Copyleft—the economics of Linux and other open source software

Mikko Mustonen*

University of Helsinki, Department of Economics, FPPE, PO Box 54, FIN-00014 Helsinki, Finland

Received 27 June 2002; accepted 1 August 2002

Abstract

An astonishing phenomenon is the rise of a large population of programmers who, seemingly against economic logic, develop and distribute software that is copylefted: freely available but licensed. Our paper suggests that the occupational choices of programmers based on reputation incentives determine the qualities of programs. A monopolist has to take into account the free software in the consumer market. When software implementation costs are low, the monopolist accepts the copylefted program in the market. We explain the simultaneous existence of commercial and copylefted programs, like Windows and Linux, and also why commercial alternatives to copyleft programs may not exist.

© 2002 Elsevier Science B.V. All rights reserved.

JEL Classification: D23; L11; L15

Keywords: Property rights; Copyleft; Linux; Open source software

1. Introduction

Copyleft is a novel licensing scheme. It facilitates open and decentralized software development. Its key feature is that once a program is licensed by the inventor, the subsequent programs based on the original must also be licensed similarly. As investments in computer software are becoming quite large representing, for example, 11 percent of total national investments in Sweden in 1998 (Jagren and Morell, 2000), the incentives for software development work and the

*Tel.: +358-400-200-480.
E-mail address: mikko.mustonen@helsinki.fi (M. Mustonen).

0167-6245/02/$ – see front matter © 2002 Elsevier Science B.V. All rights reserved.
doi:10.1016/S0167-6245(02)00090-2
associated efficiency aspects are an increasingly important issue. In this paper we develop a positive economic model of software development in the presence of copyleft and analyze how it affects commercial software markets. For programmers, copyleft creates incentives alternative to employment allowing them to signal their abilities and receive complementary income.

We raise the question how the existence of a copyleft program affects the behavior of a monopoly firm with a copyright program who invests in the quality of its program at the development stage. It turns out that the firm can influence the occupational choices of programmers through the wage policy. The programmers choose whether to be employed by the copyright firm or to join the copyleft community developing a copyleft program and receive complementary income based on the acquired reputation. These choices in turn determine the qualities of the firm's copyright program and the copyleft program at the development stage. At the output stage, the firm supplies a program protected by copyright. It has a market of its own and there are no substitutes except the rival program eventually developed by the copyleft community. Consumers value quality and determine whether to buy the copyright program or acquire the copyleft program or not to buy any. This creates a trade-off, which the firm can try to exploit: it can pay a higher wage at the development stage increasing costs but also increasing revenue by improving the quality of the copyright program while reducing the quality of the competing copyleft program. We analyze the monopolist's profit-maximizing behavior when consumers face non-negative implementation costs, i.e. costs of installing and learning the program. We characterize a monopolist entering the industry. We also analyze the welfare consequences of policy actions. The paper also has implications on the theory of vertical product differentiation. A novel industry structure arises: the decision of a single firm, the monopolist, determines the qualities of all (two) products in the market. To focus on the effects copyleft licensing has on labour and product markets, the model abstracts from network effects in consumer markets.

The main results are as follows. Program implementation cost is critical and determines whether the software firm has to take into account the existence of the copyleft program in the market. If the cost is low, there are consumers preferring the copyleft program, providing a constraint on profit maximization. There is an intermediate interval for implementation cost, at which the optimal price for the firm just deters the marginal consumer from acquiring the copyleft program. When the implementation cost is high, the firm is free to charge the optimal monopoly price. The larger the consumer market compared to the population of programmers

---

1 The model is applicable in many markets but the reader may feel more familiar with it by imagining William Gates as the monopolist and his program as Windows. The copyleft community could then be the programmers working on the project initiated by Linus Torvalds and Linux the copyleft program.

2 We plan to address this issue in the future.
is, the more the firm employs programmers and the smaller is the copyleft community. The firm employs many programmers if their complementary income from copyleft work is low or if the consumers’ valuation of program quality is high. We show that there is a threshold market size below which the monopolist will not develop a copyright program. Casual empiricism suggests that the results coincide with the structures in software markets. The implementation costs of programs have no doubt decreased, resulting in greater use of free copyleft programs like Linux. On the other hand, there are applications, for example in networking, where consumers exclusively resort to copyleft programs.

Our analysis on copyleft licensing has policy implications. First, the incentives for copyleft programming are independent of the consumer market. Copyleft programmers do not pay attention to whether copyleft programs are used outside the copyleft community. It is possible that a copyleft program exists and consumers are not aware of it. Informing consumers of such a program is likely to increase welfare with presumably low costs. Another policy implication concerns the enforcement of copyright. Paradoxically, copyright is vital to the incentives of copyleft programmers as the copyleft license is based on copyright. Securing copyright can be costly and the copyleft community may not have the resources to defend its copyright against firms. If society does not support a high level of copyright protection the copyleft communities are likely to restrict the distribution of programs to consumers because the risk of copyright violations increases. This in turn decreases welfare.

2. What is copyleft?

To copyleft a program, the programmer, besides copyrighting the program to himself, also signs a General Public License (GPL) (GNU Project, 2000b) granting everyone the right to use, modify and distribute the program on the condition that the licensee also grants similar rights over the modifications he has made. Under this arrangement, everyone has to have free access to the program but it is protected from becoming someone’s private intellectual property. The strong implication is that this allows for decentralized program development. The enhancements and modifications accumulate to the basic program even if the programmers have no other affiliation with the project. There is another significant feature of copyleft. It is a device for linking the programmer and his contribution

---

3 For example DNS and Sendmail have no commercial rivals, they are ‘category killers’ (Johnson, 2001; Opensource.org, 2002).

permanently together while the contribution is publicly observable. This creates an environment where talented programmers have an incentive to signal their abilities via the copyleft community.

