AN ECONOMIC THEORY OF FREE AND OPEN SOURCE SOFTWARE: A TOUR FROM LIGHTHOUSE TO CHINESE-STYLE SOCIALISM[*]

BENJAMIN HAK-FUNG CHIAO
New York University
Dec 15, 2003 (Version B2.0), Nov 26, 2002 (First version)

ABSTRACT

The theory is that free and open source software is private property under the guise of common property. Such software is distributed mostly under the GNU General Public License. The intents in *The GNU Manifesto* suggest striking similarities between this license and communism. The resulting economic properties, however, are similar to those of Chinese-style socialism: both resulted from an increased separation of legal and economic ownership. The phenomenal growth of China in the last twenty five years and of such software in the past few years could be attributed to such separation.

I. INTRODUCTION

> Many programmers are unhappy about the commercialisation of system software. It may enable them to make more money, but it requires them to feel in conflict with other programmers in general rather than feel as comrades…
>
> Richard Stallman, *The GNU Manifesto*[1] (1985)

> The Open Source Initiative does not have a position on whether ideas can be owned, whether patents are good or bad, or any of the related controversies. We think the economic self-interest arguments for open source are strong enough that nobody needs to go on any moral crusades about it.
>
> Open Source Initiative[2] (2002)

The opposing views in the above quotations are a reflection of the Chinese economy in the last twenty five years. The theory is that free and open source software (FOSS) is

[*]My sincere thanks go to Janusz Ordover and Andrew Schotter for their support and advice. I gratefully acknowledge the comments by participants of the International Conference on Open Source held on 26 July 2003 at Academic Sinica in Taipei, and of the doctoral student seminar held on 26 November 2002 at New York University. I would also like to thank Ben Hyde and Peter Meyer for their suggestions. All remaining errors are mine. Mailing address: Room 618, 726 Broadway, New York, NY 10003. Email address: hfc204@nyu.edu. Website: http://cess.nyu.edu/hfc.

[1] Richard Stallman, The GNU Manifesto, 1985, http://www.gnu.org/gnu/manifesto.html, visited November 17, 2002.

[2] Open Source Initiative, Frequently Asked Questions, 2002, http://www.opensource.org/advocacy/faq.php, visited November 17, 2002.

private property under the guise of common property[3]. The separation of the legal and economic ownership of FOSS, as in the present Chinese economy, is our key observation (see Section III). This leads to the conclusion that FOSS satisfies the definition of private property but not common property—I will support this claim by dissecting the economic structures of FOSS licenses.

The failure of transforming software from private into common property is consistent with the failures of all true communist states. The tremendous wealth created by China's transition from communism to Chinese-style socialism is also consistent with the phenomenal growth of FOSS (see Section III). Estimates show that if the Red Hat Linux, a flagship of FOSS, had been programmed using traditional proprietary means, it would have taken over one billion U.S. dollars and the equivalent of 8,000 person-years to develop the three million lines of code[4].

My view differs from most current analyses in that I found evidence to support that FOSS, in contrast with a lighthouse, is not *necessarily* a public good because some excludabilities are observed[5]. In practice, excludabilities take the forms of trade secrets[6],

---

[3] See Section II for the definition of FOSS in this paper. For another name in place of FOSS, see *infra* note 8. For a definition of private property, see Section III. For a general discussion on common property, see Steven N. S. Cheung, "Common Property" in The New Palgrave: A Dictionary of Economics, Eatwell, Milgate, and Newman eds. 1998. Note that private and common property are two extremes of a continuum. Therefore, a more precise formulation of the argument in the text should read: FOSS is closer to the private property end than it appears to be.

[4] D. Wheeler, More than a Gigabuck: Estimating GNU/Linux's Size, 2002, http://www.dwheeler.com/sloc/redhat71-v1/redhat71sloc.html, visited November 17, 2002.

[5] This is not to make a claim that FOSS is never non-excludable. The concept of public goods is originated by Lindahl and generalized by Samuelson. There are various definitions of public goods but all of them contain the non-excludability property. See E. Lindahl, *Die Gerechtigkeit der Besteuing*. Lund: Gleerup. [English translation: Just Taxation—A Positive Solution] (1919); P. A. Samuelson, Economics: An Introductory Analysis (1964). It has been widely accepted that it is efficient for governments to provide public goods for free. However, to argue whether an economic good is excludable, we can not consider the good in isolation from its environment. As an illustration, should the government provide apples for free? Perhaps or perhaps not. The answer may be affirmative in an imaginary island in which the only possible means to defend invaders is apple (note: national security is a classic example of public good). But it will be a joke to provide them for free in the city of New York. From this perspective, one can argue that some FOSS is (or at least, in principle, can be) a national security end product. In the FOSS literature, however,

induced network congestions, and the like (see Section IV A). Using this generally neglected income generating opportunity as a constraint, I put forth a new theory in this paper. The theory is based on a model presented in Figure 1, in which the observed excludabilities are consistent with self-interested behaviors.

INSERT FIGURE 1 FROM THE LAST PAGE TO HERE

The remaining sections are organized as follows. Section II clarifies major confusions about FOSS and lays the axiomatic foundation of the theory. Section III presents the theory and draws a parallelism between FOSS and the Chinese economy. Section IV reports empirical results. Section V concludes.

II. CONFUSIONS AND AXIOMS

A. Free as in Freedom

There are two camps that promote FOSS. The Open Source Initiative (OSI) certifies open source software; the Free Software Foundation (FSF) categorizes free software. Some software is compatible with both the objectives of FSF and OSI, for example, software that is distributed under the GNU General Public License (GPL). It is a widely

---

usually only the more straightforward perspective of excludability is considered—once the code is revealed, its use is non-excludable. From this other perspective, some scholars claim that FOSS is a public good. See J. P. Johnson, Open Source Software: Private Provision of a Public Good, 11 J. Econ. Manage. Strategy (2002), and some articles in R. Hahn, Government Policy toward Open Source Software (2002). Note that it is this other perspective that the author disagrees on. Their policy recommendations may be void because the premise of non-excludability is problematic. However, in general, A imp lies B is not equivalent to not-A implies not-B.

