

Free & Open Source Software

Everything you need to know about: **FOSS 101**



Google Developer Student Clubs
PESUECC

```
$ whoami
```

```
$ whoami
```

```
| polarhive@earth
```

```
|-----
```

```
| OS: GNU/Linux
```

```
| Semester: 3.0
```

```
| Uptime: 19.99 years
```

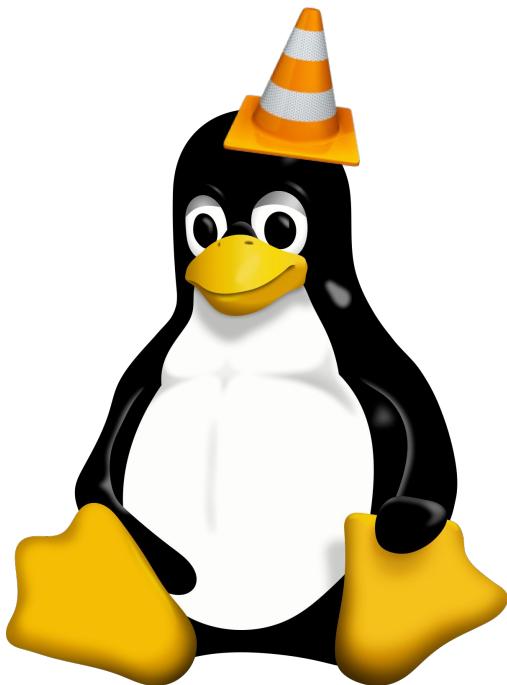
```
| Member: GDSC, HSP, ACM
```

```
| Pronouns: he/him
```

```
| Location: Bengaluru, India
```

```
| PGP: 0x78DBFBE8B325A5A9
```

What is **FOSS** all about?



Before we begin

Before we begin

I know you had lunch but...

Before we begin, grab a snack

I know you had lunch but...

Before we begin, grab a snack

Imagine you're hungry. What do you do?

You open your favorite food ordering app and **order** something from Swiggy or Zomato.

Before we begin, grab a snack

Imagine you're hungry. What do you do?

You open your favorite food ordering app and **order** something from Swiggy or Zomato.

If you're **alone**, you might gather ingredients and follow a **recipe** or ask someone to cook for you.

food.

food. tangible.

food. tangible. a real-world thing

food. tangible. a real-world thing

If I take your ice-cream



food. tangible. a real-world thing

If I take your ice-cream — I have it and you don't.



food. tangible. a real-world thing

If I take your ice-cream — I have it and you don't.

It can only be at one place at one time.



food. tangible. a real-world thing

If I take your ice-cream — I have it and you don't.

It can only be at one place at one time. Unlike software



food. tangible. a real-world thing

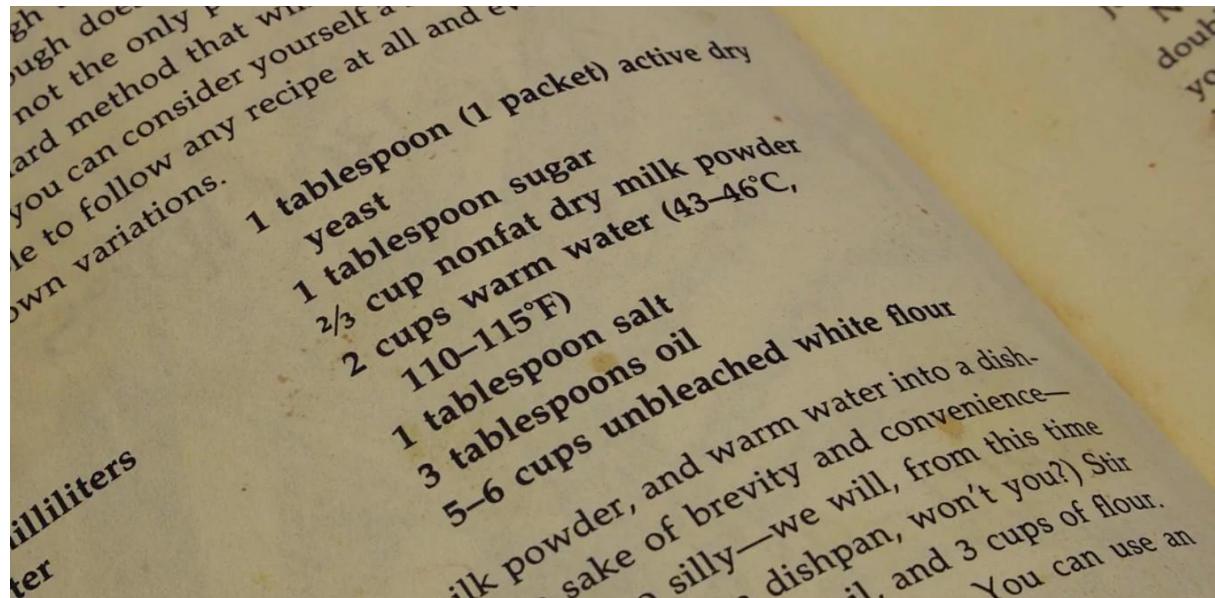
If I take your ice-cream — I have it and you don't.

It can only be at one place at one time. Unlike software; I can't make infinite copies of it and distribute it.



Recipes?

- Essentially, to make any kind of food the person cooking would follow a recipe:



Recipes?

Easy to read instructions that anyone could make use of!

- What ingredients?
- Utensils I'd need?
- How much time to cook?
- How much it'd cost if I had to order-in?

You'd know what's in it

- How many Calories?
- Potential Allergies?
- Medicinal and health benefits?
- Suits your taste?

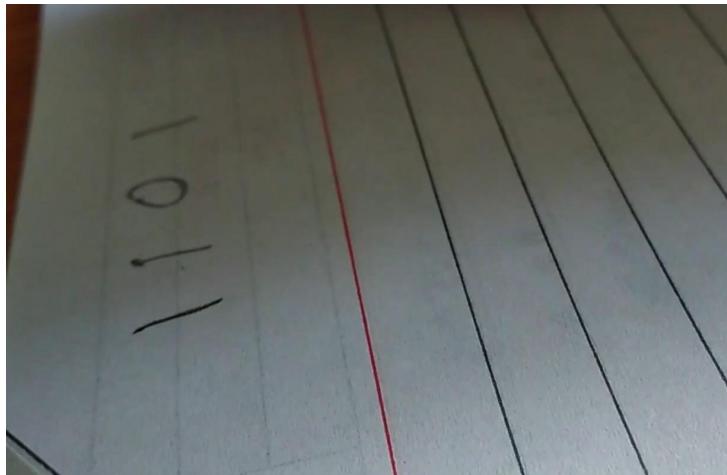
What if you **didn't**?

- Your friend orders a pizza, and you're allergic to a specific kind of mushroom.
- Somehow, you didn't check and ended up being forced to eat it.

You'd probably feel sick and be upset at your friend for not informing you about the source or contents of the ingredients.

Patterns

Imagine I have a piece of paper with the pattern **1101** written on it. If I control a lightbulb to represent this pattern, someone who has the lightbulb's switch can accurately recreate the pattern reading off the paper.



Pixels on your screen

Patterns? Look at Games, Video, Simple programs that change an LED sign board display!



What is software made of, exactly?

beep/bop

What is software made of, exactly?

01101110011001010111011001100101011100100010000001100111011011110110111
00110111001100001001000000110011101101001011101100110010100100000011
11001011011110111010100100000011101010101110000

Any Clues?

What is software made of, exactly?

0110111001100101**0**111011001100101**0**1110010**00**10000001100111**0**1101111**0**110111
1011011100110111001100001**00**10000001100111**0**1101001**0**111011001100101**0010**
000001111001**0**1101111**0**1110101**00**10000001110101**0**1110000

No?

What is software made of, exactly?

0110111001100101**0**111011001100101**0**1110010**00**10000001100111**0**1101111**0**110111
1011011100110111001100001**00**10000001100111**0**1101001**0**111011001100101**0010**
000001111001**0**1101111**0**1110101**00**10000001110101**0**1110000

24 chars

What is software made of, exactly?

**01101110011001010111011001100101011100100010000001100111011011110110111
101101110011011100110000100100000011001110110100101110110011001010010
0000011110010110111101110101001000000111010101110000**

Some repeat!

What is software made of, exactly?

```
01101110011001010111011001100101011100100010000001100111011011110110111  
101101110011011100110000100100000011001110110100101110110011001010010  
0000011110010110111101110101001000000111010101110000
```

Demo time :D



What you need – Python

link: polarhive.net/talks/qdsc-foss-101



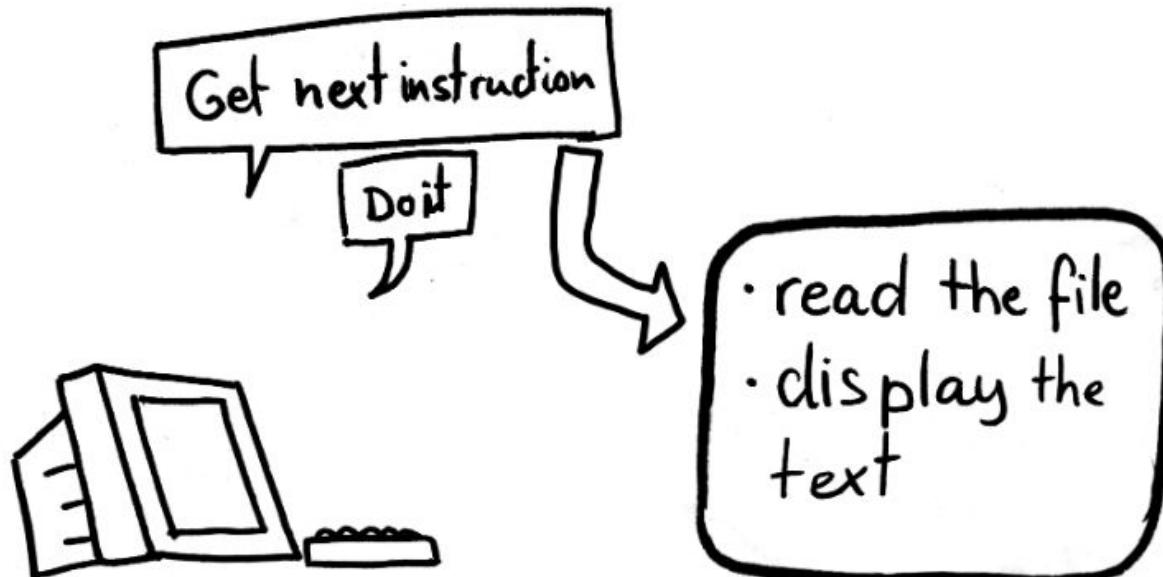
Bit Manipulation:

01101110011001010111011001100101011100100010000001100111011011110110111
101101110011011100110000100100000011001110110100101110110011001010010
0000011110010110111101110101001000000111010101110000

<https://lingojam.com/BinaryTranslator>

A universal machine

It's only job is to read the **next bit** of instruction and **process** it.



Computer Speak.

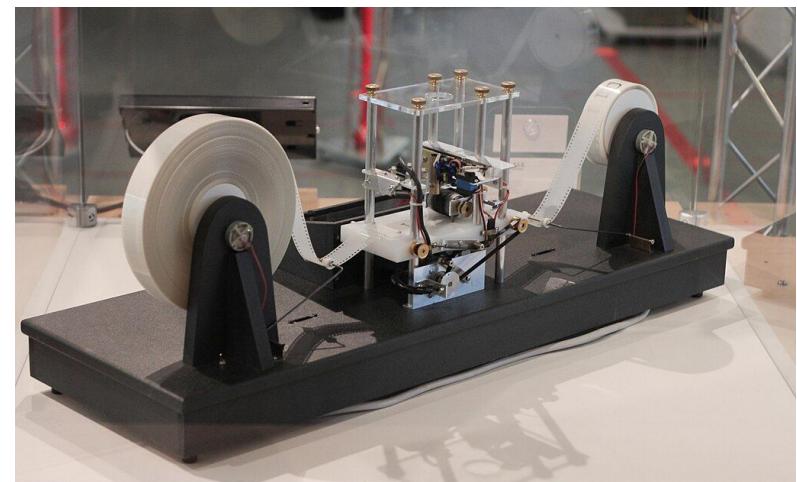
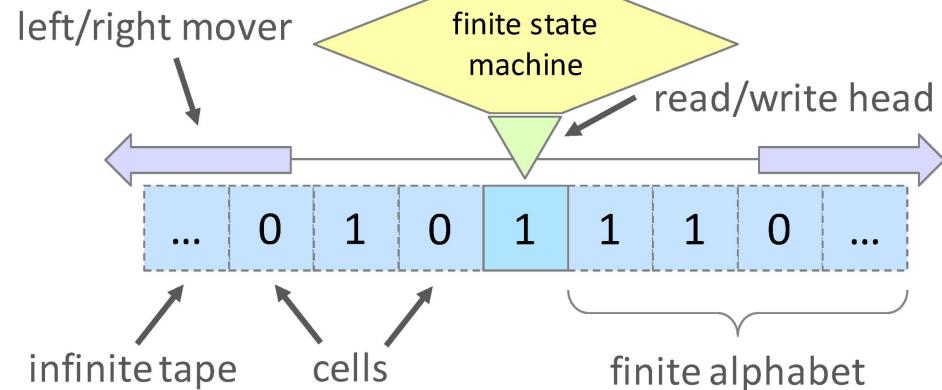
Computers are made up of electronic **circuits**, which run on machine code instructions, read in binary, **1's** and **0's**

Computer Speak.