Historically, copyleft licensing was created for ideological purposes by Richard Stallman and the Free Software Foundation (GNU Project, 2000b). However, the functions and quality of some programs like Linux and Apache have reached and in some respects surpassed those of traditional copyright protected commercial programs. The population of programmers participating in the development work is today so large that ideological motivations are inadequate to explain the phenomenon. Copyleft programs have gained significant market shares in a short time. For example, in the year 2000, the web server program Apache was used in 63 percent of web sites, which translates to almost 100 million sites (Netcraft, 2001). In the same year, the market share of Linux operating system in server use was 27 percent of 6.1 million shipments. The market share of Microsoft was 41 percent, but the increase from the previous year’s shipments was largest for Linux, 25 percent, as Microsoft’s shipments grew 20 percent (Johnston, 2001). In desktop use, the market share of Linux was 2 percent in 2000, but it has the highest growth rate. The co-existence of commercial copyright and non-commercial copyleft programs in the market is a reality. However, the low cost of copyleft programs means that the market shares for them in public market research reports are low even if the shares of users may be high. Deckmyn (2000c) reports that even though the number of users of Linux increases at twice the speed of Windows, the turnover from the Linux business is projected to be only a few percent of that of Windows in 5 years.

Even though signing the license agreement means that the creator, the programmer, cannot receive any rents from the sale of his creation, business activity based on copyleft programs is not absent. Varner (2000) categorizes the copyleft business actors into four groups: service sellers, loss leaders, widget frosting and accessorizing. A service seller provides installation and operation services for copyleft programs. Perhaps the best-known company that packages and supports Linux at the moment is Red Hat. Loss leaders distribute a copyleft program to create demand for some other copyright program, Netscape web browsers being an example. Widget frosting refers to hardware suppliers that may enhance their product with some copyleft program. Accessorizing is essentially selling complementary, but remote services or products. Recently, large computer companies (IBM, HP, SUN) have announced their support for some existing copyleft programs like Linux. SUN Microsystems has also developed programs (Staroffice) that are substitutes for existing office applications and copylefted them (see Deckmyn, 2000b). Lerner and Tirole (2000) and Subramanian (2000) discuss the intriguing phenomenon of firms engaging in copyleft work.

The economics of information tell us that there is a fundamental trade-off between information as a public good and the incentives to create new information (Arrow, 1962). Information, like a computer program, once it is created, is
practically costless to reproduce. From society’s point of view, it should be distributed freely. However, as incentives are then destroyed, there will be no creation of new information. Society’s solution to the missing incentives has been to secure various intellectual property rights to originators. In the case of a new, novel and non-obvious invention, a patent gives the inventor a temporary monopoly over the invention. Copyright protects the rights of the originator of a unique expression, for example a work of art, for a fixed period.\footnote{For a survey of the development of intellectual property rights, see David (1993).}

From the perspective of intellectual property rights, a computer program is a problematic object (Samuelson, 1993; Dam, 1995). It can be a unique expression as is a poem. Copyright would seem an appropriate method of protecting the rights of the originator. But a program, once it is running, also creates functions, like a machine. New and novel functions or uses for a function may seem to fulfill the requirements for a patent. Programming in a sense is creating and modifying algorithms. However, algorithms and ‘mathematics’ are not eligible for patenting. As a result of the first lawsuits over the property rights for software, copyright has generally become the method of securing property rights. Presumably the original motivation of the Free Software Foundation to introduce the copyleft licensing scheme was based on this development. Copyright entails a uniqueness requirement and the Free Software Foundation considered that the requirement restricted innovation in software. Recently, however, in the US patents have also been granted to programs and the EU considers a common policy on software patenting.

The rapid development of some originally copylefted programs (Linux, Apache, Mozilla, Sendmail, PERL) and the large number of participating programmers suggest that there must be powerful incentives to create and further develop copyleft programs. The licensing scheme rules out direct economic appropriation of rents based on property rights. One can see simultaneous software development, where the suppliers’ incentives are secured by property rights and on the other hand copyleft software development where programmers give away their rights at the outset. Dasgupta and David (1987, 1994) present a framework of ‘Science’ and ‘Technology’, suggesting that new knowledge is created in society under two distinctly different incentive structures.\footnote{Brooks (1994), Stephan (1996) and Stephan and Levin (1996) discuss similar issues.} In ‘Science’, peer recognition and the resulting reputation lead to complementary benefits, such as grants, positions in academic organizations or highly compensated future positions in firms. The combination of these is the incentive. Scientific recognition is achieved by making one’s contribution public to peer review as quickly as possible and acquiring priority to the new knowledge. In ‘Technology’, the incentive structure is the traditional one: maximization of profit by securing property rights. By definition, latter rights keep the new knowledge private. Dasgupta and David, having studied several fields of research, conclude that these incentive structures in many cases appear simultaneously and that they both are present in the same research areas.
Copyleft is a striking analog of the ‘Science’ while copyright belongs to the ‘Technology’. *The essential property of the copyleft licensing scheme is that it creates a particular incentive structure within the business environment. This structure has properties that are equivalent to the incentive structures of scientific communities.*

The framework of Dasgupta and David also captures a crucial element of the positive economic model of copyleft. They assert that the occupational choices of aspiring employees are the decisive factor in the relative shares of both incentive environments. Employees assess the benefits of the ‘Science’ and ‘Technology’ environments. For the same performance in the ‘Science’ environment, the expected monetary return is usually lower than in ‘Technology’ (see Stern, 1999) but because of the complementary nature of income they may be linked to less interesting activities, like for example teaching or project management. The elasticity of the compensation with regard to the performance may be much higher in the ‘Science’. An able scientist is almost certainly rewarded for his contributions because, according to the priority principle, they become public and receive the appreciation of peers. In the ‘Technology’, it is probable that the incentives in the firms do not take the contributions of individual employees fully into account, in tasks performed by groups maybe not at all. Furthermore, the secrecy inherent in private research inhibits also the accumulation of reputation to individual employees.

