

FLOSS : HISTORY AND EVOLUTION

Dr. Kuljit Kaur Chahal

Department of Computer Science

Guru Nanak Dev University, Amritsar, India

Which of the two names have you seen/heard the most?

- A. Linux
- B. GNU/Linux

Free/Libre Open Source Software (FLOSS)

- Open Source Software is dominating the software industry more than ever before:
 - More than 90% of software products contain open source components*

*Managing your open source supply chain – why and how, IEEE Computer, 53(6), 2020

Is FLOSS free?

- YES!
 - The user is free to run, copy, study, adapt, improve, distribute an open source product
 - The source code is in the public domain for all these actions
- NO!
 - Software has a price.
 - Its not free as free beer.



However, software was free as free beer when*

- Researchers were writing software and gladly sharing it with other researchers to port and use it
 - Academic principles of sharing knowledge!!!
- it was bundled with hardware in 1950s and 1960s
 - manufacturers were happy that software was making the machines useful
 - Later, they found that all the users were not into maintaining the source code on their own, and wanted help.
 - They stopped distributing source code.
 - Leading player IBM's unbundling of software and hardware.
- By mid 70s, proprietary software was dominating the scene

*there didn't exist any definition of *free software*

Why Free software?

- User should control the software rather than software controlling the user.
 - So appropriate even in the modern context?
- User should have the freedom or liberty to see, and modify.

Four levels of freedom

(0) The freedom to run the program as you wish, for whatever purpose.

(1) The freedom to study the program's “source code,” and change it, so the program does your computing as you wish. Anyone who knows programming, and has the program in source code form, can read the source code, understand its functioning, and change it too.

(2) The freedom to make and distribute exact copies when you wish. Distributing a program to users without freedom mistreats them; however, choosing not to distribute the program—using it privately—does not mistreat anyone.)

(3) The freedom to make and distribute copies of your modified versions, when you wish.

Four freedoms from 0 to 3

- The reason the current and definitive four freedoms are numbered 0, 1, 2 and 3 is historical.
- Around 1990 there were three freedoms, numbered 1, 2, and 3.
- It was then decided that the freedom to run the program needed to be mentioned explicitly.
- Rather than renumber the others, freedom 0 was added.

People who contributed to FLOSS for what it is today

Stallman-Torvalds-Raymond's hacker diaspora

- Richard Stallman
- Linus Torvalds
- Eric S. Raymond

Hackers: Heroes of the Computer Revolution

Steven Levy. *Hackers: Heroes of the Computer Revolution*.

Anchor/Doubleday. 1984. ISBN 0-385-19195-2.

A few others

- Tim O'Reilly
- Christine Peterson

The Players : An Introduction

- Richard Stallman (also known as RMS) kicked off the free software movement at MIT in the 1980s.
 - Started the GNU project and the Free Software Foundation
- Eric S. Raymond (also known as ESR) supported the rebranding of free software to “open source” in the 1990s.
 - Wrote the famous essay “The Cathedral and the Bazaar”
- And Linus Torvalds created both Linux, the open source kernel that powers many of today’s operating systems, in 1991, and Git, in 2005.
 - Linux a poster child for the early promise of large-scale collaboration,
 - Git for version control. Ironically GitHub is there due to Git.





Richard Stallman



Eric Raymond



Linus Torvalds

Launching of the GNU project in 1983

Free Unix!

Starting this Thanksgiving I am going to write a complete Unix-compatible software system called GNU (for Gnu's Not Unix), and give it away free [\[1\]](#) to everyone who can use it. Contributions of time, money, programs and equipment are greatly needed.

Launching of the GNU project in 1983

Who Am I?

I am Richard Stallman, inventor of the original much-imitated EMACS editor, now at the Artificial Intelligence Lab at MIT. I have worked extensively on compilers, editors, debuggers, command interpreters, the Incompatible Timesharing System and the Lisp Machine operating system.

Launching of the GNU project in 1983

Why I Must Write GNU

I consider that the golden rule requires that if I like a program I must share it with other people who like it. I cannot in good conscience sign a nondisclosure agreement or a software license agreement.