[6] It is the modest aim of this paper to argue that there are grounds to claim that trade secrets can still be kept in one's business though the source code of computer progra m is released because information such as complete documentation does not necessarily have to be released according to the practice and licenses of the FOSS community. In American Law Institute's Restatement of Torts (1939): "A trade secret may consist of any formula, pattern, device or compilation of information which is used in one's business, and which gives him an opportunity to obtain an advantage over competitors who do not know or use it. It may be a formula for a chemical compound, a process of manufacturing, treating or preserving material, a pattern for a machine or other device, or a list of customers. . . A substantial element of secrecy must exist, so that, except by the use of improper means, there would be difficulty in acquiring the information."

adopted software license written by Richard Stallman in 1989. I will refer to software that has been categorized as both free software and GPL-compatible software as "FOSS"[7].

Although the term "free" causes many people to think that the software does not have to be paid for, it is originally intended to imply no royalty and some form of liberty[8]. For instance, the GPL stipulates that licensees have legal permission to copy and distribute the software, either verbatim or with modifications, either gratis or for a fee. However, there is a specific meaning for a justified fee. In the first version of the GPL, Stallman writes that one "may charge a fee for the physical act of transferring a copy, and [one] may at [one's] option offer warranty protection in exchange for a fee." In a more current document, he continues to write that the software should be distributed "either gratis or [by] charging a fee for distribution"[9]. These remarks together with the following quotation from *The GNU Manifesto* convey an impression that the fee is to cover distribution costs such as expenses for diskettes, postage, and so on[10]:

> …I can give [GNU] away free to everyone who can use it…everyone will be able to obtain good system software free, just like air.

---

[7] For a list of free software or GPL-compatible software, see Richard Stallman, Various Licenses and Comments About Them, 2002, http://www.gnu.org/licenses/license-list.html, visited November 10, 2002. Some parts of Netscape 6 and 7 (Mozilla) that are distributed under the Mozilla Public License, and Apache that is distributed under the Apache License are not FOSS because they are not GPL-compatible. Version 2 of the GPL can be downloaded from http://www.gnu.org/copyleft/gpl.txt. An earlier version was written in 1989.

[8] Richard Stallman writes, "Because of the ambiguity of 'free,' people have long looked for alternatives…[The English Language] lacks a simple, unambiguous, word that means 'free,' as in freedom—'unfettered,' being the word that comes closest in meaning. Such alternatives as 'liberated,' 'freedom,' and 'open' have either the wrong meaning or some other disadvantage." See Richard Stallman, The GNU Project, 1998, http://www.gnu.org/gnu/the-gnu-project.html, visited September 1, 2000. On the other hand, the OSI refers to GPL-covered software as "open source software." The term "free software" proposed by the Free Software Foundation does not convey the wrong idea that proprietary open source software is free (in terms of liberty) for code modifications. However, it causes people to think that it is free of charge. A compromise suggested here is "royalty-free software". In general, the set of FOSS defined by OSI is different from those defined under the GPL.

[9] See Richard Stallman, The Free Software Definition, 2000, http://www.gnu.org/philosophy/free-sw.html, visited September 1, 2000.

[10] Stallman, *supra* note 1.

In the Open Source Definition v. 1.9, the OSI makes the above impression concrete by emphasizing "reasonable reproduction cost"[11]:

> The program must include source code, and must allow distribution in source code as well as compiled form. Where some form of a product is not distributed with source code, there must be a well-publicized means of obtaining the source code for no more than a *reasonable reproduction cost* [emphasis added]—preferably, downloading via the Internet without charge.

However, the provision of reasonable "distribution costs" or "reproduction costs" in FOSS licenses has never been tested in court. In practice, there is some uncertainty whether this is legally binding. For instance, one can defend his charging a high distribution price by claiming that he needs to recoup his opportunity cost of spending the resources in distribution. In fact, in the 1980s, Stallman charged $150 for each distribution of GNU Emacs—a word processor he coded and distributed under the GPL[12]. Thus, in principle, the costs of using FOSS can be prohibitively high. In 1993, Stallman clarified his stance by adding a footnote to his sentence quoted earlier:

> The wording here was careless…the intention was that nobody would have to pay for *permission* to use the GNU system…. This is another place I failed to distinguish carefully between the two different meanings of "free".

This clarification is an important step towards encouraging the growth of FOSS (see Section III, IV).

B. Self-Interested Behavior vs."Gift Economy"

It might be puzzling why software that costs billions to develop is distributed using royalty-free licenses and is sometimes developed by volunteers in addition to hired workers. Hired workers are assigned to work according to their superiors' orders;

---

[11] "The Open Source Definition is derived from the Debian Free Software Guidelines. Bruce Perens composed the original draft; it was refined using suggestions of the Debian GNU/Linux Distribution developers in an e-mail conference during most of June, 1997. They then voted to approve it as Debian's publicly stated policy. It was revised somewhat and Debian-specific references were removed at the origination of the Open Source Initiative in February 1998." See Open Source Initiative, The Open Source Definition, 2002, http://www.opensource.org/docs/definition.php, visited November 15, 2002.

[12] Stallman, The GNU Project, *supra* note 8.

volunteers are motivated not by altruism according to some economists. For instance, Lerner and Tirole put forth a theory of maximization of instantaneous and delayed benefits through reputation and the like[13]. Note that this is merely a postulate not necessarily a reflection of reality—a standard method in economics[14]. At least Linus Torvalds, the creator of the Linux kernel, claims to be motivated by fun. He has even written a book called "Just for Fun"[15]. Through out this paper the self-interested postulate is used. A direct consequence of this postulate is that there are no gifts without an expected increase in benefits. Put aside the question about the realism of this postulate, some arguments for the emergence of a "gift economy" in some well-cited articles are not logically sound[16].

Another view is that in practice, voluntary help does not violate the self-interested postulate because the costs involved are minimal. Raymond argues that programmers

[13] Lerner and Tirole, Some Simple Economics of Open Source, 52 J. Ind. Econ. (2002), at 213.

