Crux of the Matter

# The Decline of the Copyleft Free Software Movement and Its Ideological Limitations

## A Marxist Critique of the Free Software Movement

● Abhijit A.M.

The Free Software Movement (FSM) that heralded freedom in the software world, for the programmers and users, can be clearly seen losing its fight against the Open Source 'Movement' (OSM) which happened to be an offshoot of the FSM itself, and against the monopolistic hold of the bourgeoisie over software. The corporate takeover of the FSM and the expected corporate adoption of the OSM has taken place, on a large scale. The reasons for the failure of the Free Software Movement in achieving the utopia perceived by many lie in its ideological weaknesses with roots in anarchism.

After briefly introducing the FSM and OSM, we demonstrate how the software world is today dominated by Open Source/Free Software, identify the anarchist ideological roots of the FSM, demonstrate how the software bourgeoisie today benefits from FSM and OSM and drives both, and argue that this failure results from the anarchist misgivings about capitalism.

## 1. A Short History of the Free Software and Open-Source Movements

### *1.1 Initial Days*

Software, since its inception, was developed in a fashion that involved sharing of source code. The sharing of source code and subsequent improvements by the users and developers of the code was a common thing. It was not perceived to be a commodity to be sold separately. This was the story upto 1970s. In the initial days of software industry, companies like IBM sold software bundled with their mainframe computer hardware, practically free of charge. However, with the advent of Personal Computers in the late 70s and software required to run them, things started to change. Unix was a popular operating system developed at AT&T Bell Labs, and it had already spread to many universities in the USA, with many students and academics contributing to improve it. However, AT&T Bell Labs started claiming ownership and control of the Unix operating system source code since 1979, after Bill Gates wrote his infamous 'An Open Letter to the Hobbyists'[1] in 1976 where he appealed to convert software into a commodity and against the 'hobbyist' programmers, and added to the efforts of the software industry that had started converting software into a separately saleable commodity. In retaliation, a movement arose to *liberate* software. Led by Richard M Stallman (*aka* RMS), then a programmer at a MIT laboratory, the Free Software Movement (free here refers to freedom, not cost) appealed to software programmers (hackers) to write software that will grant its users the 4 essential freedoms to "(*0*) *to run the program,* (*1*) *to study and change the program in source code form,* (*2*) *to redistribute exact copies, and* (*3*) *to distribute modified versions.*[2]

## *1.2 RMS, FSF, GNU*

Stallman started the GNU (GNU's Not Unix, a recursive acronym) project in 1983 to write a Unix-like operating system that will be free and will remain free. Stallman wrote the GNU GPL (General Public License), against the prevailing proprietary licenses that privatized and commodified software, a legal agreement which could be used to release a software as Free Software, but the licence also mandated that anyone using GPL licensed code in their own code would have to release their code also under GPL. This was to ensure that a GPL licensed code could never be used to make and release non-free (again, free as in freedom) source code. The GPL intended to enforce freedom. This was called *copyleft*. A programmer, or a set of programmers, could claim their copyright (!) over their code, and release the software under GPL license, which granted the four freedoms to users, programmers and everyone, but said that the license cannot be modified. Now, anyone who re-released the same or modified version of the code was mandated to release it under GPL, otherwise it would be a violation of the *copyright* of the original programmers. The GPL was considered as a clever *legal hack*.

The popularity of GPL continued, and thousands of projects were released later under the GPL. Stallman, himself a prolific programmer, was and is revered by programmers' community for his enormous coding contributions to major projects like Emacs, GNU Debugger (GDB), GCC Compiler collection, etc. and he is recognized widely for his stewardship of the Free Software Movement. Though mired in multiple personal and political controversies in last decade or so, RMS's contributions to the growth of FSM remain undisputed.

[1] **Gates, Bill.** 1976. 'Open Letter to Hobbyists'. *Online.* https://web.archive.org/web/20220920082550/https://www.digibarn.com/collections/newsletters/homebrew/V2_01/gatesletter.html

[2] **GNU.** 'Philosophy of the GNU Project'. *GNU.org.* https://www.gnu.org/philosophy/philosophy.html

The Free Software Foundation (FSF) was established in 1985, and it works to "secure freedom for computer users by promoting the development and use of free (as in freedom) software and documentation". FSF continues to operate till day. It has led the GNU project, chartered many new licenses like the GNU AGPL, GNU LGPL, etc. and fought various legal and ideological battles for freedom of software users. Its significant campaigns include the right to repair (the freedom to employ any person or firms to repair things), against surveillance, against defective by design technologies (designs which create technologies that become obsolete or enforce particular vendors, etc.), against software patents, for open document standards, etc. It has also participated in various technology forums, like the artificial intelligence safety consortium or the European parliamentary forums, to fight against the big corporations trying to restrict software users' freedoms, and for more democratic World Wide Web.

The GNU operating system did not materialize; however, the Linux kernel (kernel, the core part of an Operating System) written by a hobbyist student Linus Torvalds, in 1991 was released with GNU's code, and hence under GPL, and became quite popular. The Linux kernel grew, with the contributions of thousands of programmers all over the world, and became one of the most stable, capable, scalable kernels by 1998. This led to the proliferation of hundreds of GNU/Linux Operating Systems Like Debian, Red Hat, Suse, Fedora, Ubuntu, etc. over next 30 years. Some of the world's best programmers led many significant projects in Free Software (FS) and developed industrial strength software like Linux kernel, the MySQL database engine, the GCC Compiler collection, the GNU software codebase, the Apache web server, the DNS servers, etc. which are very core as building blocks of the internet and software ecosystems.

The concept of free software with the 4 freedoms, led to development of software not in a competitive commercial way, but naturally in the way of forming a community of developers who would, in different possible ways, communicate and collaborate with each other to develop the software, often using different more or less democratic processes. Thus, one can find a *community* of developers, users, supporters around each of the thousands of free software projects. These communities typically communicate using email-lists, or online forums, or chat servers, or many different messaging systems to drive the projects. In the words of Eben Moglen,

> The development of the Linux kernel proved that the Internet made it possible to aggregate collections of programmers far larger than any commercial manufacturer could afford, joined almost non-hierarchically in a development project ultimately involving more than one million lines of computer code – a scale of collaboration among geographically dispersed unpaid volunteers previously unimaginable in human history.[3]

### *1.3 GPL, Copyleft and Its Significance*

Fight against the monopolistic control by multinational corporations was always

[3] **Moglen, Eben.** 1999. 'Anarchism Triumphant: Free Software and the Death of Copyright'. *First Monday.* https://firstmonday.org/ojs/index.php/fm/article/view/684/594

an intrinsic part of the copyleft movement. Arguing that ownership of software went against individual's freedom, RMS said in 'Free Software, Free Society', "When a program has an owner, the users lose freedom to control part of their own lives."[4]

Eben Moglen, says in '*The dotCommunist Manifesto'* (*2003*) "**Proprietary software is the moral equivalent of slavery.**"[5]

The GPL was always seen by the software businesses, rightly, as being anti-commercial, as is evident from the multiple assessments now available. Craig Mundie, Microsoft Senior Vice President said in 2011,

> This viral aspect of the GPL poses a threat to the intellectual property of any organization making use of it. It also fundamentally undermines the independent commercial software sector because it effectively makes it impossible to distribute software on a basis where recipients pay for the product rather than just the cost of distribution.
>
> The GPL asserts that any product derived from source code licensed under it becomes subject to the GPL itself. When the resulting software product is distributed, the creator must make all of the source code available, at no additional charge. This effectively makes it impossible for commercial software companies to include source code that is licensed under the GPL into their products, since by doing so, they are constrained to give away the fruits of their labour....[6]

The US government assessments also talked something[7] similar. In 'Government Policy toward Open-Source Software' ( Edited by Robert W. Hahn), in 2002, Evans D says, "The GPL effectively prevents profit-making firms from using any of the code since all derivative products must also be distributed under the GPL license."[8]

## *1.4 The Split, and OSM*

The Open-Source Movement (if it could be called a 'Movement'!) started in the late 1990s. Programmers Bruce Perens, Eric Raymond, Christine Peterson, Jon Hall, Larry Augustin, and others led it. The Open-Source Initiative (OSI) was started in 1998, by Eric Raymond and Bruce Perens. Initially they claimed that they had no principled differences with Free Software Movement. To quote Eric Raymond,

> The real disagreement between OSI and FSF, the real axis of discord between those who speak of 'open source' and 'free software', is

[4] **GNU.** 'Free Software, Free Society: Selected Essays of Richard M. Stallman'. *GNU.org.* https://www.gnu.org/philosophy/fsfs/rms-essays.pdf

[5] Moglen, Eben. January 2003. 'The dotCommunist Manifesto'. Columbia Law School. https://moglen.law.columbia.edu/publications/dcm.html

[6] Mundie, Craig. May 3, 2001. 'Speech Transcript – The New York University Stern School of Business'. Microsoft News. https://news.microsoft.com/speeches/speech-transcript-craig-mundie-the-new-york-university-stern-school-of-business/

[7] Hahn, Robert W. 2002. 'Politics and Programming', in Government Policy toward Open Source Software, AEI-Brookings Joint Center for Regulatory Studies, Washington D.C. , p. 47. https://www.aei.org/wp-content/uploads/2011/10/Govt%20Policy%20Toward%20Open%20Source%20Software.pdf?x85095

[8] **Daffara, Carlo.** 2009. 'The Small/Medium Enterprises Guide to Open Source Software'. *Scribd.* https://www.scribd.com/doc/96574077/Floss-Guide

> not over principles. It's over tactics and rhetoric. The open-source movement is largely composed not of people who reject RMS's ideals, but rather of people who reject his *rhetoric*....
> If RMS's rhetoric had been effective outside the hacker community, we'd have gotten where we are now five or ten years sooner and OSI would have been completely unnecessary[9]

Eric Raymond, a major proponent of the Open-Source Movement, wrote in 1997 one of the much-celebrated articles 'The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary'[10] which advocated Open Source as a much better development methodology. Raymond compared the Cathedral model, employed by Free Software projects like Emacs and also proprietary software that focussed on contributions from limited set of *selected programmers* in between two releases, while the projects like Linux kernel invited contributions from practically anyone, developing code over the Internet, which he called the Bazaar model. Raymond argued that the Bazaar model led to faster development, better code and the Cathedral model could no longer compete with it. He popularized Linus Torvalds's quote that "Given enough eyeballs, all bugs are shallow" as Linus's law, and an advertisement of the quality of the Open-Source Software.

