Open Source Licensing

Abram Hindle
abram.hindle@ualberta.ca
Department of Computing Science
University of Alberta
http://softwareprocess.es/

Introduction

- *Opensource is the defacto standard for communicating code.*

- *Regardless of your status, as a programmer you will interact with the code of others.*

- I am not a lawyer.

- We are going to talk about opensource and licensing issues with opensource.

Not Covered

- Public Domain
  - Not internationally recognized (France)
  - Use CC0 License Instead
- Proprietary Stuff
  - Proprietary Licenses
  - Shrinkwrap Licenses
  - Terms of Service
  - Too case specific

Copyright

- When you make a work the copyright is the right you gain to your work.

- Copyright limits what OTHERS can do with your work. You have exclusive rights to your work:
  - Copy, reproduce and sell a work
  - Import/Export a Work
  - Create Derivative Works
  - Perform or Display Works
  - Assign rights to others
  - ...

Copyright Act of Canada
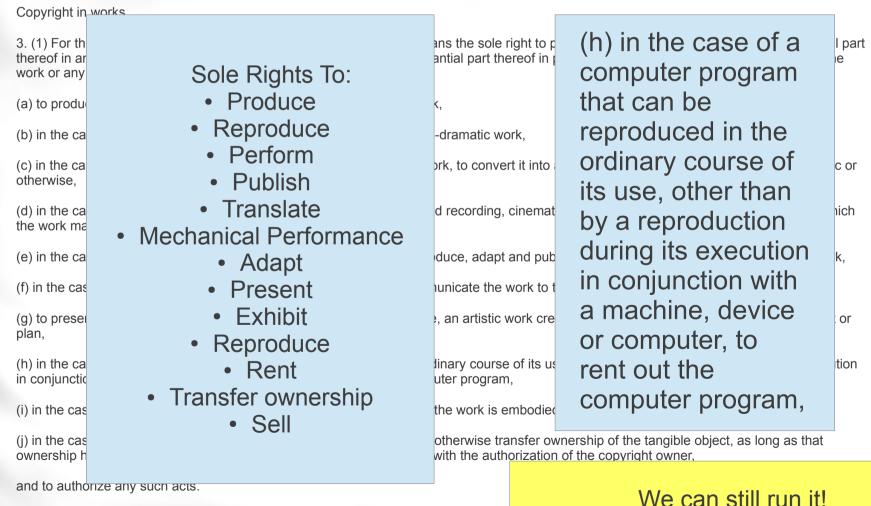
- ## http://laws-lois.justice.gc.ca/PDF/C-42.pdf

Copyright in works

3. (1) For the purposes of this Act, "copyright", in relation to a work, means the sole right to produce or reproduce the work or any substantial part thereof in any material form whatever, to perform the work or any substantial part thereof in public or, if the work is unpublished, to publish the work or any substantial part thereof, and includes the sole right

(a) to produce, reproduce, perform or publish any translation of the work,

(b) in the case of a dramatic work, to convert it into a novel or other non-dramatic work,

(c) in the case of a novel or other non-dramatic work, or of an artistic work, to convert it into a dramatic work, by way of performance in public or otherwise,

(d) in the case of a literary, dramatic or musical work, to make any sound recording, cinematograph film or other contrivance by means of which the work may be mechanically reproduced or performed,

(e) in the case of any literary, dramatic, musical or artistic work, to reproduce, adapt and publicly present the work as a cinematographic work,

(f) in the case of any literary, dramatic, musical or artistic work, to communicate the work to the public by telecommunication,

(g) to present at a public exhibition, for a purpose other than sale or hire, an artistic work created after June 7, 1988, other than a map, chart or plan,

(h) in the case of a computer program that can be reproduced in the ordinary course of its use, other than by a reproduction during its execution in conjunction with a machine, device or computer, to rent out the computer program,

(i) in the case of a musical work, to rent out a sound recording in which the work is embodied, and

(j) in the case of a work that is in the form of a tangible object, to sell or otherwise transfer ownership of the tangible object, as long as that ownership has never previously been transferred in or outside Canada with the authorization of the copyright owner,

and to authorize any such acts.

