

MultiROM

The Multiboot Modification For Android Devices

Vojtěch Boček (vbocek@gmail.com, <http://tasssadar.github.io>)

"Why does one need to go through this lengthy process of backing up restoring all the data on the device when trying out a different operating system?"

"Desktop computers have supported multiboot for years, why isn't the same possible on Android devices?"

1 Introduction

A fairly large community of enthusiasts and developers has formed around Google Android™ powered devices. Since Android is open-source, the community soon started making modified versions of it, these are called "ROMs" [1]. Principle of Android ROMs can be compared to the Linux distributions - some have more features, some change the looks of user interface and so on. And because some Android devices themselves are fairly open compared to others, developers of new mobile operating systems soon started using them as testing devices for their creations (since they don't have their own hardware made yet).

As a result, there are a lot of operating systems you can run on Android devices (this varies depending on the exact device). Unlike desktop computers however, Android devices don't support multiboot, you can't install more than one operating system at a time. Instead, one has to go through a lengthy procedure of backing up all the data, installing another operating system and restoring everything again when going back.

While "average users" won't ever hit the problem, a multiboot modification could save a lot of time for enthusiasts, developers and other individuals working with Android devices on a deeper level.

2 Engineering goal

The goal of this project is to develop a software modification which adds multiboot feature to Android devices, can boot any operating system available for particular device and supports unlimited number of secondary systems installed. It should also do so in a way that is easy to use. The resulting code will be released as Open Source Software under the GNU GPLv3 license.

3 Procedures

1. Determine the best method of implementing multiboot on relatively closed Android devices[2]. Using Linux kernel syscall "kexec"[3] seems feasible.
2. Implement proof-of-concept code, without any extensive GUI or management tools.
3. Implement a boot manager with GUI.
4. Add management of secondary system installed on the device into existing open source TeamWin Recovery Project[4].
5. Release the project on XDA-developers[5] forums for public to use.
6. Develop an Android application which will automatically install and update all parts of this modification.
7. Add support for more Android devices.
8. Maintain the project.

4 Risk assessment

As this is purely a software project, no severe risks are present. I will however be modifying low-level parts of device's software, which could possibly result in broken hardware. Parts of open-source software will also be used, so honoring their licenses should not be neglected.

5 Bibliography

- [1] RUSSAKOVSKII, Artem. *Custom ROMs for Android explained*. URL: <http://www.androidpolice.com/2010/05/01/custom-roms-for-android-explained-and-why-you-want-them/> (visited on 01/10/2014).
- [2] *Android boot process description*. URL: <http://www.androidenea.com/2009/06/android-boot-process-from-power-on.html> (visited on 01/10/2014).
- [3] KASICK, Mike. *Using Kexec syscall on Android devices*. URL: <http://forum.xda-developers.com/showthread.php?t=1266827> (visited on 01/10/2014).
- [4] *TeamWin Projects - TWRP 2.8 - TeamWin*. URL: <http://teamw.in/project/twrp2> (visited on 01/10/2014).
- [5] *XDA-Developers Android forums*. URL: <http://www.xda-developers.com/> (visited on 01/10/2014).