Criticising the OSM, Richard Stallman wrote,

> The terms 'free software' and 'open source' stand for almost the same range of programs. However, they say deeply different things about those programs, based on different values. The free software movement campaigns for freedom for the users of computing; it is a movement for freedom and justice. By contrast, the open-source idea values mainly practical advantage and does not campaign for principles. This is why we do not agree with open source, and do not use that term.[11]

It did not take long for the OSM to reveal its true colours. Eric Raymond said in 2008, "I think that if a programmer wants to write a program and sell it, it's neither my business nor anyone else's but his customer's what the terms of sale are."[12]

Raymond wrote in *Cathedral and the Bazaar*,

> Our success after Netscape would depend on replacing the negative FSF stereotypes with positive stereotypes of our own—pragmatic tales, sweet to managers' and investors' ears, of higher reliability and lower cost and better features.

The OSM clearly tried to appease to the bourgeoisie. Linux Torvalds said, as quoted in an article on Oracle, in 2017, "It's very important to have companies in open

[9] Raymond, Eric. 1999. 'Shut Up And Show Them The Code'. *Linux Today.* https://web.archive.org/web/20030901104127/http://www.linuxtoday.com/developer/1999062802310NWSM (accessed July 15, 2025).

[10] Raymond, Eric. 1999. *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary.* O'Reilly. https://monoskop.org/images/e/e0/

[11] Stallman, Richard. 2024. 'Why Open Source Misses the Point of Free Software'. *GNU.org.* https://www.gnu.org/philosophy/open-source-misses-the-point.en.html

[12] Raymond, Eric. 2008. 'Why I Hate Proprietary Software'. http://esr.ibiblio.org/?p=556

source. It’s one thing I have been very happy about,” Torvalds said. “Linux came out of the free software movement, and there was a lot of anti-commercial sentiment."[13]

Advocacy of the term ‘Open Source’ and its commercial and technical benefits, ‘better quality, higher reliability, lower costs, and increased choice’, along with advocacy of a *better* model of development remains the mainstay of the OSM.

Thus, the FSM criticized the OSM for not insisting on freedom and justice as core values, for not being political, and in effect for compromising on principle of freedom. However, as we shall see later in the essay, in practice both the movements often walked together.

As a side-note, the Open-Source Movement also found its offshoots in other fields of life. O’Reilly started publishing its books under the Open Publication License and giving away free (of cost) PDFs of the books they published. The Wikipedia, which uses the Creative Commons License, can be seen as an example of the Bazaar model, where anyone can write or edit an encyclopaedic article.

### *1.5 FOSS, and the ‘Compromise’*

The term FOSS (Free and Open-Source Software), a highly prevalent term today, became popular in 1990s and 2000s, and referred to both Open Source and Free Software, and suggested neutrality between the two camps. Expectedly, the FSF and RMS had reservations about the term as it seemed to underplay the freedom aspect of software, and suggested an alternative term FLOSS (Free/Libre and Open-Source Software) to emphasize on the freedom aspect of the software sufficiently. To quote RMS,

> Others use the term ‘FOSS,’ which stands for ‘Free and Open-Source Software.’ This is meant to mean the same thing as ‘FLOSS,’ but it is less clear, since it fails to explain that ‘free’ refers to *freedom*. It also makes ‘free software’ less visible than ‘open source,’ since it presents ‘open source’ prominently but splits ‘free software’ apart.
>
> ‘Free and Open-Source Software’ is misleading in another way: it suggests that ‘free and open source’ names a single point of view, rather than mentioning two different ones. This conceptualization of the field is an obstacle to understanding the fact that free software and open source are different political positions that disagree fundamentally.
>
> Thus, if you want to be neutral between free software and open source, and clear about them, the way to achieve that is to say ‘FLOSS,’ not ‘FOSS.’
>
> We in the free software movement don't use either of these terms, because we don't want to be neutral on the political question. We stand for freedom, and we show it every time—by saying ‘free’ and ‘libre’—or ‘free (libre).’[14]

[13] Oracle. ‘Big Data Open Source’. *Oracle.com.* https://www.oracle.com/in/big-data/what-is-big-data/open-source/

[14] Stallman, Richard. 2013. ‘FLOSS and FOSS’. *GNU.org.* https://www.gnu.org/philosophy/floss-and-foss.en.html

It is quite clear that the FSM did not take antagonistic stand against the OSM, but one of a friendly contradiction. The OSM and the FSM have historically walked the path together. Many FOSS projects were worked upon together by people from both the camps, many of them released under the copyleft GPL like licenses.

The fact that Richard Stallman accepted the Linus Torvalds award in 1999 and Bradley Kuhn, executive director of FSF accepted the 2012 O'Reilly Open-Source Award, allude to some of the *results* of this 'walking together'.

RMS as a firm believer in 'freedom' always believed that FS is about ethics (and OSM is about commercial benefits) and hence also believed in the freedom of the multinational corporations or the OSM, he was never antagonistic to the idea of letting them *choose* what they want. In fact, arguing that FSM is a better development methodology was also the part of his argument. Comparing FS vs proprietary software to a free road and toll-road, he says,

> To apply the same argument to software development, I will now show that having 'toll booths' for useful software programs costs society dearly: it makes the programs more expensive to construct, more expensive to distribute, and less satisfying and efficient to use. It will follow that program construction should be encouraged in some other way. Then I will go on to explain other methods of encouraging and (to the extent actually necessary) funding software development.[15]

### *1.6 Copyleft vs. Non-Copyleft FOSS*

The GNU GPL licence, pioneered by Richard Stallman granted its users the rights to use, study and modify, copy, and redistribute the software, while mandating that any other software created using a GPL software, must be redistributed under GPL only. This created a trap for the software companies which wanted to build their software, to save on development costs, by using readily available GPL software. Such companies were invariably required to release their software under GPL. One reason why the bourgeoisie and the corporate world in particular always resisted this copyleft license was in the fear of losing their monopoly. One can say that GPL was the idea that 'touch me, and you have to become like me' in software world, a *virus of freedom.* Leaked *Halloween Documents* of Microsoft were found to say, "**The GPL is a viral license. It poses a threat to the intellectual property model of software development."**[16]

Eben Moglen, mentions in *Anarchism Triumphant*,

> ...As a visionary designer Richard Stallman created more than Emacs, GDB, or GNU. He created the General Public License...
> ...Section 2(b) of the GPL is sometimes called 'restrictive,' but its intention is liberating. It creates a commons, to which anyone may

[15] Stallman, Richard. 2021. 'Why Software Should Be Free'. *GNU.org.* https://www.gnu.org/philosophy/shouldbefree.en.html

[16] Valloppillil, Vinod, et al. 1998. 'Halloween Document I (Version 1.14)'. https://web.archive.org/web/20020125002335/https://opensource.org/halloween/halloween1.html

> add but from which no one may subtract...[17]

However, after the OSM took roots, the period of the 1990s saw many other licenses gain popularity, which were free software licenses in essence (as they granted their users the 4 essential freedoms), but did not mandate the propagation of freedom. Such licences, which came to be known as *permissive* licenses, allowed anyone including a multinational corporation, to use a free software to make their own software, without mandating them to release their code as free software. Examples of such licenses include the MIT license, the Apache license, etc. For example, the Apache 2.0 license says that you can add *"Your own copyright statement to your modifications and may provide additional or different license terms and conditions."* Notably the FSF grants accreditation to such licenses as Free Software Licenses, even though they do not mandate propagation of freedom. Some proponents of the permissive licenses, even called the copyleft licenses as "restrictive" licenses.

It should be noted that licences like MIT, Apache, apart from the fact that they allow using the code without mandating the passage of freedom, also allow patenting the software and placing a warranty. It is quite obvious that these *free software* licenses are more appealing to bourgeois interests.

Two more variants of the GPL, the Affero GPL (AGPL) and the GPL version 3 (GPLv3) were released later by GNU to cover more cases of software under copyleft umbrella, particularly after it was found that the businesses found ways to use copyleft software and still deny essential freedoms to users. AGPL mandated the distribution of software accessed over network, and GPLv3 mandated that users must be able to run the modified versions of software on a given hardware (anti-tivoization), more patent retaliation, and protection against Digital Rights Management (DRM).

The preference of the bourgeoisie for the so-called permissive licenses was for everyone to see. Apart from the stated benefits of being *easy* and *simple,* the obvious benefits of further commercialization and corporatization are obvious. Although, often coming under the garb of promoting "more freedom", the permissive licences always meant "more freedom" to the bourgeoisie. Many in the FOSS movement have echoed the same *sentiments*.

Tim O'Reilly said, "I think the BSD-style licenses are both more effective at creating more value in the world, and a better morality,..."[18]

Mike Olso, who built the Berkeley Database software said in support of dual licensing of the software, "**The GPL was kind of the poison, and we would sell you the antidote. If you preferred not to infect your source code with the GPL, you could buy a different license...**"[19]

**Bjørn Reese and Daniel Stanberg say while listing many benefits of co-operation with corporates, in 2001, "We see copyleft as a hindrance to this cooperation,**

[17] Moglen, Eben. 1999. 'Anarchism Triumphant: Free Software and the Death of Copyright'. *First Monday,* 4(8). https://doi.org/10.5210/fm.v4i8.684

[18] Bustillos, Maria. 2015. 'The GNU Manifesto Turns Thirty'. *The New Yorker.* https://www.newyorker.com/business/currency/the-gnu-manifesto-turns-thirty

[19] Metz, Cade. 2012. 'Open Sourcers Drop Software Religion for Common Sense'. *Wired.* https://www.wired.com/2012/02/cloudera-and-apache/

**as it only addresses the concerns of one side. It is our impression that copyleft licenses are selected much too often due to an unfounded fear of exploitation by commercial corporations. In the following we will explain why we choose not to release our open-source software under a copyleft license."**[20]

## 2. Prevalence of FLOSS Today

FLOSS has outgrown the proprietary software world today. A large number of major FLOSS projects like GNU/Linux Operating systems (Ubuntu, Debian, Fedora, Kali, etc), Libreoffice, VLC, Google Chrome, Android, etc. are part of everyday personal computer usage, and the internet infrastructure is largely powered by FLOSS software like GNU/Linux servers, web servers like Apache/Nginx, Database servers like MySQL/PostgreSQL/MongoDB, content management systems like Drupal/WordPress, DNS Servers, or programming languages like Python, Ruby, PHP, JavaScript, etc. Many important software projects like Signal Messenger, The Tor Browser, cloud software NextCloud, video conferencing software like Jitsi, social media platform software like Diaspora, which help protect privacy, and ensure security of end users have been FLOSS projects. It can be argued that every other major software technology development that has taken place in last two decades has been FLOSS. As per a report, FLOSS is now used in 92 percent of all the applications.

