

Free Software and Open Hardware Licenses — a Short Guide for People in a Hurry —

Predrag Pejović, Nadica Miljković,
Miloš Cvetanović, Milica Ševkušić

Aims of this paper . . .

Main (minimalist) aims:

1. free software **is not equal** to freeware!
2. to clarify and understand copyleft

Auxiliary aims:

1. clarify that software is different than material products . . .
2. that business models of software development are specific . . .
3. to review common free software licenses . . .
4. to analyze possible generalizations . . .

Disclaimer

- ▶ the authors are **not** lawyers!
- ▶ neither solely lawyers could solve the problem ...
- ▶ optimal education for this task is the background of Eben Moglen: law professor, lawyer, legal historian, programmer, computer user, ...
- ▶ not (m)any people having such background available ...
- ▶ we focus to “human readable layer” (inspired by CC licenses)
- ▶ essential ideas and their consequences without details of legal code, no legalese used
- ▶ from the people who primarily use computers ...
- ▶ but is there a person nowadays who produced more software than he/she uses?
- ▶ verbal presentation: no equations, no diagrams ... strange!
- ▶ **just sharing our thoughts on the subject**

Introduction 1

- ▶ one bit doubles the combinatorial space
- ▶ bit by bit, things changed significantly
- ▶ changes slow and gradual enough not to be noticed as great, but fast enough that we might find us lost in digital space
- ▶ industrial revolution in the world of nonmaterial objects, ideas
- ▶ human mind relieved from algorithmic tasks
- ▶ emergence of “products” without material carrier
- ▶ is this really new?
- ▶ emergence of products with zero marginal cost
- ▶ emergence of products that do not wear out
- ▶ significantly increased ability of common people to communicate, copy, distribute, share . . .

Introduction 2

- ▶ quite a new environment!
- ▶ requires different business models
- ▶ has there ever been a good business model in science?
- ▶ one approach is to treat software in the same way as material objects, bricks or potatoes, for example
- ▶ but, since the software is different, you have to restrict users!
- ▶ they should not be free to copy the software, to start with
- ▶ ... and enforcing this lead us to a number of paradoxes
- ▶ introduced control for copying, is it going to stop there?

Introduction 3

- ▶ trade secret, closing the code; could you trust the program?
- ▶ at least, you cannot build upon
- ▶ to build upon cement and bricks to create houses needs constant supply of cement and bricks
- ▶ to build upon a source code you need just one copy
- ▶ fairly different?
- ▶ potential for a single donation for software development to close the market niche
- ▶ revolt of programmers: the rise of free software
- ▶ and free software respects freedom of its users

Free Software Definition

four freedoms:

- freedom 0** The freedom to run the program as you wish, for any purpose.
- freedom 1** The freedom to study how the program works, and change it so it does your computing as you wish. Access to the source code is a precondition for this.
- freedom 2** The freedom to redistribute copies so you can help others.
- freedom 3** The freedom to distribute copies of your modified versions to others. Access to the source code is a precondition for this.

Free Software and/or Open Source Software?

- ▶ just creates confusion nowadays . . .
- ▶ essentially the same thing
- ▶ basic idea to “improve marketing” by not raising ethical issues related to proprietary software
- ▶ conspiracy theorists would interpret this as an outside attempt to divide enthusiasts and to weaken the movement
- ▶ our standing: the same license, the same category
- ▶ both groups overwhelmingly used GPL
- ▶ nowadays FOSS or FLOSS
- ▶ we’ll treat both camps as the same
- ▶ . . . since they really are . . .
- ▶ I (I ! = we) use “free software”
- ▶ because I am “Stallmanist”

GNU

- ▶ it seems that our initial conditions are different . . .
- ▶ for those who are not familiar, Gnu is Not Unix
- ▶ recursive acronym, some fun to create . . .
- ▶ announced on September 27, 1983
- ▶ Linux is GNU/Linux!
- ▶ essential start of free software movement is here!
- ▶ GNU GPL is named after GNU
- ▶ basic idea to follow UNIX philosophy and create a free operating system
- ▶ really brave idea at that time . . .
- ▶ but unless you do not have high goals you'll never reach them
- ▶ nowadays, a reality
- ▶ done.

Classification of Free Software Licenses

1. restrictive

- ▶ require derivative works to be released under the same license
- ▶ “viral licenses”
- ▶ brilliant piece of logic
- ▶ the least acceptable license in the proprietary world
- ▶ not acceptable for some funding sources

2. permissive

- ▶ “we, as authors, provide you with all four freedoms, just read the disclaimer”
- ▶ authorship preserved
- ▶ disclaimer important, well justified
- ▶ acceptable for proprietary ecosystem, at least as less evil
- ▶ some packages turned out to be a source of significant profit in the proprietary world, SPICE, BSD, ...

