# Bootstrappable builds

Ricardo Wurmus

September 18, 2017

# Recipe for yoghurt:

add yoghurt to milk

- "yoghurt software" undermines software freedom
- ignore: use existing binaries to build new binaries
- retrace history, save old tools from bit rot
- write new alternative language implementations

# Java

- JDK is written in Java
- GCJ needs ECJ (Java), bundles pre-compiled GNU Classpath :-(

## JDK bootstrap

1. Jikes (C++) -> SableVM (Java + C)
2. old Ant (Java) -> old ECJ (Java)
3. ECJ -> GNU Classpath 0.99 -> JamVM
4. JamVM -> GNU Classpath devel (for Java ~1.6) -> JamVM'
5. OpenJDK via IcedTea 1.x -> 2.x -> 3.x

## Java libraries

- cultural problem: download/bundle pre-built packages
- only leaf nodes are built from source
- dependency cycles are common
- cannot use modern build tools

# Haskell

- All versions of GHC are written in Haskell
- need GHC `(n-1)` to build GHC `n`
- history is littered with defunct Haskell systems: nhc98, Yale Haskell, Hugs
- failed: use Hugs to build nhc98 to build GHC see: `https://elephly.net/posts/2017-01-09-bootstrapping-haskell-part-1.html`
- next: use Hugs interpreter to build patched GHC directly
- future: revive and extend Yale Haskell?

# The elephant in the room. . .

[width=.9]./elephly

# GCC

- needs C++ compiler since version 4.7
- is there a path from reasonably auditable source to GCC?

# Stage0 and Mes

## Stage0

`https://git.savannah.nongnu.org/git/stage0.git`

- `hex.0`: self-hosting hex assembler that we consider to be source ($<$ 300 bytes)
- M0 macro assembler written in .0
- M1 macro assembler written in M0
- a hex2 linker written in M0

## Mes

`https://gitlab.com/janneke/mes`

- mes.c: a scheme interpreter prototyped in C ~1400 Lines
- mescc.scm: a C compiler written in Scheme (uses Nyacc C99 parser in Scheme)
- mes.M1: this scheme interpreter in annotated M1 assembly

# Current status

- stage0: hex.0, M0 done; M1, `hex2_linker` prototyped in C
- tcc compiled with mescc correctly compiles: `int main () {return 42;}`
- mes+mescc.scm are mutually self-hosting
- during development we run mescc.scm on Guile (mes is slooowww)
- tcc compiled with GCC is known to compile GCC

# Open issues

- fix mescc.scm so that tcc can correctly compile GCC
- fix bootstrap-loops: (Nyacc?, mes.M1?, psyntax.pp?)
- make GCC bootstrappable again, remove [need for] tcc stage?
- stage1/2 LISP, FORTH?
- integrate with GuixSD
- $x86_{64}$, arm?

# Contact

- `#bootstrappable`, `#guix` on freenode
- `http://bootstrappable.org`
- `mailto:bootstrappable@freelists.org`