GitHub, the major, or in a sense the de-facto source code repository (the GitHub itself was never a FOSS project!) for FLOSS software, in 2022 hosted 85.7 million source projects' source code, had 94 million users, and 3.5 billion code contributions. These figures are themselves sufficient to demonstrate the very large scale at which FLOSS is being developed today.

Many estimates of the net-worth of the FLOSS software have been tried. According to the Linux Foundation Networking (LFN) Study in 2020, using COCOMO Model (that uses no. of lines of Code, Labour wages per month, etc. as parameters) some major projects (like FDIO, OpenDayLight, OPENFV, ONAP, OPX, PNDA, CNTT, Tungsten Fabric) with 87 Million lines of code, and estimated 7,01,282 Hours of effort could be valued upto $7.3 billion.[21] García-García and de Magdaleno estimated the cost of the Linux kernel, as of 2010, to 1.2 billion Euros.[22] According to Carlo Daffara, the lower bound of savings that OSS does bring to the European economy is at least 116B€. [23] According to the Linux Foundation, the cost of building a standard Linux distribution (in 2008) would have been over $10 billion, using COCOMO Model.[24] To compare this with the total size of the software industry, according to industry analyst Gartner, the size of the worldwide software industry in 2013 was US$407.3 billion. Thus, the FLOSS software was already comparable

[20] Reese, Bjørn and Daniel Stenberg. 2001. 'Working Without Copyleft'. Originally *O'Reilly.* https://daniel.haxx.se/transition.html

[21] Joshipura, Arpit. 2020. 'Quantifying the Value of Open Source Software'. *LFNetworking.* https://lfnetworking.org/quantifying-the-value-of-open-source-software/

[22] Lemire, Daniel. 2014. 'The Financial Value of Open Source Software'. *Daniel Lemire's Blog.* http://lemire.me/blog/2014/04/14/the-financial-value-of-open-source-software/

[23] Daffara, Carlo. 2011. 'The Economic Value of Open Source Software'. https://web.archive.org/web/20110708140850/http://carlodaffara.conecta.it/the-economic-value-of-open-source-software/

[24] Bauwens, Michel. 2009. 'Estimating the Development Cost of Open Source Software'. *P2P Foundation.* https://blog.p2pfoundation.net/estimating-the-development-cost-of-open-source-software/2009/09/28

in size to proprietary software way back in 2009. The amount of "savings" brought by FLOSS to the capitalist class run in billions of dollars. It is estimated that open source contributed £10.5 to £43.15 billion in value to the UK economy in 2021.[25]

As we will discuss ahead, FLOSS has often been likened to a "Commons". It is obvious that the worth of this commons is likely to be in trillions of dollars today.

# 3. The Bourgeoisie Has Triumphed

## *3.1 Corporates in FOSS Development*

The release of the Halloween documents, a confidential Microsoft document listing the threats of FOSS in 1998[26], clearly indicated the companies like Microsoft saw FOSS as a threat, but the engineers appreciated the FOSS development model. Microsoft CEO Steve Ballmer in 2001 referred to Linux as "a cancer that attaches itself in an intellectual property sense to everything it touches" and called some other day, Linux users 'a bunch of communist thieves'.[27]

However, Surprise! Satya Nadella, the CEO of Microsoft went on to say in 2015 that "Microsoft Loves Linux"[28]. Microsoft today hosts opensource.microsoft.com and has become one of the major contributor and supporter of the FOSS! This somersault, which is a characteristic of not only Microsoft but almost every other large software corporation (netflix.github.io, opensource.fb.com, opensource.apple.com, opensource.google, opensource.adobe.com, www.ibm.com/opensource/, netapp.io/, symantec.github.io, 01.org by Intel, github.com/twitter, opensource.oracle.com, consumer.huawei.com/en/opensource, linkedin.github.io, etc). According to a Znet Survey, 97 percent of UK businesses were using some form of Open Source Software.[29], while according to opensource.com 65 percent of companies are contributing to open source projects.[30] According to David Habusha, the VP of product at WhiteSource the top ten open-source projects today are managed by Facebook, Google, and Microsoft and 60 per cent to 80 per cent of every modern application's software stack consists of open-source code."[31]

According to a survey,

> IT firms are paying developers to contribute to open source. Microsoft purchased the GitHub code repository in 2018 for US$7.5 billion. It is maximising this investment: commits or code changes using the @microsoft email domain were by far the most numerous

[25] McCurdy, Will. 2021. 'Open Source Contributes 'Between £10.5bn to £43.15bn' to UK Economy'. *National Technology News*. https://nationaltechnology.co.uk/Open_Source_Contributes_Between_10.5bn_To_43bn_To_UK_Economy.php

[26] 'Halloween Document I (Version 1.14)'.

[27] Greene, Thomas C. 2001. 'Ballmer: 'Linux Is a Cancer''. *The Register*. https://www.theregister.com/2001/06/02/ballmer_linux_is_a_cancer/

[28] Microsoft Windows Server Team. 2015. 'Microsoft Loves Linux'. https://www.microsoft.com/en-us/windows-server/blog/2015/05/06/microsoft-loves-linux/

[29] Tung, Liam. 2021. 'Open Source Is Everywhere in Business. Here's What's Driving Adoption'. *ZD Net*. https://www.zdnet.com/article/open-source-is-everywhere-in-business-heres-whats-driving-adoption/

[30] LeClair, Haidee. 2016. '65% of Companies Are Contributing to Open Source Projects'. *OpenSource.com*. https://opensource.com/business/16/5/2016-future-open-source-survey

[31] Claburn, Thomas. 2020. 'Copy-left Behind: Permissive MIT, Apache Open-Source Licenses on the Up as Developers Snub GNU's GPL'. *The Register*. https://www.theregister.com/2020/01/17/mit_apache_versus_gpl/

in top GitHub repositories in 2015 -2019.[32]

To some, this is a victory of the FLOSS movement. They argue that the corporates were forced to adopt the FOSS, as they could not beat it in the competition. A very popular video on YouTube, "Truth Happens" about the success of Red Hat Linux operating system said, "First they ignore you, ... then they laugh at you, ... then they fight you, ... then you win."[33]

The truth unfortunately is the opposite. As we will argue ahead, the corporate world has recognized what the OSM tried to convince it, that the Open-Source development methodology is indeed a better way of organizing production and extracting more surplus value. The corporates can be seen increasingly controlling the development of the FOSS. As per a survey of GitHub repositories, "...as is the case for the Linux Kernel. Linux has by far the highest number of commits overall, 73% of which are authored by firm employees…"[34]

We will discuss this aspect further under various subheads.

### *3.2 The Rise of Non-Copyleft FOSS*

As discussed earlier, the non-copyleft Free Software licences, are indeed the ones which appeal to the commercial interests. The corporate 'interest' in FOSS is evident from the rapid growth of the use of non-copyleft free software licences like MIT and Apache over copyleft licenses. The use of permissive licenses on GitHub has increased from 41 percent in 2012 to 78 percent in 2021, while the usage of copyleft licenses decreased from 59 percent to 22 percent. While simplicity, flexibility, reduced barriers to entry, avoidance of legal complexities are some of the reasons often cited for this growth of permissive license usage, the fact cannot be ignored that all these reasons essentially underline *reduction in costs,* and the wide support of the corporate sector, the influence of major large projects (also supported by large corporations) like jQuery, Ruby on Rails, X11, Node.js, TensorFlow, all Apache projects, Pandas, Bootstrap, Kubernetes, etc. under permissive licenses is undoubtedly a major factor.

Corporates now see the existing huge body of FOSS, as a quick way of not only developing required skill-set among their workers, but also of developing complex and large software systems, as all they need to do is combine the existing FOSS projects, customize and add some parts of code on top of it. Large corporations like Facebook or Twitter, are running on software code that was largely borrowed from the FOSS projects.

"If most of the code comprising your product or service isn't open source software, it's highly likely that you're wasting effort and cash reinventing the wheel."[35]

[32] O'Neil, Mathieu, et al. 2021. 'The Coproduction of Open Source Software by Volunteers and Big Tech Firms'. *Digital Commons Policy Council and News & Media Research Center.* https://researchsystem.canberra.edu.au/ws/portalfiles/portal/50699042/apo_nid312607_0.pdf

[33] Red Hat. 'Truth Happens Remix' Video. Youtube. https://www.youtube.com/watch?v=5EkkMfjetEY&t=181s

[34] O'Neil, Mathieu, et. al. May 2024. 'Co-producing industrial public goods on GitHub: Selective firm cooperation, volunteer-employee labour and participation inequality'. New Media and Society. May 2024.26(5):2556-2592 DOI:10.1177/14614448221090474

[35] Spinellis, D. 2019. 'How to select open source components'. IEEE Computer. 52, pp. 103–106. http://dx.doi.org/10.1109/MC.2019.2940809

Mark Shuttleworth, the founder of Canonical, the company that leads the development of popular Ubuntu Operating System said in 2006,

> There are many examples of companies and countries that have improved their competitiveness and efficiency by adopting open source strategies. The creation of skills through all levels is of fundamental importance to both companies and countries.[36]

### *3.3 Acquisition and Control of FOSS projects by Corporates*

Many major open-source projects were acquired by large corporations. Microsoft purchased GitHub, the largest repository of open-source projects, hosting millions of projects, in 2018. Note that it was not a FOSS project, but a proprietary project that allowed FOSS to be distributed and hosted easily. These millions of software projects with their codes, have also arguably helped Microsoft train the OpenAI's ChatGPT artificial intelligence software, which can now generate software code of reasonable complexity. IBM purchased Red Hat, one of the biggest commercial vendors for a major Linux distribution, in 2019. Oracle purchased Sun Microsystems, then an owner of one of the largest Database software, MySQL, in 2009. The Linux Foundation Core Infrastructure Initiative is funded by Amazon, Google, IBM, Microsoft, Facebook, and Intel. These acquisitions and controls show that the biggest corporates now have a keen interest in controlling the direction of development of these FOSS projects.