Computers are made up of electronic **circuits**, which run on machine code instructions, read in binary, **1's** and **0's**

These instructions, which can often be extremely long sequences of bits, and are collectively called programs.

Familiar?



A universal machine

> A computer does not have a **mind of its own**.

If you give it a piece of code that triggers a **bomb**, wipes your hard-drive, or **spy** on you — it will simply, run it.

> It will simply, run it.

<wannacry, crowdstrike computers crashing because of faulty code>



Let's run a binary-blob

link: polarhive.net/talks/qdsc-foss-101



Let's run a binary-blob

Demo: <open vim, show the compiled binary file garbage>

\$ source-code

Source code is the recipe (**often** kept **secret**) is a **prerequisite** that helps you build new-programs and modify existing ones.

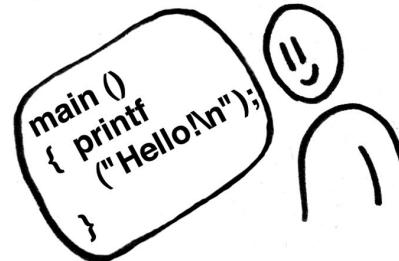
The idea of free-software

All software comes with its binary pre-packaged. A machine **does not need source code for it to run**, but you need it to re-make software and edit it to your liking.

non free software :



free software:



The idea of free-software

All software comes with its binary pre-packaged. A machine **does not need source code for it to run**, but you need it to re-make software and edit it to your liking.

You do-not like **ads** on Spotify, YouTube or even your calculator app. These apps don't really come with source code, instructions on how to block these anti-features.

The idea of free-software

All software comes with its binary pre-packaged. A machine **does not need source code for it to run**, but you need it to re-make software and edit it to your liking.

You do-not like **ads** on Spotify, YouTube or even your calculator app. These apps don't really come with source code, instructions on how to block these anti-features.

In a way, by taking your **freedom**, because you know an ad-free version of the same app's binary code exists on your very machine.

\$ bad things

In the previous example if I **blindly** showed you the binary-code to the sample program and **did not** provide source code. It would be nearly impossible for a beginner to figure out what it was doing.

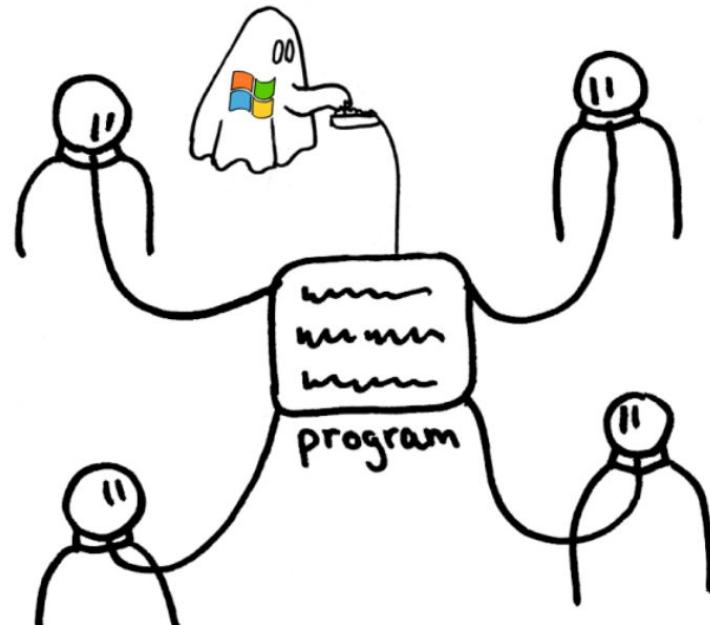
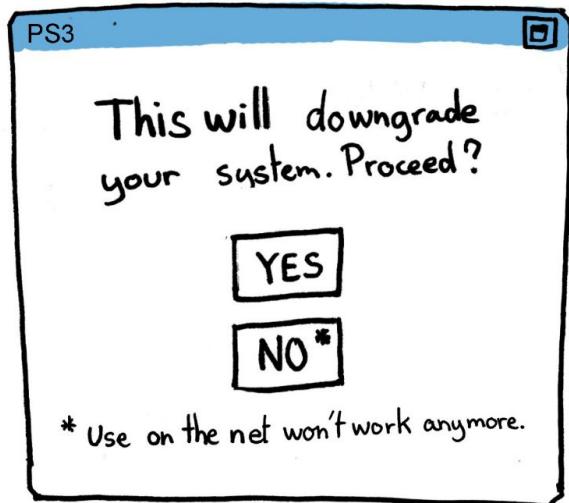
What **prevents** bad things like this from happening??

> Good? Bad? Your computer will simply, run it

The image is a collage of screenshots from different software interfaces, likely illustrating various types of cyber threats or software interfaces:

- Wana Decryptor (Ransomware):** A screenshot showing a red-themed interface with a padlock icon. It displays a message: "Oops, you're locked!" and "What Happened to My Files?". It shows a progress bar for decryption and a timer: "Payment will be raised on 5/16/2017 00:47:55" and "Time Left 02:23:57:37". It also includes sections for "How Do I Pay?" and "Check Payment".
- Steam Account Alert:** A screenshot from Steam showing an account has been locked for posting or upvoting suspicious content. It provides instructions to secure the account and details about the lock.
- Adobe Creative Cloud Pricing:** Screenshots for Adobe Lightroom (Lr) and Photoshop (Ps) showing monthly subscription plans at \$9.99 and \$20.99 respectively. Both offer a free month if paid annually.
- Adobe Creative Suite:** A screenshot for Adobe Premiere Pro (Pr) showing a monthly plan at \$20.99/mo. It includes a note about professional video and film editing.
- All Apps:** A screenshot for Adobe Creative Cloud showing a monthly plan at \$52.99/mo. It includes a note about getting all 20+ creative desktop and mobile apps including Photoshop, Illustrator, InDesign, Premiere Pro, and XD.
- CrowdStrike Flock:** A screenshot of a mobile device displaying a restart screen with the text: "Your device ran into a problem and needs to restart. We're just collecting some error info, and then we'll restart for you." It shows a QR code and the CrowdStrike logo.

Either the **user** controls the program or the **program** controls the **user**



This is called **proprietary** software

The **antithesis** to FOSS software. Software that is solely owned and distributed by the **proprietor** usually — <\$MEGACORPs>

Microsoft Office, Adobe Suite, Windows, MacOS, iOS, Gaming console OSs, etc..

Maybe I **have** source-code but what now?

Having access to source code doesn't always mean you get to **use** it or distribute it **legally**.

Maybe I **have** source-code but what now?

Having access to source code doesn't always mean you get to **use** it or distribute it **legally**.

Say you **somewhat** obtained leaked source-code for a popular app everyone is using.

By default everything **published** on the internet is copyrighted and **restricts** you from changing or modifying it without accepting a LICENSE.

This is **still** called **proprietary** software

Viewing source code might tell you **exactly** how the program is running and you get to know if it's stealing data or get a peek at security flaws in it.

But you **can not** get to modify or distribute it! Only the original authors can fix those flaws.

Copyright Law



- > Berne convention
 - Creative works in a **fixed medium** be **automatic**, and last for at least **50** years after the author's death for any work except for photographic and cinematographic works. Photographic works are tied to a minimum of 25 years.

Copyright Law

Is coding art?



Copyright Law

I guess I'll leave that question to you :))



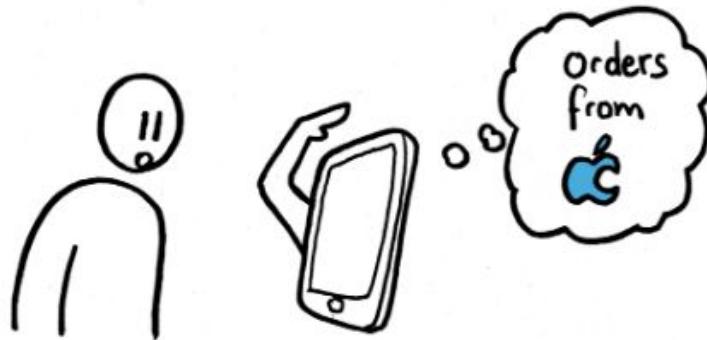
So far, my **proprietary** software works so good?

Sure, you can assume the binary-code you run **today** works as, intended.

But sooner or later, it is designed to go **obsolete**. What if the original programmer stops updates, charges extra-for software lock-ins. Like paying a subscription for heated seats on a BMW.

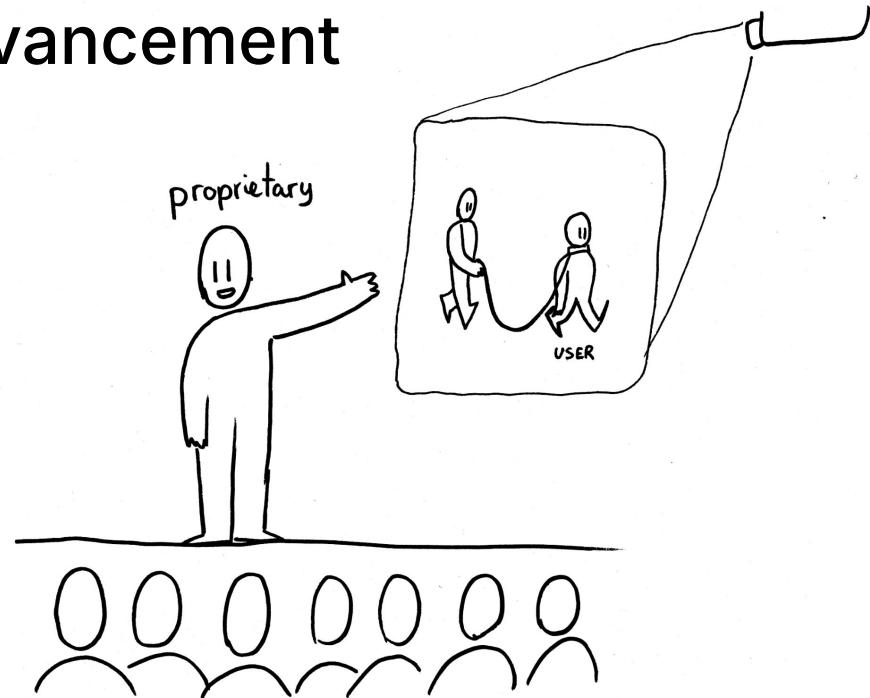
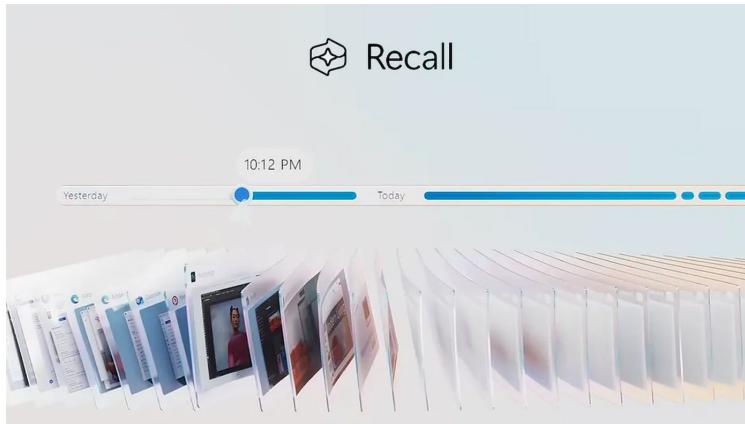
Hostile business models

When working around **proprietary** ecosystems, the manufacturer has the last say.



Presenting our latest advancement

In spying on our users!
Rebranding it as a feature™



Sounds familiar? 🤔

- Ather Pro Pack
- BMW's heated seat subscription
- Tesla's Autopilot
- Paywalling hardware features even after paying full price!
- iCloud photo scanning
- Playstore's 30% cut
- Xbox Pass
- ChatGPT ratelimits?
- Adobe Creative Cloud
- Deprecated Windows XP, 7, 8, 10, and soon 11..

Do you own your scooter?

Pro ⚡

Core performance, layered with
smarts and intelligence.

Ather Ride Assist

5 modes to match your riding style -
SmartEco™, Eco, Ride, Sport and Warp.

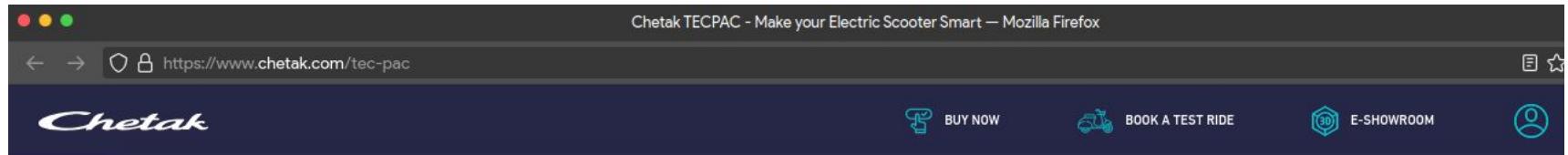
- ✓ Ride Modes ⓘ
- ✓ Auto Indicator cut-off
- ✓ Dashboard - Auto Brightness
- ✓ Call & Music Control
- ✓ AutoHold™ ⓘ
- ✓ Interactive UI ⓘ
- ✓ Coasting Regen
- ✓ Emergency Stop Signal

- ✓ TrueRange™ Display
- ✓ Guide-me-home-lights
- ✓ Vehicle FallSafe™

Ather Connect™

Atherstack™ OTA Updates

Or does your scooter own you?

