Free Beer

Written by speakers at FSCONS 2008

Edited by Stian Rødven Eide
LaTeX by Patrik Willard

November 29, 2010

Free Beer 1.0

This book was written during a year and finished in a hurry.
All texts are based on speeches that were held during FSCONS 2008.
A complete list of reference links can be found on freebeer.fscons.org.

FSCONS, the Free Society Conference and Nordic Summit, is a annual event taking place in Göteborg, Sweden. Please visit fscons.org for more information.

Johan Söderberg

8

Hackers GNUnited!

8.1 The political left and the politics of hackers

In this article I will look at hacking from a trade union perspective. The political significance of computer hacking has puzzled the old left, though there are some communicating bodies between the hacker movement and traditional, social movements. Most noticeable are those groups within the computer underground calling themselves 'hacktivists'. They want to apply their computer skills in furthering an already established political agenda, such as feminism or environmentalism[29]. More challenging is making sense of the political agenda of the mainstream of the hacker movement. One immediately comes up against the question of does the computer underground qualify as a social movement at all. Many hackers, perhaps the majority, would say that this is not the case. At best, politics is held to be secondary to the joy of playing with computer technology[30]. Even so, out of this passionate affirmation of computers have grown ideas with political ramifications. For instance, hackers who otherwise do not consider themselves as 'political' tend nevertheless to be opposed to software patents and state surveillance on the Internet, to mention just two examples. Indeed, these viewpoints are so widely shared in the computer underground that they look more like commonsense than political stances. Some issues, such as campaigns against the expansion of intellectual property laws and the defence of freedom of speech, have been added to political agendas and are actively promoted by hacker lobby groups, two examples of which are the Free Software Foundation and the Electronic Frontier Foundation. These organisations are clearly involved in politics, though they claim that these interests cut along different axes than the traditional right-left divide. When social scientists have analysed the assumptions which lay behind the public statements of these hacker lobby groups however, they have usually found a close affinity with liberalism[31].

A couple of leftist writers have broken ranks in that they do not interpret hacking as a liberal ideology. Quite to the contrary, they believe that the hacker movement could revitalise the old struggles of the left, not just for individual freedom but also against injustice and inequality. The most renowned insider who has voiced such opinions about hacking is Eben Moglen. He is a law professor and was for a long time a senior figure in the Free Software Foundation. Moglen is also the author of *The DotCommunism Manifesto*, where he predicted that the anarchism of free software development would replace capitalist firms as the most efficient mode for organising production in the future[32]. The media scholar Richard Barbrook reasoned in a similar way when he was debunking the hype about 'free markets in cyberspace' which was touted in the 1990s. Instead he presented his own vision of a high-tech, anarchistic gift economy. The impulse to give would follow automatically from the fact that people on the Internet had a self-interest in sharing information freely rather than trading it on a market[33]. Arguably, the rise of Napster and later generations of file-sharing technologies could be said to have proven Barbrook right. Even more iconoclastic in his embrace of socialist rhetoric is the Slovenian philosopher Slavoj Zizek. He has paraphrased Lenin's endorsement of electricity by stating, tongue-in-cheek, that 'socialism equals free access to the Internet plus power to the Soviets'[34]. At least a few old-time communists are taking this idea seriously. They believe that computer technology has provided the missing link which at last could make a planned economy a viable alternative to the market economy[35].