[14] For a discussion that is compatible with this approach, see Milton Friedman, The Methodology of Positive Economics (1953). The self-interested postulate is at least justified in a sense that this is historically useful. Although the existence of altruistic behaviour is not news, economists have long ignored it on purpose. In The Theory of Moral Sentiments, Adam Smith, who used only self- love of people as a basis for his economic theories, already observed some "anomalies": "How selfish soever man may be supposed, there are evidently some principles in his nature, which interest him in the fortunes of others, and render their happiness necessary to him, though he derives nothing from it except the pleasure of seeing it." In The GNU Manifesto, Stallman's finding corroborates Smith's view on some principles of human nature. He asserts that "[the] fundamental act of friendship among programmers is the sharing of programs." In 1776, however, Smith develops the self-interested postulate as a cornerstone of economics in his Wealth of Nations: "It is not from the benevolence of the butcher, the brewer, or the baker that we expect our dinner, but from their regard to their own interest. We address ourselves, not to their humanity but to their self-love, and never talk to them of our own necessities but of their advantages." See Adam Smith, The Theory of Moral Sentiments (1759), Stallman, *supra* note 1, and Adam Smith, The Wealth of Nations (1776).

[15] Linux Torvalds and David Diamond, Just for Fun—the Story of an Accidental Revolutionary (2001).

[16] Raymond writes, "Gift cultures are adaptations not to scarcity but to abundance…social status is determined not by what you control but by what you give away…it is quite clear that the society of open-source hackers is in fact a gift culture. Within it, there is no serious shortage of the 'survival necessities'—disk space, network bandwidth, computing power. Software is freely shared. This abundance creates a situation in which the only available measure of competitive success is reputation among ones' peer." Even if the "survival necessities" he mentioned are indeed abundant, it does not imply that gifts are the *only* means to success because other economic goods are still scarce according to a basic tenet of economics. The making of the gifts takes up other economic goods such as time and resources of the programmers. See Eric Raymond, Homesteading the Noosphere?, The Cathedral & the Bazaar—Musings on Linux and Open Source by an Accidental Revolutionary (1999), at 99.

who originally wanted to "scratch an itch" to satisfy their own needs, can submit their code without incurring too much extra cost[17]. In addition, due to the highly modular design of some FOSS, subroutines of the GNU/Linux can, loosely speaking, be divided into small and separate files. So controllers of the repository (the source code) do not incur huge costs to incorporate these contributions. Moreover, in reality, firms, redistributors, and open source contributors have obtained income (see Section IV A). And the evolution of FOSS licenses is generally consistent with the motivation to obtain additional income (see Section IV B). This further challenges the use of the alternative altruistic motivation.

III. FOSS AND CHINESE-STYLE SOCIALISM: SEPARATION OF LEGAL AND ECONOMIC OWNERSHIP

Sharing code and attempting to abolish private property are *not* without similarities. Stallman refers to the distribution of software through FOSS licenses as "copylefting"[18]. He writes, "[c]opyleft says that anyone who redistributes the software, with or without changes, must pass along the freedom to further copy and change it." In *The GNU Manifesto,* Stallman writes about his intents of inventing the GPL[19]:

> Many programmers are unhappy about the commercialization of system software. It may enable them to make more money, but it requires them to feel in conflict with other programmers in general rather than feel as comrades. *The fundamental act of friendship among programmers is the sharing of programs* [emphasis added]; marketing arrangements now typically used essentially forbid programmers to treat others as friends. The purchaser of software must choose between friendship and obeying the law. Naturally, many decide that friendship is more important. But those who believe in law often do not feel at ease with either choice. They become cynical and think that programming is just a way of making money.

[17] Eric Raymond, *Ibid*.
[18] Richard Stallman, What is Copyleft, 1996, http://www.gnu.org/copyleft/, visited November 19, 2002.
[19] Stallman, *supra* note 1.

Benkler concludes that "no one 'owns' [FOSS] in the traditional sense of being able to command how it is used or developed, or to control its disposition[20]." If one interprets the abolition of this kind of "ownership" to be equivalent to the abolition of private property, then one should conclude that Stallman's intents have striking similarities with those of communism. This is because, in 1848, Marx and Engels wrote in *The Manifesto of the Communist Party*[21]:

> …the theory of the Communists may be summed up in the single sentence: Abolition of private property.

In economics, however, there is a distinction between the legal and economic ownership. Alchian writes that economic owners are persons who are assigned private property rights. He defines private property rights as socially enforced rights to select rules of an economic good[22]. Later, Cheung refines it based on three economic (not legal) rights[23]:

> Any productive resource is a private property if, within well-defined limits, its owner has:
> (i) the right to exclude others so that he alone may decide on its use;
> (ii) the right to extract exclusive income from its use; and
> (iii) the right to transfer or sell the property or resource to any one as he sees fit.

Barzel distinguishes the property rights defined by Alchian and Cheung as economic property rights, and those defined traditionally in economic literature as what the state assigns to a person, as legal property rights[24]. Using this distinction, Benkler's use of the word "own" implies economic ownership. Similar to Benkler's view, Johnson explicitly mentions that property rights are absent in FOSS[25]. If Johnson uses the legal property rights definition, then it is inconsistent because licensors usually have copyrights on

[20] Yochai Benkler, Coase's Penguin, or Linux and the Nature of the Firm, 112 Yale L. J. Winter (2002).
[21] Karl Marx and Friedrich Engels, The Manifesto of the Communist Party (1848).
[22] Armen Alchian, Some Economics of Property Rights, Il Politico (1965).
[23] Steven N. S. Cheung, Will China Go 'Capitalist': An Economic Analysis of Property Rights and Institutional Change (1982)
[24] Barzel, Yoram, Economic Analysis of Property Rights (1997), at 3.
[25] Johnson, *supra* note 5, at 661.

FOSS. In property and copyright laws, copyright owners are the owners of resources that are copyrighted in the sense that the law protects their rights to the exclusive and alienable use of those resources, except under conditions such as those stipulated in the fair use doctrine. In addition, under the Berne Convention, whenever the works are published with the consent of the authors, and are reasonably available to the public, the countries that have signed this treaty will enforce the intellectual property rights of its creators. On the other hand, if Johnson uses the economic property rights definition, then his claim that no economic ownership exists is problematic. According to Alchian-Cheung's definition, you can have some, if not all, economic ownership on FOSS you copylefted by excluding access through trade secrets, and other means (see Section IV) [26]. In the FOSS community, Raymond asserts that the owner(s) of a software project are those who have the exclusive right, recognized by the community at large, to re-distribute modified versions of that software. Raymond's assertion is not inconsistent with Alchian-Cheung's general definition of property rights.