The increasing control that the multi-national corporations funding the community-led FLOSS projects exert on the development, and on the developers of the FLOSS is also becoming evident. According to BoringCactus, a blog (quoted verbatim)

> ...big mozilla layoffs that included a lot of people working on cool and important shit. the consensus i'm seeing is that it reflects mozilla's search for profit over impact, mismanagement, and disproportionate executive compensation. this is taking place in a larger trend of corporatization of open source over the past several years,...[37]

### *3.4 Software as a Service (SaaS) Threat*

The next level of threat to FLOSS arose with the emergence of Software as a Service (SaaS) products.

Traditionally users installed software like Microsoft Office, or LibreOffice on their devices, and executed it locally on that device. However, from the late-1980s or early-1990s, the client-server model of software architecture started becoming popular. In this model, the software was divided in two parts. First, a server, which was running on a remote networked computer, and another, a client, running on the networked PC. Together, both the client and server delivered a service like

[36] Otter, Alastair. Jan 30, 2006. 'Go Open Source puts R3m into building Linux channel'. Tectonic Blog. https://web.archive.org/web/20220701052316/https://tectonic.co.za/?p=840

[37] Author Unknown. 2020. 'Post-Open Source'. *Boring Cactus Blog*. https://www.boringcactus.com/2020/08/13/post-open-source.html

accessing a website (e.g. Apache Web Server and Firefox browser), a database, etc. In this model the server, a more critical part of the software, was not required to be distributed.

Today web-based software services like Google Docs, or online photo editing applications like Canva, etc. have become extremely popular. In this model, which is an extension of the client-server model, end-users get only a service using a client software that runs *in* their internet browsers like Firefox or Google Chrome. Since the emergence of this SaaS model, and also the erstwhile client-server model, the entire FOSS ecosystem, including the copyleft FS movement have been pushed back by a wide margin, as the reciprocity of GPL cannot be not mandated now.

Most interestingly the servers running such services are using FOSS software to a large extent but are not required to deliver the software to the end-users. Licenses like AGPL tried to circumvent it, but have met limited success, and in fact companies like Google have banned its employees from using AGPL.[38] The SaaS model has enabled the IT companies to turn FOSS into practically a proprietary software. This could not have been foreseen by the proponents of the GPL and later efforts to find legal hacks like GPL have not found much success.

### *3.5 Bonhomie Between Corporates and Communities*

The FSM always stood *only* for software freedom and thus it never had any apprehensions about working with the businesses, if they agreed to the terms of FS. The FSM activists, organizations, and groups can often be seen attending and organizing events sponsored by the large corporations and smaller businesses. To cite a few examples, Apache Software Foundation sponsors include Google Microsoft, Facebook, IBM, Red Hat, Twitter, Comcast, Lease web reliable hosting, etc. The Free Software Foundation itself is sponsored by likes of Alibaba, Barracuda, Red Hat, etc. IBM, Intel, Google, and Microsoft, for example, are all “platinum donors” to the Linux Foundation, which employs developers to work on the Linux kernel full time. Thus, some of the largest FLOSS projects and communities are indeed sponsored by software sector bourgeoisie.

### *3.6 FOSS Model of Development in the Service of the Corporates*

The FOSS model is now being used in interesting and “innovative” ways by the businesses to further their interests and make proprietary or profit-making use of the software.

One interesting example of the collaboration among technology giants is the Data Transfer Initiative[39] (Apache licensed) jointly driven by Google, Apple, Facebook, which aims to help users move their data from one platform to another. Another, similar, but a larger scale example is that of OpenStack, which is a core software that runs many of the cloud servers today. It was developed as a result of collaboration among the biggest storage/virtualization companies, 675 in total, coming together. The most interesting fact is that these rival companies have come together to develop a software that will be free software (under Apache license)! Clearly, all these companies do see a higher rate of profit in “investing”

[38] Google. n.d. ‘AGPL Policy’. https://opensource.google/docs/using/agpl-policy

[39] Data Transfer Initiative. n.d. https://dtinit.org/

in non-copy-left free software paradigm. All these companies benefit from the 'gratis' (free as in free of cost) contributions of programmers.

Further, such companies now get to dictate the direction of development of the software, as many core programmers working on the software get paid by these corporations. As per a report on Vice,

> ...According to Moore, there are over 1,200 IBM employees contributing to 1,000 open source projects on both company time and in their freetime. Moore said that 'many' IBM employees work on open source projects such as Linux, Kubernetes, Java, and Node. js full-time and that the company offers annual rewards to top IBM developers in various open source communities.
> ...Google employees that work full time on open source projects, he said that the company has over 2,000 active open source projects on GitHub. In particular, DiBona highlighted work on the programming language Go and the container software Kubernetes as examples of widely used open source projects created at Google that employees work on everyday.
> ...Android operating system, which is found on 86 percent of all smartphones sold in the world. Android is open source, but almost all of the development work on the OS is done internally at Google. At the same time, Google is also paying engineers to develop some of the proprietary applications that are generally seen as the biggest selling point of an Android operating system such as Google Maps and Gmail.[40]

Further, according to another report on FreeCodeCamp,

> Microsoft appears to have ~1,300 employees actively pushing code to 825 top repositories on GitHub.
> Google displays ~900 employees active on GitHub, who are pushing code to ~1,100 top repositories.[41]

Another interesting phenomenon, of corporate interest in FLOSS is the Google Summer of Code (GSoC). As a part of GSoC, Google invites contributions from students programmers all over the world into *third party* FLOSS projects. These projects have on the face of it, no association with Google. These contributions are done over a period of 12 to 20 weeks and the participants are required to solve some problems faced by the FLOSS communities. There is a tremendous competition among college students all over the world, to get selected for GSoC. On successful completion of the task, Google (!) pays the programmers "stipends" depending on the PPP of the country, for example up to $3000 for India. Why is Google doing this? Being one of the largest consumers of code, being one of the

[40] Oberhaus, Daniel. 2019. 'The Internet Was Built on the Free Labour of Open Source Developers. Is That Sustainable?'. *Vice.* https://www.vice.com/en/article/the-internet-was-built-on-the-free-labor-of-open-source-developers-is-that-sustainable/

[41] Hoffa, Felipe. 2017. 'Who Contributed the Most to Open Source in 2017 and 2018? Let's Analyze GitHub's Data and Find Out'. *freeCodeCamp.* https://www.freecodecamp.org/news/the-top-contributors-to-github-2017-be98ab854e87

largest software corporations that needs to solve multitude of coding problems, Google is clearly investing into FLOSS projects, as it benefits from most of them getting developed.

Further, the larger the corporation, the larger is its ability to benefit from FOSS. Larger corporations can also be seen deriving benefit from work of smaller companies or communities. According to David Habusha, "Companies behind popular open-source projects like ElasticSearch, Redis Lab, Docker and many more have blamed the big three cloud providers for monetizing their open-source projects without giving back to the open-source community,"[42] One should note here that all these three projects are under non-copyleft Free Software licenses.

It is argued, on other side, that the use of Affero GPL (AGPL) for cloud software (like Nextcloud, ownCloud – private cloud software) shows the opposite trend. According to Paul Berg, an open-source licensing consultant,

> On the other side of the spectrum though, particularly in the area of cloud computing, we are seeing a resurgence of interest in extremely strong copyleft licensing, such as the AGPL, which is even less permissive than the GPL, because it has stronger guarantees that consumers of that software will remain members of the community rather than simply extend and repackage the software for their own sole benefit.[43]

Here, one should note that private cloud offerings are best utilised, with the additional (add-on) software plugins that make the cloud more attractive. In such a scenario, inter-operability of the add-ons become more a pressing issue than keeping some part of the code-private, and hence it *does* benefit large corporations, because while they add some add-ons, they also get inter-operability with free-of-cost add-ons of other developers.

### *3.7 More Profits to the Corporates*

A recent 2023 survey by Linux Foundation says,

> The median economic value of OSS is 1-2 times its cost...
> ...perceived benefits of utilizing OSS significantly exceed these costs for the large majority of organizations that use OSS...
> Indeed, some organizations adopt OSS even when its costs seem to be higher than alternatives because of the improvements in the speed of creating and deploying the code...
> Beyond cost savings, the study also found that OSS promotes innovation within organizations. Respondents reported that OSS encourages collaboration and knowledge sharing among developers, which can lead to new and innovative ideas. Moreover, OSS allows developers to customize software to fit their needs, increasing efficiency and productivity. The survey also found that OSS improves software quality. Respondents reported that using OSS leads to better code quality, as the community-driven nature of

[42] 'Copy-left Behind: Permissive MIT, Apache Open-Source Licenses on the Up as Developers Snub GNU's GPL'

[43] *Op. sit.*

> OSS encourages peer review and testing. Furthermore, OSS allows organizations to quickly fix bugs and release updates, resulting in more reliable and secure software.

Many examples can be cited where the FOSS communities and the corporates can be seen acknowledging the benefits of improved worker efficiency, in fact at times enforced improvements in worker efficiency, and free contributions of volunteers.

Bradley Kuhn, a prominent Free Software activist, participated in discussion with Ash Dryden and said in agreement acknowledging that there is an co-option of the FSM,

> With companies and even many individuals so rabidly anti-copyleft recently, I suspect that everyone in the discussion is assuming that the *underlying license structure of these volunteer contributions is non-copyleft.*
> ...copyleft ensures that volunteer contributions do have, for lack of a better term, some strings attached: the requirement that even big and powerful companies that use the code treat the lowly volunteer contributor as a true equal.
> These days, most developers, even though they are required to use some Free Software as part of their jobs, usually are assigned work on some non-Free Software that interacts with that Free Software *'volunteer to work on Free Software so you can get a job working on some proprietary software'. That practice is a complete corruption and cooption of the Free Software culture.* (*emphasis ours*)

The benefits of cost-saving by availability of 'gratis' copies of FLOSS to the software bourgeoisie is immediately evident. Many instances can be found where the corporate big honchos talk of the savings brought to them by FLOSS. A Forbes article, by John McCann (CEO, NodeSource) says,

> Barclays, a British international banking and financial services provider, found that using open source software reduces their software development costs by 90%.
> Netflix is able to maintain their cheap monthly prices for consumers because their platform is entirely built on open source software.[44]

While the corporates benefit from FOSS, they are not mandated to 'give back'. Bruce Perens admitted in 2023, in 'The Register" that "...So I feel that IBM has gotten everything it wants from the open source developer community now, and we've received something of a middle finger from them."[45]

He, an Open Source proponent himself, failing to acknowledge the real character of Open Source Software itself, further admitted about *open source* that

> is that Open Source has completely failed to serve the common

[44] McCann, Joe. 2018. 'Why Pay For Something When It's Free?'. *Forbes.* https://www.forbes.com/sites/forbestechcouncil/2018/02/22/why-pay-for-something-when-its-free/

[45] Claburn, Thomas. 2023. 'What Comes After Open Source? Bruce Perens Is Working on It'. *The Register.* https://www.theregister.com/2023/12/27/bruce_perens_post_open/

> person. For the most part, if they use us at all they do so through a proprietary software company's systems, like Apple iOS or Google Android, both of which use Open Source for infrastructure but the apps are mostly proprietary. The common person doesn't know about Open Source, they don't know about the freedoms we promote which are increasingly in their interest. Indeed, Open Source is used today to surveil and even oppress them.

