

Sharing Dual-Licensed Drivers between Linux and FreeBSD

As a silicon vendor, allowing device driver source code to be shared between Linux, FreeBSD, and other operating systems brings several benefits, including a potentially increased market, and additional collaboration effort resulting in increased test coverage and bug fixes.

Linux and FreeBSD are both Open Source UNIX-like operating systems. Both have a long development history and are maintained by sizable development teams consisting of professional, volunteer, academic and hobbyist contributors. Both are capable of high performance in demanding production applications.

One area where they differ is in the license: Linux is licensed under the GNU General Public License (GPL) while FreeBSD uses the permissive Berkeley Software Distribution (BSD) license. The GPL is a reciprocal, "share-alike" license, sometimes called a viral license: derivative works of a GPL project must also be made available under the same license terms. In contrast, the BSD license allows FreeBSD to be used as a component of other projects but does not require that the derivative source code be shared. Some companies build products using unmodified FreeBSD, or share their modifications; examples include Netgate (pfSense), Netflix (streaming content distribution) and iXsystems (TrueNAS). Other companies like NetApp, Sony, and Apple build products that reuse parts of FreeBSD in proprietary software.

It is possible to share driver source code between multiple operating systems to reduce development costs. In order to do so there are at least two aspects to consider: license compatibility, and architecture and interface compatibility.

License Compatibility

The Linux kernel (like many large projects, including FreeBSD) has a license on the software as a whole, but includes individual components or files under their own, possibly differing license. These files may carry only a permissive license that is GPL-compatible, or may be explicitly dual-licensed, allowing a recipient of the file to use it under their choice of license.

The Software Freedom Law Center also published a set of guidelines on maintaining permissive licensed files in a GPL project:

<https://softwarefreedom.org/resources/2007/gpl-non-gpl-collaboration.html>. When a file is used in a larger GPL work but intended to be shared with other, non-GPL projects they suggest applying a permissive license only: “If such a developer is using a license like the modified BSD license or the ISC license, where there is an established and widespread community understanding that the terms permit incorporation into larger programs covered by the GPL, the developer should simply use the permissive license without any further reference to the GPL.”

That said, the SFLC document was published in 2007 and there is now a broader understanding of dual licensing. The Linux Kernel “license rules” documentation <https://www.kernel.org/doc/html/v5.0/process/license-rules.html> has explicit discussion of dual licensing: “individual files can be provided under a dual license, e.g. one of the compatible GPL variants and alternatively under a permissive license like BSD, MIT etc.”

The ISC, BSD, and MIT licenses are largely functionally equivalent.

Interface Compatibility

The second issue with code reuse between operating systems relates to the interfaces provided by and/or required by the code in question. This may be achieved in several ways; one is to implement core functionality in a reusable library, and augment it with an

operating-system-dependent layer. It may be that the core library can be shared between Linux, Windows, and FreeBSD, providing for significant reuse.

For the specific case of code sharing with FreeBSD, another option is the use of the “linuxkpi” layer. This interface layer provides API-compatible implementations of many Linux kernel interfaces, greatly simplifying the adaptation of Linux driver source to FreeBSD.

A third option is to provide permissively-licensed or dual-licensed source, but forgo interface compatibility altogether. In the future an interested party can use that source as a starting point for a ported driver.

Code-sharing Examples between Linux and FreeBSD

Intel i915 graphics drivers

License: MIT

Linux source reference:

<https://github.com/torvalds/linux/tree/master/drivers/gpu/drm/i915>

FreeBSD source reference:

<https://github.com/freebsd/drm-kmod/tree/master/drivers/gpu/drm/i915>

The Intel i915 graphics drivers (like much of the drm graphics stack) is provided under a permissive license, generally MIT.

In the past (up to early 2010s) FreeBSD maintained a bespoke version of the Intel drivers customized for the FreeBSD kernel. More recently we use the existing drivers verbatim via the linuxkpi layer.

Intel iwl wireless driver

License: SPDX-License-Identifier: GPL-2.0 OR BSD-3-Clause

Linux source reference:

<https://github.com/torvalds/linux/tree/master/drivers/net/wireless/intel/iwlwifi>

FreeBSD source reference:

<https://github.com/freebsd/freebsd-src/tree/main/sys/contrib/dev/iwlwifi>

The FreeBSD Foundation recently funded Bjoern Zeeb to port iwlwifi to FreeBSD using the linuxkpi layer. Source is dual-licensed in the Linux tree, and we have submitted some small fixes found during the porting effort, such as

<https://github.com/torvalds/linux/commit/ff1676391aa9a59c36c755724807f349a6b5a1b6>

NXP SJA1105

License: `SPDX-License-Identifier: BSD-3-Clause`

Linux source reference:

https://github.com/torvalds/linux/tree/master/drivers/net/dsa/sja1105/sja1105_spi

FreeBSD source reference: N/A

NXP SJA1105 5-port L2 switch driver. It is not (yet) ported to FreeBSD, but is an example of the BSD license used without dual licensing.

Microsemi VSC85xx PHYs

License: MIT

Linux source reference:

<https://github.com/torvalds/linux/tree/master/drivers/net/phy/mscc>

FreeBSD source reference: N/A

Not yet ported to FreeBSD. An example of a dual-licensed (GPL and BSD) file from another organization, in this case Microchip.

Cut and paste template for other examples

License: MIT

Linux source reference:

https://github.com/torvalds/linux/tree/master/drivers/net/dsa/sja1105/sja1105_spi

FreeBSD source reference: <https://github.com/freebsd/freebsd-src/tree/main/>

Conclusion

At the end of the day, creating dual licensed drivers provides silicon vendors with a long list of benefits. These efforts can reduce development costs, and increase collaboration between operating system developers leading to the possibility of greater share of the market, higher test coverage or increased bug fixes. Making it a win-win for everyone involved.

- Sharing Dual-Licensed Drivers between Linux and FreeBSD
 - License Compatibility
 - Interface Compatibility
 - Code-sharing Examples between Linux and FreeBSD
 - Conclusion

Expand all

Back to top

Go to bottom

Select a repo