Literature on copyleft or open source software is limited. Lerner and Tirole (2000), however, do provide an extensive survey of case studies of projects in which the development mode is decentralized and the licensing of the programs is based on copyleft. They analyze the motivation and incentives of the programmers engaged in copyleft activity, concluding that skill signalling is an essential incentive for copyleft work. Lerner and Tirole concentrate on the effectiveness, longevity and structure of decentralized open source software projects in comparison to the traditional hierarchical commercial projects. Johnson (2001) also focuses on the effects copyleft licensing has on program creation. He models copyleft activity as private provision of a public good: each programmer receives utility from the new code developed by each of them. A single programmer can either contribute to the project or ‘free-ride’, receiving utility from the work by others. He shows that free-riding becomes more common and ultimately a constraining factor in program development as the size of the project increases. He also analyzes and compares the welfare implications of both the copyleft device and the traditional software with copyright, finding that neither leads to the welfare optimum. Johnson acknowledges the signalling incentives in the copyleft work but regards the public good nature of copyleft as dominant. Dalle and Juillien (1999) view copyleft licensing as an ‘anti-patent’ system enhancing creativity in society. They acknowledge the skill-signalling incentive of copyleft programming but consider the expected profits from future programs created under copyright protection to be an important motive for programmers. They introduce an
evolutionary adoption model to study the diffusion of a copyleft and a copyright program to users. However, none of these papers consider the implications the copyleft programs have on the markets for copyright programs. Their focus is on the implications of copyleft for labor markets and software creation.

As a method of securing intellectual property rights, copyright has received considerable attention in the literature. Landes and Posner (1989) provide a seminal model, where the level of copyright protection is endogenous. It is either a decision variable optimized by the social planner or is determined by decentralized markets. If protection is strong, the right owners face little competition from unauthorized copies in the market and profits are high. The trade-off is that a high level of copyright protection inhibits authors of new works from utilizing existing works and the costs of creating new works are higher. Koboldt (1995) develops these ideas further in a market where the original work is sold at a positive price and a substitute copy is sold at the cost of copying. These costs include the production cost and the cost of being caught violating the copyright. Takeyama (1994, 1997) and Shy and Thisse (1999) analyze the incentives of protecting the copyright of programs with costly technical methods. Their models include both copyright programs and unauthorized cheap copies. They find, using different models and assumptions, that with programs exhibiting network externalities, the unauthorized copying may increase the profits of the copyright holders.

The structure of this paper is the following: in Section 3 we first develop the cost function of the monopolist at the development stage based on the incentive structure when a copyleft community exists. We then analyze the market for programs and solve for the optimal behavior of the monopolist in the presence of a copyleft program. Based on the results we present some policy suggestions and finally conclude in Section 4.

3. The model

3.1. Stages

There are two stages. At the first stage a monopoly firm and a copyleft community are engaged in programming projects. At the second stage the consumers value programs and the firm sets a profit maximizing price for its program. Consumers and the firm take the qualities of the competing programs as given. Programs are produced and consumers choose one of the programs or neither of them. The model is solved by backward induction. The time line is in Fig. 1.

The monopolist invests in the quality of his product, a program, at the development stage. The quality is dependent on the programming output. Programmers’ ability is heterogeneous and unobservable to the monopolist. Copyleft licensing is an alternative for programmers who can be engaged in
copyleft work and receive complementary income based on ability. To signal their skills, the programmers in the copyleft community develop a program that is a substitute for the monopolist’s program. The monopolist faces convex costs of program development as a result of the existence of the copyleft community.

At the output stage, consumers buy the monopolist’s program, acquire the copyleft program or no program. Consumers face implementation cost for the program. This cost includes the effort of acquiring the physical media, the effort of installing the program in the computer, conversion and rearrangement of data and learning the program.

Copyright protection ensures that the monopolist can price his product without the threat of illegal copies. Copyright protection, however, is also essential for the copyleft community. The copyleft license grants the copyright to the original programmer. The robustness of this property right is crucial to the incentives of programmers even though the copyleft license allows the free use and further development of the program. Copyright protection in our model is perfect. We work with simple functional forms to obtain sharp results.

3.2. Programmers’ occupational choice

A programmer has several career alternatives. He can be hired by a software firm, become an entrepreneur starting a business or become engaged in copyleft programming work. Based on Dasgupta and David (1987, 1994), we assume that career choices lead to different incentive structures. Employment in the firm is compensated by an equal wage for all programmers, as ability is unobservable. This can be supported by casual empiricism of software development environments, which indicates that the process of creating a competitive program in a centralized development environment requires sophisticated teamwork and project

---

7 While new programs have more complex properties, much of the programming effort is directed to enhancing ‘ease-of-use’. A general trend in society is an increase in computer skills or ‘literacy’, both because of public education and user experience. This tends to decrease the implementation cost.