### *3.8 How is FLOSS Now Benefitting the Software Bourgeoisie*

While the investments in automation and scale of infrastructure, that is *constant capital* in the IT industry kept increasing over years, *FLOSS* has helped the software bourgeoisie achieve much less expenditure on the *variable capital,* that is wages, in various ways.

Among the various costs that the software industry incurs, including the cost of purchasing labour-power mainly in the fields of support, product documentation, customization of software, training of personnel, against the costs of constant capital in infrastructure (hosting, bandwidth), other supporting software-hardware and tools, one can see that the wages component is relatively reduced with the use of OSS to small or large extent. In fact, the costs of other software tools have also reduced, given they are OSS themselves.

It can be clearly seen today that it is the large multinational software corporations which are controlling and driving the growth of the FOSS movement and have become some of the major *freeloaders* of the efforts of millions of programmers all over the world. They now benefit from (a) free-of-cost available billions of lines of code; (b) the increased productivity of their employees due the training provided by the FLOSS software and technically better quality of the FLOSS software; (c) gratis contributions in terms of reporting bugs, fixing bugs, and writing new code by a large pool of volunteers; and, (d) savings on the cost of licensing from other companies, had the software been a proprietary software.

### *3.9 New Licenses, That are 'Anti-Capitalist'*

The recognition that copyleft or non-copyleft Free Software, and Open-Source software is being co-opted by the capitalist class was not lost to many. Efforts were and are being made to redefine new licenses, although without much adoption, to compel the software bourgeoisie pay in monetary terms, for the commercial use of the FOSS. The copy*far*left licenses are examples of this.

A wiki article on the Peer Production License (PPL) says,

> The peer production license is an example of the Copyfair type of license, in which only other commoners, cooperatives and nonprofits can share and re-use the material, but not commercial entities intent on making profit through the commons without explicit reciprocity. This fork on the original text of the Creative Commons non-commercial variant makes the PPL an explicitly anti-capitalist version of the CC-NC.
>
> ...the basic pragmatics is that that any organisation that employs

> wage-labour, that is, is not a collective, co-operative or individual, would need to buy a license to distribute or otherwise use the works of the copyright collective. It is exactly the same as non-commercial, except that commons-based commercial use is allowed.[46]

It should be noted, that these licenses only make the programmers beneficiaries of the commerce, effectively making the workers claimants to wages, or ostensibly a share in the surplus value and in no way socialize the total wealth. The adoption of such licenses has been low.

What is the problem with these 'anti-capitalist' licenses? That they are not anti-capital! The whole model is based on cooperation among individual producers and a kind of equalitarian or egalitarian approach towards wages. As we discuss later in this essay, even these licenses assume, suggest and hence support the existence of generalised commodity economy that is market economy, and hence capitalism. They assume that collective revenue (the 'income' from commodification of their software) collection amounts to collective property (hence anti-capitalist!), as they ignore the necessity of control over other necessary means of production that are in the hands of the bourgeoisie. They fail to realize that if their work does not find any commercial use, then the only way for the members of the collective to survive is to work as a wage-labour somewhere else. They fail to recognise that even a collective employs(!) wage-labour, as the share of revenue its members receive is both wages, and share of the surplus value they have generated themselves, one realized through the commercial use of their software. Further, it seems to assume that the software bourgeoisie will be compelled to use their software and accept the terms of their *copyfarleft* license, which is possible only if highly skilled workers are able to form a collective and offer a competing price to some other solution that the bourgeoisie can get developed themselves by paying wage-labour. Thus this entire effort remains circumscribed by the boundaries of capitalist economy, in the absence of a revolutionary ideology and revolutionary organization challenging capitalist private property in general. Licenses like this remain yet another romantic idea of challenging capitalism.

### *3.10 Crowdfunding Fails*

The question of how the programmers will earn, was always treated as secondary to the needs of 'freedom' by RMS and the FSM. RMS argues saying "The question 'How can we pay programmers?' becomes an easier question when we realize that it's not a matter of paying them a fortune. A mere living is easier to raise"[47]. He further argues that hardware manufacturers, university programming projects, getting paid for software services on free software, funding by non-profits or charities like the GNU could be some ways of paying the programmers, and programming is "fun" and comparable to music and art. This, surely, is not a recipe to employ millions of trained programmers who are out in the labour market scouting for a job. Moreover, in capitalist society even musicians and artists do

[46] Copyfarleft. *P2PF Wiki.* https://wiki.p2pfoundation.net/Copyfarleft#:~:text=%22the%20basic%20pragmatics%20is%20that%20that%20any%20organisation,non-commercial%2C%20except%20that%20commons-based%20commercial%20use%20is%20allowed.

[47] 'Why Software Should Be Free'

productive labour and get paid (sometimes handsomely, at other in meagre fashion). The comparison itself smacks of a complete lack of understanding of the essence of present mode of production and social formation as a whole.

Capitalism keeps on transforming everything into a commodity. This includes works of music and art, too. In capitalism, many programmers, or musicians and artists, choosing to live a life beyond the commodified labour market can only be termed as a romanticist idea. As we will demonstrate later in this essay, free software programmers, who chose to live a frugal life, which is commendable, remains a minority trend. Moreover, this in itself does not prove a thing regarding the general character of the present society and system.

We can see James Turner saying, (originally in O'Reilly Radar as 'FOSS isn't always the answer', which was re-published as 'Proprietary software isn't evil' in 2011)[48],

> But they need to accept the ground rules that most of us live in a capitalist society, we have the right to raise and provide for a family, and that until we all wake up in a FOSS developer's paradise, we have to live and work inside of that context. Drop me a note or post a comment. I'd love to hear how a proprietary-free software world could work.

The idea of 'crowd-funding', a word which does not mean the same as working class or working masses because crowd can mean anyone including a corporate, was often seen as a way to fund *volunteer* programmers who wrote FOSS. However, expectedly, experience shows that 85 percent of the crowdsourcing campaigns fail.[49]

According to some estimates, the largest share of FOSS contributors comes from the USA, and Europe[50], with USA, Germany, France and UK contributing more than 50 percent of the work-force, while countries like India (4 percent) and China (2 percent) contribute very small to the pool. This indicates that it is possible for programmers in large relatively more developed capitalist countries, to find many reasonable ways of making a living, which includes getting paid to work on FOSS projects, and also contribute to FOSS in their *free time*, but it remains difficult for programmers in relatively less developed capitalist countries like India.

### *3.11 Voluntarism Fails: Its Jobs Over FSM for the Programmers*

The hope that FSM had, that a large number of programmers would 'rebel' and write copyleft code, to challenge the monopoly of the big multinationals has not fructified. It can also be seen from the studies showing motivation that programmers have, in contributing to FLOSS. According to a recent study in 2020, 'The Shifting Sands of Motivation: Revisiting What Drives Contributors in

[48] O'Reilly Media. Jul 22, 2011. 'Proprietary software isn't evil'. Forbes. https://www.forbes.com/sites/oreillymedia/2011/07/22/proprietary-software-isnt-evil/

[49] Arora Project. Oct 29, 2021. 'The Dark Side of Crowdfunding: Why Campaigns Fail'. https://www.aroraproject.co/2021/10/29/the-dark-side-of-crowdfunding-why-campaigns-fail/

[50] Nagle, Frank, et al. n.d. 'Report on the 2020 FOSS Contributor Survey'. *The Linux Foundation & The Laboratory for Innovation Science at Harvard.* https://8112310.fs1.hubspotusercontent-na1.net/hubfs/8112310/2020FOSSContributorSurveyReport_121020.pdf

Open Source'[51], the major factors that motivate the developers to contribute to FLOSS projects are neither ideological, nor political, but ones around personal development and, hence, better job opportunities. According to the survey, among different factors, “I want to develop and improve my skills” ranks first, while “I want to limit the power of large software companies” ranks sixteenth , “I dislike proprietary software and want to defeat them” ranks eighteenth, while “I'm paid to contribute” ranks twentieth . Ideology and *altruism* remain the factors at the bottom of the chain.

This once again shows that FLOSS projects have become a training ground for more efficient workforce, thus raising the overall labour productivity in software world, and thus indirectly benefitting the large corporations with higher rate of surplus value.

### *3.12 Free Labour in the Service of Capital*

The bourgeoisie always tries to push down the wages and would be delighted to benefit from free labour. Marx says in *Capital*, Volume-1,

> But if the labourers could live on air they could not be bought at any price. The zero of their cost is therefore a limit in a mathematical sense, always beyond reach, although we can always approximate more and more nearly to it. The constant tendency of capital is to force the cost of labour back towards this zero.[52]

The free (as in gratis) contributions of millions of volunteers in the world, are now serving the interests of the very software bourgeoisie that the FSM wanted to defeat. In a sarcastic tone, commenting on the ‘remuneration' that FLOSS programmers get, an article by Abraham de Jesus says, "*...this movement convinces many of it's participates that their contributions of free labour will be equally rewarded or that, when exploitation is enjoyable, it magically stops being exploitation."* [53] (*emphasis ours*)

According to a blog by BoringCactus (quoted verbatim) [54]

> *what's good for corporations is, of course, bad for people. random individual contributors almost never get paid for their work*, even when a corporation or several will profit substantially from those changes. maintainers of vital infrastructure libraries generally only get paid if they wrote the library for or under the control of the company they worked for anyway. professional, corporate maintainers can offer more to the community since they're getting paid for it, which heightens expectations on independent maintainers and leads to maintainer burnout.[54] (*emphasis ours*)

[51] Marco, Igor, Bianca, Georg, Gregorio, Christoph, Igor, and Anita. 2020. ‘The Shifting Sands of Motivation: A Survey with OSS Contributors'. https://doi.org/10.5281/zenodo.4453904

[52] Marx, Karl, and Frederick Engels. 2010. *Collected Works*, Vol. 35. Lawrence & Wishart, London, p. 596.

