Write a Bourne-shell compiler front-end for Guile

Efraim Flashner
efraim@flashner.co.il

Who am I?

My name is Efraim, I'm currently living in Jerusalem, Israel. I'm a 3rd year Computer Science student at the Jerusalem College of Technology (Israel has the college/university split). One of my classes was on functional programming, where we covered prolog and lisp. I found the idea of functional programming interesting, and at the time even tried writing small programs using them.

I switched full-time to Linux in 2010, trying out Ubuntu and then switching to Debian, and currently I'm waiting on a blob-free wireless mini-pcie/usb before switching my laptop over to GuixSD. I haven't had a lot of opportunity in Israel to meet up with Free Software or Linux people, and I've worked hard to make up for that by reading lots of online tutorials and experimenting on my machines. I first discovered Guix around June of last year and got involved shortly after.

The Plan:

As it currently stands, looking in gnu/packages/bootstrap/$ARCH/, in addition to guile there's bash, mkdir, tar and xz. From the manual we have the bootstrapping methodology (https://www.gnu.org/software/guix/manual/html_node/Bootstrapping.html) which also uses gcc, binutils, coreutils, libc, etc. We also have guix/build/bournish.scm to act as a rescue-shell of sorts and in use in the initrd.

By extending/building upon bournish we can lessen the number of bootstrap binaries that we need to bootstrap a working system AND we get a more powerful shell to use when something™ happens during boot and we're dropped down to a rescue shell.

Currently the idea on the GSoC idea wiki is to extend bournish so that it can cover everything needed by a configure script. This would of course allow us to decrease the number of binaries we need to get started. Looking at the list of bootstrap binaries, I'm most interested in including enough functionality to remove our dependance on bash which to me seems like a good security idea, as surely we wouldn't want to have a vulnerability found for our bootstrap version of bash.

The Plan, Part 2:

From now until the start of GSoC I'd work on bettering my knowledge of guile and more closely analyzing the bootstrap process. I've had a look at Wikipedia's entry on Unix shells and I'll be grabbing the source of a few of them, including ash, scsh and busybox, for inspiration.

For the coding I plan on implementing the different features that we'd need for replacing a number of the bootstrap binaries. As a stretch-goal I'd like to see if it is possible to switch to using it as my user shell in place of bash. Part of this would of course come with deciding on how to use it to replace bootstrap binaries; whether to have additional binary outputs for the different binaries being replaced or to have one monolithic binary.