But these positive affirmations of hacking and computer technology are probably minority opinions within the traditional left. There is a deeply rooted suspicion among leftist intellectuals towards computer technology and, by extension, its most zealot users, i.e. hackers. The Internet's origin in American cold war institutions is sufficient to put off many progressive thinkers[36, 37]. Add to that the hype surrounding the Internet in the mid-1990s. It gave new lease to the old chestnut about the ’Information Age’. This notion dates back to the 1950s and conservative American sociologists who set out to disprove the continued relevance of class conflicts. By announcing an end to industrial society, they wanted to prove that tensions between the classes had been dissolved and the ideological struggle between liberalism and socialism was becoming obsolete. Consequently, left-leaning scholars have protested against notions about the rise of an Information Age and insisted on the continued existence of industrialism, capitalism, and class conflict[38]. To make this point they have only to call attention to the inhuman conditions under which computer electronics are manufactured in export zones in third world countries[39]. A report from 2008 has documented how girls in China as young as 16 years old are working twelve to fifteen hours a day, six or seven days a week, and barely earning a living[40]. These findings resonate with the historical circumstance that punched cards, numerical control machinery, mainframes, and other embryos of modern computers were instrumental in making blue-collar workers redundant and degrading craft skills at the point of production[41, 42].

Now, having briefly outlined the perplexed relation between the traditional left and the political thrust of hackers, this article will proceed by examining the political significance of hackers in the light of an old debate about factory machinery and labour. The Braverman Debate, as it is known after the author who started the controversy, harks back to the 1970s. Harry Braverman published a book where he argued that the deskilling of labour was an inherent quality of capitalism. The reason was that managers strove to become independent of highly skilled workers in order to keep wages down and unions politically weak. Braverman found support for his hypothesis in the writings of the pioneers of management philosophy. The pivotal figure among them, Winston Taylor, had laid the foundation of what is now known as ’scientific management’ or ’Taylorism’. A central idea of scientific management is that the shop-floor ought to be restructured in such a way that tasks can be done with simple routines requiring a minimum of skills from employees. Taylor argued that this could be done through the introduction of factory machinery. Braverman showed how this strategy was being deployed in heavy industry during the mid twentieth century.

This insight can serve as a lens for looking at the political significance of computer machinery and the hacking of it. The novelty of this argument is that its analysis of hackers is formulated from a production-oriented perspective, as opposed to a consumer rights perspective. It will be argued that the rise of Free and

Open Source Software (FOSS) can be traced back to the industrial conflict between managers and workers. Furthermore, the similarity between the struggle of workers against factory machinery and the struggle of the hacker movement against proprietary software will be highlighted. Free access to source code, a key concern of hackers, contradicts the factory system and the logic of scientific management in computer programming[43]. Though the situation of programmers compared to blue-collar workers is very different in many respects, the article notes that both groups are preoccupied with the goal of preserving skills and worker autonomy in the face of rapid technological change. Hackers' demand that source code should be freely accessible can be interpreted as part of a strategy which is aimed at preserving the programmer's know-how and his control over the tools of his trade.

8.2 The machine at work

The ambivalent feelings of enthusiasm and fear which computer technology often evokes among people have a historical precedent. At the dawn of the industrial revolution, it was hotly debated in all quarters of society what mechanisation would do to the human being, both socially and spiritually[44]. Even some of the forerunners of liberal economic theory, such as David Riccardo, admitted that the working class had good reasons for being resentful of factory machinery[45]. The wretchedness which befell workers who were subjugated under machinery and factory discipline was vividly described by James Kay, a social reformer who worked as a doctor in the slums:

> "While the engine runs the people must work – men, women and children are yoked together with iron and steam. The animal machine – breakable in the best case, subject to a thousand sources of suffering – is chained to the iron machine, which knows no suffering and no weariness."[46]