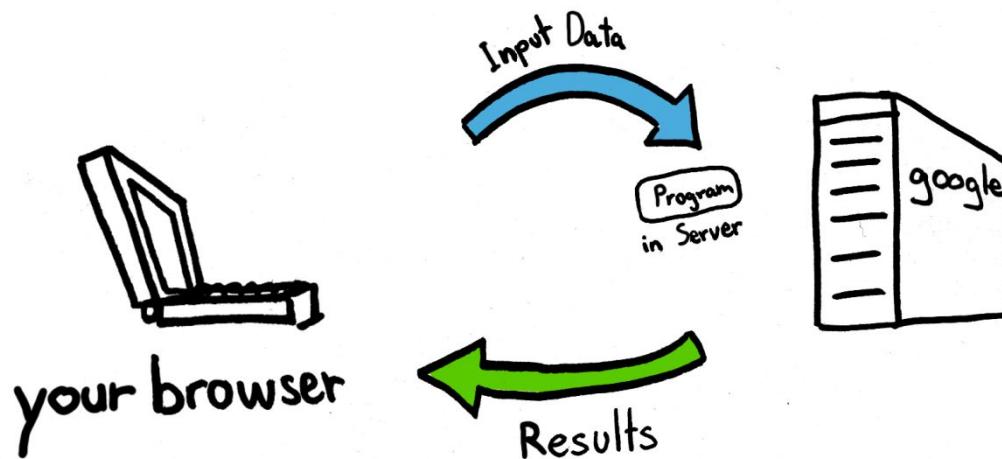


*Purchase of **TecPac** is not mandatory, however it is advised that customers purchase TecPac to avail enhanced vehicle features and capabilities of connected vehicle. If a **TecPac is not purchased, the vehicle capabilities / performance and Chetak App functionality will be limited.**

Service as a Software Substitute (SaaS)

Using a service as a substitute for running your copy of a program.

Service as a Software Substitute



Service as a Software Substitute (SaaS)

Someone sets up a network server that does certain computing activities—for instance, modifying a photo, **translating text** into another language — then invites users to let that server do their own computing for them instead of locally running software

Free? Free of cost?

LIBRE ^{not} GRATIS

Free in context refers to **freedom** or (free'd software if you'd like), software that puts the user in control, and lets you run your programs the way you want it to.

What makes software — free / non-free (**proprietary**)

The 4 **essential** software-freedoms:

What makes software — **free** / non-free (**proprietary**)

The 4 **essential** software-freedoms:

- The freedom to **run** the program as you wish, for **any** purpose

What makes software — free / non-free (**proprietary**)

The 4 **essential** software-freedoms:

- The freedom to **run** the program as you wish, for **any** purpose
- 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.

What makes software — free / non-free (**proprietary**)

The 4 **essential** software-freedoms:

- The freedom to **run** the program as you wish, for **any** purpose
- 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.
- The freedom to **redistribute** copies so you can help others.

What makes software — free / non-free (**proprietary**)

The 4 **essential** software-freedoms:

- The freedom to **run** the program as you wish, for **any** purpose
- 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.
- The freedom to **redistribute** copies so you can help others.
- The freedom to distribute copies **of your modified versions** to others.

What makes software — free / non-free (**proprietary**)

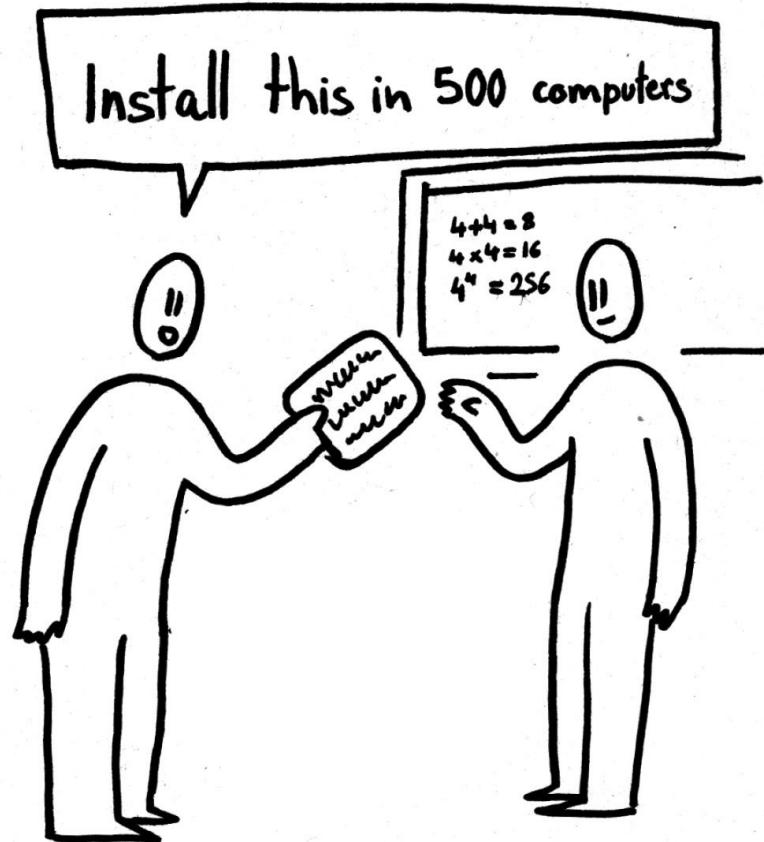
The 4 **essential** software-freedoms:

- The freedom to **run** the program as you wish, for **any** purpose
- 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.
- The freedom to **redistribute** copies so you can help others.
- The freedom to distribute copies **of your modified versions** to others. 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.

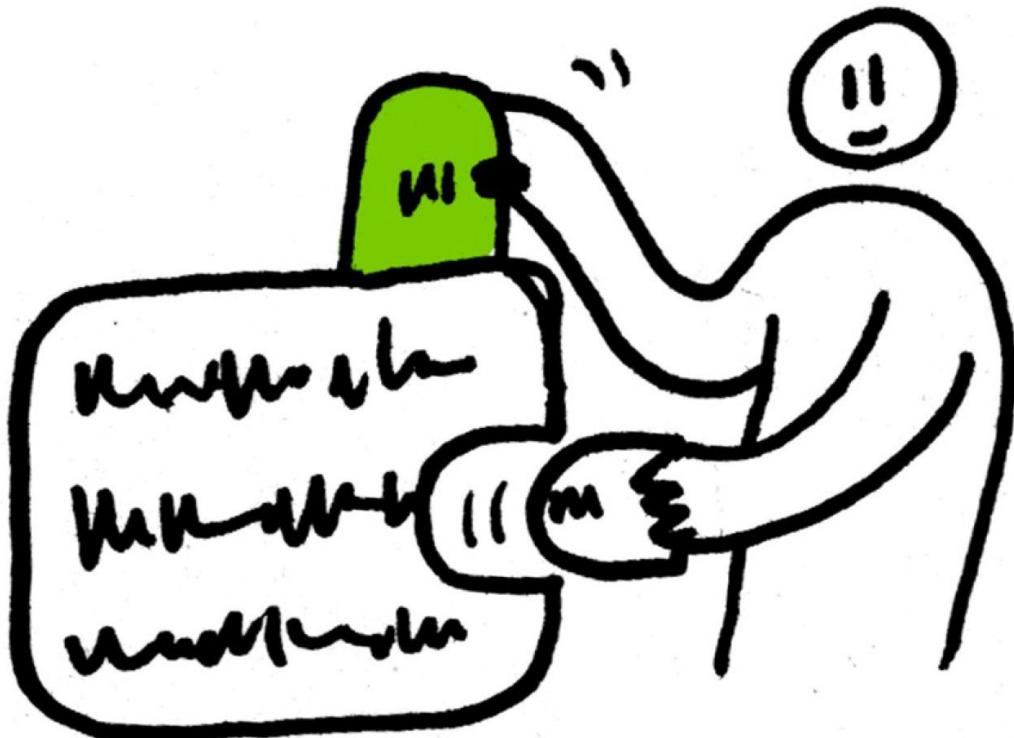
What makes software — **free** / non-free (**proprietary**)

The 4 **essential** software-freedoms: (In true programmer fashion we start counting from 0)

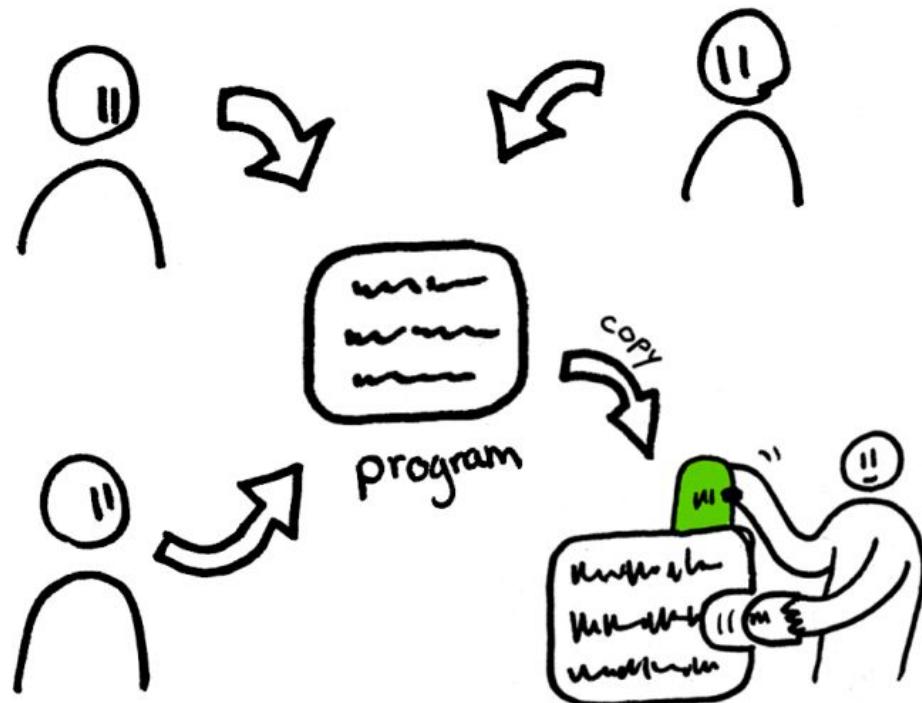
Freedom 0
Run it



Freedom 1 – change it



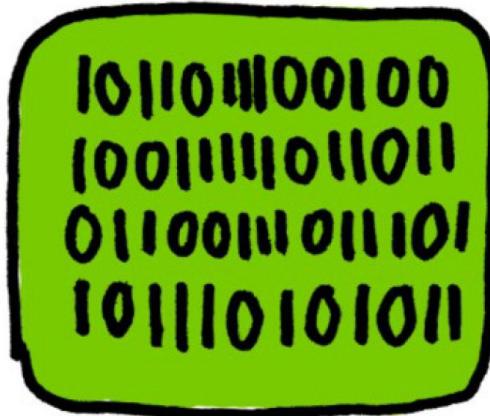
Pull source code and make changes **locally**



Access to source-code is a **precondition**

```
main()
{
printf("Hello,
world!\n");
}
```

Source



101101100100
1001111011011
0110011011101
101110101011

Executable

Distribute non-modified copies if you wish, for **any** price.



Freedom 2
Redistribute it

Distribute modified copies if you wish, for **any** price



Freedom 3
Redistribute
with changes

4 ESSENTIAL FREEDOMS OF SOFTWARE



0 To run the software when ever you wish & for what ever purpose.



1 To study the source code & make modifications to the software.



2 To give or sell copies of the software to other people.



3 To give or sell copies of your modified versions of the software.

You have the 4 essential freedoms with other useful items that belong to you. Clothing, Food, Simple Electrical Devices. But most software companies do not want you to have these essential freedoms with software, running on your various devices. Taking away your control over your own devices.

SWITCH INSTEAD TO FREE SOFTWARE!

www.GNU.org

Copyright © 2021
Jeison Yehuda Amihud

Used under the CC-BY-SA-4.0 license

Back to that **food** analogy

What if I don't know how to cook? Or I don't have time to learn how to cook?

Not a programmer ?



What if you **weren't** a cook?

You would ask someone who knew how to cook, order in or hire someone to cook for you!

Not a programmer ?



What's important is the **recipe**

If you want to make changes to your software, access to source code and the 4 freedoms is a prerequisite.

Licenses

Since all code is automatically protected by copyright, a **LICENSE** file serves as a legal document or tool to specify what you can and cannot do with the source code.

Proprietary Licenses



(aka.ms/genuine). Certain updates, support, and other services might only be offered to users of genuine Microsoft software.

6. **Updates.** The software periodically checks for system and app updates, and downloads and installs them for you. You may obtain updates only from Microsoft or authorized sources, and Microsoft may need to update your system to provide you with those updates. By accepting this agreement, you agree to receive these types of automatic updates without any additional notice.
7. **Downgrade Rights.** If you acquired a device from a manufacturer or installer with a Professional version of Windows preinstalled on it and it is configured to run in full feature mode, you

Tells you what you **can't** do with your software.

