Practical, verifiable software freedom with GuixSD

David Thompson

Sunday, March 25th, 2018

about me

GNU Guix contributor since 2013

GNU Guile user and contributor since 2012

Day job: DevOps (AWS, Ruby)

the four freedoms

0: The freedom to run the program as you wish, for any purpose

the four freedoms

1: The freedom to study how the program works, and change it so it does your computing as you wish

the four freedoms

2: The freedom to redistribute copies so you can help your neighbor

the four freedoms

3: The freedom to distribute copies of your modified versions to others

the four freedoms

a wonderful set of rights, but often difficult to exercise in practice

common issues

figuring out how to view the exact source for a running program is tricky

- ??

common issues

building from source is difficult or sometimes impossible

- non-standard build system
- build scripts make assumptions that aren't true for your system

common issues

sharing source or binaries has many pitfalls

- dependency hell
- incompatible libraries between systems
- high barrier to entry for common package managers

freedom: embedded

GuixSD removes many of the common barriers that prevent users from exercising their four freedoms

what is guixsd?

<logo>

fully-free GNU/Linux distribution with an advanced package manager and system upgrade mechanism

what is guix?

GuixSD's package manager

"functional" package manager

atomic updates and rollbacks

unprivileged package management

Users can build and install software without root privileges

unprivileged package management

Each user may have one or more “profiles”, a union of many packages.

Use cases:

- Alyssa and Ben use different versions of Emacs
- Alyssa hacks on 2 Ruby projects that require different versions

experiment without fear

```
guix package --roll-back

guix system reconfigure
```

inspecting source code

```
guix build --source
```

inspecting dependency graph

```
guix graph
```

<small dependency graph image>

sharing system configurations

<minimal code example here>

sharing packages

```
guix build -L ~/daves-packages foo
```

<minimal package example>

sharing development environments

`guix environment`

<minimal environment file example>

sharing binaries

Use `guix publish` to spawn a web server that allows other Guix users to download the software you've already built

```
guix build \
     --substitute-urls=http://guix.dthompson.us:8080 \
     hello
```

reproducible builds

did anyone attend "You think you're not a target? A tale of three developers. . . " by Chris Lamb yesterday?

```
guix build --rounds=N
```

verifying binaries

```
guix challenge
```

customize packages

```
guix edit
```

```
guix build
```

interoperate with other systems

```
guix pack
```

literally: embedded

GuixSD now runs on the Beaglebone Black single-board computer!

the freedom to contribute

GNU Guix is a welcoming community:

we have a code of conduct and enforce it

we have started seeking new contributors via Outreachy

we participate in Google Summer of Code every year

oh, and no copyright assignment (in case you were wondering)

thanks!

docs, past talks, source code, mailing list/IRC info, etc.:

https://gnu.org/s/guix

credits