Early management writers like Andrew Ure and Charles Babbage welcomed this opportunity and advised factory owners how to design machinery in order to keep workers docile and industrious[47, 48]. Their testimonies informed Karl Marx's analysis of capitalism. He denounced factory machinery as 'capital's material mode of existence'. But he also qualified his critique against technology by adding that: "It took time and experience before the workers learned to distinguish between machinery and its employment by capital, and therefore to transfer their attacks from the material instruments of production to the form of society which utilises those instruments."[49]. Thus Marx renounced the strategy of machine breaking which had been the hallmark of the Luddites. The Luddites consisted of combers, weavers, and artisans who felt that their trade was threatened by the introduction of new looms and a subsequent reorganisation of the textile industry. Nightly raids were conducted to smash wool mills and weaving frames owned by 'master weavers'. These activities culminated in 1811-1813 and at one time the English Crown had to deploy 14,400 soldiers in the region to crush the nightly insurgencies. Quite remarkably, more English soldiers were mobilised against the Luddites than had been sent to Portugal four years earlier to face Napoleon's army[50]. In his classic re-examination of the Luddite uprising, Eric Hobsbawm showed that the breaking of machines was not a futile resistance against technology and progress, as it was later made out to have been. Instead he interpreted it as a method of 'collective bargaining by riot'. Breaking the machinery was one option, but workers could also put pressure on their employers by setting fire to the warehouse or sending anonymous threats. Hobsbawm concluded that, if judged by the ability of workers to preserve their wages and working conditions, they had been moderately successful[51].

The misreading of the Luddite rebellion as deranged, irresponsible, and, most importantly, as having nothing at all to do with politics, resembles the portrayal of hackers in news media today. Andrew Ross has protested against the image of the hacker as a petty criminal, a juvenile prankster, or, alternatively, a yuppie of the Information Age. He stresses that spontaneous sabotages by employees contributes to most of the computer downtime in offices. These attacks often go unreported since managers prefer to blame external adversaries. With this observation in the back of his mind, he suggests a much broader definition of hacking:

> "While only a small number of computer users would categorize themselves as 'hackers', there are defensible reasons for extending the restricted definition of *hacking* down and across the case hierarchy of systems analysts, designers, programmers, and operators to include all high-tech workers – no matter how inexpert – who can interrupt, upset, and redirect the smooth flow of structured communications that dictates their position in the social networks of exchange and determines the pace of their work schedules."[52]

Andrew Ross' suspicion is confirmed by studies conducted by employers' organisations. Personnel crashing the computer equipment of their employers is a more common, more costly, and more dreaded scenario for firms than the intrusion by external computer users. According to a survey in 1998 conducted jointly by Computer Security Initiative and the FBI, the average cost of a successful computer attack in the U.S. by an outsider was $56,000. In comparison, the average cost of malicious acts by insiders (i.e. employees) was estimated to $2.7 million[53]. The fondness of employees for attacking the computer systems of their employers underlines the role of computerisation in transforming the working conditions of white-collar office workers. Ross' comparison with sabotage will certainly raise some objections among 'real' hackers. Those of the hacker movement who want to be 'fit for the drawing room' try to counter the negative media stereotype of hackers by differentiating between original hackers and so-called crackers. The former name is reserved for creative uses of technology which contributes to socially useful software projects. The negative connotations of computer crime are reserved for the latter group[1].

These efforts at improving the public relations of hackers merely underline the historical parallel with labour militancy suggested above. The trade union movement too has rewritten its own history so that sabotage, wildcat strikes and acts of violence are left out of the picture. Indeed, unions have been very successful in formalising the conflict between labour and capital into a matter of institutionalised bargaining. The case could be made, nonetheless, that the collective bargaining position of labour still relies on the unspoken threat of sabotage, strikes and riots[54]. In the same way, I understand the distinction between hackers and crackers to be a discursive construction that does not accurately portray the historical roots and the actual overlapping of the subculture. Rather, it seeks to redefine the meaning of hacking and steer it in one particular direction. In spite of the success of this rhetoric, it is nevertheless the case that the release of warez, the breaking of encryptions, and the cracking of corporate servers play a part in the larger struggle to keep information free.