The distinction between the meanings of legal and economic ownership implies a separation of the legal and economic ownership of FOSS—legal ownership belongs to intellectual property owners such as licensors holding the copyrights, and economic ownership to individuals such as potential trade secret holders and some licensees. This is the key message of this paper. In general, the separation will never be complete. Nor do

[26] However, if one observes that trade secrets are deliberately released, should we still say that this "non-excludable" FOSS is private property? The answer should still be affirmative, because it is a choice, and by definition, must be preferred as a result of the constrained maximization of an economic decision maker. It is important to note that private property made non-excludable should still be considered private property. Note that this conclusion is based on the assumption that it does not incur extra cost to keep trade secrets in forms such as insufficient documentation. This assumption is justifiable in a sense that written documentation requires labor input. In other words, without this assumption, licensors might develop "non-excludable" FOSS just because finding ways to hide information in the code is costly. Relaxation of this assumption is outside the scope of this paper but deserves further study. Also, the development of exclusion devices/mechanisms that incurs cost is outside the scope of this paper.

any things exist that are not separated at all. What matters for economic analysis, however, is the marginal effect. That is the effect of an increased separation on economic behavior.

Theoretically, an increased separation leads to several effects. First, this is more Kaldor-Hicks efficient[27]. An allocation is said to be Kaldor-Hicks efficient if there is no other feasible allocation such that it is possible for gainers to gain more than losers lose in monetary terms. A measure that improves Kaldor-Hicks efficiency is one that minimizes resources in the public domain. Barzel writes that a commodity is in the public domain if the resources used to acquire it accrue to no one[28]. The OSI recommendation that FOSS be distributed using reasonable distribution charge might be, but is not necessarily, efficient only when FOSS is non-excludable. For FOSS that is excludable, the difference between the controlled price and the willingness to pay is the maximum rent that lies in the public domain. Rent is defined as the portion of income that can be derived from a resource in which a change in rent does not affect the allocation of the resource. If one puts his newly-developed FOSS on a home PC with limited bandwidth for downloading, but is not allowed to charge a fee for distribution, users are forced to suffer congestion when everyone rushes to download the software. The value of the time, which is a rent for the software, is dissipated. Yet there is no change in the supply of the software. On the other hand, if there is no such restriction, the software can be written on a CD and mailed to buyers. People who pay more will get it first, perhaps by asking for faster

[27] Another well-known efficiency concept, the pareto optimality, will not be pursued in details here. Pareto efficiency is defined as an allocation of resources such that there are no other feasible allocations that make someone better off without making others worse off. Some authors argue that pareto optimality needs reinterpretation or is satisfied tautologically. See a discussion in Steven N. S. Cheung, The Transaction Costs Paradigm—1998 Presidential Address, Western Economic Association, Econ. Inquiry (1998).
[28] See Barzel, *supra* note 24, at 5.

shipping. Stallman's clarification, discussed in Section II, which supports the charge of distribution, in effect, facilitates the separation. This is but one of the many possible means of reducing the dissipation of rent. The conditions of no royalty and no distribution charge should not be mixed up. I have argued against the latter, but not the former. This is because the author suspects that, in principle, trade secrecy can be used to generate the same amount of income as if royalty were allowed.

Second, a generally neglected aspect is that separation reduces the potential divergence of interests between the resource owner and the project manager. Lerner and Tirole mention a similar, but not identical point[29]. I would like to add two points to their analysis. The first point is that the project manager does not necessarily prefer to choose to take the in-house software private; releasing it as FOSS might be more profitable for him. The second point is that even though this might mean a reduction in a firm's income, the software might still be released as FOSS because of the private interest of the project manager.

Let us consider the case in which a project manager has no final say in the decision regarding what license the in-house software would be bundled with. By persuading the resource owner to distribute software under FOSS licenses, a manager increases his

[29] Lerner and Tirole, *supra* note 13, at 215. They provide a reason for why the project manager "turns valuable code to the community" but does not "[take the code] private." They argue that the reason the leader of a project would turn the software into FOSS is that the leader may "meet layers of resistance within the leader's corporation. To the extent that the innovation was made while working in-house, the programmer must secure a license from the employer; and her division, which does not want to lose a key programmer, may not be supportive of her demand." They seem to be saying that "it is *more profitable* for the manager to take the technology home. However, she is likely not being able to do so. Therefore she chooses a *less profitable* option, that is, to share it with the community." In my view, sharing with the community in the form of licensing under the GPL, for example, might well be a more profitable option for the manager. They further mention, in the footnote, that it is possible that the sharing of the code will not reduce the income of the firm: "Open source projects may be seen as imposing less of a competitive threat to the firm. As a result, the firm may be less inclined to enforce its property rights on innovations turned open source." The last point they make on this is that "the firm may be unaware that the open source project is progressing." But it is unclear to us whether they mean the project manager is also unaware of this.

economic ownership on the software. This is because the manager can legally appropriate income, for example, through redistributing FOSS (see Section Ⅳ). There are several means by which the manager can generate a positive income: (i) he can develop support and documentation services in advance; (ii) he can outfox other firms in terms of provision of better products by hiding product information; (iii) he may be more credible in conveying that he knows better because he managed the product in the first place. On the other hand, there are a significant number of managers' superiors who do not care what their subordinates do. The no-liability arrangement in all FOSS licenses further enhances this. In universities, funding agencies and the universities themselves are the sources of software development budgets. But these organizations might not interfere too much with what licenses managers use. Therefore, in effect, some managers have the final say on the choice of license. This provides an incentive for managers to over-develop software in a sense that the resulting production level exceeds the production level that would be chosen by the resource owners if they had the chance to do so. However, owners might choose to allow the possibility of deviations from the desired production because it might be too costly for owners to monitor the production.

