



# Open Source Software IPR's and licenses

Timo Tuunanen

# Intellectual property rights (IPR)

## § IPR's

- Copyright (expression)
- Patents (invention)
- Trademarks

§ IPR's are originally created to protect the rights of artists (music, literature etc.)

§ In case of software a difference between expression and invention is often unclear

# Copyright

- § All software is automatically covered by copyright
  - as long as work is copyrightable
- § Copyright holder has the following rights to his work (covered by the international copyright laws)
  - Exclusive right to make copies
  - Exclusive right to prepare derivative works
  - Exclusive right to distribute copies of the original or derivative works
  - In case of literature, music, movies etc. you have an exclusive right to display the work publicly
- § Others don't have these rights and they are not allowed to perform these actions without copyright holders permission

## Copyright cont.

- § “As to copyright, a single sentence is generally accepted to be too short to qualify for copyright protection. Still, at the same time, a haiku is likely to be protected even though it's easy to write single sentences that are longer than haikus.” Arnoud Engelfriet, [debian-legal@lists.debian.org](mailto:debian-legal@lists.debian.org), 7.10.2005
- § In case of software one can't say the LOC that qualifies for copyright protection

# Patents

§ Far more complicated to obtain than copyright

§ Patent holder rights

- Right to exclude others from making products embodying your patented invention
- Right to exclude others from using products embodying your patented invention
- Right to exclude others from selling or offering for sale products embodying your patented invention
- Right to exclude others from importing products embodying your patented invention

# Trademarks

- § Purpose is to differentiate from other products
- § Can be owned, sold and licensed
- § OS licenses don't license trademarks
  - If you want to use Linux trademark in your product the license must be obtained from Linux Mark Institute

# License

- § License is simply a permit to do something that is not legal otherwise (Driving license / Software license)
- § Software license describes copyright and patent holders promise to use their intellectual property
- § Open Source licenses guarantee following rights to the user
  - Licensees are free to use Open Source software for any purpose whatsoever
  - Licensees are free to make copies of Open Source software and to distribute them without payment of royalties to a licensor
  - Licensees are free to create derivative works of Open Source software and to distribute them without payment of royalties to a licensor
  - Licensees are free to access and use the source code of Open Source software
  - Licensees are free to combine Open Source and other software

## License cont.

- § Open Source Initiative approves OS licenses. Approved licenses can be found from: <http://www.opensource.org/licenses/>
- § By contrast commercial software licenses grant users a limited right to use the program
- § Open Source license types
  - Academic licenses (for example BSD and MIT)
  - Reciprocal (for example LGPL ja MPL) licenses
    - § Extreme cases are viral licenses (for example GPL)
  - Content licenses (for example AFL, Creative Commons)
- § More than 80 % of the OS software is licensed under GPL or LGPL

# OS warranty

§ OS applications come without warranty.

- THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. **IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES** (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. (Part of BSD -license)

# License comparison

<i>Criteria</i>	<i>Free Distribution</i>	<i>Free Use</i>	<i>Open Source</i>	<i>Reciprocal</i>	<i>Viral</i>
<i>Shareware</i>	<i>X</i>				
<i>Freeware</i>	<i>X</i>	<i>X</i>			
<i>BSD</i>	<i>X</i>	<i>X</i>	<i>X</i>		
<i>LGPL</i>	<i>X</i>	<i>X</i>	<i>X</i>	<i>X</i>	
<i>GPL</i>	<i>X</i>	<i>X</i>	<i>X</i>	<i>X</i>	<i>X</i>

Table by Mikko Välimäki

# Academic licenses

- § Origins of these licenses is at universities
- § Most common academic licenses: BSD, MIT, Apache
- § Idea is to give the software to the users and let them use it any way they want
- § Permission to re-license the software.
- § Derivative works can be closed source (commercial) software
- § Typical requirements are
  - Copyright information can't be removed
  - Names of the organizations that produced the software can't be used when promoting the software
  - In binary distributions, the copyright holders must be mentioned in documentation

# MIT

- § MIT license <http://www.opensource.org/licenses/mit-license.php> permits the use of copyright holders intellectual property on two conditions:
- § Copyright notice and permission notice must not be removed
- § Copyright holder gives absolutely no warranty for the program

# The MIT License

- § Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
- § The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
- § THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

# BSD

§ BSD license <http://www.opensource.org/licenses/bsd-license.php> is very close to the MIT license

§ Some additional rules

- “Neither the name of the <ORGANIZATION> nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.”
- Distribution in binary form must include copyright notice and the license itself

# The BSD License

- § Copyright (c) <YEAR>, <OWNER>  
All rights reserved.
- § Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:
- § Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- § Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- § Neither the name of the <ORGANIZATION> nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.
- § THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

# Reciprocal licenses

- § Derivative works must be distributed under the same license as the original work
- § In GNU world, a term Copyleft is often used:  
<http://www.gnu.org/copyleft/copyleft.html>
- § Viral licenses require that all software that use the original work must be licensed under the same license
- § LGPL and MPL are reciprocal but not viral licenses
- § GPL is a viral license

# GPL introduction

- § GPL is the most common Open Source license
- § “Copyleft” is the most important idea behind GPL license
- § Adherents of the GPL suggests that this provision protects free software
- § Detractors say that this provision creates an island of software from which only GPL-licensed software can escape

# GPL contents

§ Section 2 embodies the FSF idea of "copyleft"

- " a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change"
- " b) You must cause any work that you distribute or publish, ... to be licensed as a whole at no charge to all third parties under the terms of this License"
- Subsection c) makes sure that everybody knows that they are dealing with GPL software
- "These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, ... then this License, and its terms, do not apply to those sections when you distribute them as separate works"
- "In addition, mere aggregation of another work ... on a volume of a storage or distribution medium does not bring the other work under the scope of this License. "

## GPL contents cont.

- § Section 3 requires that GPL source code is made available in one of two ways
- “a) Accompany it with the complete corresponding machine-readable source code, ... on a medium customarily used for software interchange; or,”
  - “b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code; or,”
  - “The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable.”

