

Legal Issues

From eLinux.org

Contents

- 1 Legal Issues using Linux in embedded projects
 - 1.1 Kernel is licensed GPL v2 only
 - 1.2 GPL/LGPL license version (v3) issues
 - 1.3 Signed-off-by lines and the DCO
 - 1.4 Resources for legal analysis and compliance
- 2 EXPORT_SYMBOL_GPL
 - 2.1 EXPORT_SYMBOL_GPL for kernel USB API
- 3 Binary proprietary kernel modules
- 4 Use of kernel header files in user-space
- 5 Other Links

Legal Issues using Linux in embedded projects

The intricacies of using the GPL license have been hashed out repeatedly in many other forums.

Here are some highlights of a few specific issues that come up occasionally:

Kernel is licensed GPL v2 only

The Linux kernel is licensed under the GNU General Public License, version 2.0 **ONLY!**

This is different from many other projects, which use the default wording in the license to allow GPL v2 or any later version.

This means it is unlikely that the kernel will switch to GPL version 3.0.

In September of 2006, a group of Linux kernel developers signed a position statement (<http://lwn.net/Articles/200422/>) indicating that they objected to GPL version 3.0 (as then drafted). This further indicates the unlikelyhood of any change of the kernel to the GPL v3 license.

GPL/LGPL license version (v3) issues

GPLv3 and LGPLv3 raise interesting new issues for embedded products. One aspect of GPLv3 is the "anti-TIVOization" clause, which prohibits hardware manufacturers from digitally signing software (thus preventing substitution of modified software in the field). See Tivoization (<https://wikipedia.org/wiki/Tivoization>) for more information.

The new versions of the GPL and LGPL raise issues with license compatibility with previous versions. See the following page for a matrix showing what code under which licenses and be released, under what circumstances: <https://gplv3.fsf.org/dd3-faq>

Finally, here is a statement from a developer of a library, indicating some problems using LGPLv3: <http://nmav.gnutls.org/2013/03/the-perils-of-lgplv3.html>

Many embedded companies avoid using GPLv3 and LGPLv3 software in their products, due to the additional restrictions in those licenses, compared to GPLv2 and LGPLv2.

Signed-off-by lines and the DCO

When developers contribute to the kernel, they must provide a "Signed-off-by" line, indicating that they acknowledge the licensing and declare the work (to the best of their knowledge) to be either original, or derivative of something compatible with GPL v2.

See the Developer Certificate Of Origin which is contained in the kernel's Documentation/SubmittingPatches (<http://git.kernel.org/?p=linux/kernel/git/torvalds/linux-2.6.git;a=blob;f=Documentation/SubmittingPatches>) file.

Resources for legal analysis and compliance

- The Software Freedom Law Center has a compliance guide for GPL that is useful:
 - http://www.softwarefreedom.org/resources/2014/SFLC-Guide_to_GPL_Compliance_2d_ed.pdf - October 2014
 - Note that not everyone agrees with all legal interpretations included in this document, but overall it's a good resource.
- Copyleft and the GNU General Public License: A Comprehensive Tutorial and Guide
 - <http://www.copyleft.org/guide/comprehensive-gpl-guide.html#comprehensive-gpl-guidepa1.html>

EXPORT_SYMBOL_GPL

EXPORT_SYMBOL_GPL for kernel USB API

In January of 2008, Greg Kroah Hartman submitted a patch to change the core USB API to EXPORT_SYMBOL_GPL. Here is some information about that change:

- USB: mark USB drivers as being GPL only (LWN.net) (<http://lwn.net/Articles/266724/>)
- Linux 2.6.25 without Closed Source USB Drivers (Linux Magazine) (http://www.linux-magazine.com/online/news/linux_2_6_25_without_closed_source_usb_drivers)
- USB drivers going GPL-only in 2.6.25 (LinuxWorld) (<http://www.linuxworld.com/community/?q=taxonomy/term/24>)
- the actual git commit (<http://git.kernel.org/?p=linux/kernel/git/torvalds/linux-2.6.git;a=commit;h=782e70c6fc2290a0395850e8e02583b8b62264d8>)

Binary proprietary kernel modules

One outstanding legal question with Linux, that is of particular importance in embedded, is whether or not binary (non-GPL) kernel modules violate the GPL license of the Linux kernel.

Opinions on this topic differ.

Here is an article with some interesting information:

- Encouraging closed source modules part 1: copyright and software (<http://www.networkworld.com/news/2006/120606-closed-modules1.html>)
- Encouraging closed source modules part 2: law and the module interface (<http://www.networkworld.com/news/2006/120806-closed-modules2.html>)
- Encouraging closed source modules part 3: eliminating the "API update tax" (<http://www.networkworld.com/news/2006/121106-closed-modules3.html>)

Use of kernel header files in user-space

It is allowed to use kernel header files in user space, in order for user-space programs to interact with the kernel via ordinary system calls. This is allowed without the result that the user-space program becomes a derivative work of the kernel and therefore subject to GPL.

In general, use of header files do not create derivative works, although there can be exceptions. There used to be a lot of attention paid to the amount of code (e.g. number of lines) included from a header file, but no one seems to care about that these days, and this is almost never a problem. Richard Stallman has stated that use of header files for data structures, constant definitions, and enumerations (and even small inlines) does not create a derivative work. See: <http://lkml.indiana.edu/hypermail/linux/kernel/0301.1/0362.html>

The user-space use of the kernel header files is expected and ordinary. This explicitly encompasses non-GPL software using these files, and not being affected by the GPL. In order to calm fears about using the header files directly, and to prevent leakage of kernel internal information to user-space (where it might be abused), the mainline kernel developers added an option to the kernel build system to specifically create "sanitized" headers that are deemed safe for use by user-space programs, without incurring licensing issues.

These are the "make headers_check" and "make headers_install" targets in the kernel build system.

In general, it is legally safest to use such sanitized headers (that is, headers that have specifically been stripped of large inline macros or anything not required for user space.)

This article explains how to create sanitized kernel headers using the kernel build system. <http://darmawan-salihun.blogspot.jp/2008/03/sanitizing-linux-kernel-headers-strange.html>

Note that a different process was used by the developers of the Android operating system, to sanitize headers for bionic for their system. Their process was developed around the same time as the mainline header sanitization feature.

Other Links

- <http://gpl-violations.org/> - The gpl-violations.org project tries to resolve GPL violations and raises public awareness about GPL compliance.
- <http://www.softwarefreedom.org/> - The Software Freedom Law Center provides legal representation for open source projects and publishes information on legal issues around open source.
- <http://www.linuxfoundation.org/programs/legal/compliance> - Linux Foundation's Open Compliance Program
- <http://www.binaryanalysis.org/> - A binary analysis tool for GPL compliance investigations
- <http://lwn.net/Articles/386280/> - LWN.net article about the binary analysis tool (published on 2010/05/06)
- <http://fossology.org/> - FOSSology is a framework to scan open source code: it currently scans for copyright and license information and can easily be extended.

Retrieved from "http://elinux.org/index.php?title=Legal_Issues&oldid=397251"

Category: OpenSource Licensing

-
- This page was last modified on 1 December 2015, at 12:41.
 - This page has been accessed 52,367 times.
 - Content is available under a Creative Commons Attribution-ShareAlike 3.0 Unported License unless otherwise noted.