Third, resource owners can also increase their incomes. Appropriation of the software by agents, including managers and programmers, through copylefting may not bother the resource owner. For example, incentives are more compatible. He could even reduce the manager's payment because of the extra benefits appropriated. Otherwise, he could enter a superseding contractual arrangement so that agents are unable to appropriate in-house software or its future derivatives, or provide support services.

In practice, the FOSS growth experience mirrors the growth of the Chinese economy in the last twenty five years. This link can be established because both the present Chinese economy and FOSS have resulted from a dramatic increase of legal and economic ownership separation.

Deng Xiaoping, a former leader of the Chinese Communist Party, did not take to heart the manifesto by his foreign "comrades" when he was reforming the Chinese economy starting 1978. He proposed the concept of "socialism with Chinese characteristics" (it is referred to as "Chinese-style socialism" here)[30].

Cheung later translated this into the language of economics. Using the terminologies of this paper, he essentially describes this as a separation of legal and economic ownership: "Chinese-style socialism" is a system in which legal ownership of some key productive resources belongs to the state, but economic ownership to the individuals. In short, it is a private property rights system (also known as the market-based system) with communist ideals on paper[31].

For instance, the Constitution of the People's Republic of China essentially reads that its constituents have no private legal ownership[32]:

> All power in the People's Republic of China belongs to the people.
>
> …The basis of the socialist economic system of the People's Republic of China is socialist public ownership of the means of production, namely, ownership by the whole people and collective ownership by the working people.

---

[30] Deng Xiaoping, address at the Third Plenary Session of the Central Committee of the 11th Congress of the Chinese Communist Party (1978).

[31] Steven N. S. Cheung, 'Zi Ben Zhu Yi' Yu 'She Hui Zhu Yi' De Da Tong (Similarities between 'Capitalism' and 'Socialism'). Mimeo, 1988. Reprinted in Er Shi Yi Shi Ji Kan Zhongguo De Jing Ji Ge Ming (China's Economic Revolution from the Perspective of the 21st Century), 2002. Observing the dramatic changes and growth in China, he still holds the same view on the nature of Chinese-style socialism after 15 years of this article. See his recent work, Steven N. S. Cheung, Economic Explanation, Book III (2002), at 98.

[32] P.R.C. CONST. amend. III, ch. 1, arts. 1 and 6. The present Constitution was adopted in 1982. It was later revised in 1988 and 1993. The quotation hereafter is from the 1993 amendment.

Take land as an example. The Constitution reads[33]:

> Land in cities is owned by the state. Land in the rural and suburban areas is owned by collectives except for those portions which belong to the state in accordance with the law; house sites and privately farmed plots of cropland and hilly land are also owned by collectives. The state may, in the public interest, requisition land for its use in accordance with the law. No organization or individual may appropriate, buy, sell or unlawfully transfer land in other ways.

However, land satisfies the Alchian-Cheung definition of private property though it is owned by the state because[34]:

> The right to the use of the land may be transferred in accordance with the law.
>
> …The state protects the right of citizens to own lawfully earned income, savings, houses and other lawful property.

Historically, private legal ownership of a productive resource is not a requisite for a private property rights system. Cheung was quick to realize that the three economic rights would be greatly strengthened in China shortly after the reform in 1978. Then, in 1981 (a year before his book was published), he made the celebrated prediction that China would become a capitalist country, despite facing harsh criticism from many prominent economists[35].

The same intent underlies communism and FOSS but their fates differ. Before the present Chinese economic miracle, it had suffered decades of poor economic growth and political prosecution such as the Cultural Revolution since the inception of the communist rule in 1949[36]. If FOSS has such similarities with the Chinese institution, why didn't the FOSS community go through the same ordeal before enjoying its present

[33] P.R.C. CONST. amend. III, ch. 1, art. 10.
[34] P.R.C. CONST. amend. III, ch. 1, arts. 10 and 13.
[35] Cheung, *supra* note 23.
[36] For communism, Cheung argues that not only are resources not directed to the most productive means, other forms of right systems such as ranking, corruption, and backdoor policies are formed to reduce rent dissipation. The corollary is that human right is superseded by the ranking system. *Ibid*. To what extent these activities are present in the FOSS community is an interesting question, especially because they contradict the concept of freedom, the revised ideology of Stallman after he gave up interpreting "free" as the one in "free air."

status? Is this difference a refutation of the central thesis of this paper? My answer is that the lack of armed enforcement renders the earlier regime, implicit in Stallman's utopia, non-sustainable. As a result, legal and economic ownership are separated dramatically. Therefore, FOSS has bypassed the ordeal phase at the outset.

Let me explain why enforcement is important to the operations of common properties. During China's communist days, the state monopolized the procurement and marketing of agricultural products in a low-price and compulsory manner, which turned out to have suppressed the peasant's incentive to produce. To enforce the policy, people's communes were established in 1958 to implement the agricultural collectivization movement. Commune leaders were cadres appointed by the state to ensure that people did not keep their own tools and that all commune members ate together so they would not be able to keep their food for themselves. Violators were subjected to severe reprimands and punishments[37]. Turning back to the FOSS case, the initial intent of charging a nominal distribution fee is a form of price control, but it is not strictly enforced. In such a diverse Internet environment, Stallman cannot ensure that licensors truly embrace *The GNU Manifesto* as his comrades. Although I do not exclude the possibility that the earlier regime ever worked well with Stallman's friends or a small group of people, the evidence in the next section suggests that Stallman's utopia broke down once the FOSS community started to grow.

IV. EMPIRICAL EVIDENCE

Empirical data is provided in Section IV A to support the hypothesis that some FOSS is essentially private property. It is important to note that the existence of rights, which

[37] For a detailed account of the growth experience of the Chinese economy in the last decades, see Justin Lin, The China Miracle (1996).

define private property, is a matter of degree. Thus there is a continuum between private and common property. Since quite a few scholars claim that FOSS is not private property, this section is meant to support that FOSS is closer to the end of private property than it alleged to be (see Section III). In Section IV B, the evolution of some major FOSS licenses will be analysed, which supports another hypothesis: using a Darwinian argument, those institutions that do not provide FOSS owners with returns higher than those in alternative institutions will be weeded out. Institutions are defined as the rules that govern behavior. In the case of FOSS, licenses are the main determinants of behavior. In other words, we will observe that one will eventually adjust the license terms *as if* he has maximized his economic benefits from FOSS. The licenses discussed below can be accessed from the GNU website[38].