Having said this, the reader would be right in objecting that the motivation of Luddites and workers for rejecting factory and office machinery is very different from the motivation of hackers who are fighting against proprietary software. For

[1] For instance, the Jargon file, which is considered to be the authoritative source on hacker slang, goes out of its way to distinguish between crackers and 'real' hackers: http://ur1.ca/f6o3 (accessed: 27-05-2009)

the latter group, computers reveal themselves as consumer goods and sources of stimulus. Arguably, their relation to technology is one of passion rather than hostility. Even when hackers (crackers) sabotage corporate servers, it is an act out of joy. Discontented office workers might also take some pleasure in destroying the computer of their employer, but it is still meaningful to say that their act springs from resentment against their situation. This difference in motivation does not, however, rule out the possibility that hackers share some common ground with machine breakers of old. Both are caught up in a struggle which is fought out on the terrain of technological development. It might even be that the passionate affirmation of technology by hackers offers a more subversive line of attack, in comparison to, for instance, the insurgency of Luddites. Though it is incorrect to say that Luddites were against technology *per se*, it is true that they defended an outdated technology against a new, scaled-up factory system. Thus it appears in hindsight as if their cause was doomed from the start. Hackers, in contrast, have a technology of their own to draw on. They can make a plausible claim that their model for writing code is more advanced than the ’factory model’ of developing proprietary software.

8.3 Deskilling of workers, reskilling of users

It is a strange dialectic which has led up to the current situation where hackers might reclaim computer technology from companies and government institutions. Clues as to how this situation came about can be sought in a retrospective of the so-called Braverman Debate. The controversy took place against the backdrop of the idea about the coming of a post-industrial age[55]. Two decades later, the same idea was repackaged as the ’rise of the Information Age’ or the ’Network Society’. This notion has come in many hues but invariably paints a bright future where capitalism will advance beyond class conflicts and monotonous work. Crucially, this transition has not been brought about through social struggle but owes exclusively to the inner trajectory of technological development. Harry Braverman targeted one of its key assumptions, namely that the skills of workers would be upgraded when blue-collar jobs were replaced with white-collar jobs. He insisted that the logic of capital is to deskill the workforce, irrespectively whether they are employed in a factory or in an office. Instead of a general upgrading of skills in society, he predicted that the growth of the so-called ’service economy’ would result in white-collar office workers soon confronting routinisation and deskilling just as the blue-collar factory workers had done before.

> “By far the most important in modern production is the breakdown of complex processes into simple tasks that are performed by workers whose knowledge is virtually nil, whose so-called training is brief, and who may thereby be treated as interchangeable parts.”[56]

His statement was rebutted by industrial sociologists. They acknowledged that deskilling of work is present in mature industries, but argued that this trend was counterbalanced by the establishment of new job positions with higher qualifications elsewhere in the economy. At first sight, the emergence of the programming profession seems to have proven the critics right. One of the critics, Stephen Wood, reproached Braverman for idealising the nineteenth century craft worker. Wood pointed at the spread of literacy to prove that skills have also increased in modern society[57]. His comment is intriguing since it brings into relief a subtlety that was lost in the heated exchange. It is not deskilling *per se* that is the object of capital, but to make workers replaceable. When tasks and qualifications are standardised, labour will be cheap in supply and lack political strength. From this point of view, it doesn’t really matter if skills of workers level out at a lower or higher equilibrium. Universal literacy is an example of the latter.

Literacy in this regard can be said to be analogous to present-day campaigns for computer literacy and calls for closing the ’digital gap’. In a trivial sense, skills have increased in society when more people know how to use computers. One might suspect that a strong impetus for this, however, is that computer literacy reduces a major inertia in the scheme of ’lifelong learning’, that is, the time it takes for humans to learn new skills. Once workers have acquired basic skills in navigating in a digital environment, it takes less effort to learn a new occupation when their old trade has become redundant. This somewhat cynical interpretation of computer literacy can be illustrated with a reference to the printing industry. The traditional crafts of typesetting and printmaking took many years to master and it required large and expensive facilities. The union militancy which characterised the printing industry was founded upon this knowledge monopoly of the workers. The introduction of computer-aided processes was decisive for breaking the strength of typographic workers[58]. Personal computers can be seen as an extension of this development. Software mediation allows the single skill of navigating in a graphical interface to translate into multiple other skills. With a computer running GNU/Linux and Scribus, for instance, the user is able to command the machine-language of the computer and can imitate the crafts of printmaking and typesetting. Very little training is required to use these programs compared to the time which it took for a graphical worker to master his trade. This suggests how computer literacy reduces the inertia of human learning and makes the skills of workers more interchangeable. Liberal writers interpret this development as an example of linear growth of learning and education corresponding with the so-called 'knowledge society'. From the perspective of labour process theory, quite to the contrary, the same development is seen as a degradation of the skills of workers and ultimately aimed at weakening the bargain position of trade unions.