Copyright Act of Canada

- ## http://laws-lois.justice.gc.ca/PDF/C-42.pdf

Copyright in works

3. (1) For th~~...~~ ans the sole right to p~~...~~ l part thereof in a~~...~~ antial part thereof in p~~...~~ e work or any~~...~~

(a) to produ~~...~~ k,

(b) in the ca~~...~~ -dramatic work,

(c) in the ca~~...~~ rk, to convert it into ~~...~~ c or otherwise,

(d) in the ca~~...~~ d recording, cinemat~~...~~ hich the work ma~~...~~

(e) in the ca~~...~~ k,

(f) in the cas~~...~~ oduce, adapt and pub~~...~~ unicate the work to t~~...~~

(g) to prese~~...~~ e, an artistic work cre~~...~~ or plan,

(h) in the ca~~...~~ dinary course of its u~~...~~ tion in conjunctio~~...~~ ter program,

(i) in the cas~~...~~ the work is embodie~~...~~

(j) in the cas~~...~~ otherwise transfer ownership of the tangible object, as long as that ownership h~~...~~ with the authorization of the copyright owner,

and to authorize any such acts.

Sole Rights To:
- Produce
- Reproduce
- Perform
- Publish
- Translate
- Mechanical Performance
- Adapt
- Present
- Exhibit
- Reproduce
- Rent
- Transfer ownership
- Sell

(h) in the case of a computer program that can be reproduced in the ordinary course of its use, other than by a reproduction during its execution in conjunction with a machine, device or computer, to rent out the computer program,

We can still run it!

Copyright

- You or your employer gets the copyright on your work.
  - It depends on the deal you have with your employer and when you created it and the laws in your jurisdiction.

- For open source licenses, if it is your work, you can change the license you distribute at anytime, as long as you have permission of all copyright holders (aka just you). You may relicense open source only with the permission of ALL copyright holders. You cannot relicense copies already distributed.

Copyright Versus Copyleft

- Copyright is prohibitive. It is a negative right. The creator gains, everyone else loses.

- Copyleft is a corruption of copyright, it is meant to be a positive right, a privelege you can gain by following a few simple rules.

- Copyleft is to allow you to share.

Free Software

- Coined by Richard M. Stallman
  - Current president of the Free Software Foundation (http://fsf.org/)
- A program is free software if the program's users have the four essential freedoms:
  - The freedom to run the program, for any purpose (freedom 0).
  - The freedom to study how the program works, and change it so it does your computing as you wish (freedom 1). Access to the source code is a precondition for this.
  - The freedom to redistribute copies so you can help your neighbor (freedom 2).
  - The freedom to distribute copies of your modified versions to others (freedom 3). By doing this you can give the whole community a chance to benefit from your changes. Access to the source code is a precondition for this.
- Free Software Foundation, https://www.gnu.org/philosophy/free-sw.html , 2013

Richard M. Stallman
http://www.fsf.org/about/staff-and-board/

Open Source Initiative

- Corporate Friendly Free Software, founded by Jon "maddog" Hall, Larry Augustin, Eric S. Raymond, and Bruce Perens (http://opensource.org/history)

- Arguably more pragmatic than the FSF.

- The Open Source Initiative tends to a set of licenses they consider opensource. http://opensource.org/

- Holders of the Open Source Definition http://opensource.org/osd

Open Source

- Taken from http://opensource.org/osd by the Open Source Initiative.

- Open Source Software Must:

  – Allow Free Redistribution (no cost, no limit)

  – Enable available Source Code

  – Allow Derived Works (derivative works)

  – Ensure Integrity of The Author's Source Code (allow building modified software, name changes)

  – Not Discriminate Against Persons or Groups

  – Not Discriminate Against Fields of Endeavor

  – Distribute the License (and be allowed to distribute the license)

  – Have a license that must Not Be Specific to a Product

  – Have a license that must Not Restrict Other Software

  – Have a license that must Be Technology-Neutral

FLOSS

- Free/Libre Open Source Software

- A middle ground (FLOSS or F/LOSS or FOSS)

- Software that meets for the 4 essential freedoms or the open source definition.

Debian Free Software Guidelines

- Excellent body to determine if you can freely distribute a software package.

- If they, Debian, can't distribute the package then you should be worried.

- Anything that limits commercial-use is not DFSG approved

- Anything that is DFSG-approved is pretty safe for you to distribute in any manner. It is guaranteed to be FLOSS

Creative Commons

- Good licenses for content
- Poor for software
  - Software is viewed as a special group under copyright law and much of CC licenses are inappropriate
- Good for content:
  - Manuals and Documentation
  - Story
  - Text
  - Assets (Multimedia)
- Has No Derivatives and Non-Commercial options (these are not FLOSS compatible)

Wait.. What's FLOSS again?

- Software that lets you do **anything** (Read/Write/Modify/Inspect) with it **UNTIL** you distribute it.

- Once you **distribute** it to other people or outside your organization the FLOSS license comes into play.