A. Some FOSS as Private Property

1. Exclusion Right (or Usage Right)

Red Hat bundles GNU/Linux with other FOSS to produce another FOSS—Red Hat Linux, which users can download free of charge. Red Hat also provides links to some Red Hat servers, as well as universities, non-profit organizations, and companies storing Red Hat Linux. Users can decide for themselves which servers they would like to download the software from. The prima facie interpretation is that one can obtain the software with ease, and thus gives developers no incentive to produce it. For instance, the inability to profit is the source of "inefficiency" in Johnson's argument[39]:

> Any developed [FOSS] can be freely provided over the Internet to the other user-developers and will be so provided if developed.
>
> …the GPL [guarantees] any purchaser strong rights to redistribution, and therefore might vastly restrict a developing agent's ability to profit.

[38] Stallman, Various Licenses and Comments about Them, *supra* note 7.
[39] Johnson, *supra* note 4, at 641, 661, and footnote 7.

> …free riding implies that some valuable projects will not be produced…

He argues that because of non-excludability, people do not develop and distribute enough FOSS because of the free-riding problem and thus it is inefficient. The alleged problem of the non-excludability is problematic. Over the last four years, the author measured the time needed to download various versions of Red Hat Linux, from 5.2 to 8.0. On average, it took more than 5 hours. In September 2002, the author spent about 10 hours to download this software through the Internet. This finding can be easily verified by anyone who has access to the Internet.

There are ways in which Red Hat can increase the inconvenience of the free download: (i) it can easily limit the speed of download, for example, by setting the number of connections allowed on an Apache web server or ftp server, or limiting the bandwidth; (ii) it can lengthen the time needed to publish links of organizations volunteering to store Red Hat Linux for downloading; (iii) it can increase the uncertainty over the time needed for downloading and the quality of the downloaded files. Unlike Cnet.com, which links to download sites for other software, Red Hat does not rate the connection reliability of published links. Unpublished sites face significant difficulties and incur considerable costs to claim to hold an original, virus-free version of Red Hat Linux.

Are these measures effective? One might argue that with the increase in bandwidth, the download time constraint may not be binding. However, the size of the software also increases over time. Or at least, Red Hat has an incentive to increase the size of the software. In fact, Wheeler finds that, "Red Hat Linux 7.1 represents over a 60% increase

in size…over Red Hat Linux 6.2 (which was released about one year earlier)"[40]. The proprietary Microsoft Windows 2000 Professional comes with one CD but the comparable Red Hat Linux 8.0 Professional Edition comes with five.

There are many other ways in which one can exclude easy access to the code, which will not be listed here.

2. Income Extraction Right

I attempt to present some evidence to refute the view that it is not possible for FOSS licensors or distributors to extract benefits from it. In fact, agents are able to obtain income through redistribution, support or complimentary services, and authority. First, it is possible for FOSS to be redistributed at a positive market price. Through excludability measures discussed in the preceding session, Red Hat can induce users to buy the Red Hat Linux CD packages from computer stores. These packages are generally available worldwide. I will report several list prices from Amazon.com as of March 2003. The Red Hat 8.0 Personal (Professional) Edition was $39.95 ($149.95). Other flavors of Linux provided by different companies: the SuSE Linux 8.0 Personal (Professional) Edition was $39.99 ($79.99), and the Mandrake Linux Standard (PowerPack) Edition was $32.99 ($69.99). Note that the proprietary Microsoft Windows XP Home (Professional) was listed at $199 ($299). One might argue that one buys the package not because of the time earned but due to value-added services such as documentation and support. However, documentation from famous publishers like O'Reilly is easily available and cheap. This strongly suggests that users buy the CD to get rid of the burden of the download process.

[40] Wheeler, *supra* note 4. Other forms of data are also widely available on the Internet. In general there is a clear consensus that the size of the Linux kernel, the heart of Red Hat, has been an increasing function of time.

Second, one can obtain income by providing support and complimentary services. This can be done, for example, through trade secrets such as information left out from documentation. For some FOSS, good documentation from parties other than the inventors is not widely available. Software without good documentation is hard to work on or use due to the idiosyncratic nature of algorithms and the heterogeneous nature of users' knowledge of the products. Insufficient documentation drives some authors to think that Linux is "premature". For example, the following that I have translated shows that Xue, a popular Chinese economics column writer, makes a problematic argument[41]:

> Source code is the most important asset in the software business…Software firms have the incentive to continually invest in development only if the source code belongs to them…. To date [Linux] is still in its elementary stage, even though its inception was in 1991…The popular "Linux Forums" on the Internet are merely "Installation Forums."

His view is that the nature of Linux is "premature" because the software companies, the innovators, do not have some form of ownership of the code. In Section III, I already discussed that economic ownership does not necessarily attenuate. Let alone that copyright holders have legal ownership of the code (if this is what Xue is referring to).

Third, Linus is the dictator (perhaps benevolent) of the Linux kernel—the heart of the GNU/Linux system. Under him, there are "lieutenants" deciding the operations of each section of the kernel. When there are conflicts between lieutenants, they are resolved by Linus[42]. The implication is that some leaders of hackers are able to extract implicit income based on their authority.

---

[41] Steven Xue, Zi You Ruan Jian Guo Yan Yun Yan (The Transient Phenomenon of Free Software), Securities Times, October 29, 1999.

[42] In Apache, a rotating board of leaders makes crucial decisions. Though Richard Stallman has categorized the Apache License as a GPL-incompatible free software license, it is conceivable that this kind of decision making process can take place in FOSS.

The three points mentioned above are not necessarily mutually exclusive. One could, for example, follow a two-part tariff scheme, allowing low-cost downloads and generating income based on support and complimentary services.

3. Transfer Right

The ability to sell the commercial packages is one of the transfer rights. Commercial FOSS packages are widely available for sale throughout the world.

B. Evolution of FOSS Licenses

The history of changes in some major FOSS licenses is studied here to shed light on the reinforcement of the three rights in the preceding section.