David Noble's classic study of the introduction of numerical control machinery in heavy industry in the mid twentieth century provides the missing link between Braverman's argument about deskilling and the current discussion about computers and hackers. One thing which his study sheds light on is how the universality of the computer tool was meant to work to the advantage of managers. Their hope was that it would weaken the position of all-round, skilled machinists. Special-purpose machinery had failed to replace these labourers, since initiatives had still to be taken at the shop-floor to integrate the separate stages of specialised production. In contrast, general-purpose machines simulated the versatility of human beings, thus it was better fitted to replace them[59]. This historical connection is important to stress because it is now commonplace that the universality of computer tools is assumed to be an inherent quality of information technology itself. Thus the trajectory towards universal tools has been detached from its embeddings in struggle and is instead attributed to the grace of technological development.

Saying that does not oblige us to condemn the trend towards a levelling out of productive skills and the growth of universal tools such as computers. On the contrary, in sharp contrast to the negative portrayal of Harry Braverman as a neo-Luddite, Braverman reckoned that the unification of labour power caused by machinery carried a positive potential.

> “The re-unified process in which the execution of all the steps is built into the working mechanism of a single machine would seem now to render it suitable for a collective of associated producers, none of whom need spend all of their lives at any single function and all whom can participate in the engineering, design, improvement, repair and operation of these ever more productive machines.”[60]

With a universal tool, the computer, and the near-universal skill of using the computer, the public can engage in any, and several, productive activities. It is from this angle we can start to make sense of the current trend of ’user empowerment’. In other words: Displacement of organised labour from strongholds within the capitalist production apparatus, through a combination of deskilling and reskilling, has prepared the ground for computer-aided, user-centred innovation schemes. Because programs like *Inkscape* and *Scribus*, and their proprietary equivalents, are substituting for traditional forms of typesetting and printmaking, a multitude of people can produce posters and pamphlets, instantly applicable to their local struggles. Companies have a much harder time controlling the productive activity now than when the instruments of labour were concentrated in the hands of a few, though relatively powerful, employees. What is true for graphic design equally applies to the writing of software code and the development of computer technology. Here the Janus face of software comes to the fore: the very flexibility and precision by which software code can be designed to control subordinated workers the same ease allows many more to partake in the process of writing it. Though embryonic forms of computer technology, such as numerical control machinery, were introduced at workplaces by managers in order to free them from their dependency on unionised and skilled workers; as a side-effect, computer technology has contributed to the establishment of user-centred production processes partially independent of managers and factories. The free software development community can be taken as an illustration of this.

8.4 Free software as a trade union strategy

The corporate backing of the Free and Open Source Software (FOSS) development community must be seen against the background of a restructured labour market. During the last few decades, industrial sociologists have documented a trend where the factory is losing its former status as the role model of production. The point of production has become increasingly decentralised and spread out in a network of subcontractors, freelancers, work-at-home schemes, and franchisees[61]. Companies can now add volunteer development communities to the list of heterogeneous forms for contracting labour. Or, saying it with a catchphrase, labour is outsourced and open sourced. The opportunity to drastically cut labour costs for software maintenance has attracted government institutions, vendors, service providers, and hardware manufacturers to FOSS. The savings that are made by giants such as