So that I can continue to use computers without violating my principles, I have decided to put together a sufficient body of free software so that I will be able to get along without any software that is not free.

Launching of the GNU project in 1983

Almost complete in 1990

What motivated Richard Stallman?

- His belief that
 - computer programs are easier to copy and modify.
 - Relevance of social needs
 - availability of information
 - freedom for its citizens to control their own lives.
 - the spirit of voluntary cooperation.
 - system that says you are not allowed to share or change software—is antisocial, unethical
- His stance that free software is a matter of freedom, not price.
 - Economics is important no doubt.
 - “Some people write useful software for the pleasure of writing it or for admiration and love; but if we want more software than those people write, we need to raise funds.”

“Free as in freedom, not free as in beer,”

What motivated Richard Stallman?

- Vanishing software-sharing community in the neighbourhood and a failed attempt to customize a printer in office for lack of access to its source code was the last straw .
- Stallman wanted to liberate code from proprietary use.
 - Contributions of time, money, programs and equipment are greatly needed.
- and the occasional use of the Spanish word *libre*, rather than *gratis*, to refer to free software, in order to distinguish between these two meanings.
 - "Gratis" pertains to being able to access and use the code, without a price-barrier, while "libre" pertains to being allowed to modify and re-use the code, without a permission barrier.

So I looked for a way that a programmer could do something for the good. I asked myself, was there a program or programs that I could write, so as to make a community possible once again?

The answer was clear: what was needed first was an operating system. That is the crucial software for starting to use a computer. With an operating system, you can do many things; without one, you cannot run the computer at all. With a free operating system, we could again have a community of cooperating hackers—and invite anyone to join. And anyone would be able to use a computer without starting out by conspiring to deprive his or her friends.

GNU, is a recursive acronym meaning “GNU's Not Unix”—

- GNU gets the technical ideas of Unix.
 - But gives its users freedom to read and adapt code
- The principal goal of GNU is to be free software.
- Though no technical advantage over Unix,
 - But a social advantage
 - allowing users to cooperate, and
 - an ethical advantage,
 - respecting the user's freedom.

Richard Stallman quit the job at MIT in **1984** for creating a new software-sharing community.

the Free Software Foundation (FSF)

- In 1985, a tax-exempt charity for free software development.
 - the organizational home for the GNU Project
- The FSF also took over the GNU Emacs tape distribution business;
 - later it extended this by adding other free software (both GNU and non-GNU) to the tape,
 - and by selling free manuals as well.

February 1986

G N U ' S B U L L E T I N

Volume 1 No.1

What is the Free Software Foundation?

by Richard M. Stallman

The Free Software Foundation is dedicated to eliminating restrictions on copying, redistribution, understanding and modification of software.

The word "free" in our name does not refer to price; it refers to freedom. First, the freedom to copy a program and redistribute it to your neighbors, so that they can use it as well as you. Second, the freedom to change a program, so that you can control it instead of it controlling you; for this, the source code must be made available to you.



FREE SOFTWARE FOUNDATION OF INDIA

- FSF India is a non-profit organisation committed to advocating, promoting and propagating the use and development of swatantra software in India.
- <https://fsf.org.in/>

- GNU Project.
- Programming languages, such as Perl and Python,
- and Internet software, such as BIND, sendmail, and Apache, demonstrated that free software was already ubiquitous and influential.

Linux arrives in 1991

- Linus Torvalds,
 - 21 year old Finnish computer science student.
 - Finish wunderkid

"I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones. This has been brewing since april, and is starting to get ready. I'd like any feedback on things people like/dislike in minix, as my OS resembles it somewhat (same physical layout of the file-system (due to practical reasons) among other things).

I've currently ported bash(1.08) and gcc(1.40), and things seem to work. This implies that I'll get something practical within a few months [...] Yes - it's free of any minix code, and it has a multi-threaded fs. It is NOT protable (uses 386 task switching etc), and it probably never will support anything other than AT-harddisks, as that's all I have :-).