Free-software Licenses



Free as in Freedom

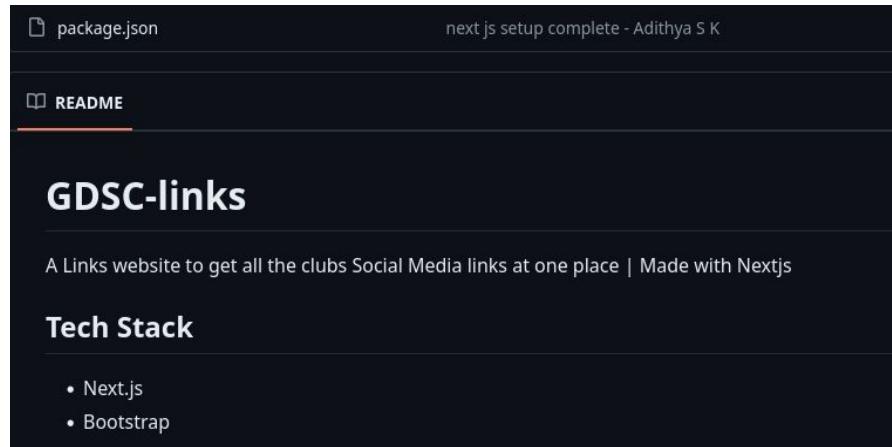


Tells you what **can** do with your software, and **legally** prevents anyone from taking away your rights

I put up my code on GitHub? So now it's FOSS?

No. If you don't **explicitly** add or mention how you are **licensing** your software, it's still proprietary software by default.

> We just had a look at the Berne convention, law.



GDSC RR repo missing a **LICENSE** file

Making it open-source but proprietary.

I put up my code on GitHub? So now it's FOSS?

No. If you don't **explicitly** add or mention how you are **licensing** your software, it's still proprietary software by default.

> We just had a look at the Berne convention, law.

A screenshot of a GitHub repository page. At the top, there are two files: "package.json" and "next.js setup complete - Adithya S K". Below them is a "README" file, which is currently selected. The content of the README file is as follows:

```
GDSC-links
A Links website to get all the clubs Social Media links at one place | Made with Nextjs
Tech Stack
• Next.js
• Bootstrap
```

A screenshot of a GitHub repository page showing the "MIT license" file. The file content is:

```
MIT License
Copyright (c) 2023 GDSC PESU
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.
```

Let's publish some F/OSS

link: choosealicense.com

Let's publish some F/OSS

link: choosealicense.com

name your project: github.com/new

Let's publish some F/OSS

link: choosealicense.com

name your project: github.com/new

pick your license: I

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Required fields are marked with an asterisk (*).

Owner * polarhive / **Repository name *** flossin 

Great repository names are short and memorable. Need inspiration? How about

License

Filter...

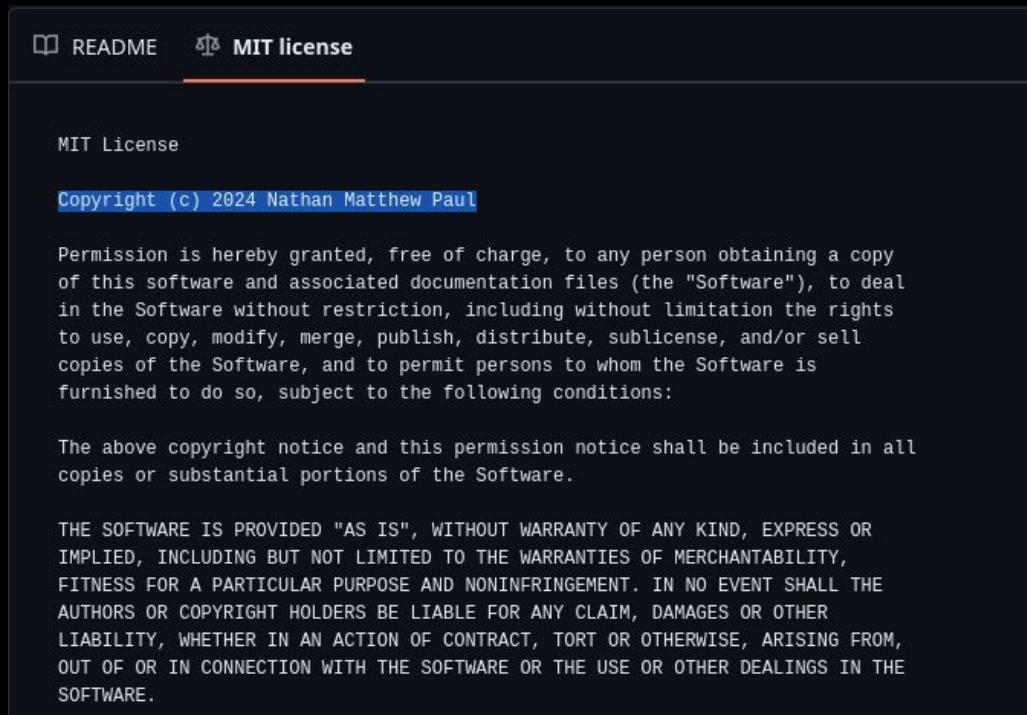
- None
- Apache License 2.0
- GNU General Public License v3.0
- MIT License
- BSD 2-Clause "Simplified" License
- BSD 3-Clause "New" or "Revised" License
- Boost Software License 1.0
- Creative Commons Zero v1.0 Universal

License: None

A license tells others what they can and can't do with your code. [Learn more about licenses](#).

Let's publish some F/OSS

Add your name to the license file:



Other ways? — SPDX identifier.

The SPDX License List includes a standardized short identifier, the full name, the license text, and a canonical permanent URL for each license and exception. spdx.org/licenses

Example:

Code

Blame

243 lines (200 loc) · 10.1 KB

Raw

```
1  # Creating a typing test tool that measures users typing speed and accuracy, providing an entertaining
2  # author: PES2UG23CS368 Nathan Matthew Paul
3  # author: PES2UG23CS371 Navneet Nayak
4  # author: PES2UG23CS381 Nevin Mathew Thomas
5  # author: PES2UG23CS390 Nilay Srivastava
6  # SPDX-License-Identifier: GPL-3.0-or-later
7  # Copyright (C) 2023
8
9  import time|
10 import threading
11 import simpleaudio as sa
12 import text_module
```

Origins: MIT AI Lab (1981)

Talented programmers such as Richard Stallman, develops **EMACS**, flourished in the AI Lab.



Origins: EMACS, the text editor

Sold \$150, per copy



GNU Emacs - GNU Project — Mozilla Firefox

Emacs 29.4 is out, download it [here!](https://www.gnu.org/savannah-checkouts/gnu/emacs/)

An extensible, customizable, free/libre text editor — and more.

At its core is an interpreter for Emacs Lisp, a dialect of the Lisp programming language with extensions to support text editing.

↓ GNU/Linux ↓ BSDs ↓ Windows ↓ MacOS

```
(defsubst hash-table-empty-p (hash-table)
  "Check whether HASH-TABLE is empty (has 0 elements)."
  (zerop (hash-table-count hash-table)))

(defsubst hash-table-keys (hash-table)
  "Return a list of keys in HASH-TABLE."
  (let ((keys '()))
    (maphash (lambda (k _v) (push k keys)) hash-table)
    keys))
```

Origins: Hacker culture

"a spirit of playful cleverness"



Origins: Hacker culture



Origins: MIT/UNIX

How did free-software come into fruition?

```
From: RMS@MIT-DZ@mit-eddie.UUCP (Richard Stallman)
Subject: new UNIX implementation
Message-ID: <771@mit-eddie.UUCP>
Date: Tue, 27-Sep-83 13:35:59 EDT
Date-Received: Thu, 29-Sep-83 07:38:11 EDT

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 to everyone who can use it. Contributions of time,
money, programs and equipment are greatly needed.

To begin with, GNU will be a kernel plus all the utilities needed to
write and run C programs: editor, shell, C compiler, linker,
assembler, and a few other things. After this we will add a text
formatter, a YACC, an Empire game, a spreadsheet, and hundreds of
other things. We hope to supply, eventually, everything useful that
normally comes with a Unix system, and anything else useful, including
```

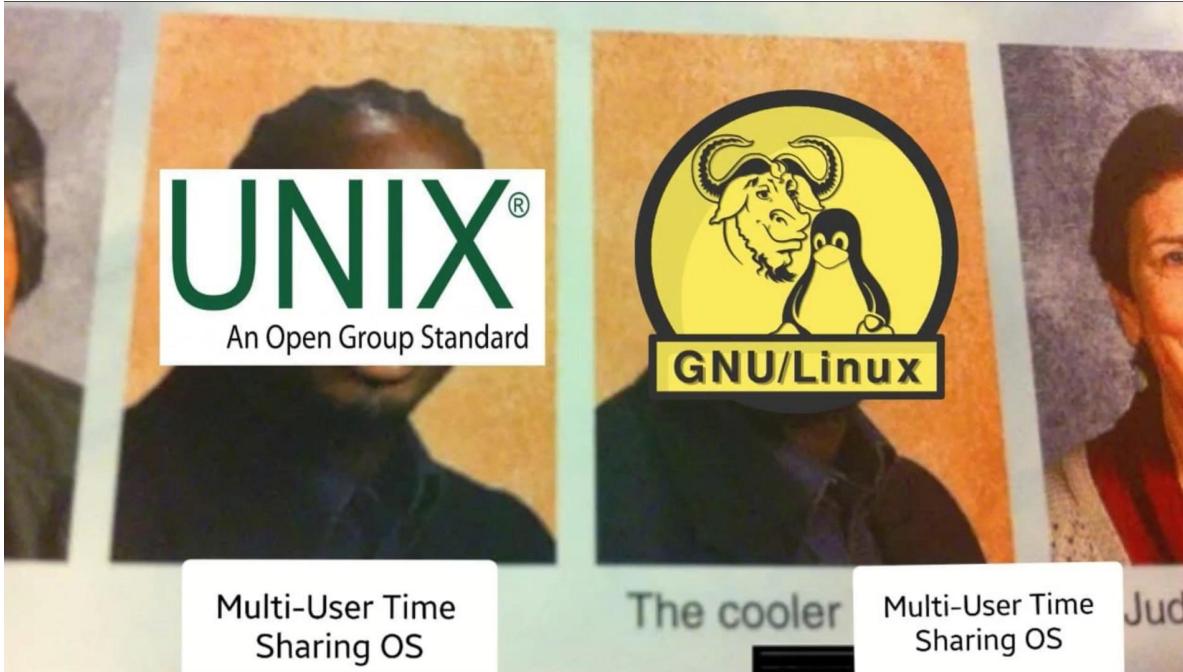
Origins: MIT/UNIX/GNU

A free-d **UNIX** Operating System.



Origins: MIT/UNIX/GNU

GNU is a play on words, a recursive acronym:



GNU IS **N**OT **U**NIX

Origins: FSF/GNU

Richard M. Stallman (RMS)



Initial Announcement

This is the original announcement of the GNU Project, posted by [Richard Stallman](#) on September 27, 1983.

The actual history of the GNU Project differs in many ways from this initial plan. For example, the beginning was delayed until January 1984. Several of the philosophical concepts of [free software](#) were not clarified until a few years later.

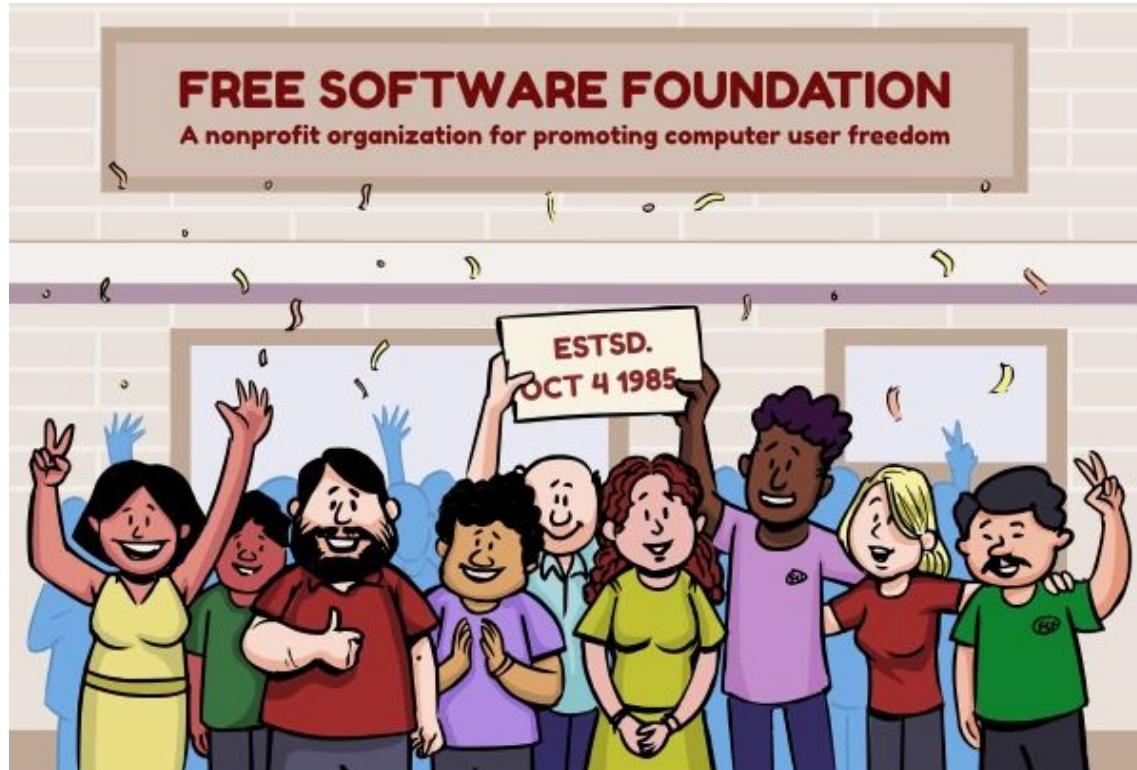
Free Unix!