Restrictive (Copyleft) Free Software Licenses

- ▶ the essential idea is **copyleft**
- ▶ the term was initiated with “Open Letter to Hobbyists” written by Bill Gates in 1976
- ▶ started as a word play in Palo Alto Tiny BASIC, “@COPYLEFT ALL WRONGS RESERVED”
- ▶ present meaning is different, with deeper meaning
- ▶ something like “all rights reversed”
- ▶ viral technique: the license spreads and preserves rights granted to the users by the program initial author
- ▶ modified version of the program, if released, should keep the same license!
- ▶ unintended use of copyright!
- ▶ but it worked!
- ▶ regardless numerous attacks and pejorative labels

GNU General Public License

- ▶ strong copyleft license
- ▶ there are no exceptions: modified versions, if released, should keep the same license
- ▶ three versions:
 1. version 1, 1989
 2. version 2, 1991
 3. version 3, 2007
- ▶ presently active versions are 2 and 3
- ▶ version 3 addressed software patents, hardware restrictions, license compatibility, DRM, ...
- ▶ not that version 3 is completely and immediately accepted ...

GNU Lesser General Public License

- ▶ weak copyleft license
- ▶ how to license free software libraries?
- ▶ if under strong copyleft, proprietary software cannot use free software libraries
- ▶ political decision . . .
- ▶ programs that use the library are not required to keep the license
- ▶ modified versions of the library are required to keep the license
- ▶ copyleft depends on the nature of the resulting derivative work

GNU Affero General Public License

- ▶ really strong copyleft license
- ▶ “network applications” and “cloud computing”
- ▶ in GNU GPL running the program does not trigger the copyleft mechanism!
- ▶ AGPL: derivative work, offered as a network application, when run on a server should provide downloading of the source code from the server
- ▶ in this manner running the program triggers the copyleft
- ▶ important in the era of cloud computing

GNU Free Documentation License

- ▶ free software needs free documentation ...
- ▶ complicated: cover texts, invariant sections ...
- ▶ a bit complex license, historical motivation ...
- ▶ similar to CC BY-SA, though not directly compatible
- ▶ requires attribution of original authors
- ▶ requires changes to be notified
- ▶ burdens when printing: original license should be printed as well, and it is a pretty big document
- ▶ regardless these facts, fairly popular, used by Wikipedia
- ▶ GNU Simpler Free Documentation License is a result, for manuals and textbooks
- ▶ compatibility issues ...

double licensing

- ▶ applicable for programs licensed under copyleft licenses
- ▶ if the derivative work does not comply with the original license special licensing terms might be negotiated
- ▶ an example: FFTW
- ▶ sounds like a fair deal

Permissive Free Software Licenses

- ▶ provide four freedoms
- ▶ do not require copyleft
- ▶ great compatibility potential
- ▶ usually just state the software creator and contain a disclaimer
- ▶ sort of acceptable in the proprietary world ...

BSD License

- ▶ permissive software licenses
- ▶ there have been several of them . . .
- ▶ 4-clause, 3-clause, 0-clause . . .
- ▶ used to contain “advertising clause”
- ▶ historically important, used by BSD
- ▶ used to license many packages included in proprietary software

MIT License

- ▶ where is Berkeley, there is MIT ...
- ▶ permissive license, very similar to BSD license without advertising clause
- ▶ really short
- ▶ a sentence that transfers rights and a disclaimer, common format
- ▶ all rights granted, just keep the note, DISCLAIMER in capital letters
- ▶ I was about to include the license, but copyright licenses are under copyright by Berne convention
- ▶ should licenses have their own licenses? They do! (as a copyright note)

Apache License

- ▶ another permissive license
- ▶ started as BSD license, and evolved as BSD license
- ▶ addition: derivative works should not keep the Apache name
- ▶ unmodified parts of the code keep the license
- ▶ historically important, Apache HTTP Server promoted application of GNU/Linux
- ▶ version 2.0 addresses software patent threats in the way favored by the FSF
- ▶ compatible with GPL 3
- ▶ widely accepted!

ISC License

- ▶ included **AFTER** the conference, upon suggestion of Vladimir Milovanović, University of Kragujevac
- ▶ similar to simplified BSD license and MIT license, but simplified by removing content unnecessary since there is Berne Convention
- ▶ really short, really simple, really liberal, really readable ...
- ▶ ... but do we need copyleft?

Software in Public Domain

- ▶ sort of early license, when it all started
- ▶ all rights transferred
- ▶ in some jurisdictions not possible to disclaim authorship, primarily in Europe
- ▶ since everything copyrightable is copyrighted by Berne convention, requires a copyright waiver, a copyright note
- ▶ usually, this contains a liability disclaimer
- ▶ an example of the effect of different jurisdictions in our www world
- ▶ sometimes good, sometimes bad, depends . . .
- ▶ requires careful thinking before qualifying!