[...] It's mostly in C, but most people wouldn't call what I write C. It uses every conceivable feature of the 386 I could find, as it was also a project to teach me about the 386. As already mentioned, it uses a MMU, for both paging (not to disk yet) and segmentation. It's the segmentation that makes it REALLY 386 dependent (every task has a 64Mb segment for code & data - max 64 tasks in 4Gb. Anybody who needs more than 64Mb/task - tough cookies). [...] Some of my "C"-files (specifically mm.c) are almost as much assembler as C. [...] Unlike minix, I also happen to LIKE interrupts, so interrupts are handled without trying to hide the reason behind them"

Linux arrives in 1991

- In 1991, the last major essential component of a Unix-like system was developed:
 - [Linux](#), the free kernel written by [Linus Torvalds](#),
 - Other people joined to collaboration to create the GNU/Linux operating system.
 - It was proprietary at first, but in 1992, he made it free software;
 - Combined Linux with the not-quite-complete GNU system resulted in a complete free operating system.
 - a substantial job in itself, of course..
 - [GNU/Linux](#), is a combination of the GNU system with Linux as the kernel.
- “Please don't fall into the practice of calling the whole system “Linux,” since that means attributing our work to someone else.”

Threats to Free Software

- Secret hardware specification.
 - To write free software to communicate with a device if there is no specification to use the device.
- Proprietary software in classroom
 - Teaching young students only the proprietary software makes the whole ecosystem dependent on it.

Counter culture

- It is a mistake to associate GNU with the term “open source”—that term was coined in 1998 by people who disagree with the Free Software Movement's ethical values.
- They use it to promote an amoral approach to the same field.
- Open source developers were frequently characterized as “hobby” developers (most famously in Bill Gates’s 1976 “Open Letter to Hobbyists,” which we’ll get to later), because the assumption was that only companies could make “real” software.
- Once companies started using open source for commercial purposes, and people realized that these “hobby projects” were able to compete with the software made by paid employees, scholars had to come up with a new framework to explain this behaviour.
- By contrast, the open source projects attracting attention in the late 1990s and early 2000s
 - the Linux kernel, which powers operating systems;
 - Apache, an HTTP server;
 - FreeBSD, an operating system;
 - GNOME, a desktop environmentwere produced by distributed groups of developers that transcended employer affiliations.

From: Free software

To: Open source Software

Use, Learn, Improve, Share

The Freely Redistributable Software Conference

- First conference on free software
 - In 1996 sponsored by the Free Software Foundation
 - With keynotes from **Linus Torvalds** and **Richard Stallman**
 - The dividing line separating Linux developers from GNU/Linux developers was largely generational.
 - Stallman representing the older, wiser contingent of ITS/Unix
 - Torvalds representing the younger, more energetic crop of Linux hackers
 - A welcome event for **Eric S. Raymond**
 - Active in the GNU Project during the 1980s

The Cathedral and the Bazaar

- The essay written by Eric S. Raymond in 1998.
- It contrasted the management styles of the GNU Project with the management style of Torvalds
- The paper drew its name from Raymond's central analogy.
 - GNU programs were "cathedrals," impressive, centrally planned monuments to the hacker ethic, built to stand the test of time.
 - Linux, on the other hand, was more like "a great babbling bazaar," a software program developed through the loose decentralizing dynamics of the Internet.
 - 'cathedral' (centralized, closed, controlled) and 'bazaar' (decentralized, open, peer-review-intensive)