8 This simplifying assumption can be relaxed as long as the complementary income is more elastic than wage with regard to productivity.
management. It is difficult to assess the ability of a single programmer even if the productivity of programmers varies greatly. The public nature of a copyleft program encourages peer review. Able programmers can build a reputation that results in future complementary income, such as partnership in a software venture, grants or academic employment. Lerner and Tirole (2000) aggregate these career concern\(^9\) and ego gratification incentives under a single heading: ‘signaling incentives’. These incentives are the stronger the more visible the performance is, the more dependent it is on effort, and the more revealing it is about talent. The empirical analysis of Stern (1999) points to that the most able of the population tend to attach themselves to the ‘Science’ community, which corresponds to copyleft in our model.

To formalize, there are \(N\) programmers with industry-specific skills, \(N_R\) employed by the firm, \(N_L\) working in the copyleft community, \(N = N_R + N_L\). Programmers’ productivity \(a_i\) is evenly distributed on the unit interval \([0,1]\). We let \(P\) denote the subjectively expected complementary income of a programmer with the highest skill. The expected complementary income for a less able programmer \(i\) is then \(a_P\). The firm, not observing the individual skills, pays a uniform wage \(w_i\) to hired programmers regardless of their productivity. Programmer \(i\) is indifferent between employment and entering the copyleft community if \(Pa_i = w\). Given \(w\) and \(P\), the level of productivity of the marginal programmer is

\[
a^* = \frac{w}{P}.
\]

The number of copyleft programmers and firm programmers depends on the threshold productivity \(a^*\). Programmers \(N_L = (1 - a^*)N\) with productivity greater than the threshold value \(a^*\) join the copyleft community and those, whose productivity is below \(a^*\), \(N_R = a^*N\), are employed by the firm.\(^{10}\) The complementary income \(P\) is assumed to be inversely related to the size of the copyleft community \(N_L\). The members of the community value new contributions more the smaller the community is. This assumption is in accordance with Lerner and Tirole (2000) and Johnson (2001) on the inner dynamics of copyleft communities. Moreover, based on case studies, Lerner and Tirole assert that a copyleft community is prone to ‘break’ if its size increases. Johnson analyses the production process within the copyleft community and finds that as the size increases, free riding becomes a restrictive factor. We define the complementary income as a decreasing function of the proportion of programmers in the copyleft community. Several case studies of copyleft programming (Lerner and Tirole, 2000) also indicate that some individuals stay in the copyleft community even if

\(^9\) Rigorous analysis of career concerns can be found in Holmström (1999) and Detriwapont et al. (1999).

\(^{10}\) Obviously there is an outside employment option, but we assume that the outside wage is below the wage levels present in this model.
alternative monetary benefits are large. Our formulation also captures the view that some programmers have ideological reasons to be engaged and to stay in copyleft programming. The functional form chosen below conforms to this notion. We thus let the expected complementary income to the most able programmer, \( P \), be a function of the threshold productivity \( a^* \) with the parameter \( \beta > 0 \) describing the level of complementary income (see Fig. 2)\(^{11}\).

\[
P(a^*) = \beta \left( \frac{-\ln(1 - a^*) - a^*}{a^*} \right)
\]  

(2)

As the firm’s wage is equal to the complementary income for the indifferent programmer, we have

\[
w(a^*) = a^* P(a^*) = \beta \left( \frac{-\ln(1 - a^*)}{a^*} - 1 \right)
\]  

(3)

In Eq. (3), \( w(a^*) \) is the wage rate the monopolist will offer to be able to hire \( a^*N \) programmers. The offered wage has a direct effect on the marginal programmer and hence on the size of the copyleft community and the complemen-

\[\text{Fig. 2. Complementary income.}\]

\(^{11}\) The parameterization looks complicated but satisfies \( P(a^* \to 1) \to \infty \), \( P(a^* \to 0) \to 1/2 \beta \) by L’Hospital’s rule and \( P' > 0 \) when \( 0 < a^* < 1 \). It has merits that become obvious when analyzing the monopolist’s profit-maximization: the solution becomes algebraically easy and intuitive and yields simple subsequent results. Replacing Eq. (3) by a linear or exponential function for complementary income decreasing in the number of copyleft programmers does not affect the qualitative results.
tary income. The occupational choices determine the program development outputs. The complementary income specification implies that the monopolist cannot suppress the copyleft community completely by his own actions. The most able members always value copyleft work more than the wage offered by the firm. The number of programmers hired by the firm is \( a^*N \) from and their average ability is \( a^*/2 \), so the total development output of the firm is

\[
X_R(a^*) = \frac{1}{2}a^*N^2 \quad (4a)
\]

Respectively, as the number of programmers in the copyleft community is \((1 - a^*)N\) and their average ability is \((1 + a^*)/2\), the development output of the copyleft community is (see Fig. 3)

\[
X_L(a^*) = \frac{1}{2}(1 - a^*)N^2 \quad (4b)
\]

The total development cost to the monopolist, \( C \), is simply the wage cost in Eq. (3) multiplied by the number of programmers employed:

\[
C(a^*) = w(a^*)N_R = w(a^*)a^*N = N\beta(-\ln(1 - a^*) - a^*) \quad (5)
\]

3.3. Market demand at the output stage

There is a consumer market supplied by a monopolist. We denote the copyright program by \( R \) and the eventual copyleft program by \( L \). There are \( M \) consumers,
who each buy at most one program. The valuation of the copyright program is evenly distributed on the interval \((0,V_R]\) and the valuation of the copyleft program on the interval \((0,V_L]\), respectively. The programs are substitutes and we assume that the ratio of the valuations is constant for all consumers:

\[
\frac{V_{Lj}}{V_{Rj}} = \frac{V_L}{V_R}, \quad \text{for all consumers } j. \tag{6}
\]