[53] de Jesus, Abraham. 2013. ‘The Tyranny of Open'. *Academia.* https://www.academia.edu/4636449/The_Tyranny_of_Open

[54] BoringCactus Blog

A large number of companies ‘allow’ or rather encourage their programmers to contribute to FLOSS projects. This is obviously done to increase the productivity of the labour and also benefit from the pool of FLOSS they create. At the same time the ‘voluntary’ contributors are essentially contributing free labour in the service of the bourgeoisie. A blogger, Isaacs wrote in 2013,

> Many employers are willing to ‘let’ their employees hack on their own projects in their “personal” time, or contribute back to projects that impact the bottom line (often with the stipulation that the copyright to the contribution remains the property of the employer). This is not ideal. *‘Free’ time isn’t free; what it costs is your life.* Spending it working deprives you of the opportunity to be a complete person....
>
> Some forego the corporate gig, in favor of being nomads and starving artists. They take on minimal employment requirements, if any, and *spend the rest of their time being productive on open source. But it’s a rough way to live....*
>
> It’s a shame that these artists and nomads are not making more money for the value they produce, and an even worse shame that such a high level of dedication and sacrifice is required for something that benefits the rest of us so much. They are capable of producing tremendous value, but most of them can’t afford the price. (*emphasis ours*)

Ash Dryden, a well-known programmer, wrote that programmers are forced to work on the OSS now (quoted verbatim). "While that on the surface is true, the imposition is there. Many jobs require open source contributions to even consider a candidate." [55]

All this, while a large number of FLOSS programmers are not rewarded monetarily at all for all the work that they do. Ash Dryden also, rightly, argues that *corporates are the freeloaders now*,

> People who are contributing their unpaid and underpaid labour are investing their time into companies that are profiting greatly and giving little back in terms of financial support.
>
> Open source originally broke us free from the shackles of proprietary software which forced us to 'pay to play' and gave us little in the way of choices for customization. Without realizing it, we've ended up in a similar scenario where we are now paying for the development of software that large companies financially benefit from with little cost to them.

## 4. The Ideological Flaws of the Free Software Movement

### *4.1 The Ideological Basis of the FSM in Anarchism*

Though the FSM was continuously accused of being a ‘communist’ movement by the software corporate world, the chief protagonist of the FSM, Richard Stallman never attributed any particular ideology to the Free Software Movement. Stallman

[55] Dryden, Ash. 2013. ‘The Ethics of Unpaid Labour and the OSS Community’. *Blog of Ash Dryden.* https://www.ashedryden.com/blog/the-ethics-of-unpaid-labor-and-the-oss-community

has always avoided questions on socialism, communism, or even being anti-capitalism. Neither does the Free Software Foundation mention its opinion on any point beyond software freedom. Stallman can be seen criticizing the mega-corporations on his personal blog, on their role in destroying environment and healthcare, and for being economic vultures, and that's it. Stallman, from his publicly expressed opinions, can best be described as a libertarian anarchist. The FSM, being a loose confederation of many individuals and groupings does not have any ideological charter or manifesto. However, occasionally Stallman could be seen making comments on ideologies like Anarchism or Communism. On one occasion, in around 2013, when asked whether the FSM is anarchist, Stallman replied,

> Free software combines aspects of catholism, aspects of communism and aspects of anarchism. The communist aspect is the idea that we all work together to make the things that we use, the catholist aspect is that everybody is welcome to start a business involved in this and make money as long as it respects the freedoms of others, and the anarchist aspect is that you do what you want to do, you contribute if you want to contribute, and what's available you use if you want to the way you want to.[56]

The above quote with extraordinary confusing mixture of ideologies shows that either Stallman lacks the understanding about communism (and definitely the Marxist perspective on Communism), political economy and the contradictions among these three ideologies, or he was just trying to be tactically safe about associating or not associating with any ideology. That Stallman is not a communist, is also evident from this bourgeois stereotypical criticism of the USSR,

> Communism as was practised in the Soviet Union was a system of central control where all activity was regimented, supposedly for the common good, but actually for the sake of the members of the Communist party. And where copying equipment was closely guarded to prevent illegal copying.... Thus, if we are to judge views by their resemblance to Russian Communism, it is the software owners who are the Communists.

The anarcho-syndicalist notion of voluntary co-operation, and not one based on a new structure of society where co-operation is the way of life, is evident when Stallman says, "I hope that the free software movement will contribute to this: at least in one area, we will replace the jungle with a more efficient system which encourages and runs on voluntary cooperation."[57]

Stallman was always open about him being idealist, "My work on free software is motivated by an idealistic goal: spreading freedom and cooperation. I want to encourage free software to spread, replacing proprietary software that forbids cooperation, and thus make our society better."[58]

[56] Stallman, Richard. 2013. 'Is Free Software Anarchist?'. *YouTube.* https://www.youtube.com/watch?v=WVqCqgmTWCk

[57] 'Why Software Should be Free'

[58] Stallman, Richard. 1998. 'Copyleft: Pragmatic Idealism'. *GNU.* https://www.gnu.org/philosophy/pragmatic.en.html

He pitched freedom, against profits, as an idealist concept,[59] "And if cynics ridicule freedom, ridicule community…if 'hard-nosed realists' say that profit is the only ideal…just ignore them, and use copyleft all the same."

Apart from RMS, the anarchist roots of the FSM can be easily seen in numerous writings of some of the major proponents of FSM. One of them is Eben Moglen, who for a long time was the legal counsel of the FSF, and is the founder of the Software Freedom Law Center. Moglen wrote in 1999, 'Anarchism Triumphant: Free Software and the Death of Copyright', clearly associating anarchism with the FSM. One of the sections of this article is titled 'Anarchism as a Mode of Production', essentially promoting FSM as an idea which can end capitalist mode of production and bring in a new 'anarchist' mode of production, against 'social inequity'. Moglen says in this article,

> ...most significant difference between political thought inside the digirati and outside it is that in the network society, *anarchism* (or more properly, anti-possessive individualism) *is a viable political philosophy...*
> Those of us who are worried about the social inequity and cultural hegemony created by this intellectually unsatisfying and morally repugnant regime are shouted down. Those doing the shouting, the dwarves and the droids, believe that these property rules are necessary not from any overt yearning for life in Murdochworld - though a little luxurious co-optation is always welcome - but because the metaphor of incentives, which they take to be not just an image but an argument, proves that these rules - despite their lamentable consequences - are necessary if we are to make good software (*emphasis ours*)

Even Eric Raymond, could not resist quoting the anarchist Peter Kropotkin and his concept of 'mutual aid' (called by him the principle of shared understanding) in his 'Cathedral and the Bazaar',

> Linus, by successfully positioning himself as the gatekeeper of a project in which the development is mostly done by others, and nurturing interest in the project until it became self-sustaining, has shown an acute grasp of Kropotkin's 'principle of shared understanding'. This quasi-economic view of the Linux world enables us to see how that understanding is applied.

Eben Moglen, in *Anarchism Triumphant*, asserted the FS as a 'commons' a term commonly used by anarchists,

> Why do people make free software if they don't get to profit? The wrong answer is embedded in numerous references to 'the hacker gift-exchange culture.'...
> *Free software, at the risk of repetition, is a commons: no reciprocity ritual is enacted there. ...* (*emphasis ours*)

[59] 'Copyleft: Pragmatic Idealism'.

And he further contended that anarchists would prove that they can be better than the propertied class (which essentially, according to us, is the bourgeoisie), essentially also relying on the voluntary spontaneous action of the software programmers in challenging the corporates. Comparing to news services he says,

> It isn't the wire services that have the advantage in covering Kosovo, that's for sure. Much less those paragons of 'intellectual' property, their television lordships. They, with their overpaid pretty people and their massive technical infrastructure, are about the only organizations in the world that can't afford to be everywhere all the time. And then they have to limit themselves to ninety seconds a story, or the eyeball hunters will go somewhere else. *So who makes better news, the propertarians or the anarchists? We shall soon see.* (*emphasis ours*)

Eric Raymond, says in *Cathedral and the Bazaar*, that the Open Source Movement will win because it will attract programmers in much larger numbers than corporates can. He points out,

> Perhaps in the end the open-source culture will triumph not because cooperation is morally right or software 'hoarding' is morally wrong (assuming you believe the latter, which neither Linus nor I do), but simply because the commercial world cannot win an evolutionary arms race with open-source communities that can put orders of magnitude more skilled time into a problem.

Many of those who were attracted to FSM believed that this ‘new” ideology can serve as a basis of transformation towards a property-less society. Wendy Liu, associated with *‘New Socialist’* says[60]

> If you take the core tenets of free software to their logical conclusion, you end up with a desire to reverse all kinds of commodification by transforming property rights in their entirety. As a result, today’s open source communities have the potential to serve as gateways to a more radical politics, one that pushes for the decommodification of not just information but also the material resources needed to sustain the production of information.

Thus, one can clearly see that the FSM based itself on ideological concepts deriving from anarchist or anarcho-syndicalist ideas of commons, absolutist notion of freedom, ‘mutual aid’ or voluntary co-operation, spontaneity or voluntarism, etc. These are reincarnations of the petty-bourgeois ideas that anarchists like Bakunin and Proudhon preached and Marx-Engels fought against. The failures of the FSM can only be understood by understanding how the basic ideas of the FSM are unscientific and hence fail the test of social practice. Among other, decentralization, internet connected social-networks as tools of democratic control and social liberation are ideas that also find significant influence in FSM and OSM

[60] Liu, Wendy. 2018. ‘Freedom Isn’t Free’. *Logic(s)*. https://logicmag.io/failure/freedom-isnt-free/

communities, however we will consider them in a separate essay later.

### *4.2 Commons Is Not Really 'Common' to All*

Those having radical hopes from the Free Software Movement fail to understand that (a) Capitalism has been privatising and commodifying commons since its inception; (b) Commons may co-exist with capitalism without causing any challenge to the mode of production by their mere existence; (c) A commons can benefit the bourgeoisie much more than the working masses; and, (d) Knowledge, and hence software, has peculiar quality of being commons by its very nature, as sharing it does not divide, diminish or displace it, but multiplies it.