Development Style

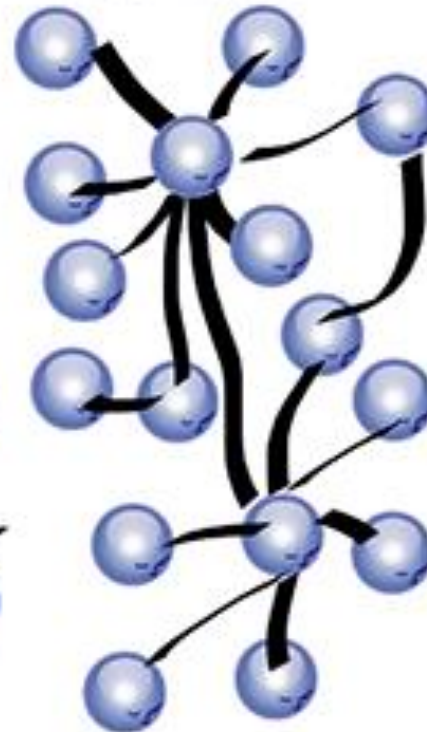
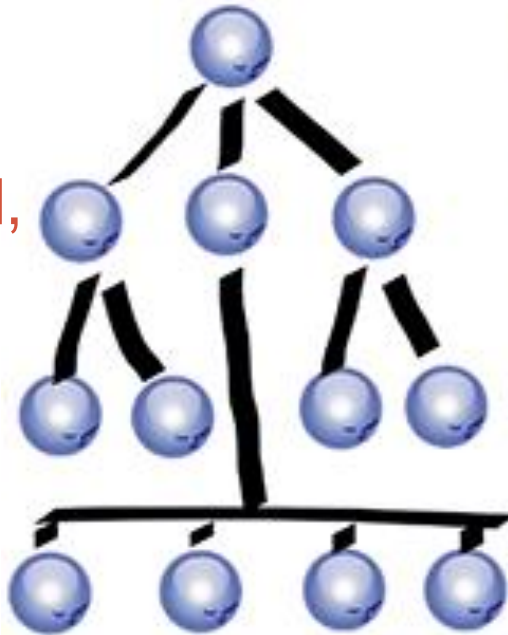
GNU

Linux

Cathedral

Bazaar

Centralized,
Closed,
Controlled



Decentralized,
Open,
Peer-review-
intensive

Cathedral vs Bazaar

- Rather a comparison of Stallman's and Torvalds's management styles.
- Stallman belongs to the cathedral style
- Torvalds is like a dinner-party host.
 - Let others lead the discussions and step in only when needed
 - Management means not to impose control but keep the ideas flowing.

Bazaar style attracted more attention:

Too many cooks spoil the broth
really!!!

- Brooks' observation an incremental person when added to a project makes it take more, not less time.
 - In 1975 book, *The Mythical Man-Month*,
 - Brooks' Law held that adding developers to a project only resulted in further project delays.
- Bazaar style software development averse
 - Raymond observed it as revolutionary.
- Torvalds's development style
 - Resulted in better software

Netscape Communications releases source code of its web browser

- Influenced by Raymond's writings
- Web browser source code released
 - codenamed "Mozilla"
- Netscape Navigator market shares of up to 80% in 1996.
- Microsoft bundles Internet Explorer with Windows 95
 - Netscape Navigator to gradually lose market share.
- In 1998 Netscape announced to distribute a large part of its navigator code as free software
 - launching the Mozilla project

R the programming language for statistics released as Free software

- Ross Ihaka and Robert Gentleman, authors of the language
- Convinced by statistician Martin Mächler to release it under the GNU General Public License

Free Software turned off businesses

- the message still wasn't getting through
 - That the word "free" in free software stood for freedom and not price
- The term interpreted as "zero cost,"
- Different options
 - Sourceware
 - Source code available
 - freely distributable
 - cooperatively developed

From: Sourceware, Source code available to: Open Source Software



Christine Peterson

executive director at Foresight Institute

Role of free software in running the Internet

- Growing commercial interest in
 - GNU/Linux, a complete operating system
 - Apache, a popular free software web server,
- Need to publicize

the Freeware Summit



- In 1998 Hosted by Tim O'Reilly,
 - publisher of O'Reilly & Associates
- the first "Open Source Summit."
- At the end of the one-day conference, attendees put the three terms-
free software, open source, or sourceware-
up for voting.
- According to O'Reilly, 9 out of the 15 attendees voted for "open source."
- In 1998, Eric Raymond proposed launching the [Open Source Initiative](#), or OSI, an organization that would police the use of the term "open source" and provide a definition for companies interested in making their own programs.
- **Profit motives separates OS movements**

Tiemann a summit attendee says.

"Open source positioned itself as being business friendly and business sensible. Free software positioned itself as morally righteous. For better or worse we figured it was more advantageous to align with the open source crowd."

The Linux world show in 1999

- Stallman preferred to stick with the term free software.
- But he didn't reject the idea of open source per se.



