Legal Best Practises

How to develop your product with Open Source

Harald Welte

gpl-violations.org
hmw-consulting.de

ST-Ericsson Workshop at ELCE 2009

Outline

1. FOSS Licenses
   - Free Software
   - Copyleft
   - The GNU GPL
2. A detailed view of the GNU GPL
   - Source Code
   - GPL and Embedded Systems
   - Practical GPL compatible source code offer
   - Automatized Tools
   - Derivative Works
3. Business Reasons for Compliance

About the speaker

- Using + playing with Linux since 1994
- Kernel development since 1999
- IT security specialist, focus on network protocol security
- Board-level Electrical Engineering
- System-level Software for PPC, ARM, x86
- IANAL, but companies not complying with the license forced me to spend lots of time with legal issues

Free Software

Definition by the FSF

Free Software has to ensure the following key freedoms:

- Freedom to use the software for any purpose
- Freedom to make copies "to help your neighbor"
- Freedom to study its functionality (source code)
- Freedom to fix it yourself (make modifications)

Copyleft

A concept to ensure Freedom

Copyleft is an idea to use copyright to ensure Softare Freedoms

- Use/claim copyright on the software
- Create a license that is permissive enough for the 4 Freedoms
- However, put some conditions/obligations in the license
  - ensure the source code will always be available
  - ensure nobody is able to remove the 4 Freedoms from the software
- Use that license for the software.

The GNU GPL

An implementation of Copyleft

The GNU General Public License (GPL)

- is a Copyleft Free Software License
- assures the original author that his work will always have the freedoms
- establishes a level of fairness: You can use my code, if you share your additions back with us.
- is a big motivation factor for many community members

Revisiting the GPLv2 License Terms

The GNU GPLv2

- Regulates distribution, not use (running the program)
- Allows distribution of source code and modified source code, if
  - The license is mentioned
  - A copy of the license text accompanies each copy
- Allows distribution of or modified binaries, if
  - The license is mentioned
  - A copy of the license text accompanies each copy
  - The source code is either included with the copy, or a writen offer is made on how the source can be obtained.

Complete Corresponding Source Code

As required by GPLv2

…complete source code means all the source code for all modules it (the software) contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable.

- For a C language program, this means
  - Source Code
  - Makefiles
  - compile-time configuration (e.g. kernel .config)
- General rule
  - Intent of the license is to enable the user to run modified versions of the program
  - If you provide everything needed for that, there will be no discussion

Modifications of GPL'd source code

The details that matter

- In the GPL, it does not matter if you have modified the GPL'd program or if you ship it unmodified.
- You always have to provide the source code!
- If you modify the source code, your changes have to be visible/identifiable
- For practical reasons, I suggest shipping original upstream tarball + a diff/patch with your changes

GPL and Embedded Systems

Interpreting the meaning

- The GNU GPLv2 was written for the GNU project, at the time this project was working on replacing individual application programs on top of a proprietary UNIX operating system kernel.
- scripts used to control compilation and installation
  - Intent: To enable the user to modify + run modified versions
  - In case of embedded systems, the "scripts used to control installation" include the software required for installing the program onto the target device

GPL and Embedded DRM

Sometimes called Tivo-ization

- Some companies want to lock down their Linux-based system, by
  - Cryptographic verification of bootloader by ROM loader
  - Cryptographic verification of kernel image by bootloader. . .
- This is problematic from a GPL pooint of view, since
  - You are depriving the user from practically exercising his right to run modified versions of the program
  - Thus, violation not of the GPLv2 wording, but likely of the GPL's intention
  - Legal outcome unclear, different scholars have different opinions, also depends on jurisdiction
- GPLv3 makes this intent explicit in the license text

Complete + Corresponding Source

For every Release you make

- Whenever you *distribute* GPL licensed software, the license applies. This includes
  - Actual sale of a physical embedded device with the software in flash
  - Download of a firmware update as a file from a website
  - Shipping of firmware updates on physical storage
  - Distribution of firmware updates e.g. by over-the-air mechanisms in DVB-S or other networks
- Every time, the conditions of the license have to be fulfilled (mention there's software under GPL, include full license text, include or offer complete corresponding source code
- For every release you ever ship (even beta release if it ever is shipped only to one customer), you need the *complete corresponding* source code.

Automatized Tools

Helping to make your job easy

Re-creating the source code used for a certain release that was made years ago is close to impossible and very time-consuming. Therefore, you should use a build system that

- always rebuilds your software from source to ensure you don't release binaries of unknown source/origin
- automatically creates the source-release for a firmware releae at the time you create your binary release
- automatically contain the information what is the original GPL'd program and what are your modifications to it.

Use a package manager like dpkg, rpm, ipkg, opkg. It will t ake care of building source packages at the time you build the binaries.

Automatized Tools

Package managers and mixed licenses

A package manager such as rpm, dpkg, ipk has further advantages

- Every package has the license information associated with it
- You can build FOSS applications/libraries and automatically generate the source offer compliant with teh license
- You can also build proprietary applications/libraries and not include the source code to those
- Codified rules rather than manual processes reduce likelihood of mistakes
- You make it easier for the community to modify and re-build the GPL parts of the system

Derivative Works

Keeping it clean

Derivative works are a question of copyright law, not the GPL

- whenever you couple a GPL and a non-GPL program tightly (e.g. static/dynamic linking), your're entering a legal grey area
- there is little or no precedent on derivative works of software
- you're violating the intention of the author. If he wanted you to link from proprietary programs, he would have used LGPL
- try to work *with* the community, rather than against it

Business reasons for compliance

Or: How to convince your managers

If you ship a product that is incompliant to the GNU GPL,

- you are committing a copyright infringement not different from shipping a product with unlicensed copies of MS Windows
- you can face civil and criminal charges in court
- civil charges include (German jurisdiction)
  - immediate cease + desist (halt of product sales)
  - information of which qantity of the product has been sold to whom
  - damages for lost revenue (see dual licensing)
- civil charges can also be filed against every distributor/store/importer

There are many cases of legal precedent now in Germany and the US.

Summary

- GPL compliance is not difficult if you think about the problem when you start product development.
- A large part of the task can be automatized by using a proper build system.
- There are questionable legal *grey areas*. To minimize the risk, I'd try to stay out of them.

Outlook

Outlook

- Blatant GPL violations in embedded devices are declining, but are likely to continue due to lack of skill or negligence.
- We'll see more *derivative works* types of GPL violations, and we'll see actual legal enforcement and preceden in this area over the next years.
- Stronger copyright protection demanded by content industry will also mean stronger protection for FOSS licenses. Imagine GPL enforcement with "three strikes" law in France ?!?