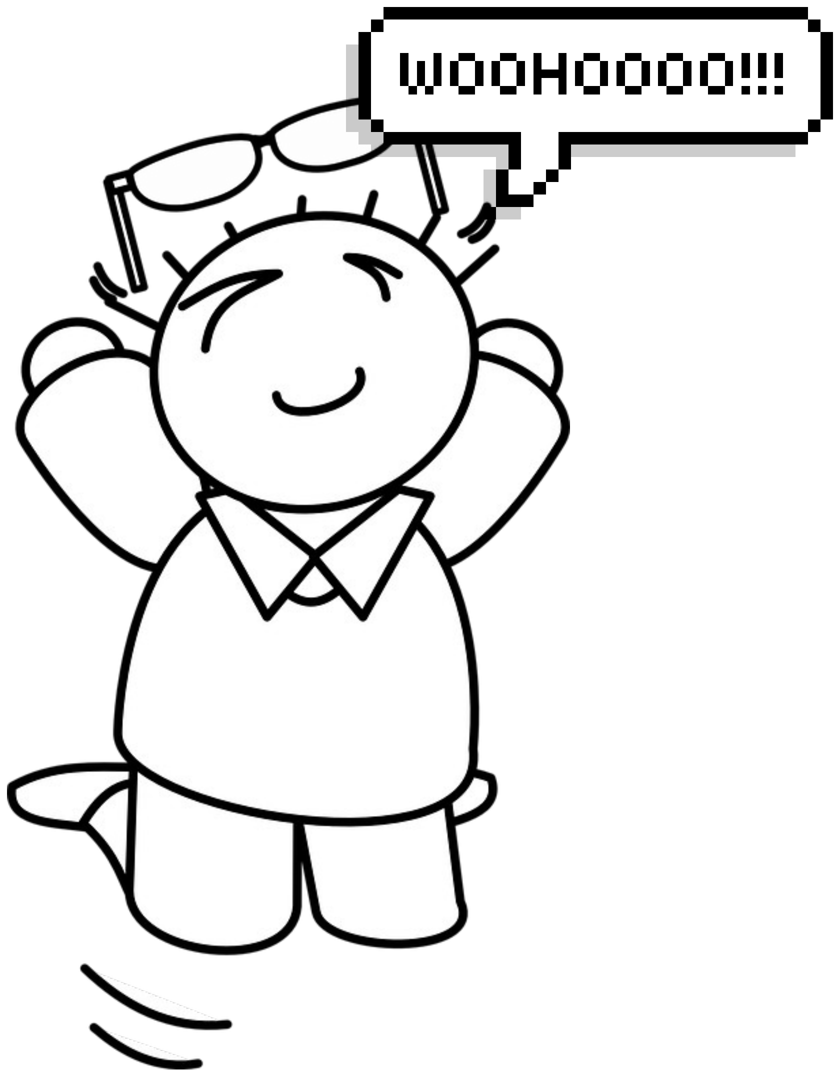
6) Create the tarballs

This process is described in (gnu packages make-bootstrap) and can be achieved with:

```
guix build –-target=i586-pc-gnu bootstrap-tarballs
```

Building the bootstrap binaries.

- PATH_MAX missing (acl, patch, sed, tar, etc..).
- Building for i686-gnu would produce static binaries that failed to run on a Hurd system. Started using i586-gnu instead.
- Two different glibc packages (`glibc/linux, glibc/hurd`). Created `glibc-for-target` to handle it and modified the `(gnu packages make-bootstrap)` module to produce the correct glibc-tarball.


WOOHOOOO!!!

Using the binaries on the new platform

1. Update `(gnu packages bootstrap)` with information on the new binaries and where to download them.
2. Add rules to the `gnu-system.am` file on how to handle the binaries.
3. Clone Guix on the new platform.
4. Run `./configure –-with-courage && make && make install`
5. Begin building!

Well not quite there yet..