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.

To begin with, GNU will be a kernel plus all the utilities needed to write and run C programs: editor, shell, C compiler, linker, assembler, and a few other things. After this we will add a text formatter, a YACC, an Empire game, a spreadsheet, and hundreds of other things. We hope to supply, eventually, everything useful that normally comes with a Unix system, and anything else useful, including on-line and hardcopy documentation.

GNU will be able to run Unix programs, but will not be identical to Unix. We will make all improvements that are convenient, based on our experience with other operating systems. In particular, we plan to have longer filenames, file version numbers, a crashproof file system, filename completion perhaps, terminal-independent display support, and eventually a Lisp-based window system through which

Origins: FSF/GNU



Free Software Foundation, the organizational home for the **GNU** Project, and provides it and other free software projects with critical infrastructure. The FSF is also the steward of the GNU family of **licenses**, including the GNU General Public License (GPL), and does GPL **compliance** work for the individual programs that make up the GNU operating system

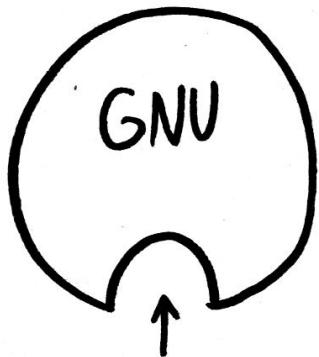
Origins: GNU/GPL/Linux



Launch of the the **GNU GPL** is at the core of the continued strength and existence of free software.

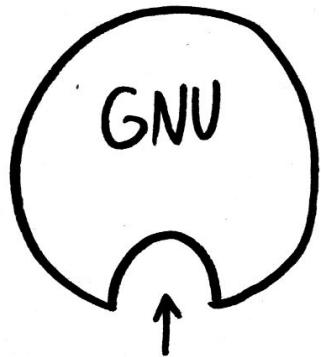
The strength of the license, and the essence of software freedom, is in its decree that any derivative or combined work is to be published under the **same** license, thereby perpetuating its freedom

Origins: GPL/GNU/Linux



The GNU project was **almost** done, except one part.

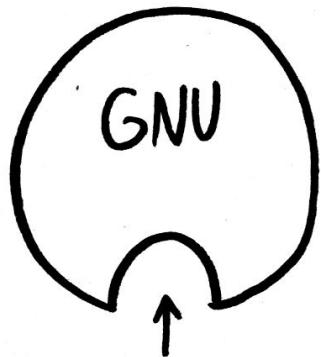
Origins: GPL/GNU/Linux



The GNU project was **almost** done, except one part.

The **kernel**

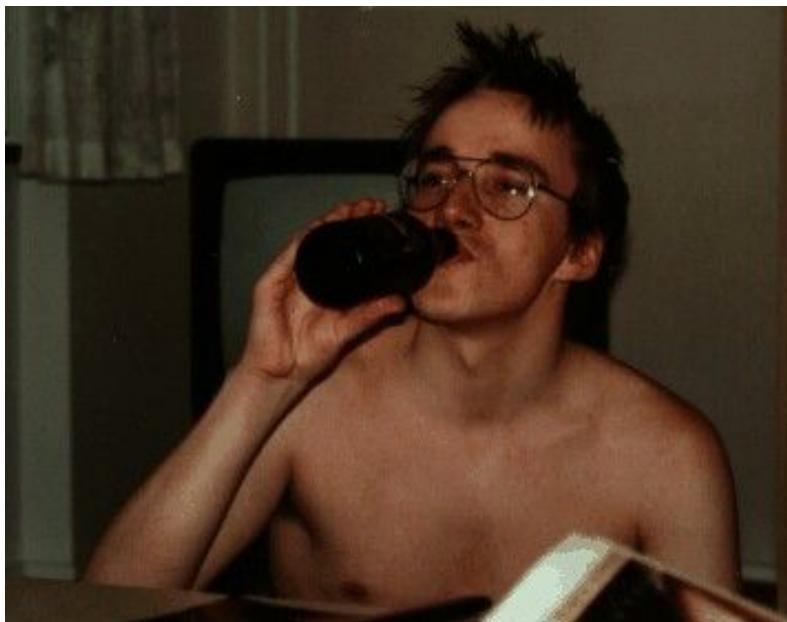
Origins: GPL/GNU/Linux



The GNU project was **almost** done, except one part.

The **kernel** is a piece of the OS that lets you interact with different kinds of hardware and manage OS resources.

Origins: GPL/GNU/Linux



Co-incidentally. At this point.

A young Finnish Computer Science bachelor was making a **toy** operating system, which was later called Linux.

Origins: FSF/GNU/Linux

Hello everybody out there using **minix**:

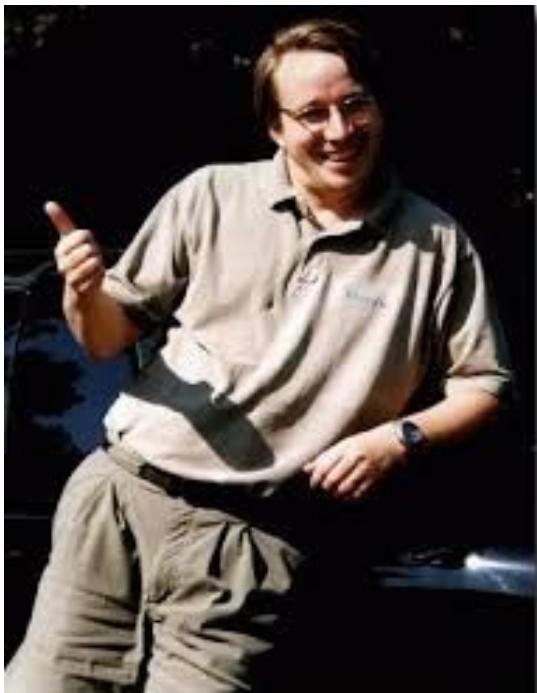
I'm doing a (free) operating system (just a hobby, won't be big and professional like **gnu**) for 386(486) AT clones. This has been brewing since april, and is starting to get ready. I'd like any feedback on things people like/dislike in minix, as my OS resembles it somewhat (same physical layout of 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, and I'd like to know what features most people would want. Any suggestions are welcome, but I won't promise I'll implement them :-)



The screenshot shows a Mozilla Firefox browser window with a green header bar. The title bar reads "MINIX: Intel's hidden in-chip operating system | ZDNET – Mozilla Firefox". The address bar shows the URL "https://www.zdnet.com/article/". The main content area is a ZDNet article titled "MINIX: Intel's hidden in-chip operating system" by Steven Vaughan-Nichols, Senior Contributing Editor. The article summary discusses MINIX as a slow, hard-to-get-at, and insecure operating system. Below the summary is a photo of Steven Vaughan-Nichols and his bio information. A progress bar at the bottom indicates "Connecting to players.brightcove.net...".

Origins: GPL/Linux

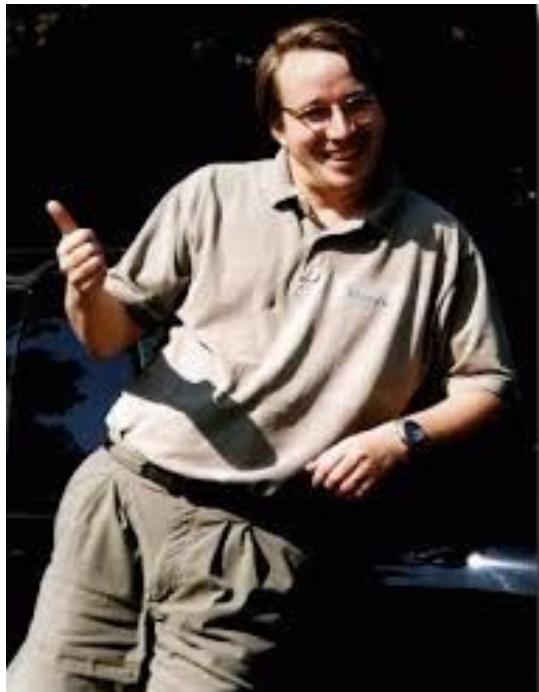


Free-ing Linux

Torvalds first encountered the GNU Project in fall of **1991** when another Swedish-speaking computer science student, Lars Wirzenius, took him to

The University of Technology to listen to free software guru **Richard Stallman's** speech.

Origins: GPL/Linux



Adopting the **GPL**

Torvalds would ultimately switch his **original license** (which forbade commercial use).

He switched to Stallman's GNU General Public License version 2 (**GPLv2**) for his Linux kernel after complaints of distributors being unable to recoup their costs due to a non-commercial clause

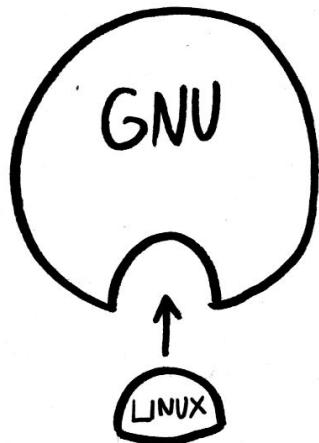
Origins: GPL/Linux



Free-software != Freeware

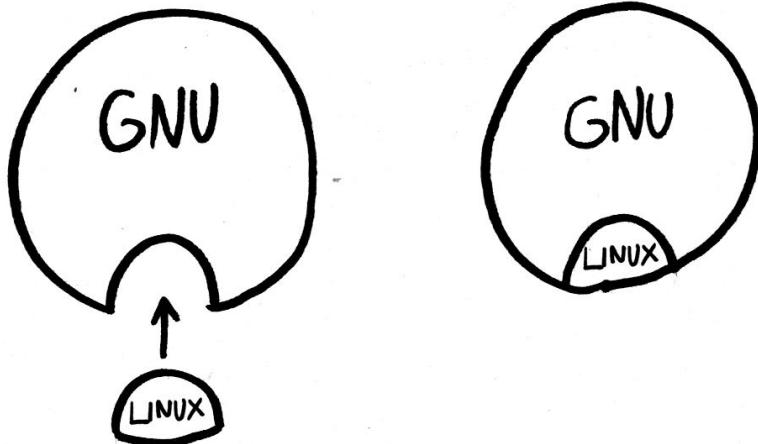
This was one of the earliest examples of how adopting a free-software license **enabled** commercialization and selling software, as hardware manufacturers.

Origins: GPL/GNU+Linux



Now that Linux had adopted the GPL, and was gaining popularity.

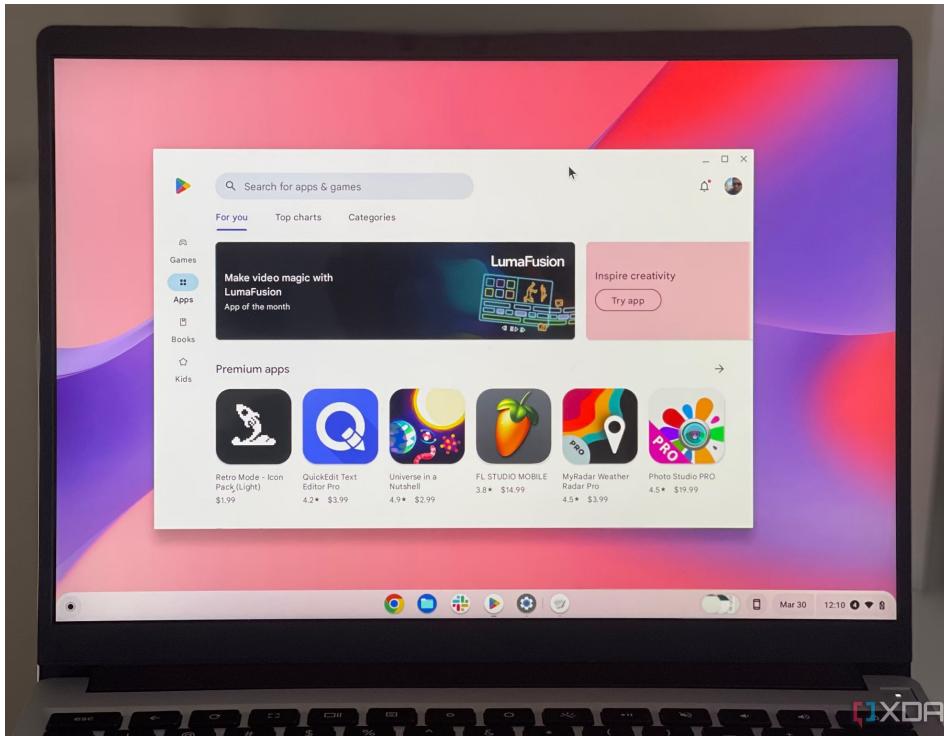
Origins: GPL/GNU+Linux



Now that Linux had adopted the GPL, and was gaining popularity.