Let the price of the copyright program be \(p\). Consumers face implementation costs for programs, \(c_R \geq 0\), \(c_L \geq 0\), respectively. We assume for the rest of the paper that the implementation costs are equal for both programs \(c_R = c_L = c\). A consumer buys the copyright program if the surplus accruing to him from it is larger than that from the copyleft program and naturally at least zero. The marginal consumer, say \(j\), is indifferent between the programs and the following condition holds:

\[
V_{Rj} = p - c = V_{Lj} - c \tag{7}
\]

Using Eq. (6) we can develop Eq. (7) into

\[
V_{Rj} = \frac{p}{1 - \frac{V_L}{V_R}} (> 0). \tag{8}
\]

From Eq. (7) we can see that the existence of the copyleft program affects the monopolist’s behavior. If the valuation of the copyleft program exceeds its implementation cost for some consumers, the monopolist has to take this into account when setting the profit-maximizing price. Consumers whose valuation is higher than that of the marginal consumer in Eq. (8) buy the copyright program. This is represented by distance OR in Fig. 4. If there exists a consumer \(k\) that receives a zero surplus from the copyright program, \(V_{Rk} = p - c = 0\), but a positive surplus from the copyleft program, \(V_{Lk} - c > 0\), the copyleft program will exist in the market. Recalling Eq. (6) we can develop the following condition for the existence of the copyleft program:

\[
p > \left(\frac{V_R}{V_L} - 1\right)c \tag{9}
\]

Fig. 4 illustrates this outcome. The distance OL−OR=RL represents the number of consumers acquiring the copyleft program and the distance OR represents the number of consumers buying the copyright program. The other possibility is that the consumer having zero surplus from the copyright program does not have positive surplus from the copyleft program. In Fig. 4, this would mean that point L is to the left of point R. In that case, while there is a programming effort by the copyleft community, no consumers will use the developed program. The monopolist can control the demand for the copyleft
Fig. 4. The marked for copyright and copyleft programs.

program in the market. The price it sets determines whether consumers will acquire the copyleft program. Apart from that, and as noted earlier, the existence of the copyleft community and even the potential threat of a substitute program will have an impact on the monopolist’s optimal price.

Because of the existence of the copyleft community, the demand function faced by the monopolist is a kinked one:

\[ q = \frac{V_R - p - c}{V_R} M, \quad \text{when} \quad p \leq \left( \frac{V_R}{V_L} - 1 \right) c, \quad (10a) \]

\[ q = \frac{V_R - V_{R'}}{V_R} M = \frac{p}{1 - \frac{V_L}{V_R}} M, \quad \text{when} \quad p \geq \left( \frac{V_R}{V_L} - 1 \right) c. \quad (10b) \]

As the production costs are zero, the revenue function for the monopolist is
Consumers value the programs observing their properties. We assume that the value of properties is increasing in the program development output. Let the valuations $V_R$, $V_L$ of the consumer with highest valuations be linear functions of the program development outputs

$$V_R = \mu X_R(a^*) = \mu \frac{1}{2} a^* N, \quad \mu > 0,$$  \hspace{1cm} (12a)

$$V_L = \mu X_L(a^*) = \mu \frac{1}{2} (1 - a^* N), \quad \mu > 0.$$  \hspace{1cm} (12b)

We turn to reporting the main findings:

**Proposition 1.** The monopolist participating in the market will hire programmers with ability $[0,a^*]$. It must be the case that the productivity of the most able programmer hired, $a^*$, exceeds a threshold level given by $a_{\text{min}}^* > (\sqrt{2})^{-1}$. Moreover, the wage the monopoly pays must exceed a threshold, $w_{\text{min}} = w(a_{\text{min}}^*) > 0.52 \beta$, to attract those programmers.

**Proof.** From Eq. (7), we note that to be present in the market the monopolist cannot allow the maximum valuation for the copyleft program to be higher than the maximum valuation for the copyright program. Otherwise, no consumers would have a positive surplus from buying the copyright program at a positive price, since they can acquire the copyleft program of the same or higher quality for free. The condition is satisfied if the total development output for the copyright program is larger than that for the copyleft program. The monopolist actions determine the outputs through the wage setting. Using Eqs. (12a) and (12b) and inserting result to Eq. (4), the condition can be expressed as:

$$V_R > V_L \Rightarrow a^* > (\sqrt{2})^{-1} = a_{\text{min}}^* \Rightarrow w(a_{\text{min}}^*) > 0.52 \beta \quad \text{Q.E.D.} \hspace{1cm} (13)$$

Maximising the revenues in Eqs. (11a) and (11b) yields the optimal price with and without the copyleft product in the market. By inserting the valuations of Eqs. (12a) and (12b), we can express the prices as functions of the given productivity of the most able programmer hired by the monopolist, $a^*$. As the implementation
cost is an interesting parameter in software markets, we solve for it from conditions of Eqs. (11a) and (11b). Inserting the respective optimal price and the valuations into the conditions in Eqs. (11a) and (11b) results in the following optimal prices for varying levels of $c$,

$$p = \frac{\mu Na^2 - 2c}{4} \text{ if } c > \frac{\mu Na^2(1 - a^2)}{6a^2 - 2} = \bar{c},$$  \hfill (14a)

$$p = \frac{2a^2 - 1}{a^2} \text{ if } c < \bar{c},$$  \hfill (14b)

$$p = \frac{\mu N(2a^2 - 1)}{4} \text{ if } c < \frac{\mu N(1 - a^2)}{4} = \underline{c}.$$  \hfill (14c)

We now prove:

**Proposition 2.** If the implementation cost is low, $c < \underline{c}$, the copyleft program will be acquired by some consumers.