Updating Guix with the new tarballs

- Create the i586-gnu directory containing guile, bash, mkdir, tar and xz.
- Update `(gnu packages bootstrap)` with the bootstrap tarball hashes and where to download them.
- Create the bootstrap dir entries in gnu-system.am so Guix can know where to find guile, bash, mkdir, tar and xz.

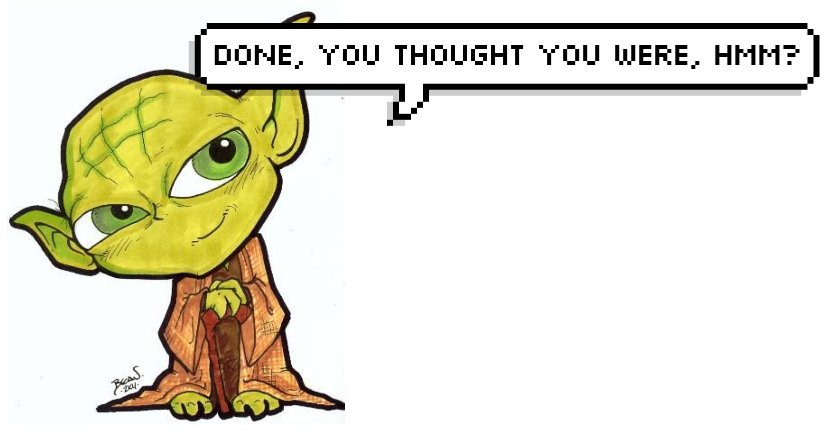
Now we are ready for the real fun.

Building Guix on GNU/Hurd

- Step 1: Run `` `./configure --with-system=i586-gnu --with-courage` ``.
- Step 2: Run `` `make` ``.
- Step 3: Create the build users and run the daemon.
- Step 4: Start building.

Building packages with Guix on GNU/Hurd

- Discovered that glibc wasn't taking into account the `“--with-headers”` argument.
- Perl could not be build because of a problem with memmove in Hurd's glibc.
- Binaries produced from `gcc-boot0` had a problematic RUNPATH. Solved with `ld-wrapper-boot0`.
- `glibc-final`'s debug output refers to `%glibc-bootstrap` while it shouldn't.



Current status

- We can cross-build to GNU/Hurd with just:

```
guix build --target=i586-pc-gnu foo
```

- 19 patches are pending integration (and more to come)
- Branch ‘wip-hurd’ can be used on a running GNU/Hurd system and it can build all the packages till the “`%final-inputs`”.
- guix-daemon cannot perform fully isolated (chroot/container) builds like it does on GNU/Linux.

Roadmap

- Port container-style features in guix-daemon
  - Hurd firmlinks instead of Linux bind-mounts
  - Sub-hurds instead of Linux name spaces
- Port GuixSD
  - Isolate Linuxisms
  - Package GNU Mach/Hurd kernel

Work in progress

- Make configure detect the correct system.
- Make the `(guix build syscalls)` module work around the not-present syscalls.
- Move Hurd's `mount()` implementation to glibc.

Thanks for mentoring, suggestions, code, debugging and patience..

- Ludovic Courtès, Samuel Thibault my two GSoC mentors for their valuable help, patience and trust!
- Marek Benc, Richard Braun, Thomas Schwinge, Mark Weaver, Justus Winter and everyone from Guix and Hurd for helping me in any way they can :-)
- This work was part of my GSoC 2015 project so I would like to thank Jose E. Marchesi for taking care of everything for the GNU Project participation and Carol Smith + Google for organizing GSoC.
- ATEI of Crete and professor George Kornaros for allowing me to travel to Brussels.

Credits

- GNU Guix logo, GFDL, http://gnu.org/s/guix/graphics
- Yoda image, http://hedbonstudios.deviantart.com/art/Chibi-Yoda-76015441
- comic speech bubble, http://wigflip.com/ds/
- cartoon person, https://pixabay.com/en/specman-man-cartoon-person-jumping-161928/