- Open Source Initiative (OSI) is a public benefit corporation
 - Founded in 1998
- Actively engaged in building open source community

<http://www.opensource.org/>

- In 1998, the Open Source Initiative (OSI) was founded
- Adopted the term *open source software* as a brand for introducing free software into the business world,
- To avoid the ambiguous term *free software* (which can be misinterpreted as free to use as well as free of charge).

The two concepts are only culturally distinct: Ideology vs Pragmatics

- The definitions of “free” and “open source” software are
 - only that the two concepts are culturally distinct.
- Free software is a “a movement for freedom and justice.”
 - code should be liberated from proprietary control.
- Free software developers has this ideology
 - They don't like to mix these terms.
- Open source advocates set pragmatic goals
 - such as standard Open source licenses offer greater liberties than the GPL.
 - Allow mixing proprietary and open-source software.”

Difference in values

- .
- For the Open Source movement
 - whether software should be open source
 - a practical question, not an ethical one.
 - non-free software is a suboptimal solution
- For the Free Software movement
 - Non-free software is a social problem and free software is the solution.

A viewpoint worth consideration

- We are not against the Open Source movement, but we don't want to be lumped in with them.
- We acknowledge that they have contributed to our community, but we created this community, and we want people to know this.
- We want people to associate our achievements with our values and our philosophy, not with theirs.
- We want to be heard, not obscured behind a group with different views.
- To prevent people from thinking we are part of them, we take pains to avoid using the word “open” to describe free software, or its contrary, “closed,” in talking about non-free software.
- So please mention the Free Software movement when you talk about the work we have done, and the software we have developed—such as the [GNU/Linux](#) operating system.

FLOSS Projects Timeline

- 1987 PERL
- 1990 Python
- 1992 GNU/Linux
- 1994 PHP
- 1994 Blender
- 1995 MySQL
- 1995 Apache web server
- 1995 R
- 1996 cURL
- 1996 PostGreSQL
- 1998 OpenSSL

2001 Eclipse
2002 OpenOffice
2003 Firefox,
WordPress
2004 Ubuntu
2005 Git
2006 Apache Hadoop
2008 Android
2013 Docker
2014 vue.js
2015 Kubernetes
2015 Tensor-Flow
2021 Apache MXNet

- The major players in computing—such as Amazon, Apple, Facebook, Google, IBM, Intel, Microsoft incorporate FLOSS .
- Among the leaders in the IT market, only Microsoft has positioned itself in clear opposition to free software and particularly software distributed under the GPL licence.

FLOSS Evolution

- Development Processes
- Hosting Platforms

Hosting Platforms (the tarball approach)