**Proof.** Inserting the threshold values of the implementation cost and the optimal prices in Eqs. (14a)–(14c) into the condition for the copyleft program in Eq. (9) shows that the condition is satisfied for the price and the associated condition in Eq. (14c). Q.E.D.

If the implementation cost is high, no consumers with a positive surplus from the copyright program receive a positive surplus from the copyleft program. The monopolist can price his program as if there were no copyleft program available. If the distribution cost is between the two threshold values, the optimal price for the monopolist is the price that just deters the marginal consumer choosing the copyright program from receiving any positive surplus from the copyleft program. Ultimately, should the implementation cost be zero, all consumers use either the copyright or copyleft program.

We solve the monopolist’s profit-maximizing problem using backward induction. At the development stage, the monopolist anticipates that the program market behaves in a manner described above. It has to decide on the ability of the most able programmer it employs. This, in turn, determines the quality of its program and the quality of the program the copyleft community creates. Consumer decisions depend on the qualities of programs and implementation cost. The cost determines the market structure. In the analysis, which follows, we concentrate on

---

12 Using the terms of the literature on industry entry and exit (Tirole, 1988, Chapter 8), we could say that the copyleft program is blockaded (Eq. (14a)), deterred (Eq. (14b)) or accommodated (Eq. (14c)) in the market.
the scenario in which the copyleft program exists in the market. Profit is the revenue at the output stage, Eq. (11), minus the labor cost at the development stage, Eq. (5). The monopolist has no other costs and that there is no discounting. Inserting in Eq. (11b) the optimal price in Eq. (14c) and the valuations of the programs in Eqs. (12a) and (12b) leads to the following profit function in terms of the decision variable $a^*$:

$$\pi(a^*) = R(a^*) - C(a^*) = M^2\frac{a^*(2a^* - 1)}{8} - N\beta(\ln(1 - a^*) - a^*),$$

when $c \leq \frac{\mu N(1 - a^*)^2}{4} = \xi$. (15)

We prove:

**Proposition 3.** When the implementation cost is sufficiently low, $(c < \xi)$, the share of programmers hired by the monopolist is an increasing function of the size of the consumer market $M$ and consumers’ valuation of quality $\mu$ but decreasing in complementary income $\beta$.

**Proof.** Solving for the monopolist’s profit maximization problem (15) yields the optimal productivity level of the most able programmer hired by the firm, $a^{**}$, which in turn determines the optimal number of employed programmers $N^{**} = a^{**}N$. The first-order condition for the monopolist’s profit maximization is

$$\frac{d\pi}{da^*} = \frac{1}{2} \mu NMa^* - \frac{N\beta a^*}{1 - a^*} = 0$$

(16)

This yields two solutions for the optimal $a^*$, $a_1^{**} = 0$ and $a_2^{**} = 1 - 2\beta/M\mu$, of which only $a_2^{**}$ satisfies the second-order condition and the requirement of Proposition 1 that the solution has to satisfy $a^{**} > (\sqrt{2})^{-1}$. We obtain:

$$a^{**} = 1 - \frac{2\beta}{M\mu}$$

(17)

Inspection of Eq. (17) proves the proposition. Q.E.D.

We prove:

**Proposition 4.** When the implementation cost is sufficiently low, $(c < \xi)$, the profit-maximising wage the monopolist pays the programmers is increasing in the

---

13 Profit functions of scenarios (14a) and (14b) are available upon request.
14 The second order condition reads

$$\frac{d^2\pi}{da^{*2}} = \frac{1}{2} \mu NM - N\beta\left(\frac{1}{(1 - a^*)^2}\right) < 0.$$
size of the consumer market \( M \), consumers’ valuation of quality \( \mu \) and programmers’ complementary income \( \beta \).

**Proof.** Inserting the optimal productivity of the most able programmer employed, \( a^{**} \), into the wage function (3) and differentiating yields the following comparative statics results:

\[
\frac{dw(a^{**})}{dM} > 0, \quad \frac{dw(a^{**})}{d\mu} > 0, \quad \frac{dw(a^{**})}{d\beta} > 0. \quad \text{Q.E.D.} \quad (18)
\]

We also show:

**Corollary 1.** When the implementation cost is sufficiently low \( (c < c) \), a profit-maximising monopolist enters the industry if \( \beta < 0.056M\mu \) and hires at least 89% of the programmers.

**Proof.** Inserting the optimal solution of the decision variable, \( a^{**} \), into the profit function (15) and setting it equal to zero yields the following condition for the model parameters:

\[
\pi(a^* = a^{**}) > 0 \Rightarrow \beta < 0.056M\mu \quad (19)
\]

The monopolist decides to enter the industry at the development stage if the condition in Eq. (19) is satisfied. Applying condition (19) to the optimal solution of the decision variable \( a^* \) in Eq. (17) allows us to characterize the employment decision of the monopolist as follows:

\[
\beta < 0.056M\mu \Rightarrow a^{**} > a^{**}_{\text{entry}} = 0.89 \quad (20)
\]

If the monopolist enters the industry, it hires the profit-maximising number of programmers, \( a^{**}N \). The condition in Eq. (19) determines a lower limit to the employment of programmers and it is 89 percent of all programmers. Q.E.D.