IBM, the U.S. Army, and Munich city, to mention a few high-profile cases, has created the space for specialised software firms to sell free software products and services. This analysis is consistent with Tiziana Terranova's critical remark that the engagement of free labour has become structural in the cultural economy. She protested against the many hopes and claims made about the trend of active media consumption, first celebrated in the cultural studies discipline from the 1980s and onwards and most recently updated with the hype around Web 2.0. In response to these often unfounded claims, Terranova responded that capital has always-already anticipated the active consumer in its business strategies[62] (2000). Her argument provides a corrective to the uncritical appraisals of the fan fiction subculture, the creative commons licence, and other expressions of 'participatory media'. Nevertheless, in my opinion, left-leaning critics like Terranova have been too eager to cry out against the economic exploitation of volunteer labour and have thus failed to see the potential for political change which also exists in some of these cases.

The relevance of my objection has to be decided on a case-by-case basis. While I concede that the interactivity of video games and the volunteer efforts of fan fiction writers is unlikely to result in any substantial political change, the interactivity and the gift-giving of free software developers cannot be tarred with the same brush. Here it must be taken into account that the software code is given away together with a clearly articulated, political goal: to make free software the standard in computing. It is true that this standpoint is not anti-commercial in a straightforward sense. As is probably known to the reader, the General Public Licence (GPL) protects the right of the user to run software for any purpose, including commercial purposes[63]. In practice, of course, this option is limited by the fact that GPL also allows sold copies to be copied and given away for free. While the free licence resides perfectly within an idealised free market, it is ungainly within the actually existing market which always presupposes quasi-monopolies and state regulations[64].

This goes some way to explain why the political right is in two minds about free software licences. Self-acclaimed libertarians, such as Eric Raymond, see the growth of open source business models as a better approximation of the free market. Behind this assessment lies an understanding of capitalism as basically identical with its institutions, i.e. private property, free markets and contracts. But that outlook disregards another possible definition of capitalism which puts stress on capital as self-expansion of money, or, in other words, accumulation. The latter viewpoint is central to Marx's analysis of capitalism, but it is also closer to the concerns of the ’captains of industry’. With that in mind, it can be interesting to take notice of market research which *claims that the adoption of FOSS* applications by businesses are eating into the annual revenues of proprietary software vendors by $60 billion per year. Crucially, the losses to proprietary software companies are disproportionate to the size of new FOSS markets, for the simple reason that a lot of it is not paid for.[2]. Hence, the opposition against FOSS from parts of the industry is not necessarily as misplaced as it has often been made out to be. This opposition reached a climax in the court case between the SCO Group and corporate vendors of GNU/Linux which came to an end in 2007. During the court case, the executive officer of the SCO Group, Darl McBride, wrote an open letter to the American Congress where he accused his competitors of being naïve in supporting FOSS licences: ’Despite this, we are determined to see these legal cases through to the end because we are firm in our belief that the unchecked spread of Open Source software, under the GPL, is a much more serious threat to our capitalist system than U.S. corporations realize.’[3].

At the very least, these worries among some parts of the computer industry show that free software developers cannot be written off as mere unsuspecting victims of commercial exploitation. Perhaps it would be more justified to say that hackers, by freely offering up their labour, are blackmailing corporations into adopting and spreading the FOSS development model. No company answering to the market imperative of lowest costs can afford to argue against free (as in free beer) labour. My hypothesis is that advocacy for free licences can be interpreted in the light of an emerging profession of computer programmers. This suggestion is far from obvious since the identity of the hacker is tied up with the notion of being a hobbyist, or, in other words, a non-professional, non-employee. Contradicting this self-image, however, numbers have it that the majority of the people contributing to free software projects are either working in the computer industry or are in training to become computer professionals[66]. Hence, it is not so far-fetched to connect the dots between hackers and the labour market that awaits them. Indeed, this line of reasoning has already been attempted in Josh Lerner and Jean Tirole’s famous article[67]. They wanted to square the supposed altruism of free software developers with the assumption in neo-classical economic theory about the ’rational economic man’. The two authors concluded that hackers are giving away code