To reinforce exclusion right, measures that require the surrender of trade secrets protection will be weeded out. An example is the scripting language PERL. It once used the Artistic License version 1 but later adopted a disjunction of the GPL and Artistic License version 2 for PERL 6. Though versions 1 and 2 of the Artistic License require that a "clear documentation" of code modifications be provided, PERL 6 programmers are instead allowed to distribute modifications using the GPL, which does not require the provision of documentation.

For income extraction right, Stallman's clarification of the fee discussed in Section II, in effect, provides an endorsement for people to generate income from FOSS. Artistic License follows suit—in version 2, the term "reasonable" attached to "distribution cost" is removed.

For transfer right, provisions that require contributors to give up copyrights will be removed because otherwise contributors are foreclosed the option of selling their copyrights. In the Artistic License version 1, programmers are required either to: (i) give

up their copyright on contributed code. Licensors are then able to decide what to incorporate so as to maximize "artistic control"; (ii) follow some redistribution restrictions such as redistribution only within the same corporation; (iii) follow some technical specifications to avoid confusion among users regarding the original and modified software. Artistic License version 2, however, is considerably less restrictive. In essence, it allows one to distribute the modified software to any one as long as one provides the licensees the key rights that the licensors have, which are encompassed in the idea of copylefting.

VI. CONCLUSION

In their economic analysis of property rights in computer software, Baumol *et al.* consider the optimal intellectual property policy for the software industry—copyright and trade secrecy are analysed as if they are disjoint methods to protect an intellectual property[43]. This is consistent with the view that open sourcing would attenuate economic ownership. This paper, however, suggests that we should not consider copyright and trade secrecy to be mutually exclusive methods. Given a choice of license, licensors are able to choose to disclose how much code to be released as copyrighted materials (and thus choosing the method of copyright over trade secrecy to protect this part of the intellectual property); the remaining information may be left as a trade secret. It is important to note that source code is but one of the many trade secrets available. Measures and evidence abound, in which agents can take advantage of the institutions of FOSS licenses to establish economic ownership through trade secrets on copylefted

[43] For a discussion on the relative merits of copyright, trade secrecy, and direct government policy in encouraging investment in software development, see William Baumol *et al.*, Economics of Property Rights as Applied to Computer Software and Databases, prepared for the National Commission on New Technological Uses of Copyrighted Works, June 1977.

software. The evolution of some FOSS licenses is shown to be generally consistent with the motive of increasing economic benefits.

There was seldom congestion in the sea when Mill invented the famous lighthouse in economics. To use a metaphor, technology now allows us to adjust the size of the harbor with ease. The GPL originated at the time when the mode of software distribution was through magnetic tapes because network transfers were not feasible. Though the advent of the Internet allows for these transfers, distributors are also equipped with the ability to adjust the excludability of a network. Hence, FOSS is not necessarily a public good. If transaction costs such as bargaining, information, and monitoring costs, are not prohibitive, it is not Kaldor-Hicks efficient to uniformly impose that all FOSS be distributed only under reasonable distribution cost. Rent dissipation occurs when the use of FOSS excludes others' use, but distributors are not allowed to charge more.

This paper indicates that FOSS can be compared to Chinese-style socialism. The recent growth experience of China, especially regarding rent dissipation, could be further studied to shed light on FOSS, or vice versa.

BIBLIOGRAPHY

Alchian, Armen. "Some Economics of Property Rights." *Il Politico* 30 (4) (1965).

Barzel, Yoram. *Economic Analysis of Property Rights.* Cambridge: Cambridge University Press, 1997.

Baumol, William J.; Braunstein, Yale M.; Fischer D.M.; and Ordover, Janusz A. "Economics of Property Rights as Applied to Computer Software and Databases." Prepared for the National Commission on New Technological Uses of Copyrighted

Works, June 1977. Reprinted in part in *Technology and Copyright*, edited by R. H. Dreyfuss, p. 140. Lemond Publications, 1978.

Benkler, Yochai. "Coase's Penguin, or Linux and the Nature of the Firm." *The Yale Law Journal* 112 (2002).

Cheung, Steven N. S. *Will China Go 'Capitalist': An Economic Analysis of Property Rights and Institutional Change*. London: The Institute of Economic Affairs. Harper Paper, 1982.

Cheung, Steven N. S. "'Zi Ben Zhu Yi' Yu 'She Hui Zhu Yi' De Da Tong (Similarities between 'Capitalism' and 'Socialism')." Mimeo, 1988. Reprinted in *Er Shi Yi Shi Ji Kan Zhongguo De Jing Ji Ge Ming* (China's Economic Revolution from the Perspective of the 21st Century). Hong Kong: Arcadia Press, 2002.

Cheung, Steven N. S. "Common Property." In *The New Palgrave: A Dictionary of Economics*, edited by J. Eatwell, M. Milgate, and P. Newman. London: Macmillian Press, 1998.

Cheung, Steven N. S. "The Transaction Costs Paradigm: 1998 Presidential Address, Western Economic Association." *Economic Inquiry* (1998): 514–21.

Deng, Xiaoping. Address at the Third Plenary Session of the Central Committee of the 11th Congress of the Chinese Communist Party (December 13, 1978).

Friedman, Milton. "The Methodology of Positive Economics." In *Essays in Positive Economics*, pp. 3–43. London: The Chicago University Press, 1953.

Hahn, Robert. *Government Policy toward Open Source Software*. Washington, D.C.: AEI-Brookings Joint Center for Regulatory Studies, 2002.

Johnson, Justin P. "Open Source Software: Private Provision of a Public Good." *Journal of Economics & Management Strategy* 11 (2002): 637–62.

Lerner, Josh, and Tirole, Jean. "Some Simple Economics of Open Source." *Journal of Industrial Economics* 52 (2002): 197–234.

Lin, Justin. *The China Miracle: Development Strategy and Economic Reform*. Hong Kong: The Chinese University Press, 1996.

Lindahl, E. *Die Gerechtigkeit der Besteuing*. Lund: Gleerup, 1919. [English translation: Just Taxation—A Positive Solution. In *Classics in the Theory of Public Finance*, edited by R. A. Musgrave and A. T. Peacock. London: Macmillan, 1958.]