When the condition (19) is not satisfied we can interpret it as a scenario in which the level of complementary income in the software application area is large compared to the market. The approach taken by Johnson (2001), who analyses the copyleft activity as private provision of a public good, fits well to that situation. In his model, the programmers are also the only users of the software and the consumer market is not present.

We also find

**Corollary 2.** If the implementation cost, \( c \), is lower than \( c_{\text{max}} = 0.21\mu N \) or less than 13% of the highest valuation of the copyright program, \( V_c \), both the copyright

\[15\] We use a numerical method to solve for the equation.
and copyleft program can co-exist in the market. The lower bound \( c_{\text{max}} \) is half of the highest valuation of the copyleft program, \( V_L \).

**Proof.** The requirements for the co-existence of the copyright and copyleft programs in the market are that the monopolist makes a positive profit and that the implementation cost is below the threshold value \( c \) defined in Eq. (15). It is decreasing in \( a^* \) and reaches its maximum at the lowest possible value of \( a^* \). Inserting the condition for industry entry in Eq. (19) into the equation for the lower limit of the implementation cost in Eq. (15) yields the proposition:

\[
\frac{\text{max entry}}{c} = 0.21 \mu N. \tag{21}
\]

To calculate the ratio of the threshold implementation cost \( c_{\text{max}} \) to the maximum valuation of the monopolist’s program \( V_R \), we conclude that the lower limit in Eq. (15) is decreasing and the maximum valuation of the program in Eq. (12a) is increasing in the decision variable, \( a^* \). The highest ratio obtains in the minimum value of the decision variable.

\[
\frac{c_{\text{max}}}{V_R(a^* = 0.89)} = \frac{V_R(a^* = 0.89)}{0.13} \tag{22}
\]

Comparing the equations for the lower limit of the implementation cost in Eq. (15) and for the maximum valuation of the copyleft product (12b), we note that their ratio is a constant and always \( 1/2 \). Q.E.D.

If the implementation cost is higher than the value indicated in Eq. (21), the market is served either by the monopolist or by the copyleft community. In condition (21), we can see that the threshold value of the implementation cost is increasing in the total programming resource, \( N \). Many determinants of the actual implementation cost are independent of the quality of the programs and more or less constant. This implies that consumers are more likely to use copyleft programs in software application areas where the programming resource is large.

4. Discussion

We turn to policy issues with the following suggestions. First, when implementation cost is low \( (c < c) \), informing consumers about an unknown substitute copyleft program increases welfare\(^{16}\). To illustrate, let us assume two scenarios in the market. The first involves a monopolist supplying the market. There exists also

\(^{16}\) The Chinese and recently the German government are both promoting the use of Linux operating system. Their motives seem to coincide with the ones described here. (China joins the Linux bandwagon, 2000, IBM signs Linux deal with Germany, 2002).
a copyleft community of programmers. However, the consumers are not aware of the substitute copyleft program available at a low distribution cost. This assumption is realistic since the copyleft programmers are indifferent as to whether the consumers use their program or not. The firm maximizes profits as if the scenario is the one defined in Eq. (14a) regardless of the implementation cost. The second scenario is the one defined by Eq. (15) and analyzed in the proof of Proposition 3. In that scenario the consumers have full information on the substitute copyleft program. To make the analysis more tractable, we assume that the implementation cost is zero, \( c = 0 \).

Maximization of profit in both scenarios yields the following optimal productivities of the most able hired programmers (we denote the first scenario by M, the second by CL):

\[
\begin{align*}
a^*_M &= 1 - \frac{4\beta}{M\mu} \\
a^*_CL &= a^{**} = 1 - \frac{2\beta}{M\mu}
\end{align*}
\]  

Comparison of Eqs. (23a) and (23b) shows that the monopolist will always hire less programmers and the quality of the copyright product will be lower when consumers are unaware of the copyleft substitute. When consumers are not aware of the potential copyleft program, welfare, that is, the sum of the firm’s profit and consumer surplus from the copyright program is \( W_M \). If consumers are aware of the copyleft product, some of them derive surplus from its use. The welfare measure is then \( W_{CL} \). Comparing welfare measures shows that that the inequality \( W_{CL} > W_M \) holds when (we denote \( Z = 2\beta/M\mu \)) \( Z < 1.53 \). On the other hand, the profit in the monopoly scenario is higher, \( \pi_M > \pi_{CL} \) when \( Z < 2 \ln 2 \approx 1.39 \). According to Proposition 1 and Eq. (17) the possible values of \( Z \) are \( 0 < Z < 1 - (\sqrt{2})^{-1} \approx 0.297 \). Welfare is thus always higher when consumers are aware of the existence of the copyleft program and the monopolist has no incentive to inform the consumers of the potential substitute copyleft program. An intriguing outcome is that the market may become unprofitable for the monopolist when consumers learn about the substitute copyleft program. The market and cost parameters that result in profit when the monopolist is alone in the market may not fulfill the entry condition in Corollary 1. This implies that when the monopolist anticipates a policy of informing consumers of copyleft programs the condition for commitment to the programming project at the development stage is the one in Corollary 1.