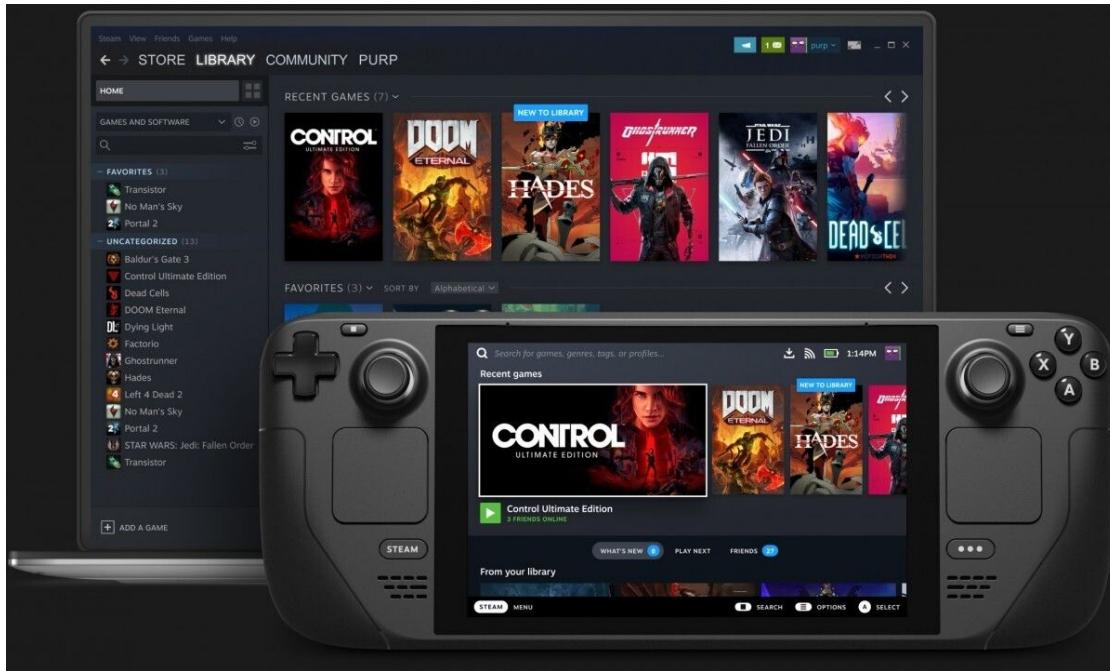
People began using the combo **GNU+Linux**: to run a fully working operating system, on hardware they already owned.

Modern variations of this combo



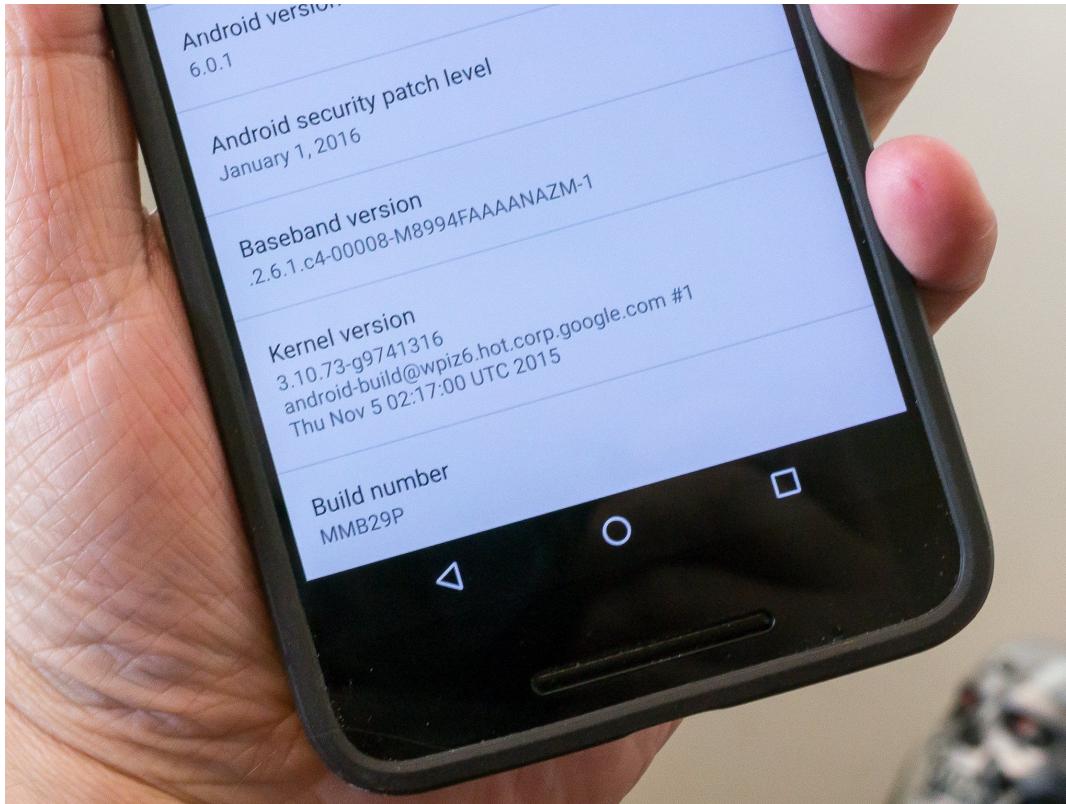
Chrome OS

Modern variations of this combo



Steam OS

Modern variations of this combo



Android

Modern variations of this combo



Linux distribution family
sorted by base distro with derivatives



Linux desktop



Debian /
Ubuntu



Arch Linux



RH Fedora



Slackware



Gentoo

Why so many variations?

Well, that's the whole point.

Free-software allows you to tinker and redistribute according to your needs.



Linux distribution family
sorted by base distro with derivatives



Debian /
Ubuntu



Arch Linux



RH Fedora



Slackware



Gentoo

FOSS is only as effective as the copyright-law

North Korea deploys its own state-run Linux distro called RedStar OS.
Freedom? Source-code? Good luck fighting Kim in court!

Red Star OS

[Article](#) [Talk](#)

[Language](#)

[Watch](#)

[History](#)

[Edit](#)

[More](#)

Red Star OS (Korean: 빨간별; MR: Pulgünbyölf) is a North Korean Linux distribution, with development first starting in 1998 at the Korea Computer Center (KCC). Prior to its release, computers in North Korea typically used Red Hat Linux^[3] and Windows XP.^[4]

Version 3.0 was released in the summer of 2013, but as of 2014, version 1.0 continues to be more widely used^[citation needed]. It is offered only in a Korean language edition, localized with North Korean terminology and spelling.^[5]

[Contents](#)

Specifications

Red Star OS features a modified Mozilla Firefox browser called *Naenara* ("My country" in Korean), which is used for browsing the *Naenara* web portal on North Korea's national intranet known as *Kwangmyong*. Naenara comes with two search engines. Other software includes a text editor, an office suite, an e-mail client, audio and video players, a file sharing program, and video games.^{[6][7][8]} Version 3, like its predecessors, runs *Wine*, a piece of software that allows Windows programs to be run under Linux.^[9]



The desktop of Red Star OS 3.0, localized with North Korean terminology and spelling

Developer [Korea Computer Center, North Korea](#)

OS family [Unix-like](#) (desktop and server), Android

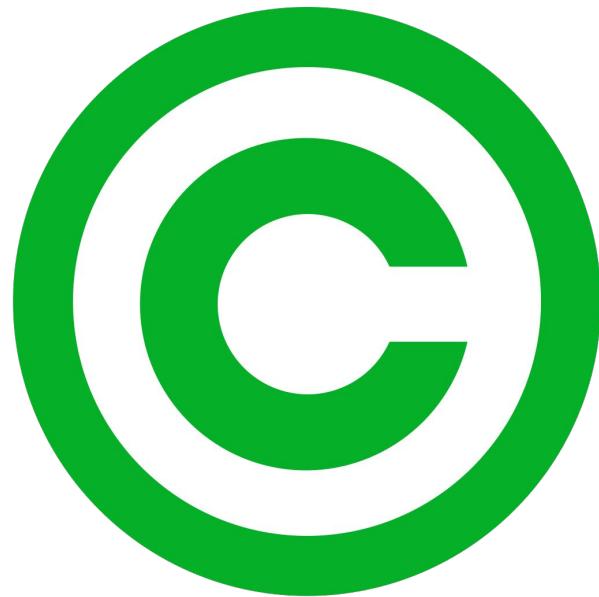
Swatantra Software? Mukt / Libre software

I feel **free** is probably a bad naming choice. They should have called it **free-d** or **libre** software, to prevent confusions with **gratis**–free of cost.

Free? Free of cost?

The **free** in *Free and Open Source* software does not talk about it's price. You can **sell** or **repackage** FOSS software at any price you want.

Free in context refers to **freedom** or (free'd software if you'd like), software that puts the user in control, and lets you run your programs the way you want it to.



All rights reserved



All **wrongs** reversed

Copyleft ©

- Copyleft is a play on the term **copyright**
- Instead of explicitly outlining what users **can't** do with software.

It allows users to use it freely as long as they share and uphold the **same licensing terms** to anyone they distribute modified-copies of their binaries.

Virality of Copyleft

If you received a '**libre/free**' copy of the program. You **must not distribute proprietary copies** of the program and grant the same freedoms you received to all your **downstream** users, so that they might do the same to you, if you want to use their improvements.

These are copylefted licenses: GNU GPL, Mozilla's MPL and such.

Pushover licensing

MIT licenses do not have a clause for you just need to credit the upstream author and add the MIT clauses to your **LICENSE** file

This is **still** free-software, but can be forked to include non-free additions and is often exploited by corporate giants by not releasing the source code of their forks and added features, making these forks **proprietary**.

Pushover vs Copyleft



I want it simple and permissive.

The [MIT License](#) is short and to the point. It lets people do almost anything they want with your project, like making and distributing closed source versions.

[Babel](#), [.NET Core](#), and [Rails](#) use the MIT License.



I care about sharing improvements.

The [GNU GPLv3](#) also lets people do almost anything they want with your project, *except* distributing closed source versions.

[Ansible](#), [Bash](#), and [GIMP](#) use the GNU GPLv3.

VSCODE/Code OSS. Chromium/Chrome, AOSP/PixelOS

The screenshot shows a GitHub repository page for 'VSCODE/Code OSS'. The top navigation bar includes links for Code, Issues (4.9k), Pull requests (228), Actions, Projects (2), Wiki (highlighted with an orange underline), Security, and Insights.

Differences between the repository and Visual Studio Code

Brigit Murtaugh edited this page 13 days ago · 8 revisions

This article outlines the differences between the github.com/microsoft/vscode GitHub project (which we refer to as **Code - oss**) and Microsoft's [Visual Studio Code](#) distribution.

The **Code - oss** Repository

The **Code - oss** repository is where we (Microsoft) develop the open source editor upon which we build the [Visual Studio Code](#) product. We contribute source code and manage issues in this repository. We also maintain the [wiki](#), publish the [Visual Studio Code roadmap](#), [monthly iteration plans](#), and [endgame plans](#) for the product. The source code in this repository is available to everyone under a standard [MIT license](#).

Visual Studio Code

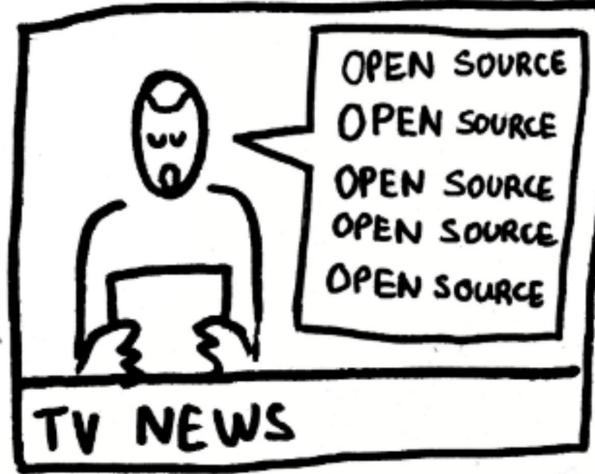
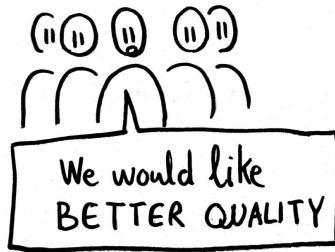
[Visual Studio Code](#) is a distribution of the **Code - oss** repository with Microsoft specific customizations, including additional source code and extensions, [released under a traditional Microsoft product license](#).

F/OSS

Misconceptions.
Wording.
History.

Free software
activists

Open source
boosters



F/OSS

Open source is a **development methodology**; free software is a **social movement**.

- All **free-software is open-source** software.
- But all open-source software **may not** be **free-software**.
- Better call such projects out!

Also known as: source-available, ethical-source, business-source license, all are forms of non-free licenses.

Making money in FOSS

Selling Expertise and services.

Bounties and stipends.

IT / Support.

Working in the Govt.

Research programs.

Joining FOSS companies.

Free? Free of cost?