Marx, Karl, and Engels, Friedrich. *Manifest der Kommunistischen Partei*, 1848. [English translation: "The Manifesto of the Communist Party." In *Marx: The Revolutions of 1848.* Harmondsworth: Penguin Books, 1973.]

Open Source Initiative. *Frequently Asked Questions*, 2002, http://www.opensource.org/advocacy/faq.php, visited November 17, 2002.

Open Source Initiative. *OSI Certification Mark and Program*, 2002, http://www.opensource.org/docs/certification_mark.php#approval, visited November, 15, 2002.

Raymond, Eric. "Homesteading the Noosphere?" In *The Cathedral & The Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*, pp. 79-135. Cambridge, Mass.: O'Reilly, 1999.

Samuelson, P. A. *Economics: An Introductory Analysis*, 6th ed. New York: McGraw-Hill, 1964.

Smith, Adam. *The Theory of Moral Sentiments.* In *Liberty Classics*, edited by D. D. Raphael and A. L. Mactie. Indianapolis: Liberty Press, [1759] 1976.

Smith, Adam. *The Wealth of Nations*. In *Harvard Classics*, edited by C. J. Bullock. Vol. 10. New York: P. F. Collier, [1776] 1909.

Stallman, Richard. *The GNU Manifesto,* 1985, http://www.gnu.org/gnu/manifesto.html, visited November 17, 2002.

Stallman, Richard. *The GNU Project*, 1998, http://www.gnu.org/gnu/the-gnu-project.html, visited September 1, 2000.

Stallman, Richard. *Various Licenses and Comments about Them*, 2002, http://www.gnu.org/licenses/license-list.html, visited November 10, 2002.

Stallman, Richard. *What is Copyleft?*, 1996, http://www.gnu.org/copyleft/, visited November 19, 2002.

Stallman, Richard. *The Free Software Definition*, 2000, http://www.gnu.org/philosophy/free-sw.html, visited September 1, 2000.

Torvalds, Linux, and Diamond, David. *Just for Fun—the Story of an Accidental Revolutionary*. New York: HarperCollins Publishers, 2001.

Wheeler, D. *More than a Gigabuck: Estimating GNU/Linux's Size*, 2002, http://www.dwheeler.com/sloc/redhat71-v1/redhat71sloc.html, visited November 17, 2002.

Xue, Steven. *Zi You Ruan Jian Guo Yan Yun Yan* (The Transient Phenomenon of Free Software). *Securities Times*, October 29, 1999.

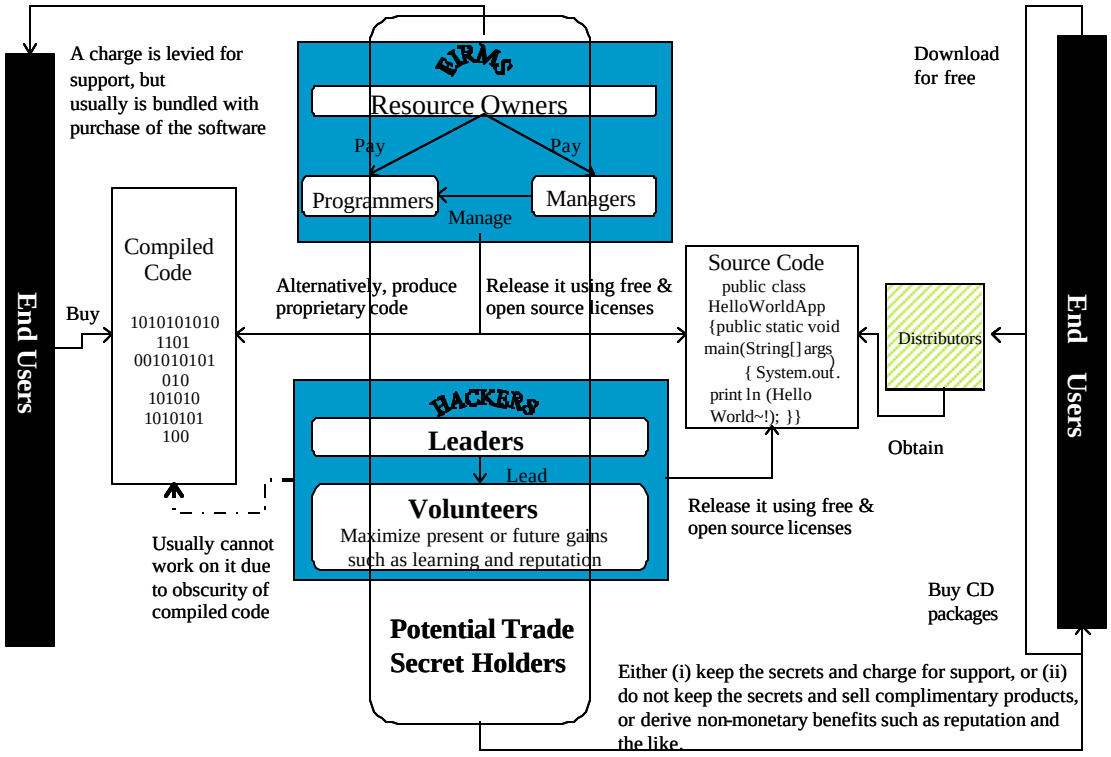
End Users
A charge is levied for support, but usually is bundled with purchase of the software
Buy
Compiled Code
1010101010 1101 001010101 010 101010 1010101 100
Usually cannot work on it due to obscurity of compiled code
FIRMS
Resource Owners
Pay
Pay
Programmers
Managers
Manage
Alternatively, produce proprietary code
Release it using free & open source licenses
HACKERS
Leaders
Lead
Volunteers
Maximize present or future gains such as learning and reputation
Potential Trade Secret Holders
Source Code
public class HelloWorldApp {public static void main(String[] args) { System.out.println (Hello World~!); }}
Release it using free & open source licenses
Distributors
Obtain
Download for free
End Users
Buy CD packages
Either (i) keep the secrets and charge for support, or (ii) do not keep the secrets and sell complimentary products, or derive non-monetary benefits such as reputation and the like.

FIGURE 1