In an existing class society, where exploiters and exploited exist, its the exploiters who stand to benefit from the so-called 'commons' more. The exploiters, by the very nature of their dominant position can use common (non-commodified) resources also as a ground of exploitation, or claim a larger share of the product of the commons, or simply benefit from the fact that the commons serve as a support for the exploited to survive and weaken their political consciousness and unity.

The process of primitive accumulation of capital, a process that is constitutive of capital-labour contradiction and yet continues during advanced capitalist accumulation as well, the dispossession of the direct producers from the means of production (either individually or commonly held) that constitutes the polar opposites of capital and wage-labour, necessarily involves the privatization of the existing 'commons', besides depriving the individual simple commodity producers from their means of production. Many examples of capitalism destroying the 'commons' and converting them into private property and commodifying them exist. The Enclosure Movement in England, from the 15th to the 19th century, led by the emerging agricultural bourgeoisie displaced millions of peasants from the 'commons', that is, pastures, forests, water bodies, agricultural land and created enclosed private land tracts of millions of acres. Similarly, the Scottish Highlands were converted from 'commons' to private property by forceful evictions. The colonial rule by the European imperialists, resulted in destruction of 'commons' all around the world: the Indian forests getting snatched from the Adivasis, the capture of the land of the Australian Aboriginals, the land grabs in Africa and Latin America, are some of the examples. Water, traditional knowledge, seeds, works of art like songs, music and stories, oceans, and even air have been privatized by capitalism.

The misconception that communal properties will escape the brunt of capitalism has plagued many, including the Narodniks in Tsarist Russia. The Narodniks placed their hopes in the 'mir' lands, the communal lands, hoping for their growth leading to communism(!). Lenin destroyed these misconceptions by demonstrating how capitalism was slowly growing in Russian agriculture, destroying the commons. The Stolypin reforms launched in 1907 in Tsarist Russia gave the final impetus to privatization of 'mir' lands, and validated Lenin's theorizations.

Compare commons to free-speech. In bourgeois democratic societies, speech is 'free', but those who own the means of communication have far more freedom of speech than those who don't. Similarly, those who have more means of accessing the commons will benefit more from it, than those who don't, *practically* converting it into a *privatized* resource. Further, an in-principle access, to the commons does not guarantee *actual* access, as it depends on many other economic factors. It should also be understood that just the 'commons' in themselves do not exhaust all means of production and subsistence. A commonly held land like 'mir' does not automatically guarantee commonly held horses, ploughs and other means of production. Similarly, a digital commons like software is never the complete set of means of production. An individual programmer may have tools of production like a PC or a laptop, but development and deployment of any large-scale utility software demands means of production like server-farms, electricity generators, land, etc. which is owned by the bourgeoisie.

For example, with FOSS one can in-principle run his/her own server for his/her own access to social media, or email, or data, but one may not do so because it costs money. One can also, in-principle, build AI-tools using FOSS, but the access to computational resources costs money. Anyone can in-principle run GNU/Linux on their laptop but will not do so because it costs time to learn or money to hire someone to fix issues. Thus, an isolated commons, without the complete socialization of the means of production, remains a rosy island accessible to few in a capitalist class society.

Large corporations benefit more from the commons, compared to the smaller companies. A report, itself financed by Alfred P. Sloan Foundation and the Ford Foundation's Critical Digital Infrastructure fund (2019-2020) says

> We analyse how the IT press portrays this coproduction: the issue of volunteer labour is absent. We show that large and small IT firm employee presentations at open source conferences convey opposed visions of digital infrastructure, business models, and the firm-community relationship.
>
> The IT news media, big tech firms and commercial foundations define firms and projects as a unified 'community.' Yet big tech firms such as Amazon are using cloud computing and Software as a Service to transform open source software, which is intended to be shared and modified, into closed assets.[61]

The hope that Eben Moglen puts in software commons, in *Anarchism Triumphant*, when he says that "no reciprocity ritual is enacted there" (as quoted above) is obviously misplaced, as the 'no reciprocity ritual' is exactly what the bourgeoisie loves and has been doing to the maximum possible extent.

This, however, does not negate the significance of the digital commons in the sense that they do stand to benefit the people, to the extent that some people can gain access to these commons. However, only with socialization of all the means of production, can the digital commons realize its true potential.

### *4.3 Unscientific Conception of Property*

The anarchists do not understood what 'property' is. They do not understood that the ownership has a class character and there is nothing called abstract ownership

[61] 'The Coproduction of Open Source Software by Volunteers and Big Tech Firms'

of resources. They do not understood that ownership of means of production has also changed its class character with the historical development of society, for example capitalist ownership of land is different from feudal ownership, as the very notion and purpose of this ownership is different and it serves a different role in the economic structure. That is why they can not understand the concept of socialist ownership as well and imagine the 'end' of ownership with a revolution. The famous debate between Proudhon, and Marx highlights this significant different.

Proudhon says, in 'What Is Property? Or, an Inquiry into the Principle of Right and Government' "*Property is robbery!*" [62]

In reply to which, Marx says,

> The upshot is at best that the bourgeois legal conceptions of "*theft*" apply equally well to the "*honest*" gains of the bourgeois himself. On the other hand, since "*theft*" as a forcible violation of property *presupposes the existence of property*, Proudhon entangled himself in all sorts of fantasies, obscure even to himself, about *true bourgeois property.* [63]

*It is this inability to see the development of the forms of property, that makes the anarchists believe that the direction of the development of the concept of property can be decided at will, that creating a 'commons' that serves the people and is against the bourgeois rule is possible within the framework of capitalist economy and that is the way to a 'property-less' social structure.*

*They fail to see that capitalism, while it leads to socialization of the production, it also intensifies the private appropriation and thus stands to benefit more from any commons. They also fail to see that existence of certain forms of commons is not antithetical to the existence of capitalism.*

### *4.4 The Peculiarity of Software against Other Commodities and as Commons*

The peculiarity of software as a commodity is not lost to most people. Software, being essentially knowledge, and produced in 'digital form' always was easily replicable without inviting much of practical cost. The cost of replicating software to a minimum (ignoring the costs of use of electricity, cost of the copying device like computer, training, etc.) is the amortized long-run cost of the storage devices over their life-time. According to current estimates, the cost of storing 1MB data on a USB thumb-drive of size 64 GB can be as low as 0.0002 paisa.[64] This, even in 1991 when the cost of storage was nearly 20,000 times more, comes out to be 5 paisa. Considering that it can be written-to multiple times, the cost goes down by many orders of magnitudes. Thus practically, the cost *can be considered* tending to 'zero'. Software as 'commons' thus stands differently from other commons based

[62] Proudhon, Pierre-Joseph. 1840. 'What is Property?'. *The Anarchist Library.* https://theanarchistlibrary.org/library/pierre-joseph-proudhon-what-is-property-an-inquiry-into-the-principle-of-right-and-of-governmen

[63] Marx, Karl, and Frederick Engels. 1865. 'On Proudhon', Letter to J B Schweizer. *Collected Works*, Vol. 20. Lawrence & Wishart, London, p. 28.

[64] A 64 GB pen-drive costs approximately Rs. 500 today. Considering the lifespan of 10 years, ignoring number of reads/writes, the cost per 1 MB per day comes out to be 500/ (64 * 1024 * 10 * 365) = 0.000002 Rs. = 0.0002 paisa

on material property, because software can be replicated without any significant cost.

Thus, as against other commodities, one can say that when one shares a commodity it divides, or diminishes, or displaces itself, but when one shares a song or music, shares knowledge or software, it multiplies, if it is freely shareable. It is an absurdity of capitalism that it commodifies things like these, too, but then what is any exploitative system like capitalism if not an absurdity!

However, for the same reason, FOSS also stands to benefit more to those who already have higher control over means of production and the markets. That is why this type of 'commons' can co-exist along with capitalist private property, without reducing its access to people, but at the same time benefitting more and more to the 'propertied classes' (in the words of the anarchists!) that is the bourgeoisie. The *akshay patra* of the Free Software ends up giving much more to those who use it not only as means of consumption, but also as means of production, and control production *in general.*

The rise of client-server programming, SaaS, and now AI are clear trends which over-score the importance of non-software means of production in producing more software. Further, the growth of SaaS, or AI led to the destruction of the *distribution* or *need to copy* characteristic of software, endangering the whole basis of the conception of Free software, which was supposed to be distributed freely. The fact that distribution requires control over the supply chains, and not only on the end product, and that the bourgeoisie also controls the supply chains (read: servers, server-farms, electricity distribution, networking equipment, etc.) for software is a realization that came to FSM quite late.

### *4.5 Misconceptions about Voluntarism*

Anarchists, unable to understand that society develops through the fundamental contradiction between development of productive forces and the relations of production, and capitalism is a stage in this process, also fail to see that a worker-capitalist relation is not based on choice, but a *historically given* relationship in a capitalist society. Anarchism fails to understand that though a worker is free to choose his/her employer, he/she is also 'free' from the ownership of the means of production and consumption and is thus compelled to work. The worker is not a slave, but the working *class* is structurally subordinated to the capitalist *class* as a whole, as wage-slave. Thus, voluntarism can remain a choice only for a limited set of skilled workers in capitalism, who can have access to means of consumption in other ways. A very large number of software professionals are forced to do wage-slavery for software bourgeoisie is therefore not essentially due to lack of their motivation to contribute to the *commons,* but just a material reality of the capitalist society. It is due to this fact, that the corporates also have the power to drive the decision-making in FOSS communities. In the opinion of Morten Rand-Hendriksen[65],

[65] Rand-Hendriksen, Morten. 2019. 'On the Corporate Takeover of the Cathedral and the Bazaar'. *Blog: A Repository of Morten Rand-Hendriksen's Thoughts.* https://mor10.com/on-the-corporate-takeover-of-the-cathedral-and-the-bazaar/

> Particularly, we need to talk about how "decisions are made by those who show up" should be amended to read "decisions are made by those who can afford to show up" and what that means for our industry.
> ...open source project like Android which is now completely dominated by Google and Samsung.
> ...Who can afford to make significant contributions to open source projects? By and large employees at large corporations. Who can afford to speak at Open Source conferences? The same people. Who sponsors Open Source projects and conferences? The same corporations.
> ...To put an edge on it, big corporations are buying out all the space of both the cathedral and the bazaar.