## GPL contents cont.

- § Beginning of the section 4 identifies the license as the exclusive license for use of the licensed software
  - “You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License.”
- § Any violation of GPL will terminate rights given by GPL.
  - “Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License”

# Interpretation of GPL

§ Interpretations of GPL can be found from GNU web pages: GPL FAQ

<http://www.gnu.org/licenses/gpl-faq.html>

- Same distribution (for example CD) can contain both GPL and otherwise licensed programs
- A company is running a modified version of a GPL'ed program on a web site?
  - § The GPL permits anyone to make a modified version and use it without ever distributing it to others. What this company is doing is a special case of that. Therefore, the company does not have to release the modified sources.
- If I port my program to GNU/Linux, does that mean I have to release it as Free Software under the GPL or some other Free Software license?
  - § In general, the answer is no
- Does the GPL allow me to distribute a modified or beta version under a nondisclosure agreement?
  - § No
- Does the GPL allow me to develop a modified version under a nondisclosure agreement?
  - § Yes. For instance, you can accept a contract to develop changes and agree not to release your changes until the client says ok. This is permitted because in this case no GPL-covered code is being distributed under an NDA

# Interpretation of GPL cont.

## § Derivative work according to FSF

- Subclassing is creating a derivative work
- If I add a module to a GPL-covered program, do I have to use the GPL as the license for my module?
  - § The GPL says that the whole combined program has to be released under the GPL. So your module has to be available for use under the GPL.
- If a library is released under the GPL (not the LGPL), does that mean that any program which uses it has to be under the GPL?
  - § Yes
- If the program uses fork and exec to invoke plug-ins, then the plug-ins are separate programs, so the license for the main program makes no requirements for them
- If the program dynamically links plug-ins, and they make function calls to each other and share data structures, we believe they form a single program

# Interpretation of GPL cont.

## § Derivative work according to FSF cont.

### – What constitutes combining two parts into one program?

- § This is a legal question, which ultimately judges will decide. We believe that a proper criterion depends both on the mechanism of communication (exec, pipes, rpc, function calls within a shared address space, etc.) and the semantics of the communication (what kinds of information are interchanged).
- § If the modules are included in the same executable file, they are definitely combined in one program. If modules are designed to run linked together in a shared address space, that almost surely means combining them into one program
- § By contrast, pipes, sockets and command-line arguments are communication mechanisms normally used between two separate programs. So when they are used for communication, the modules normally are separate programs. But if the semantics of the communication are intimate enough, exchanging complex internal data structures, that too could be a basis to consider the two parts as combined into a larger program.

# LGPL

- § LGPL, Lesser General Public License (formerly Library General Public License) is mostly a copy from GPL
- § Section 5 provides the critical definition of the “work that uses the Library”
  - “5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a “work that uses the Library”. Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License”
  - “However, linking a “work that uses the Library” with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a “work that uses the library”. The executable is therefore covered by this License.”
- § What actually makes a “work that uses the Library” or a “work based on the Library” is unclear
  - Most common interpretation is that dynamic linking is ok while static is not
  - FSF-licensed libraries may not be dynamically linked, while libraries affiliated with Linus Thorvalds and the Linux project may be.
  - Because of the complexity of such problems, users facing these questions should contact the licensor of the Library in question.

# LGPL cont.

- § LGPL has some ill-defined sections that should not be found from general-purpose software license
- § Library vs. library is one such example
  - Library a term used to refer licensed program. (can be programming library or any other work)
  - library means a programming library as software developers see it
  - However section 2a) says that: "The modified work must itself be a software library"
    - § This should have read something like: "The modified work must itself be a software library if the Library is itself a library"
- § Difference between library and Library is described in several places in LGPL
  - "These sections are impenetrable maze of technological babble"  
(Lawrence Rosen)

# License compatibility

- § Licenses must be compatible in order to create works that contain software that is licensed under different licenses
- § List of GPL compatible and incompatible licenses can be found from FSF web pages: <http://www.fsf.org/licensing/licenses/index.html#SoftwareLicenses>
- § Known GPL incompatible licenses
  - Mozilla Public License (MPL)
  - Xfree 86 1.1 -license
  - Original BSD license
  - Apache license

# References

- § The Open Source Definition <http://www.opensource.org/docs/definition.php>
- § Free Software Definition <http://www.fsf.org/licensing/essays/free-sw.html>
- § Debian Free Software Guide:  
[http://www.debian.org/social\\_contract#guidelines](http://www.debian.org/social_contract#guidelines)
- § Categories of Free and Non-Free Software  
<http://www.fsf.org/licensing/essays/categories.html>
- § Open Source Licensing, Software Freedom and Intellectual Property Law,  
Lawrence Rosen, Prentice Hall, 2005
- § Understanding Open Source & Free Software Licensing, Andrew M. St.  
Laurent, O'Reilly, 2004
- § OSI approved OS licenses: [www.opensource.org/licenses](http://www.opensource.org/licenses)
- § GPL FAQ: <http://www.gnu.org/licenses/gpl-faq.html>