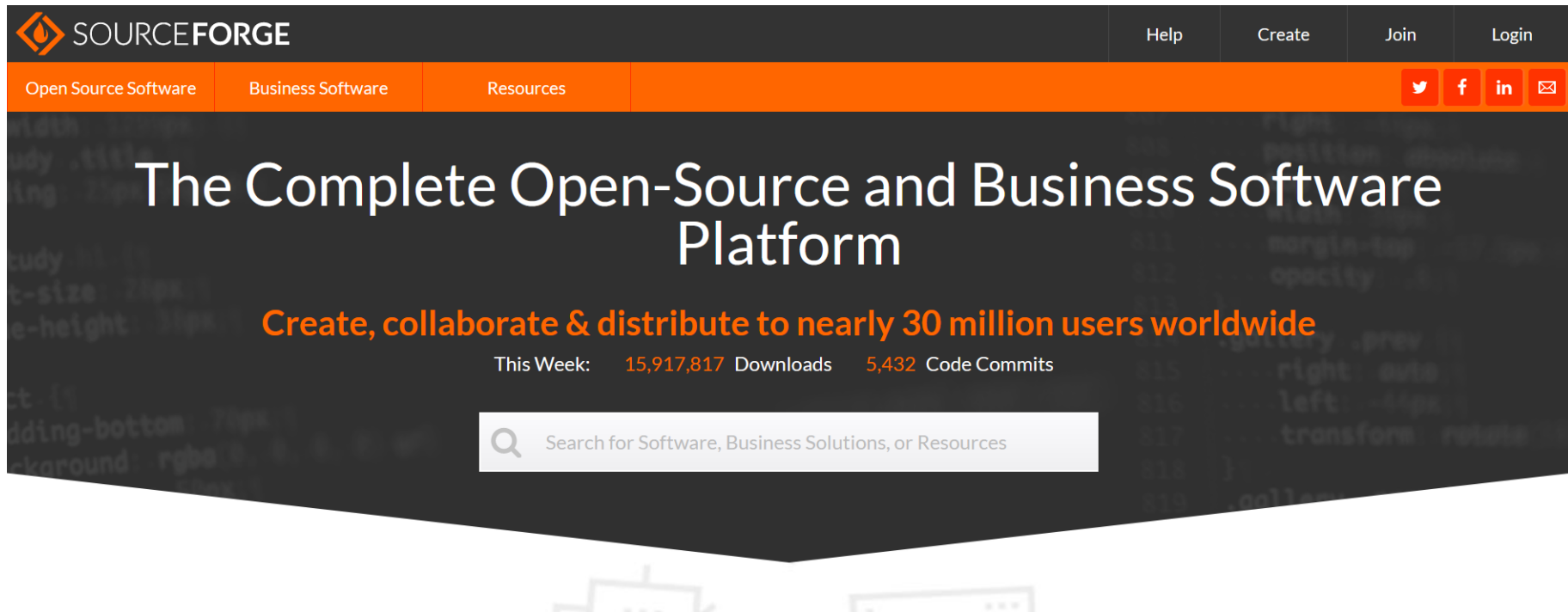
- Open source code was distributed in the form of a .tar file as a “tarball”.
- Developers used mailing lists to communicate and collaborate.

Hosting Platforms (source management systems)

- CVS (Concurrent Version System) in 1980s.
 - It uses a central repository accessed through a client/server system.
 - The site administrator decides who has access to the repository, or to which parts of the repository.
- Subversion

Development support sites

- <https://sourceforge.net/>



Development support sites

- <https://www.openhub.net/>

SYNOPSIS® | Black Duck Open Hub

Follow @OH

Sign In

Join Now

Projects

People

Organizations

Tools

Blog

Discover, Track and Compare Open Source

Search Projects...



Indexing 498,435 open source projects

Join Now


- ◆ Claim your contributions
- ◆ Manage your project's data
- ◆ Highlight your use of FOSS

What's New



Development support sites

- <https://savannah.gnu.org/>



Not Logged in

Login
New User

This Page

Language
Clean Reload
Printer Version

Search

in Projects ▾

Search

Hosted Projects

Hosting requirements
Register New Project
Full List
Contributors Wanted
Statistics

Site Help

User Docs: FAQ

Welcome to **Savannah**, the software forge for people committed to free software:

- We host free projects that run on free operating systems and without any proprietary software dependencies.
[Hosting requirements](#)
- Our service runs with 100% free software, including itself.
- [Why choose Savannah?](#)

Savannah aims to be a central point for development, maintenance and distribution of official GNU software. In addition, for projects that support free software but are not part of GNU, we provide savannah.nongnu.org.

If you would like to use Savannah to host your project, then go to the **Register New Project** menu entry.

We strongly recommend all Savannah users subscribe to this mailing list:

- [savannah-announce](#): low-volume notifications of important issues and changes at Savannah.

And this mailing list is a place for Savannah users to communicate and ask questions:

- [savannah-users](#): help with using Savannah in general (not with a specific project).

Happy hacking!

