Software Development on Linux Systems

4002-XXX-XX

By

Cody Van De Mark

Today

- History of Open Source
- How Open Source Works
- Open Source Software Development Cycle
- Licensing, Trademarks and Software Ownership

History of Open Source

- Open source principles have been around
  - Science, cooking recipes, religion, education, etc have been open source for hundreds of years occasionally having closed source periods
- Modern forms of open source had roots in the early 1900s when monopolies were still big in the U.S.
- When Henry Ford won a lawsuit against patents and monopolies, he and others agreed to open share patents without money exchange or legal battles

History of Open Source

- The automobile industry has been openly sharing patents since propelling the industry forward and providing mechanical enhancements for the rest of the world

- IBM and other groups began to follow suit, sharing design information and source code in the 1950s and 1960s before compiled executables existed

- The only option was to share the source code to distribute it

History of Open Source

- During the 1960s, those with early access to networks and Internet-like technologies began to standardize network communication as an open standard

- All of these, however, were unlicensed and still dealt with patent information

- Open source software itself, finally started to occur in the 1970s

History of Open Source

- In the 1970s manufacturers began to heavily lock down software to prevent usage on other manufacturers' hardware, to charge for usage by time and to prevent redistribution

- During the 1970s, software time-bombs were invented to stop functionality of software after a set period of time, unless the user paid for more time

- Discontentment with software patents, proprietary software, software monopolies and restricted computer access inspired a change in software licensing

History of Open Source

- On the east coast, Richard Stallman created the GNU project in 1983 and the Free Software Foundation in 1985;

  Both of these aimed to create free open source software and licensing

  These were launched as initiatives to change the ethics and politics of software development

History of Open Source

- The GNU Project was designed to create free software that users could modify, copy and distribute
- GNU has four principles:
  - Freedom to use the software for any purpose
  - Freedom to access and study the source code
  - Freedom to distribute the source openly to anyone
  - Freedom to modify or improve the software

History of Open Source

- At the same time, University of California at Berkeley started development of an open source operating system called Berkeley Software Distribution (BSD) which was similar to Unix, but had significant improvements

- BSD was based off of Unix source code, but was written without any Unix code; BSD became as stable open source operating system

History of Open Source

- Several years later in 1991, BSD was mostly halted under legal conflicts with Unix; These conflicts held up development for two years until BSD could prove that none of the code was from Unix

- By then, the GNU project had created a plethora of open source tools and most of the components for an operating system;

  They only lacked a kernel as theirs was not complete and BSD was unavailable seemingly slowing the growth of open source

History of Open Source

- Also in 1991, Linus Torvalds, a computer science student at the University of Helsinki, Finland, was in disagreement with the design decisions of the operating system called Minix

  Minix was a proprietary operating system that was intended for students

- As a result, Linus Torvalds started an open source operating system as a hobby to learn about 386 hardware that intentionally took different design approaches than Minix

History of Open Source

- As the Linux Kernel was released in 1992, it provided an available kernel for the accumulated GNU components of the last decade

- As BSD was unavailable due to lawsuits, open source developers turned to Linux as an available kernel

- The GNU components were combined with Linux to create the system you know today, GNU/Linux

  You commonly hear it referred to as Linux for short

History of Open Source

- GNU/Linux provided the first usage open source operating system

- This provided an operating system that developers could use freely and contribute to each other

- A system free of any patents, copyright issues and legalities allowed the evolution of open source software to begin

  Since then, open source software has evolved quickly

Open Source Evolution

- Since GNU/Linux was released in 1992, open source software has rapidly developed
- Debian and Slackware were released in 1993, openSuse in 1994 and Red Hat in 1995
- By 2007, Debian GNU/Linux contained seventy-three thousand man-years of work in the distribution without including software that can be added through the repository

How Open Source Works

- Open source software is the combined efforts of thousands of developers through the freedom to use, modify and redistribute

- Open source is a direction chosen by the developers of software

- Once software is licensed to an open source license, it is then legally bound to the restrictions of that license;

  Any developer who uses, modifies or redistributes the software is bound to the the requirements of that license

How Open Source Works

- Open source software is maintained by the community itself
- These may be
  - individual contributors
  - students
  - industry professionals
  - community members
  - Etc
- New software or replacement software is chosen by the community

How Open Source Works

- New software may be started individually and picked up by a community, or form a community

- Other times software is developed out of the need for it or the social implication

- Once the source code is made, there are systems in place for distributing the code the communities you want;