FS activists also fail to see that only in capitalism voluntarism is possible. In any pre-capitalist society, given that any toiling person was not free to choose whom he/she works for, the separation from means of production was also not the norm for those who toiled. Not only this, but the work was also forced upon the serf or slave, with varying degrees of coercion. The conception that one will *volunteer* in production, assumes that a choice exists to work without wages, and hence this idea can originate only in a capitalist society. The type of voluntarism that FSM envisages does not apply to a socialist society, as the products of labour of any worker would now belong to the working people and not to an individual, and hence voluntarism would take form of working *more* for the people as a whole, like the soviet *subbotnik*.

All those harbouring revolutionary ideas of social change through Free Software, fail to realize that the uprising 'from below' is not possible, without a revolutionary ideology and revolutionary organization and FSM has no revolutionary ideology of social change as it does not even identify itself with that larger social change and it stems from its ideology itself that it is anti-organization in the revolutionary sense.

### *4.6 Freedom is Not an Absolute Idea*

The absolutist idea about freedom has always plagued anarchists, including the Free Software anarchists. The Free Software movement sees only the freedom of the programmer and the user as its end goal. However, many who see it as a tool of complete social liberation are completely lost.

Engles says, in Anti-Dühring[66],

> Freedom of the will therefore means nothing but the capacity to make decisions with knowledge of the subject. Therefore the freer a man's judgment is in relation to a definite question, the greater is the necessity with which the content of this judgment will be determined; while the uncertainty, founded on ignorance, which seems to make an arbitrary choice among many different and

[66] Marx, Karl, and Frederick Engels. n.d. 'Anti-Dühring'. *Collected Works*, Vol. 25. Lawrence & Wishart, London, pp. 105–106.

> conflicting possible decisions, shows precisely by this that it is not free, that it is controlled by the very object it should itself control. Freedom therefore consists in the control over ourselves and over external nature, a control founded on knowledge of natural necessity; it is therefore necessarily a product of historical development.

Lenin quotes the summarization by Engels, in Anti-Dühring, on the relationship between freedom and necessity, "To him, freedom is the insight into necessity [die Einsicht in die Notwendigkeit]. Necessity is blind only in so far as it is not understood [begriffen]."[67]

The FS activists fail to see that the conception of freedom they cherish is in turn controlled by historical development, and that what they have is a *bourgeois* idea of freedom; that they and their idea of freedom are in turn controlled by "*the very object it should itself control*". The FSM finds it unnecessary to develop an *insight into necessity* that is understanding the laws of motion of the capitalist society, and history in general. As a result, the FSM fails to realize that there is a *necessity* for the programmer to earn wages, and the *necessity* for the employer to control means of production under the capitalist mode of production, and the FSM does not find it necessary to challenge this relation of production, thus limiting itself to a restricted notion of freedom. It does not see freedom in the liberation of the working class, and hence software working class, by *liberating* the means of production from the hands of the bourgeoisie and putting them in the hands of the people. Thus, its conception of freedom remains a bourgeois conception.

### *4.7 Fighting the Commodification of Just Software is Not Enough*

Marx begins *Capital*, Volume-1, by saying, "The wealth of those societies in which the capitalist mode of production prevails, presents itself as 'an immense accumulation of commodities', " its unit being a single commodity."[68]

Ernst Mandel, in his introduction to the *Capital*, Volume-1, says, "Capitalist production is generalized commodity production. Generalized commodity production fully unfolds trends and contradictions which are latent in every one of its basic ' cells', the commodities."[69]

Reducing capitalism to monopolies has been one of the biggest superstitions, always glorified and propagated by social-democrats and anarchists alike. The anarchists, and FSM, while opposing commodification of software, do not take any stand against capitalism or market economy in general which is the mechanism of implementing a generalized commodity producing economic system. Thus, it is content with the fact that the means of consumption that a software worker needs for his/her survival and the means of production that a software enterprise needs, remain commodities and at the mercy of market forces.

As a result, the FSM has always been in a very paradoxical position on its association with any other movement or institutions. FSM activists often take political stands on not participating in a software event if the organizers use any

[67] Lenin. 1908. 'Materialism and Emperio-Criticism'. *Collected Works*, Vol. 14. Progress Publishers, 1961, p. 187.

[68] Marx, Karl, and Frederick Engels. n.d. 'Capital'. *Collected Works*, Vol. 35. Lawrence & Wishart, London, p. 45.

[69] Marx, Karl. 1976. *Capital*, Volume 1. Penguin, p. 14.

non-free software to run their software systems. At the same time, they have no issues in corporates sponsoring such events, or non-free software being used for any of the other organizational aspects like travel, accommodation, food, etc. It is this ideological compromise inherent in the definition of FSM that also helps it become a close friend of the OSM.

This not only makes the movement essentially stand in support of capitalism, it also makes it fail to realize that the market economy will constantly create pressures to keep commodifying software in various ways, like it has done using technological advancements like client-server, SaaS, or AI.

### *4.8 Why Does the OSM 'Win'?*

The OSM has effectively been utilized by the market forces, as demonstrated above, to further its own goals of maximizing profits. The surrender by FSM before the OSM is a direct consequence of the ideological lacunae of the FSM, where, while it correctly stands against appropriation of software knowledge as private property, it does not stand against the *existence* of private property in general, and does not stand against an economic system that is a 'immense collection of commodities' and markets and is structurally based on private property, that is capitalism. That is why the FSM always stood in a friendly contradiction with the OSM, and not in an antagonistic one. It is this subjective opportunism, that has progressively led to an objective opportunism for many a people in the FSM.

Thus, corporate funding is not something that FSM stands against, but it stands only against the corporate monopoly ownership. It fails to realize that it is not that the corporate funding is given to the FSM *despite* its anti-monopoly stand, but essentially *due to* its pro-market stand. The monopolies are too well-aware that market economy will eventually led to creation of more and more monopolistic corporations and individual efforts will not be able to counter it.

## 5. A Critical Assessment of the FSM

Despite its ideological lacunae, the FSM has achieved many a feat.

A. The development of a very large body of software, through voluntary co-operation among informed software workers, using a variety of democratic forms demonstrates that workers can voluntarily organize production, and suggests multiple mechanisms for organization of production, with workers control, in a socialist society.

B. It has demonstrated that the products of co-operative work of a community of workers can be equally competent or better than the product of a capitalist enterprise.

C. It has created a digital commons and demonstrated that it can be sustained even against capitalist onslaught for privatizing and monopolizing everything possible.

D. It has created tools and technologies that help social activists, civil-democratic liberty activists, FS activists, and common people fight for preservation of their democratic, civil rights against the state and for privacy on the web.

E. Many anarchists, inspired by the FSM, have created small but resilient digital infrastructures (servers, websites, email-servers, cloud platforms, etc.) that help activists all over the world preserve their privacy against surveillance by state and non-state actors.

F. It has resisted software patents to a large extent, thus enabling free access to software knowledge for a considerable section of the people in the world.

G. It has lead to further development of productive forces in software and computer technologies by raising the efficiency of the workforce.

H. However, a large number of proponents of the FSM have, due to its ideological weakness rooted in anarchism, failed to recognize that

i. Capitalism tends to convert everything into a commodity, including the 'commons', and even the commons stands to benefit the capitalist class much more than it benefits the working masses under capitalism, and the existence of the commons itself is not antithetical to capitalism.

ii. The working class is a wage-slave of the capitalist class, as it must sell its labour-power continuously to make a living, and hence voluntarism in production cannot lead to change of social structures.

iii. The "uprising from the below" and spontaneity does not work, in the absence of a revolutionary ideology for revolutionary social change, and a revolutionary organization.

iv. The characteristic of software that it can be easily copied, and multiplies when copied, makes it a suitable candidate for digital commons to a certain extent, however, this candidacy is obstructed by the limits posed by capitalism. As long as the means of production and consumption are not socialized in general this potential will not be realized fully.

v. That property is not a historically generic or amorphous concept and its forms evolve through laws of historical materialism to reach a stage where no private property exists.

vi. Any fight against commodification of software is an incomplete fight unless one fights against commodification of all means of production in software and thus essentially leading to a fight against commodification of means of production in general.

vii. Freedom for software, its users, programmers and all the people is possible in truest sense only by realizing the *necessity,* that is to say, the laws of social change, or, Historical Materialism, which is the science of history and the revolutionary tool of the proletariat for *freeing* all means of production from the control of the bourgeoisie.

As it stands against commodification of software, for the creation of a digital commons, for the control of production by workers, and is not explicitly antithetical to the idea of social ownership of the means of production, the FSM should be seen as a mass movement standing on wrong ideological grounds, one with a friendly contradiction by the Marxists. For any true FS activist who also stands against capitalism, the path should lead to a stand against market economy in general, and hence ultimately to Marxism. The OSM, on the other hand, should be seen from the position of an antagonistic contradiction, as it does not stand, in principle, against commodification and markets and promotes commercialization and profiteering.

In conclusion, we can also say that the FSM is *objectively* a political movement, *but not political enough*. It started as a challenge to the monopoly capitalist control of software world but never challenged either software capitalism or capitalism in general, and ended up being circumscribed and co-opted by it. FSM has an idealistic, one-sided, narrow view of capitalism, while the OSM is a more 'realist' one, and hence wins within the boundaries of capitalist system and economy. However, this is not the end of the FSM. The FSM will retain its existence in the form a minority trend against OSM, as a petty-bourgeois ideology that can keep re-emerging within capitalist social structure, but limited by its competition with OSM, with both of these movements sustaining owing to the large scale digital commons already created and due to the benefits it brings to software bourgeoisie and, secondarily, also to working masses. Both movements will continue to exist also due to the fact that the software bourgeoisie, too, now understands the importance of Open Source model of development that also legally necessitates the existence of already created Free Software within the copyright regime.

For the FSM to grow its strength and existence, it is imperative to reinvigorate its fight against the OSM, and the spirit to defeat monopolistic corporate control of software; the resistance to commercialization of software needs to be pursued politically more explicitly and not implicitly in an 'economist' fashion by focussing on the production and distribution process. Moreover, without a general opposition to the capitalist system and capitalist mode of production, and with mere opposition to the symptoms of monopoly capitalism and big capital in general, the movement is bound to be circumscribed by the basic logic of capitalism, as we have already seen. It must depart from the incorrect political and philosophical positions that it has assumed, in order to be able to develop a really revolutionary and subversive character.

After this ebb, the true fillip to the FSM is possible now only with the rise of the larger working-class movement, that aims to destroy private property not only in software (impossible anyway!), but in general.