Our second suggestion is that copyright enforcement is important for copyleft activity. Copyleft incentives rely on copyright enforcement. The literature on copyright (e.g. Besen and Raskind, 1991; Landes and Posner, 1989) generally considers the enforcement of the copyright to fall on the responsibility of the

---

\[17\] The welfare functions and calculations are available upon request.
author. Landes and Posner assume in their model that copyright is not perfect and that increasing the degree of protection is costly. This means that authors, usually firms, with considerable resources are able to defend their intellectual property. Copyleft programmers or communities, however, do not usually possess such resources. As we noted earlier, copyleft programmers are indifferent as to whether consumers use the copyleft programs or not. However, if the distribution of the copyleft program outside the copyleft community results in violations of the ‘collective’ copyright of the program they may prefer not to allow for it. The economics of copyright protection analyses the optimal level of copyright protection (Landes and Posner, 1989; Koboldt, 1995; Johnson, 1985; Novos and Waldman, 1984). There is a tradeoff between stronger incentives to create new works when copyright protection is high and the increased opportunity to create derivative works and lower control costs when protection is low. The existence of copyleft communities is an additional factor in this analysis. The incentives in copyleft activity are strengthened by strong institutional copyright protection. As the consumption of programs created by the copyleft community seems to increase welfare, this promotes stronger copyright protection. We considered copyleft activity analogous to ‘Science’ in the Introduction. It is interesting to note that scientific publications are an institution that largely protects the intellectual property rights (in the sense of priority) of the scientific community. Copyleft communities presently lack such institutions. The increased economic significance of copyleft software has already resulted in a discussion on the need for such institutions (see Deckmyn, 2000a).

The novel contribution of this paper is the analysis of the impact of copyleft licensing on both the development environment and the consumer market for programs. The effect of copyleft on the incentives and conduct of programmers has not been left unnoticed in the literature. In our model, we extend the analysis by introducing a monopolist supplying a copyright protected program. Copyleft activity forces him to face constraints in the programmer labor market and competition from a substitute copyleft program in the consumer market. The role of the copyleft program is dependent on the level of consumer implementation cost for programs. When the cost is sufficiently low, some consumers choose to use the copyleft program and the monopolist has to take this into account in pricing his program. The presence of copyleft activity also creates a barrier to the market entry of the monopolist. The larger the consumer market is compared to the programming population, the larger is the share of the programmer population, which the monopolist hires and the smaller is the copyleft community. If the market size is small and consumer valuations are low, the monopolist may decide not to enter the market. Only the copyleft program is then available. This result coincides with some real-world phenomena. In certain markets, like those for some network utility programs, the supply consists entirely of copyleft programs. Our results imply that the monopolist may not be able to exercise full monopoly power in the market if a copyleft program exists. Schmalensee (2000) analyses the
personal computer operating system market in the US and finds that the market leader, Microsoft, is in practice a monopoly, but does not use monopoly pricing. This deviation is a result of several factors but our analysis provides the explanation that the ‘invisible’ competition from Linux affects the pricing of Windows.

For programmers, copyleft licensing creates an alternative incentive structure reminiscent of scientific research. The assumptions of our model mean that the most able programmers join the copyleft community. There is parallel empirical evidence supporting this finding (see Stern, 1999). Furthermore, even if case studies (Lerner and Tirole, 2000) of copyleft program projects imply that some programmers choose to engage in copyleft programming instead of highly paid copyright programming, modelling of the complementary income needs further empirical study of the programmers’ incentives. In our model, the consumer market is served by a monopolist. This is clearly a simplification. The program business is sequential in nature: first there is the development stage and then the output of the program. For the firms to recoup the programming investment in the market, imperfect competition of some degree is required. Relaxing the monopoly assumption and assuming an oligopoly does not change the qualitative results. In more general terms, the partial equilibrium nature of our model hides some important issues. Here we assume that the complementary income is an outside option. Looking at the whole economy in general equilibrium terms raises the question of the determination of the complementary income. A copyleft community may be present in each sub-industry and copyleft products may dominate some markets. In this environment we can ask who provides the income for the copyleft programmers and what consequences it may have. In taxing suppliers of copyright programs, society has to take into account the substitute nature of the copyleft programming, which it may support from public funds.

There are numerous avenues for future research concerning our topic. Our model does not address network effects for programs. These are an important property of programs and in the case of copyleft a new issue arises: the network for the copyright program is the number of consumers using it, while the network for the copyleft program is the number of consumers using it and the programmers in the copyleft community. This creates an environment in which a copyleft program may enter a market dominated by copyright programs more easily if it has a large number of developers. The incentive structure of copyleft programmers may also have other implications. In the standard literature on signaling models where potential employees signal their ability with the level of education it is assumed that the amount of education acquired does not affect the employer’s profit. Copyleft programming, if used as a signal of ability, may have a negative impact on the employer’s profit. The copyleft community develops a substitute program in a decentralized manner and this affects the market for the program of

\footnote{See for example Chapter 11 of Hirschleifer and Riley (1992).}
the employer. He will take this into account and this may change the results of the signaling analysis.

Acknowledgements

I thank Vesa Kanniainen for advice and encouragement. The valuable comments of Vincenzo Denicoló and Tuomas Takalo at the FPPE March 2001 workshop are appreciated. The comments of seminar participants at the Internet Economics Workshop, Berlin, May 2001, at the ZEW Conference on The Economics of ICT, Mannheim, June 2001, at The VATT and ETLA conference on Antitrust Issues In Network Industries, Helsinki, August 2001 and at EARIE 2001, Dublin, September 2001 are appreciated. The financial support of the Academy of Finland was vital for this work. The remaining errors are mine.

References

Dalle, J.-M., Juillien, N., 1999. Nt vs. Linux, or some explorations into the economics of free software, mimeo, ENS Cachan & IDHE-CNRS.