- It decides if you must:
  - Distribute the source code (GPL)
  - Give users the same freedom reciprocally (GPL)

- But almost always you must:
  - Credit the the contributions of others.
    - In source code
    - Or in a README or about message

Popular Licenses

- Popular licenses according to the OSI (http://opensource.org/licenses) :
  - Apache License 2.0
  - BSD 3-Clause "New" or "Revised" license
  - BSD 2-Clause "Simplified" or "FreeBSD" license
  - GNU General Public License (GPL)
  - GNU Library or "Lesser" General Public License (LGPL)
  - MIT license
  - Mozilla Public License 2.0
  - Common Development and Distribution License
  - Eclipse Public License

Kinds of Licenses

- Copyleft/Reciprocal
  - Everyone has the same rights
  - Rights are passed downstream to all end-users
  - Can never be closed or proprietary
  - Usually forces availability of source code
  - "Keep it free for everyone for ever and attribute me!"
- Permissive
  - Downstream users do not get the same rights
  - Only requires distributed to attribute
  - Source code doesn't have to be shipped
  - "Do anything, but attribute me"
- Both cases:
  - Attribution is required!

GPL License

- Copyleft

- Reciprocal (or viral)

- All users of GPL software enjoy the same rights

- Always open

- Must attribute the author

- All derivatives must be GPL-3 compatible

- Source code must be shared if it is asked for.

- GPL3 also grants end-users to relevant patents of the copyright holders.

BSD License

- Permissive
- You can link BSD to proprietary code
- You don't have to distribute the BSD source code.
- Downstream users of BSD do not enjoy the same rights.
- Closable
- Must attribute the author
- Source code DOES NOT have to shared if it is asked for.
- No patent clauses

MIT License

- Permissive
- You can link MIT to proprietary code
- You don't have to distribute the MIT source code.
- Downstream users of MIT do not enjoy the same rights.
- Closable
- Must attribute the author
- Source code DOES NOT have to shared if it is asked for.
- No patent clauses
- Very simple

Apache2 License

- Permissive
- You can link Apache2 to proprietary code
- You don't have to distribute the Apache2 source code.
- Downstream users of Apache2 do not enjoy the same rights.
- Closable
- Must attribute the author
- Source code DOES NOT have to shared if it is asked for.
- **PATENT Indemnity**
- **Best permissive alternative to GPL3**

GPL is Viral?

- Viral implies some kind of infection, your use of someone else's code is always intentional. If you can't abide by their license you have no right to the code. You have a choice!

- The end-code must be GPL compatible, this means GPL compatible licenses can have different terms but still integrate GPL components.

LICENSE File

- Either in your README or in the LICENSE file place the LICENSE

```
            GNU GENERAL PUBLIC LICENSE
               Version 2, June 1991

 Copyright (C) 1989, 1991 Free Software Foundation, Inc.
         51 Franklin St, Fifth Floor, Boston, MA  02110-1301  USA
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.

                     Preamble

   The licenses for most software are designed to take away your
 freedom to share and change it.  By contrast, the GNU General Public
 License is intended to guarantee your freedom to share and change free
 software--to make sure the software is free for all its users.  This
 General Public License applies to most of the Free Software
 Foundation's software and to any other program whose authors commit to
 using it.  (Some other Free Software Foundation software is covered by
```

License Header

- Go at the head of a file. Typically your license provides an example.

- It states the author and the copyright.

```
# Copyright (c) 2012 Jed Barlow
#
# This program is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 2 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program.  If not, see <http://www.gnu.org/licenses/>.
#
```

License Header

Copyright © 2009-2011 Brandon Mathis

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.

MIT License

Practical Use

- How do I use opensource software?

- How do I distribute opensource software?

- How do I distribute derivatives to opensource software?

How do I use opensource software?

- Anyway you want:
  - Read it
  - Critique it
  - Fork it
  - Change it
  - Run it
  - Build deadly robots with saw arms and no remorse
    - (note: as a Ualberta student you are not allowed to do this according to your community pledge)
- Any endevor, any use, otherwise it isn't opensource (e.g., many free games out there do not allow commercial use or commercial derivatives).

How do I distribute opensource software?

- Part I: You didn't make it.

- Depending on the license, when you distribute it, you call its name, and you attribute the original authors. This should be in file headers and a README document.

- The license should be clearly marked somewhere in the distribution and in the software.

- **Authors must be attributed.**

- You can distribute it via any means you want, and you can sell it, or charge money for the service of distributing it. You can even put ads on a webpage that distributes the software.

- In some cases (such as the GPL) the end-user of the software may ask you for source code. It is your responsibility to provide it for a period after you distribute it. If that user distributes the software again, it is their responsibility.
  - I asked EKEN to give me the kernel source code they used for their Android tablet. They didn't respond but VIA did. VIA is a good company :)

- Usually when you receive an open source distribution, there's nothing you need to do with it to distribute it again. Just watch out though, sometimes you are on the hook for the source code.