  Others may port the code to other communities

How Open Source Works

- Usage:
  - Usage is almost entirely unlimited, except for prevention of making the code closed source
  - All open source licenses provide you with the ability to use the software freely for personal use
  - Most open source licenses allow you to use the software freely for any use including commercial

How Open Source Works

- Modification:
    - Open source licenses give you the ability to modify, improve or fork source code as long as
        - it is licensed to the same open source license
        - The license is provided
        - Credits/Copyrights of previous developers have been maintained

How Open Source Works

- Distribution:
  - Open source licenses allow you to
    - freely copy and distribute source code and binaries as long as the license and copyright information has been maintained
    - freely copy and distribute modified source code and binaries as long as the license and copyright information has been maintained, and the modifications are open sourced

Open Source Software Development

- 1. Starts with an idea
- 2. Idea implemented into an existing project or as a new project whether it is complete or not
- *2.1 Source code is committed back to an existing project, if applicable*
- 3. Source code is distributed as development, testing or experimental package
- 4. Steps are repeated by the same or a different developer

Open Source Development Cycle

- Initiate
    - Problem Discovery & Description
    - Determining Development Team
    - Identifying a solution and work plan
- Execution
    - Code Development and Testing
    - Code Review
    - Code Commit and Documentation
- Release
    - Release Management, Bug Tracking and Repeat Process

Licensing

- The entire open source environment is dependent on open source licenses to legally bind people to keep code open source

- Without licensing, companies/individuals have
  - Taken open source code and sold it as their own
  - Improved open source code and sold the improvements as proprietary software
  - Incorporated open source code into their infrastructure without giving back
  - Leveraged open source code to create competing projects in order to erase the open source project

Licensing

- Open Source Licenses can apply to
  - Source Code
  - Binaries
  - Documentation
  - Engineering
  - Music
  - Literature
  - Philosophy
  - Performing Art
  - Visual Art
  - & more

Licensing

- *Libre* - Free as in Freedom (Open Source)
  - *Free as in Freedom* refers to little or no restrictions on your usage
  - You have the freedom to use, modify and distribute
- *Gratis* – Free as in Beer (Closed Source)
  - *Free as in Beer* refers to not having any price, but not without restriction
  - You may not be allowed to modify the software, distribute the software or use the software as you intend to

Licensing

- Popular Open Source Licenses
  - Apache License
  - BSD License
  - Common Development and Distribution License (CDDL)
  - Creative Commons Licenses
  - Eclipse Public License
  - General Public License (GPL)
  - “Lesser” General Public License (LGPL)
  - MIT License
  - Mozilla Public License

GNU General Public License (GPL) v3

- Allows you to
  - Use, modify and distribute source code freely
  - Distribute modified source code as long as the derivative is GPL or GPL compatible
  - Distribute code as long as the source code and license are attached
- Does not allow
  - Linking code to proprietary software
  - Distribution with software that is not GPL compatible unless the two clearly are not linked together
  - Distribution of modified source code if the derivative is not GPL compatible

GPL Compatibility

- The GPL requires code distributed with GPL code to be GPL compatible

- Due to the nature of this, many licenses can not be distributed along with GPL legally

  Ensure your license versions (version numbers) are GPL compatible if you distribute GPL code

- The GPL is legally binding and distributing non-GPL compatible code with GPL code has legal implications for misuse

“Lesser” Public License (LGPL) v3

- Allows you to
  - Use, modify and distribute source code freely
  - Distribute modified source code as long as the derivative is GPL compatible or LGPL compatible
  - Link code to proprietary software (IE: Software can be called by proprietary software, but is not part of the proprietary software)
  - Distribution of the work with proprietary software **if** the source code/license to the open source software is provided **with** changes (if any were made)

- Does not allow distribution
  - of modified source code if the derivative is not GPL or LGPL compatible
  - of the work without the source/license

Apache License 2.0

- GPL Compatible
- Allows you to
  - Link with proprietary software
  - Distribute the work with proprietary software as long as the the open source code is attributed and has the license
- Does not allow
  - Distribute modified source code with
    - “Apache” in the name or use of any marks from the Apache Software Foundation
    - Implication that you wrote the open source code
    - Implication that the software was endorsed by the Apache Software Foundation

Modified BSD License

- GPL Compatible
- Allows you to
  - Link with proprietary software
  - Distribute the work with proprietary software as long as the the open source code is attributed and has the license
  - Distribute the work as long as the open source code is attributed and has the license
- Does not allow
  - Implication that the open source software developers or organization endorsed any derivative open source or closed source software

Common Development and Distribution License (CDDL 1.0)