[2] The market research rapport referred to is called Trends in Open Source and has been published by the Standish Group. Because access to the material is restricted, information about it comes from news media[65]

[3] http://ur1.ca/f6o4 (accessed: 01-11-2009)

for nothing in order to create a reputation for themselves and improve their chances for employment at a later date. Without denying that such cases may exist, I disagree with the assumption of methodological individualism that underpins their thinking. When I say that free software licences might be beneficial to the labour interests of computer programmers, I do not mean that this is a rationally calculated strategy or that it is an exhaustive explanation as to why hackers license their software under GPL. Furthermore, in contrast to Lerner and Tirole, I do not think that those labour interests are pursued exclusively through individual strategies. In addition to improving their own reputation, individual hackers are contributing to changing the labour market for programmers as a collective.

It sounds counter-intuitive that programmers would improve their bargaining strength vis-a-vis firms by giving away their work to potential employers. Let me start by returning to an insight of Harry Braverman. He stressed that the very outlay of the factory put the machine operator at a disadvantage. The worker could only employ skills when given access to the machinery. Unfortunately, the scale and mode of organisation of the factory was already biased towards hierarchy. The capitalist had an advantage due to the ownership of the machines and buildings, without which the workers could not employ their abilities. The only bargain chips that the workers had were their skills and intimate knowledge of the production process. This was also how Braverman explained the tendency that capitalists are pushing for technologies which reduce skilled labour. What has happened since Harry Braverman made his analysis in the 1970s is that the large-scale Fordist machine park has grown obsolete in many sectors of the economy. This is particularly true in the computer industry. Productive tools (computers, communication networks, software algorithms, and information content) are available in such quantities that they have become a common standard instead of being a competitive edge against other proprietors (capitalists) and a threshold towards non-possessors (workers). A horde of industrial sociologists and management philosophers have written about this trend since the early 1980s[68]. It is a truism in this body of literature to claim that the employees, not the machine park, are nowadays the most valuable resource of the modern corporation. The claim is clouded in rhetoric, but the validity of the statement can be tested against the adoption of ’non-disclosure agreements’ within the computer industry. It is here stated that the employee is not allowed to pass on sensitive information about the firm. Another kind of clauses which are sometimes included in the employment contract to much the same effect, i.e. to prevent leakages, forbid the programmer from working with similar tasks for a competitor after having left his current employer. These agreements can be taken as testimonies that the knowledge and skills of the programmers have indeed become increasingly precious to the firm to exercise control over. I will argue that these practices, though they formally have very little to do with copyright law, nevertheless brace up my claim that proprietary and free licences affect the bargaining position of software developers.

The justification for these different kind of contractual agreements is the necessity of preventing trade secrets from leaking to competitors. However, as a side-effect, the programmers are prevented from moving freely to similar positions in their trade. Since the programmer becomes a specialist in the field in which he has been working, he might have difficulties in finding a job in a different position. The significance of this observation becomes clearer against the background of Sean O'Riain's ethnographic study of a group of software technicians working in a computer firm in Ireland. It has proved to be very difficult for trade unions to organise these workers. Since jobs are provided on a work-for-hire basis, the collective strategies of unions lack purchase. One of O'Riain's conclusions is that mobility has instead become the chief means by which the employees negotiate their working conditions and salaries[69]. With awareness of this fact, the significance of the contractual agreements mentioned above must be reconsidered. The limitations which they put on the ability of employees to 'vote with their feet' means that the firms get the advantage back. As to what extent non-disclosure agreements and other clauses are actually used in the Machiavellian way sketched out here is something which remains to be investigated empirically. What interests me in this article, however, is that the very same argument can be applied to proprietary software licences more generally.