A Table (a bit simplified, a bit overgeneralized)

license	pub. dom.	BSD	MIT	ISC	Apache	LGPL	GPL	AGPL	
liability disclaimer	✓	✓	✓	✓	✓	✓	✓	✓	permissive
recognizes authorship		✓	✓	✓	✓	✓	✓	✓	
same license for unmodified parts					✓	✓	✓	✓	
copyleft for modified versions						✓	✓	✓	restrictive
general copyleft							✓	✓	
public use triggers copyleft								✓	

License Compatibility

- ▶ a **real** issue
- ▶ could software under license A be used with software under license B and under what license the resulting software might be released?
- ▶ careful reading and precise wording matters here!
- ▶ license proliferation is a problem!
- ▶ unless you have a good reason, do not create your own license
- ▶ for common licenses there are compatibility tables
- ▶ in general, permissive software licenses are compatible with copyleft licenses, not the opposite
- ▶ ... but be careful!
- ▶ careful reading and consulting are strongly advised!

Freeware is NOT Free Software

- ▶ already heard this?
- ▶ people tend to forget and/or to mishear
- ▶ let us repeat:
 1. free software provides its users the four freedoms from the definition stated at the beginning
 2. freeware is the software distributed free of charge
- ▶ okay, do you find any difference?
- ▶ or better to ask, do you find any similarity?
- ▶ the notions are different, don't you agree?
- ▶ so why people treat them as synonyms so frequently?
- ▶ is money the only thing that matters?

Transition: Open Instruction Set Architectures

- ▶ let's get closer to the bare metal . . .
- ▶ the closest that you can get from the software side is the ISA
- ▶ should it be open and free?
- ▶ the case of RISC-V
- ▶ Berkeley, Krste Asanovic, . . .
- ▶ specification is open, some cores are open, some cores are not
- ▶ work in progress and in rapid expansion . . .
- ▶ my guess is that this is the future
- ▶ we'll see in the years to come
- ▶ however, keep an eye on RISC-V

Open Hardware 1

- ▶ what hardware?
- ▶ at first, computer hardware, from the ISA down to silicon
- ▶ open cores emerged ...
- ▶ open toolchains emerged, seems to be here to stay
- ▶ could you trust closed hardware any more?
- ▶ does malware tend to migrate to the hardware level?
- ▶ under what conditions you could trust your hardware?
- ▶ it seems that even hardware design should be open
- ▶ at least if you care to trust it

Open Hardware 2

- ▶ let's generalize a bit ...
- ▶ the case of Arduino success
- ▶ GPL for the software and CC-BY-SA for the design files
- ▶ unexpected winners!
- ▶ what is Arduino? software, hardware, ecosystem, community?
- ▶ another topic: single board computers?
- ▶ do open hardware projects take the market rapidly?

Open Hardware 3

- ▶ not just limited to computers . . .
- ▶ popular in scientific instrumentation
- ▶ CERN being one of the leading institutions in developing the concept; copyleft is a legal issue
- ▶ popular in 3D printing designs
- ▶ WikiHouse project
- ▶ ideas tend to generalize and spread
- ▶ design files could be exchanged easily
- ▶ would open experiences create new standards?

Generalization: Creative Commons Licenses

- ▶ complete success of open ideas in software
- ▶ with some phase delay, gaining success in hardware
- ▶ how about other areas of human activity?
- ▶ creative works became digital, available to redistribute, remix, and build upon easily
- ▶ after the experience with software, Creative Commons adapted those ideas for other creative works, resulting in Creative Commons licenses
- ▶ standardized licenses, designed by a respectful legal team
- ▶ new dimensions: attribution, share alike, no derivative works, non commercial, and combinations of these features; share-alike is copyleft
- ▶ three layers of a license: lawyer readable (legal code), human readable, machine readable; great to have human readable!
- ▶ something really complete; live and maintained!

Conclusions 1

- ▶ material objects and **non**material objects are different!
- ▶ do the same business models apply for the both of the types?
- ▶ industrial revolution(s) affected material objects
- ▶ digital technology revolutionized nonmaterial world
- ▶ in the area of software, generalization of business models appropriate for material objects lead to a number of paradoxes
- ▶ are proprietary business models socially efficient?
- ▶ the future will give us all the answers
- ▶ in the meantime, we covered free software licenses on a human readable level
- ▶ we covered rapidly expanding area affected by the ideas of free software: open hardware and open culture

Conclusions 2

- ▶ please remember the free software definition
- ▶ please remember that free software is not freeware
- ▶ please remember how copyleft works
- ▶ **enjoy being free by using free software!**