The **free** in *Free and Open Source* software does not talk about it's price. You can **sell** or **repackage** FOSS software at any price you want.

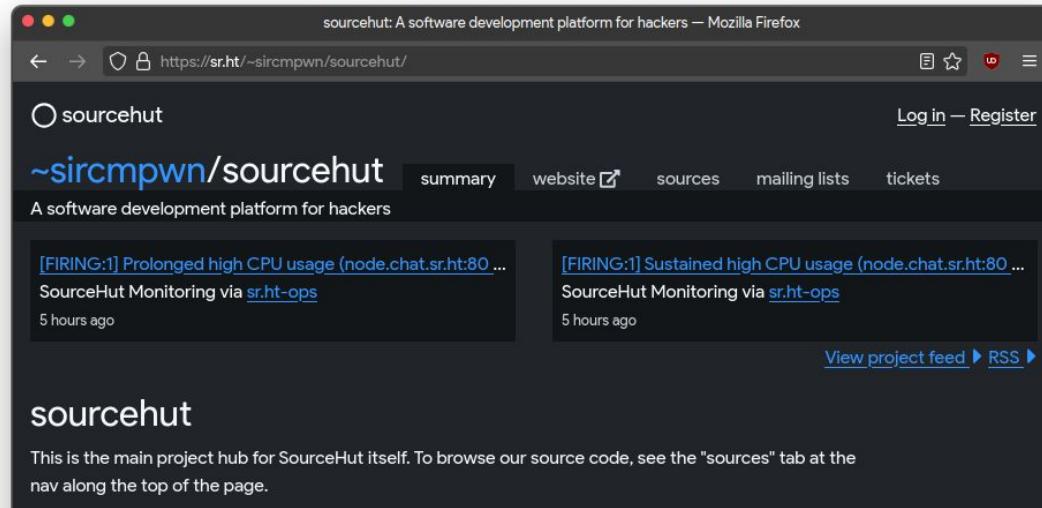
Free in context refers to **freedom** or (free'd software if you'd like), software that puts the user in control, and lets you run your programs the way you want it to.

To **FOSS** is to surrendering your monopoly over commercialization:

drewdevault.com/2021/01/20/FOSS-is-to-surrender-your-monopoly.html

Case Study: sourcehut.org

Developer platform that lets users host code, deploy using CI/CD runners, hosted real-time chat-rooms, ticketing as well as mailing lists. Think of it like GitHub – but **actually** FOSS.



A screenshot of a Mozilla Firefox browser window displaying the sourcehut.org website. The title bar reads "sourcehut: A software development platform for hackers — Mozilla Firefox". The address bar shows the URL "https://sr.ht/~sircmpwn/sourcehut/". The page itself has a dark theme with white text. At the top left is the "sourcehut" logo. On the right side, there are "Log in" and "Register" links. Below the header, the project name "[~sircmpwn/sourcehut](#)" is displayed, followed by a subtitle "A software development platform for hackers". A navigation bar below the subtitle includes "summary", "website", "sources", "mailing lists", and "tickets". Two recent monitoring alerts are listed under the "sources" section:

- [FIRING:1] Prolonged high CPU usage (node.chat.sr.ht:80 ...) - SourceHut Monitoring via [sr.ht-ops](#) 5 hours ago
- [FIRING:1] Sustained high CPU usage (node.chat.sr.ht:80 ...) - SourceHut Monitoring via [sr.ht-ops](#) 5 hours ago

At the bottom of the page, there is a link to "View project feed" and an RSS feed icon.

Case Study: sourcehut.org

Ask users for to subscribe in-exchange for a highly valued service.

The screenshot shows a Mozilla Firefox browser window displaying the SourceHut's 2022 Financial report. The URL in the address bar is <https://sourcehut.org/blog/2023-03-27-2022-financial-report/>. The page title is "SourceHut's 2022 Financial report". The header includes links for sourcehut, blog, pricing, community, and merchandise, along with a "Go to sr.ht" button. A blue banner at the top says "SourceHut's 2022 Financial report". Below the banner, a text block states: "In summary, SourceHut continues to be sustainable, profitable, and growing. We continued to grow YoY between 2021 and 2022. We were able to make some significant investments, particularly in establishing a new datacenter presence in Europe, and we intend to expand this investment over the course of 2023 and 2024, and we have plenty of capital to invest in this and other business initiatives." A small disclaimer below reads: "Disclaimer: this report is a summarized approximation of our financials, and is not used for tax purposes." A section titled "2022 revenue sources" lists two main streams: paid subscriptions for the forge platform and contracts via consulting arm. It notes that forge platform revenue is the basis for financial planning. A table at the bottom shows invoice details for 2022, with a total of 12,965 invoices issued.

6,530	\$ 2	(paid monthly)	
2,456	\$ 5	(paid monthly)	
917	\$10	(paid monthly)	
2,141	\$ 2	(paid yearly)	
604	\$ 5	(paid yearly)	
227	\$10	(paid yearly)	

Consulting and Expertise

Bounties and stipends.

IT / Support.

Working in the Govt.

Research programs.

Joining FOSS companies.

Case Study: sourcehut.org

The image displays two side-by-side screenshots of a Mozilla Firefox browser window. Both windows show the same web page from <https://sourcehut.org/blog/2022-08-23-how-does-our-consultancy-work/>. The left window shows the initial content, and the right window shows a scroll-down view of the page.

Initial Content (Left Window):

- Header:** How does SourceHut's free software consultancy work? — Mozilla Firefox
- Page Header:** sourcehut blog pricing community merchandise Go to sr.ht ▾
- Date:** August 23, 2022 by Drew DeVault
- Title:** How does SourceHut's free software consultancy work?
- Text:** SourceHut offers a [consultancy](#) in which we offer our services as experts in exclusively on free software development. Through this work we have developed Wayland, graphics, virtual reality, email, and more, entirely through free software. Our work can take the form of developing new free software from scratch, or to projects that our client depends on, offering what sage advice we can spare above.
- Text:** How does this actually work in practice? Following a discussion in our off-to-thought that it would be nice to write about the actual logistics of our operation, self free software consulting services are better equipped to do so, and so know better what to expect from us.
- Text:** The process begins with client on-boarding. We don't do much in the way of mostly come to us. When someone new comes to us with an idea, the first thing is alignment on the project goals and needs, on our process, and regarding our requirements. We evaluate proposals based on a few criteria:
 - Does this project benefit the free software ecosystem?
 - Does our team have the necessary experience?
 - Are the right engineers available to work on it?
- Text:** Roughly half of our proposals clear this stage. In addition to a reasonably complete project proposal, we need to establish with the client an understanding of our requirements, namely:

Scrolled Content (Right Window):

- Text:** Roughly half of our proposals clear this stage. In addition to a reasonably complete project proposal, we need to establish with the client an understanding of our requirements, namely:
 - All work is published under a free software or open source license
 - All work is conducted in public — no NDAs
- Text:** Though we require the results to be released to the public as free software, we're open to assigning copyright to the client. Not all clients want this, though. If the customer has a clear idea of what's required, a few emails back and forth are suitable to establish a consensus. If a deeper analysis is needed, we might ask for a temporary contract to cover costs of initial research and planning.
- Text:** Following this, we'll draft a contract and seek consensus on the expected pricing and effort required. Our contracts are short and simple, generally covering matters like logistics for payment and the contract period, but also explicitly addressing our requirements regarding subjects like software licenses. Sometimes the client prefers to use their own contract templates, which we're open to, but generally this requires more negotiation to remove "standard" clauses for things like NDAs.
- Text:** We bill for each of the engineers we plan to bring on board separately. Our pricing is essentially a function of two competing factors: we target free software and small- to medium-sized businesses, which pushes our prices down, and we offer high-end services from an experienced team, which pushes up. Right now this balance settles on a Ⓜ\$250/hour and \$150/hour standard base rate for our senior and junior engineers respectively.
- Text:** The rates are generally only somewhat negotiable — but we may tweak our rates depending on things like our perception of the positive impact a contract might have for free software, or the means of the client. An early-stage open hardware vendor might get a reduced rate for our work developing graphics drivers for them, for example. We will generally expect a deposit from unfamiliar customers.
- Text:** Once everything is signed and it's off to the races, the execution of these contracts involves a non-traditional planning style. We push back against suggestions like waterfall or agile, and prefer instead to work like a free software project often does — with broad long-term goals, loosely defined, and narrower short-term goals, more concretely defined. We also set research and planning as an objective.

Bounties, stipends

IT / Support.

Working in the Govt.

Research programs.

Joining FOSS companies.

Bounties, stipends

Some software foundations and companies offer grants for open source work. Sometimes, grants can be paid out to individuals without setting up a legal entity for the project.

Bounties, stipends

- GitHub Sponsors Matching Fund (a matching program)
- fossunited.org/grants for organizing FOSS events
- [Read the Docs](#) received a grant from Mozilla Open Source Support
- [OpenMRS](#) work was funded by Stripe's Open-Source Retreat
- [Libraries.io](#) received a grant from the Sloan Foundation
- [Python Software Foundation](#) grants for Python-related work
- [nqi.eu](#) an EU initiative
- [codeforgovtech.in](#) Indian gov

Vulnerability reward programs

- hackerone.com/bug-bounty-programs
- Google and Alphabet Vulnerability Reward Program (VRP)

Challenges

Comma AI – Self driving car calibration challenge

A screenshot of a GitHub repository page for 'commaai/jobs'. The repository has 10 forks and 135 stars. It contains a README file, a README.md file, and a design.md file. A commit by 'adeebshihadeh' adds an intern role. The repository is public and has a master branch. On the right side, there are sections for 'About', 'Releases', 'Packages', and 'Contributors'. A large banner at the bottom says 'WE'RE HIRING'.

A screenshot of the GitHub README for the 'commaai/calib_challenge' repository. The README includes sections for 'Deliverable' and 'Evaluation'. The 'Deliverable' section specifies that the deliverable is a 5-label array of pitch and yaw angles for frames 5.txt to 9.txt. It provides an email address for submission: givemeajob@comma.ai. The 'Evaluation' section states that mean squared error against ground truth labels will be evaluated, with errors for frames below 4m/s ignored. It also mentions an eval script for testing solutions. A 'Hints' section lists tips: keeping the goal in mind, looking at plots before submitting, and being cautious with ML due to the small dataset. A bold '\$500 Prize CLAIMED' section indicates that the first submission under 25% error will receive a \$500 prize.

Challenges

- github.com/tigerbeetle/viewstamped-replication-made-famous
 - > A \$20k consensus challenge based on TigerBeetle's implementation of the pioneering Viewstamped Replication protocol.

Fiscal hosts, crowdfunding

- openpledge.io
- opencollective.com
- liberapay.com
- patreon.com
- buymeacoffee.com

> A fiscal host can help offset tax-burdens by pooling contributions and can offer financial support for individuals and small organizations.

IT / Support

Working in the Govt.

Research programs at Universities Labs.

Joining FOSS companies.

IT / Support

Software needs change, and clients would need services and maintenance upgrades.

If, say the original company who made a program **dies**, you can offer a **fork** or maintain the codebase **internally**.

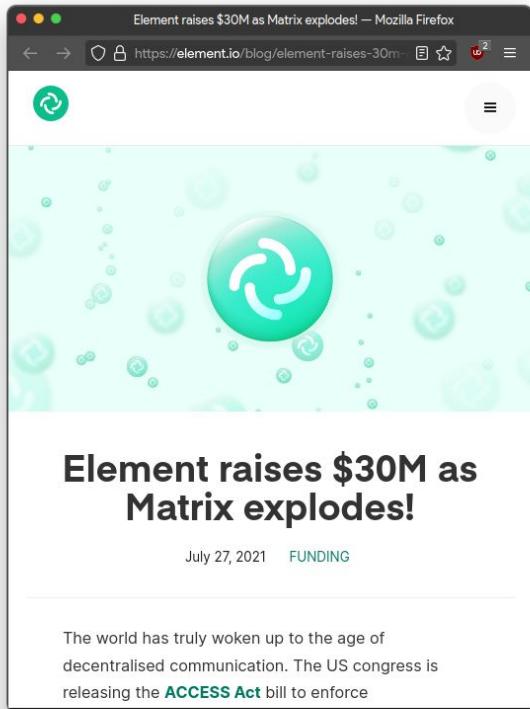
IT / Support

Software needs change, and clients would need services and maintenance upgrades.

You can choose to offer services and tech-support for software that you yourself use, like Ubuntu, RHEL and other FOSS software. You can re-sell your copies as well. Free software allows anyone who has a copy to make commercialization possible.

If, say the original company who made a program dies, you can offer a fork or maintain the codebase internally.

