

Linux, Licensing, and Drivers -- an Overview

Andy Wilson

(andrew.wilson@intel.com)

Intel Open Source Technology Center

25 July 2006



Agenda

- Quick intro
- Quick review of GPL
- Issues around corporate contributions to Linux
- The importance of process and training
- Open source and patents
- Questions



Quick Intro

- SW developer by background
- Developed Intel's 1st free/open source product in 1990
- Developed Intel's open source training material in 1999
- Helped design Intel's OSS approval process
- Currently a member of the corporate internal open source approval team
- Carrier Grade Linux steering chair 2002-2004, Mobile Linux Initiative steering rep 2005-present
- Not a lawyer! Nothing in this presentation is intended as legal advice, and you should always consult counsel when designing your own process and materials
- Terminology note:
 - this presentation uses “open source” as a synonym for “free and open source SW”
 - “GPL” means “GPL version 2” except where noted



What is success (from a corporate point of view)

- Your products which require good Linux support have it
 - Preferably at time of general availability or as close as possible
- Being seen as a good citizen of the open source community
- Demonstrating that any proprietary Linux SW is properly licensed and copyright-clean
- Ensuring your patent portfolio is used judiciously



What is success (from an open source point of view)

- Full support of modern HW with open source drivers
- Drivers should be fully Linux-native, not ported from other OSs
- Being able to defend against future SCO-type cases



How you can accomplish it

- Understanding of community norms and being willing to work with developers
- Training
- Process
- Tools



The big issues

- GPL, especially as it applies to Linux drivers
- Availability of HW and documentation for pre-release systems
- Standards and SIGs



GPL in one foil

- Of the 20 or so licenses in a Linux distro, GPL is the biggie -- covers the kernel, GNU tools, GNOME UI, etc.
- GPL is a copyright license which is triggered by distribution, e.g. making copies other than for your personal use.
- Big idea == “freedom.” You cannot restrict the freedom of users downstream from you.
- Must make sources available to downstream users.
- Developers of Linux give you millions of lines of code to use for free, with the one caveat that you must publish any improvements under the same license they used.
 - The fundamental “social contract” of Linux and open source
- GPL itself is five pages of unintelligible gobbledygook, with thousands of pages of commentaries. Remember the big principle of preserving freedom, seek legal guidance on the fine points of compliance.



Difficult areas of GPL

- Patent license (is there one?)
 - GPL2 is silent on patents except for “give me liberty or give me death” provision
- Does dynamic linking spread GPL? GPL2 does not say explicitly.
 - A vital question with respect to drivers implemented as loadable kernel modules
- Is there a cure for breach of GPL?
- Conversely, can a GPL license, once granted, ever be rescinded?
- Is the “spreads by linking” clause enforceable?
- Is GPL enforceable at all, or does it exceed privileges given by Congress to copyright owners?



Summary of important GPL court cases

One preliminary finding by a regional court in Germany indicating GPL may be enforceable



More GPL and Linux difficulties

- Compounding the uncertainty ... GPL is a copyright license. Any copyright holder could conceivably have standing to sue.
- Linus Torvalds has opined multiple times (and multiple ways) about whether GPL applies to loadable kernel modules.
- Linus's opinion, while vitally important, is not the last word. There are 500+ other copyright holders in the kernel.
- In the absence of case law or consensus among copyright holders, your lawyers will give you highly conservative advice



Who are the GPL Police?

- Free SW Foundation, as GPL license stewards, investigates alleged GPL violations, even for non-FSF projects
- GPLviolations.org (Harald Welte) also actively looks for abusers
- Module loader in 2.4 and later kernels sets the “tainted”



**Closed source Linux
kernel modules
are illegal.**



GPLv3 -- More complications

- Kernel is licensed under GPLv2 only, not “or any later version”
- Jan. draft GPLv3 clarifies some issues, creates others
 - Adds an explicit patent grant
 - Which is still unclear in the Jan. draft
 - Unambiguously states GPLv3 spreads through dynamic linking
 - Adds stringent anti-DRM provision
 - Can't implement copy protection in GPL-licensed code
 - Can't apply copy protection or module signing to GPL-licensed code
- Torvalds' public statements on Jan. draft negative
- Distinct possibility kernel may stay on GPLv2



Implications of GPL issues

- Gray areas, turbulence, and controversy surround non-GPL drivers.
- Gray areas are unlikely to be resolved soon, and GPLv3 may actually make the situation more unclear.
- Companies who do not enjoy voluntarily subjecting themselves to turbulence and controversy will GPL Linux drivers where they can.