- Not GPL Compatible
- Allows you to
  - Link with proprietary software
  - Distribute or modify open source code as long as it is still CDDL
  - Distribute the work with proprietary software as long as the the open source code is attributed and has the license
  - License each file individually, instead of the entire project
  - Link with code from other licenses
- Does not allow
  - Suing fellow community members over patent related issues
  - Modification or distribution without responsibility for claims and damages

Eclipse Public License (1.0)

- Not GPL Compatible
- Allows you to
  - Link with proprietary software
  - Distribute or modify open source code as long as it is still EPL compatible and you grant any recipients patent rights you own specific to the software
  - Distribute the work with proprietary software as long as the the open source code is included and has the license
  - Have patent rights specific to the open source software to use and distribute the code if it is under a patent

Eclipse Public License (1.0)

- Does not allow
  - Suing fellow community members over patent related issues
  - Distribution of code that infringes on patents or other intellectual property that has not been granted to you
  - Distribution of modified open source code without responsibility for the implications of the modified code
  - Acquirement of EPL code without assuming the necessary responsibility to receive that code

MIT License

- GPL Compatible
- Allows you to
    - Distribute the work as long as the open source code is attributed and has the license
    - Modify and distribute the software as long as the source code and license are included
    - Distribute the source code with software under any license as long as the open source code is included with the license
- Does not allow
    - Usage of the copyright holder's name in advertising or promotion of sale or other dealings without written permission

Mozilla Public License (MPL 2.0)

- GPL Compatible
- Allows you to
  - Link with proprietary software
  - Distribute or modify open source code as long as it is still MPL
  - Distribute the work with proprietary software as long as the the open source code is attributed and has the license
  - Distribute with code under any license as long as the MPL is still preserved and each MPL file is marked so
- Does not allow
  - Distribution of executables without the source code and license
  - Derivative Licenses for unrelated code that include “Mozilla”

Creative Commons Licenses

- Creative Commons is a license for works that fall under copyright (IE: Intellectual Property, Information, Works, Arts, etc)

- Creative Commons Licenses are non-exclusive and non-revocable

  They are not exclusive to any party and can not be revoked or terminated after they have been licensed

- Certain Creative Commons Licenses can be combined with Commercial Licenses

Creative Commons Licenses

- Do not cover
  - Software
- Do cover
  - Any and all work falling under copyright

    Examples: Websites, Blogs, Articles, Movies, Music, Documentation, Books, Plays, Movies, Videos, TV episodes, photographs, Visual Media, Artwork, Animation, etc

Creative Commons Licenses

- Attribution – copy, distribute, display, perform and modify as long as the author/licensor is credited as the license requires
- Noncommercial – copy, distribute, display, perform and modify only under noncommercial purposes and use
- No Derivative Works – copy, distribute, display and perform the work verbatim only. No modifications or derivatives are allowed
- Share-alike – distribute derivatives only under licenses identical to the original license of the original work

Creative Commons Licenses

- Non-attribution CC licenses have been retired. All Licenses contain attribution

- Licenses
  - Attribution alone (by) license
  - Attribution and No Derivative Works (by-nd) license
  - Attribution and Share-alike (by-sa) license
  - Attribution and Noncommercial (by-nc) license
  - Attribution, Noncommercial, Share-alike (by-nc-sa) license
  - Attribution, Noncommerical, No Derivative Works (by-nc-nd) license

Trademarks

- It is important to note that Trademarks are separate from open source software

- Having access to the source code does not grant any right over the trademarks

- No derivatives may include the Registered Trademarks of the original without permission from the original owner

- Trademarks can include device, brand, label, name, signature, word, letter, number, shape, packaging, coloring, smells, sounds movements or combination of those

Trademarks

- Examples
    - Apache or Mozilla name
    - Apache or Mozilla logos
    - Linux name
    - MySQL name
- In most cases, open source Trademarks have fair use policies specified by the Trademark Holder
- Apache, Eclipse and Mozilla have more strict policies than most as they are organizations

Software Ownership

- The original author or authors are the owners of the software

  These are the only ones able to modify the license or license terms

- Contributors or developers, thereafter are allowed to claim attribution and copyright to the software, but are unable to modify the license or terms without written permission from the owner(s);

  Though you may be able to license your work separately or as proprietary software, you may not alter the terms or relicense the software without written permission from the owner(s)

Software Ownership

- For the project, licensing may seem complex at first;

  You must follow the license terms of the software you leverage

  You may choose the license you want for the project

- If you work in groups on your project, you must choose an open source license together as individual ownership and terms of usage can not be determined separately by each owner