

GSoC 2013 Proposal - Bootstrappable Debian

Gustavo P. Alkmim

May 3, 2013

1 Name

Gustavo Prado Alkmim

2 Contact/Email

Email: alkmim@ic.unicamp.br

IRC: alkmim (#debian-bootstrap)

3 Background

- **General:** In 2009, I graduated in Computer Science at the Federal University of Lavras (UFLA/Brazil) in which I worked with live-migration of virtual machines using Xen and iSCSI. I was a member of Tecnolivre, a cooperative specialized on open-source based solutions []. In 2012, I finished my master at State University of Campinas (Unicamp/Brazil), working with Virtual Network Mapping. Now I'm finishing the first year of my doctorate at the same university working with Mobile Cloud Computing.
- **Linux and Programming:** I started using Linux very early during my undergraduate studies. I am one of the administrators of the LRC (Network Computer Laboratory) at Computer Institute, also in Unicamp, in which we use just Debian and Ubuntu Linux systems. I have advanced skills on C++ and shell script and basic skills on Python and Java. If necessary, I have no problem starting to learn another programming language. I also had a good foundation in functional languages in my undergraduate studies using Haskell.

- **Experience with the proposed project:** I worked with the bootstrap problem on Debian at GSoC 2011. So, I have some knowledge on how to solve cycles, how to crossbuild package and how to hack the packages in order to solve cross-build and cyclic dependencies problems. Also I learned about the staged-build process, which will be necessary to complete the project. As a result of GSoC 2011, I started hacking on Debian and the results can be found here []. Last year I had a good experience on patching a crosstoolchain from Ubuntu to work in Debian. The built packages may be found here []. This year, while preparing for GSoC I started learning Botch and improved my skills on how to build packages with reduced dependencies. As a result of this study I got some patches for building packages with reduced dependencies using the Build Profiles. These patches can be found here [].

4 Project title

Bootstrappable Debian

5 Project details

5.1 Introduction

Every time a new architecture is developed, the Debian Linux needs to be ported to be used on it. Therefore, porting any operational system to a new architecture is always a difficult task since the bootstrap needs to be done 'from scratch'. While doing the first bootstrap of an architecture some challenges need to be faced. In order to do the bootstrap to a new architecture we need to have packages built that can run in the new architecture. This can be done by using multiarch or cross-built packages. Therefore we don't need to have all packages of system crossbuilt or multiarch. We just need to have a minimal build system that is sufficient to satisfy the build-dependencies of the rest of the packages.

Meanwhile, this raises another challenge that is the cyclic build-dependencies. In order to build/crossbuild packages that have cyclic build-dependencies we need to break the cycle(s) avoiding one or more build-dependencies. This is also a very difficult task because patches need to be applied. Another problem that raises when doing the bootstrap of a new architecture is while building packages inside the minimal build system. Despite all the effort in

trying to crossbuild and have multiarch versions of all packages needed to solve the build-dependencies of the rest of the packages, some of them are very hard to be crossbuilt. So, we need to decide whether we should build the packages inside the minimal system or crossbuild them.

For previous ports, people tended to use non-Debian tools (such as Yocto, Gentoo and OpenEmbedded) to get a basic rootfs image of the target arch then do native building within that. Currently, Debian is improving its toolchain and patches for packages in order to enable the bootstrap of a Debian system using just a Debian system, instead of other OS. But a lot of work must be conclude before the bootstrap becomes easy to be done for non-expert in hacking Debian packages being able to bootstrap a Debian system. Every time someone needs to do a Debian bootstrap to a new port he needs to patch a lot of packages in order to solve all cyclic dependencies and crossbuild problems. Also, this process is manual and considering we have a high number of packages that needs to be in the basic system, the bootstrap is far from being an easy process.

5.2 Objectives

The Bootstrappable Debian project has the objective to enable the bootstrap of new Debian ports. As explained previously it is a very difficult project since we still have a lot of problems to solve. A very good explanation about the Debian bootstrap can be found in [1] and a detailed TODO list can be found in [2].

We can divide the project in three main works:

1. **Build a tool that enables an automated bootstrap of new Debian ports.** The main functionality of this tools is to build the packages for a Debian OS for a specified architecture from scratch. The exact parameters that the tool should manage are not defined yet, but this will be discussed during the project. One possibility is that the user should specify the set of packages that should be included in the new environment. To accomplish that, the tool should use the Botch toolchain [3] to determine which packages are available for the minimal build system (that need to be crossbuilt or are multiarch), which packages needs to be built inside the new environment natively and the build order of those packages.
2. **Patch all packages needed to enable the crossbuild process to be done in practice.** This is a hard step of the project because to enable the crossbuild part of the process we need to modify some

packages sources in Debian. This needs to be done to solve cyclic dependencies and file bugs in some packages related with the crossbuild.

3. **Patch all packages that are going to be built inside the minimal build system.** As soon as we got a minimal build system we can start building the packages of the new port natively. While building packages inside the minimal build system, we might lack some build-dependencies that have not being crossbuilt (e.g. they are very hard to crossbuild). So we need to build those packages avoiding the build-dependencies we don't have. This might be accomplished using the new Build Profile standard that enable the addition of information about staged builds. More about this can be found here [].

Build Profiles

5.3 Testing the Automated Bootstrap

In order to test the crossbuilt Debian system, it is plain to use QEMU to emulate the armhf architecture to test if the bootstrap process for this architecture has been done successfully.

6 Synopsis

The objective of this project is to enable the automated bootstrap of new Debian ports. We can summarize all the project in two items:

- Make a tool to manage bootstrap process (including instructions of how to use it)
- Write patches and file bugs filed for all the multiarch, cross-build and profile patches needed for the bootstrap process.

As described in the project description from the Debian ideas list [], this is a very big problem and possible all of this can't be accomplished in just one GSoC.

7 Benefits to Debian

As soon as this project is finished all the bootstrap process is gonna be easier even for non-experts in building Debian packages. Also, if we can do it in an automatic way, the effort to do this will be reduced. If all

patches be accepted by the upstream maintainers, we will avoid solving cyclic dependencies or reduced dependencies problems every time a port is done.

8 Deliverables

At the end of the Google Summer of Code, the following results should be obtained:

1. Automated Bootstrap Tool
2. Patches for all packages needed to allow the bootstrap of a base Debian system.

9 Project schedule

10 Exams and other commitments

I have no class or exams in the GSoC period.

11 Other summer plans

Since I'm a phd student, I have to work on GSoC while working on my thesis, but I already have time reserved and my advisor knows about it so this will not be problem.

12 Why Debian?

I choose Debian because I am familiar with the development environment. I worked in a related project in 2011 and I got very interested on the problem of make an automatic bootstrappable Debian.

13 Are you applying for other projects in SoC?

Yes. I'm also applying for the RouteFlow project but i'm giving a lot more effort to Debian project them to the other.