Are closed source or tamper-proofed drivers ever justified?

- (Purely ATW's opinion!)
- For compliance with laws and regulations
 - Which can be misguided, but should be challenged through the political process rather than through SW activism
- For safety and security
 - I personally would like the option of increasing my “freedom” from Trojans and viruses where SW or HW protection technology exists
 - I would also like to know medical or comm. infrastructure devices are tamper-proofed
- To protect premium content
 - The rights of GPL copyright holders to put conditions on GPL code do not trump the rights of copyright owners to put conditions on viewing content
 - Users should have the freedom to decide whether to view copy-protected content



Closed source drivers are not justified...

- (Also purely ATW's opinion!)
- “Because we think they might be valuable and we don't want to share them”
 - Remember the open source social contract. Don't put things you don't want to share in the kernel.
- “Because we know they infringe copyrights or patents”
 - Remember basic business ethics!
- “Because they contain workarounds for errata in the hardware”
 - Remember the FDIV Pentium® processor flaw (we do)



The importance of training and process

- Even if you have no Linux products or think you have zero open source inside your company, you still need training and a process.
 - College engineering curriculum may cover patent writing, but usually does not cover SW licensing and copyrights
 - WWW is filled with misinformation about OSS licensing
 - This is the golden age of plagiarism ... ask yourself how many gigabytes of MP3s your average engineer has downloaded in his/her lifetime
 - Developers need guidance on how to respect copyrights and OSS licenses
 - Managers need to understand issues around OSS licensing and the community
- Basic level of training educates developers and managers about free and open source software, and how it is different than “public domain.”
- At Intel, the copyright basics web-based training is now mandatory in all Intel divisions which write software.
- Advanced level, instructor-led training covers special provisions of the GPL and LGPL, and trains students to analyze more complex licensing scenarios. The instructor-led class is mandatory for all teams who work with the Linux kernel or other GPL code.
- All Linux code, open or closed source, is subject to internal review by an expert panel.



Open Source SW & Patents

- Newer open source licenses (MPL, EPL, Apache 2.0) explicitly address patent licensing
- Classic open source licenses (GPL, BSD/MIT) do not*
 - *(modulo some convoluted language around liberty-or-death in GPL 2.0)
 - Creates a situation where there *may* be a patent license of undefined scope in code you release under these classic licenses
 - Doesn't mean you shouldn't release patented code under these licenses
 - Doesn't mean you can't file for patents on code you release under these licenses (although releasing as open source does start the US 12 month disclosure clock ticking and may prevent non-US filing)

- Does mean you should be careful. Legal and SW developers need to identify exactly which patents (issued or filed) apply to



OSS and Pre-release HW/SW

- Dueling interests
 - Companies have an interest in seeding open source developers with advance HW to enable early SW support
 - Companies also have an interest in controlling access to prerelease HW
 - Maintain their competitive advantage
 - Be able to yank back & recycle prototype HW when production units are released
 - GPL adds another wrinkle -- can't release GPL code under NDA
- Best known method is to provide developers with HW under NDA, SW under GPL or a time-release license, documentation edited for SW developer consumption
 - Problem: Corporate developers will sign NDAs, individual developers are often leery
 - I would like to learn more about uptake under the OSDL HW NDA program and any key lessons learned.



Standards, SIGs, and Open Source

- Some standards and SIGs have OSS friendly IP licensing terms (e.g. W3C 'new' royalty-free patent policy); most do not (member-only access to specs, RAND or FRAND IP policies, other barriers to access)
- Even some companies among the most open-source friendly are still actively promoting SIGs with OSS-unfriendly terms
- This despite growing awareness that having a good, open source implementation is one of the best known methods for promulgating a standard
 - See TCP/IP, HTTP, etc.
- Worst-case scenarios in SIG licensing can leave Linux without GPL implementations of a spec
 - DVD



Questions?