Case Study: Element / Matrix

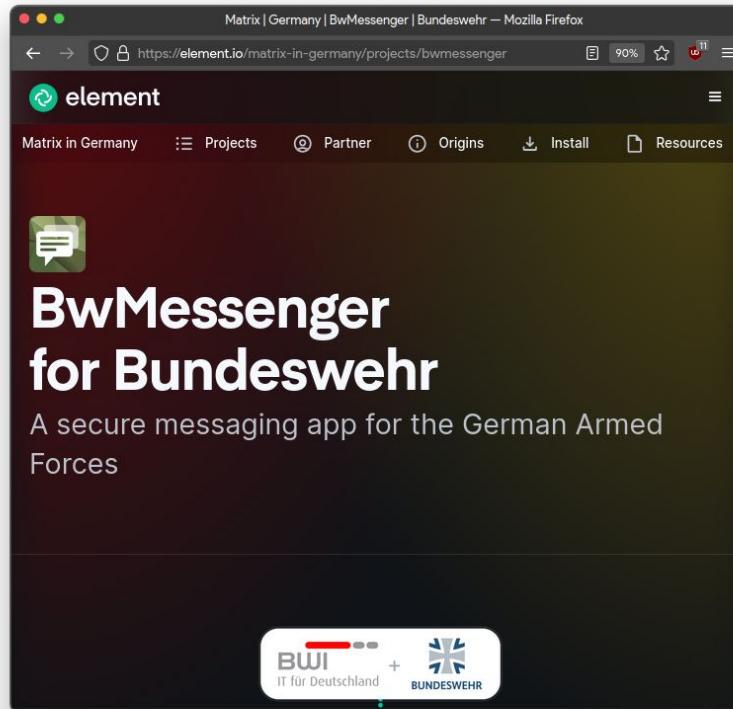


A screenshot of a Mozilla Firefox browser window showing a blog post on element.io. The title of the post is "Element raises \$30M as Matrix explodes!". The post discusses the rise of decentralized communication and the introduction of the ACCESS Act bill. The Element logo, a teal circle with a white circular arrow, is prominently displayed at the top.

Element raises \$30M as Matrix explodes!

July 27, 2021 FUNDING

The world has truly woken up to the age of decentralised communication. The US congress is releasing the [ACCESS Act](#) bill to enforce



A screenshot of a Mozilla Firefox browser window showing the BwMessenger project page on element.io. The page features a dark background with a green speech bubble icon. The title is "BwMessenger for Bundeswehr" and the subtitle is "A secure messaging app for the German Armed Forces". At the bottom, there are logos for BWI IT für Deutschland and the Bundeswehr.

Matrix | Germany | BwMessenger | Bundeswehr — Mozilla Firefox

element

Matrix in Germany Projects Partner Origins Install Resources

BwMessenger for Bundeswehr

A secure messaging app for the German Armed Forces

BWI IT für Deutschland + BUNDESWEHR

Case Study: Element / Matrix

Matrix.org - Support Matrix — Mozilla Firefox

matrix

Support Matrix

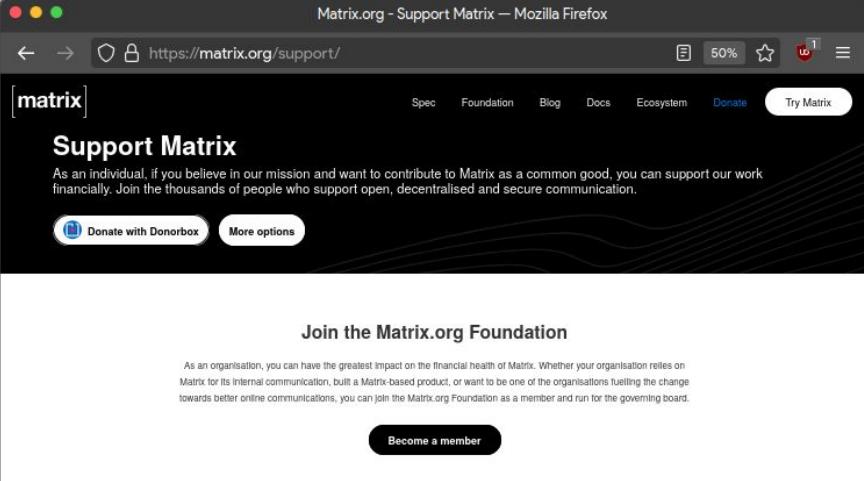
As an individual, if you believe in our mission and want to contribute to Matrix as a common good, you can support our work financially. Join the thousands of people who support open, decentralised and secure communication.

Donate with Donorbox More options

Join the Matrix.org Foundation

As an organisation, you can have the greatest impact on the financial health of Matrix. Whether your organisation relies on Matrix for its internal communication, built a Matrix-based product, or want to be one of the organisations fueling the change towards better online communications, you can join the Matrix.org Foundation as a member and run for the governing board.

Become a member



Matrix.org - Support Matrix — Mozilla Firefox

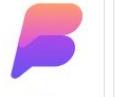
matrix

Platinum Members



Element

Gold Members



Beeper Futurewei Technologies

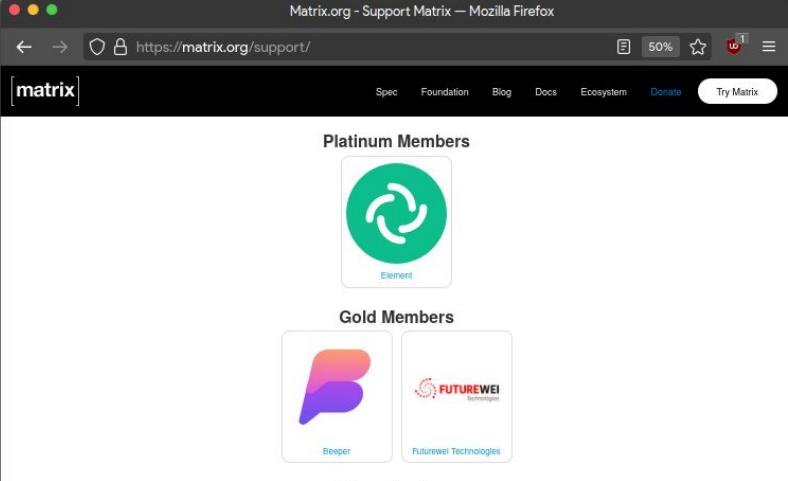
Silver Members



Fairkom Fomedly Fractal Networks gematik



IndieHosters Verji Tech AS XWiki



Case Study



Home About Docs Code Community [Shop](#)



USB



Computers



Keyboards



Phones



Routers



Keypads



Auto

Liberated Computers

Free software-friendly laptops for students, journalists, teachers and developers.

Built for running GNU/Linux from ground up. Now - you won't need to use proprietary software or drivers anymore.

These laptops run Debian GNU/Linux (main), PureOS, Trisquel or any FSF-endorsed distribution out-of-the-box. They can also run Ubuntu and FreeBSD, if you like.



Past Keyboard Builds



Granite DSA Keyset



ICE SA Keyset



Custom DSA Keyset



Troubled Minds SA Keyset



Modern Selectrics SA Keyset



Deep Space SA Keyset



Case Study: mostlyharmless.io/libretech.shop

Libre Tech Shop – Ethical and Hackable Products + Convenience — Mozilla Firefox

https://libretech.shop

Liberated Products

(The prices are exclusive of 18% GST.)

Computers

Product	Description	Price (₹)	Action
Liberated Computer - LC9010	Dell desktop tower	40,000.00	Select options
LC440p - Liberated Computer	ThinkPad laptop	27,000.00	Select options
Liberated Computer - LC230T	ThinkPad laptop	26,500.00	Select options
Liberated Computer - LC230	ThinkPad laptop	30,000.00	Select options

Case Study: Purism / Elementary OS

Purism- Librem 14 — Mozilla Firefox
https://puri.sm/products/librem-14/

Purism | Librem 14 Services Products News & Events About Support PureOS Tech Specs Buy Now

Same Footprint, Larger Display
The same, perfect laptop size – now with a bigger screen.



The thoughtful, capable, and ethical replacement for Windows and macOS



Pay What You Can:

\$ 10 \$ 20 \$ 30

\$ Custom

Purchase elementary OS

elementary OS 7.1 Horus (3.0 GB)
Recommended System Specs | FAQ

Joining FOSS companies

- RedHat
- GitLab
- SUSE
- {Dell, Lenovo, Intel} have teams working on the kernel and bringing up driver modules for existing and new product lines.

Joining FOSS companies

- RedHat
- GitLab
- SUSE
- {Dell, Lenovo, Intel} have teams working on the kernel and bringing up driver modules for existing and new product lines.
- Entrepreneurship, handling licensing and **freelancing**

FOSS companies

- To make money in FOSS, build a solid business.
- Unlike your proprietary vendor locked consumers, once you sell your software, you can not take advantage of your consumers or lose their trust.

FOSS companies

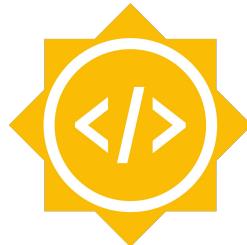
- To make money in FOSS, build a solid business.
- Unlike your proprietary vendor locked consumers, once you sell your software, you can not take advantage of your consumers or lose their trust.
- Like **EMACS**, which sold at \$150 back in the '80s, you have to keep delivering solid releases, with features users would pay for, and maintain the same codebase quality.

FOSS companies

- To make money in FOSS, build a solid business.
- Unlike your proprietary vendor locked consumers, once you sell your software, you can not take advantage of your consumers or lose their trust.
- Like **EMACS**, which sold at \$150 back in the '80s, you have to keep delivering solid releases, with features users would pay for, and maintain the same codebase quality.
- You do not have a **monopoly** over commercialization, the users — once they buy the software, get to decide who they call for services and feature upgrades.

How to contribute to FOSS

FOSS software prospers, because of the culture it was built around. GSoC, Linux Foundation, Outreachy, EU Grants, GitHub sponsorships, College clubs, and online communities

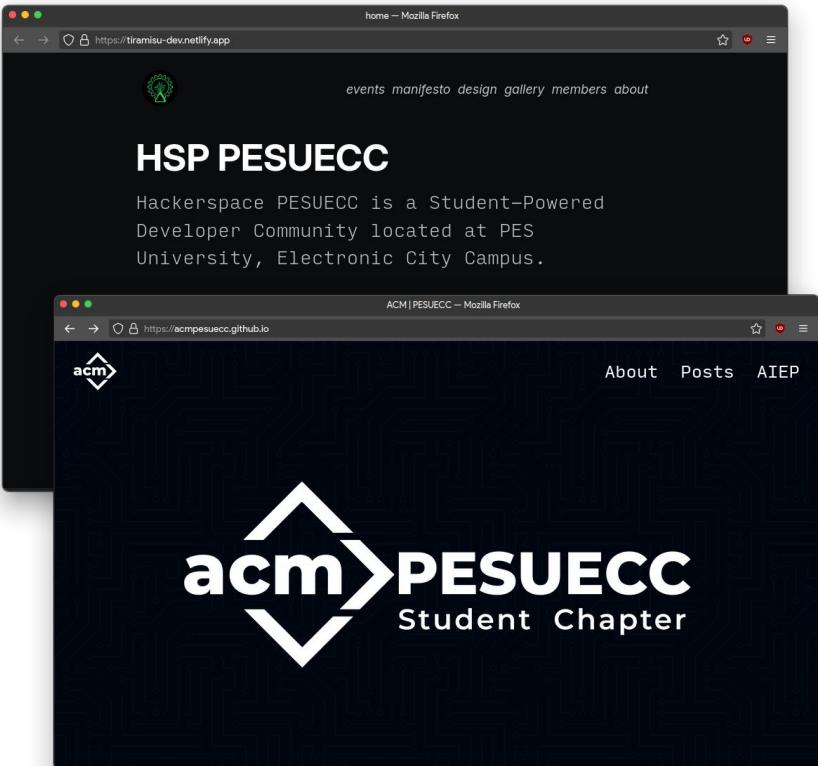


Google
Summer of Code



FOSS at campus

- GDSC :D
- HSP, ECC
- ACM Student Chapter, ECC
- IEEE RAS
- Silicon, ECC
- Parallax, ECC (Godot, Blender)
- Maaya / Aatmatrisha websites



Hacks!

- InFeynite: ECE Dept
- Override: **GDSC**
- Hacknight: **ACM**
- DeltaTime: **Parallax**
- Hashcode: **PIL**
- Kalpana: **IEEE**
- inGenius: CSE Dept
- Tilde: **HSP**
- AIEP: **ACM**

FOSS communities @ India

- fossunited.org / IndiaFOSS
- fsf.org.in
- fsmi.in
- fsci.in
- codeforgovtech.in / C4GT
- gnulinuxindia.sh
- swecha.org / IIIT-H
- smc.org.in / Swathanthra Malayalam Computing

FOSS communities @ **World**

- fsf.org / LibrePlanet
- fsfe.org / PBPC
- -----
- Fedora release parties
- NixOS user-groups
- Hacktoberfest / Hacknight
- LinuxFest / SXSW
- LinuxFoundation / CNCF

Fin.

- Attribution for the illustrations (slides) by Christian Noguera, Valentin Pasquier, [Richard Stallman](#) © 2014. Used under [CC-BY-3.0](#)
- [Poster](#) by [Jeison Yehuda Amihud](#) © 2021 [CC-BY-SA-4.0](#)
- Blender™ Foundation (Logo) (Fairuse)
- GDSC PESU-ECC (Fairuse)
- Mozilla Corporation (Logo) [MPL 2.0](#)
- Chromium (Logo): [CC-BY-2.5](#)
- Krita (Logo): Wolthera van Hövell tot Westerflier and Timothée Giet [CC-BY-SA-4.0](#)
- git (Logo) [Jason Long](#) [CC-BY-3.0](#)
- Tux (Logo): lewing@isc.tamu.edu Larry Ewing and [GIMP](#)
- Python (Logo): python.org [GPLv3](#)

Questions?



polarhive.net/contact

My classroom's at #307