Latest News

git server updated

posted by rwp, Fri 31 Dec 2021 04:08:42 AM UTC - 0 replies

Savannah Statistics

25537 registered users
3836 hosted projects
- 0 Extras for official GNU software
- 26 GUG
- 3341 non-GNU software and documentation
- 401 Official GNU software
- 30 www.gnu.org portions
- 38 www.gnu.org translation teams
+ 11 registrations pending

Help Wanted

- 67 Developer
- 7 Project Manager
- 4 Unix Admin
- 16 Doc Writer
- 13 Tester
- 1 Support Manager
- 11 Graphic/Other Designer
- 25 Translator
- 4 Other

Most Popular Items

- bug #55093: Add LUKS2 support, 1039 votes
- sr #108134: Use XDG dirs instead of \$HOME, 450 votes
- bug #23435: Flash content rendered above..., 377 votes

Hosting Platforms

(The Arrival of Git)

- Git, the most popular version control system today, was released in 2005.
 - Created by Linus Torvalds
- Before that, developers primarily used *centralized* version control systems, such as Subversion or CVS, if they used version control at all.
- These systems weren't designed for decentralized collaboration at scale.

Hosting Platforms (The Arrival of Github)

- GitHub founded in 2008
- GitHub itself is not open source.
 - acquired by Microsoft in 2018
 - Microsoft CEO called Open Source “a cancer that attaches itself in an intellectual property sense to everything it touches” in 2001
- GitHub’s next-closest competitor, GitLab, launched in 2011, is open source.
- However, GitHub is still by far the dominant market player.

Hosting Platforms

Convenience triumphing over personal values

- Developers choose GitHub over GitLab
 - comfortable to use, and
 - better tool support.
- Developers doesn't seem much care about using the distinction between free and open source software.
- Sharing code is more like sharing any other form of content nowadays.
- For them developing solutions looks more important than the business model of the supporting platform.

Development Processes

- Community Supported Projects
 - Projects' described as a *commons*, meaning a resource that is owned, used, and managed by a community.
 - The Cathedral Style
 - The Bazaar Style
- Foundation Backed Projects
- Corporate Supported Projects

Community Based Projects

- Successful free/libre open source projects operated like communities,
 - relying upon self-governed rules, rather than outside intervention,
- The community members are volunteers motivated
 - For fulfilling a need,
 - For learning, for earning reputation in the community as well as
 - By psychological needs involving self-fulfillment, basic fun and enjoyment

The Cathedral Style

- Highly centralised
 - a few people are responsible for the software's design and implementation.
- Well defined tasks
 - Volunteers assigned tasks.
- Strictly scheduled releases
- For example, GNU

The Bazaar Style

- No absolute authority to control the processes or to strictly plan what has to happen.
- At the same time, participants' roles can change continuously (sellers can become clients) and with no outward indication.
- For example, Linux

Foundation Based Projects

- Clustered projects sponsored by foundation
 - Common license model
 - Mix of paid and volunteer developers
- Apache; Eclipse, LibreOffice
- Meritocracy
 - E.g. Contributors with important and continuous contributions only can become members of the Apache Foundation.
- Motivation
 - To improve some system
 - To solve a problem
- Funding
 - Public funding
 - Private not-for-profit funding
 - Indirect funding

Corporate involvement in OSS Projects

- Sharing the burden
 - Developing solutions jointly for common problems
 - The advantages are obvious: benefits for all at divided costs.
- In other cases, companies may also seek to collaborate in free projects promoted by volunteers,
 - The GNOME Foundation
- make volunteers collaborate with their own free projects.
 - Red Hat

Corporate involvement in OSS Projects

- Usually those external entities determine how the funds are to be spent, and where the development efforts are headed.
- The developer just follows those more or less strict guidelines.
- In some sense, it could be said that the external entity 'sponsors' the development of some given open source software.
- Three models, according to who funds the project and why.
 - public funding,
 - 'needed improvement' funding, and
 - indirect funding.
- Today each of these kinds of free software business is practiced by a number of corporations.

Practice of supporting the OSS developers

- If open source code is like infrastructure, we want to measure its value based on a combination of
 - *dependencies* (Who uses the code?) and
 - *substitutability* (If this code disappeared, how hard would it be to replace?).
- Sponsorships managed through Patreon and GitHub Sponsors (a platform-native sponsorship product launched by GitHub in 2019).

A newer practice of open source developers raising money

- from their fans and users,
- independent not just of a salaried job but of a specific project.

Sponsors



Sponsoring



Sindre Sorhus

sindresorhus

Follow

♡ Sponsor

<https://github.com/sponsors/sindresorhus>