How do I distribute opensource software?

- Part II: You made all of it.

- Choose a license, read the instructions of the license. Apply the license to header to the comments of all the relevant files.

- Add your name to copyright notices of the files.

- Write your name down in the README and the license that you assume the files have.

- Copy your license to a LICENSE file.

- Distribute the software and if someone asks you should give them the source code.

How do I distribute derivatives of opensource software?

- Part III: You made some of it!

- You included some other opensource software didn't you?

- **First**, ensure that you **license** your source code under a license compatible with the other component's **licenses**. (GPL compatible licenses are the best licenses to use to ensure the greatest compatibility)

How do I distribute derivatives of opensource software?

- Part III: You made some of it!

- **Second**, make sure you have the source code of the components on hand just in case someone asks.

- Add your name to copyright notices of the files you changed and added.

- Write your name down in the README and the license that you assume the files have. Don't change the license of contributor files.

- Copy your license to a LICENSE file. Tack it on the end and tell the end users which files are under your new LICENSE (or just use the old license).

- Distribute the software and if someone asks you should give them the source code.

How do I distribute opensource software?

- Summary
  - Make sure the license is clear in the distribution
    - LICENSE file is a good start
  - Make sure the headers and LICENSE file have your name in copyright
  - Make sure that you attribute all work that wasn't yours.
  - Keep source code on hand just in case you have to distribute it.

Respectful Rules to Follow

- You may never change the license header of a file, except to update copyright data or to add your name.

- If you copy code from a file, bring the license header with it.

- It is a huge hassle to change a license, so just leave it there. If you don't like it, don't use it, or don't modify the file.

- Make adapters instead of modifying the code.

- Almost every license requires attribution! Be sure to mention the person, not just the product.

- Don't modify the terms of the license.

How to stay compliant

- Can end-users figure out all of the copyright in this software?

- Did you attribute the people, rather than the URL?

- Don't change the license. **SUPER COMPLICATED**

- If you are adding on to an existing project, try to use the same license as the project.

I want to use Open Source but not Open Source my Code

- First consider why you need to do this?

- Q1: Are you going to distribute your software outside of yourself or your organization?

  – No? Then don't worry, the open source licenses aren't relevant. Go ahead.

  – Yes? Well the permissive licenses and inbetween licenses are built to allow for this. Just follow their rules and you won't have to share.

- Q2: Software as a service.

  – Only the GPL Afferro license requires that services open their code. The sad fact is that a service is running on your machines, thus the end users are not being distributed the software. So the license doesn't matter (except GPL Afferro). Plenty of startups, businesses and corporations are built on taking FLOSS code and contributing nothing back (although some do).

Clever Technical Measures

- Firstly, The courts care about **intent**.

- If you are trying to avoid a license, technical measures can be used to exploit the licensed software, **but** avoiding the license might be circumvention (a tort).

- Furthermore there is strong language in Canada and USA against circumvention of copyright.

- Avoid the situation, don't do it. You're not a lawyer, or a judge.

Economic Motivation

- But programmers need to make money!
  - Open Source enables a free market of maintenance, any developer can be paid to fix the problem.
  - Lots of software is not sold directly to consumers or companies. It is bundled as a contract or service.
  - Not everything should/can be an app/consumer product.
  - Expertise enables consulting as well.

Recommendations For You

- You as an individual need to showcase your work. But you should benefit from commercial exploitation should you not?

  - Use the GPL-3

  - If a business wants the code they will pay you to relicense it.

  - It is a strong opensource license.

Recommendations For Business

- Your boss probably doesn't want to opensource the code :(

- But BSD/MIT/Apache2/LGPL are all licenses that won't cause your organization a lot of grief.

Recommendations

- Bruce Perens of the OSI suggests for

- Copyleft License

  - GPLv3
    - Ensures all software that integrates your work is free
    - Ensures patent indemnity (GPL2 doesn't mention patents)

- Permissive License

  - Apache 2
    - Clearer but more complicated than the BSD license
    - Provides patent indemnity

- "In-between License"

  - LGPLv3

- Bruce Perens, 2009
  http://www.datamation.com/osrc/article.php/12068_3803101_2/Bruce-Perens-How-Many-Open-Source-Licenses-Do-You-Need.htm

Conclusions

- FLOSS – Free/Libre Open Source Software

  - Free for any use

  - License matters when you distribute the software again.

  - No sharing or distributing? No problem!

- Keep license headers, keep copyrights, attribute the people, not just the code.

- Lots of code to build on, licensing isn't that complicated if you're just respectful.