Intellectual property[4] too is justified by the necessity of firms to protect their knowledge from competitors. A complementary justification is that intellectual property is required so that producers can charge for information from consumer markets. But intellectual property is also likely to affect the relation between the firm and its employees, a subject which is less often discussed. A case can be made that proprietary licenses prevents the mobility of employees. It ensures that the knowledge of employed programmers is locked up in a proprietary standard

[4] Many critics of copyright and patent law reject the words 'intellectual property'. In their opinion, the words are loaded with connotations that mislead the public. Instead they advocate the words 'intellectual monopoly'. I am unconvinced by this argument though there is no space to develop my counter-position here. It suffices to say that I will use the words 'intellectual property' in the article as I think that the association with other kinds of property is entirely justified

owned by the firm. A parallel can be drawn with how the blue-collar worker depends on the machine park owned by the industrialist. Without access to the factory the worker cannot employ his skills productively. In the computer industry, as was mentioned before, most of the tools that the programmer is working with are available as cheap consumer goods (computers, etc.). Hence, the company holds no advantage over the worker by providing these facilities. But when the source code is locked up behind copyrights and software patents, large amounts of capital are required to access the programming tools. As a consequence, the software licence grants the firm an edge over the labourer/programmer. This theoretical reasoning is harder to prove empirically than the claim made before that clauses in the employment contract might be used to restrict the mobility of programmers. Even so, it might be of an order of magnitude greater in importance to the working conditions in the computer sector. Indeed, this production-oriented aspect of proprietary licences might be as significant as the officially touted justifications for intellectual property law, i.e. to regulate the relation between the firm and its customers and competitors. If I am correct in my reasoning so far, then the General Public Licence should be read in the same light. I was led to this thought when reading Glyn Moody's authoritative study of the FOSS development model. He makes the following observation concerning the exceptional conditions for firms specialised in selling services in connection to free software:

> "Because the 'product' is open source, and freely available, businesses must necessarily be based around a different kind of scarcity: the skills of the people who write and service that software."[70]

In other words, when the source code has been made publicly available to everyone under the GPL, the only things which remain scarce on the market are the skills required to employ the software tools productively. And this resource is inevitably the faculty of 'living labour', to follow Karl Marx's terminology. It is thus that the programmers can get an edge over the employer when they are bargaining over salary and working conditions. The free licence levels the playing field by ensuring that everyone has equal access to the source code. Terranova and like-minded scholars are correct in pointing out that multinational companies have a much better starting position when exploiting the commercial value of free software applications than any individual programmer. The savings that IBM makes from running Apache on its servers are, measured in absolute numbers, many times greater than the windfalls bestowed on any programmer who has contributed to the project. Still, at a second reading, the programmer might be better off if there exists a labour market for free software developers, compared to there being no such occupation available. By publishing software under free licences, the individual hacker is not merely improving his own reputation and employment prospects, a point which has previously been stressed by Lerner and Tirole. He also contributes to the establishment of a labour market where the rules of the game are rewritten, for him and for everyone else, in his trade. It can be interpreted as a kind of collective action adapted to a time of rampant individualism.

It remains to be seen if the establishment of a labour market in free software development translates into better working conditions, higher salaries and other benefits otherwise associated with trade union activism. Such a hypothesis needs to be substantiated with empirical data. Comparative research of people freelancing as free software programmers and those who work with proprietary software is much wanted. Such a comparison must not, however, focus exclusively on monetary aspects. As important is the subjective side of programming. An example hereof is the consistent finding that hackers report that it is more fun to participate in free software projects than it is to work with proprietary software code[66]. Neither do I believe that stealth union strategies are the sole explanation as to why hackers publish under GPL. Quite possibly, concerns about civil liberties and the anti-authoritarian ethos within the hacker subculture are more important factors. Hackers are a much too heterogeneous bunch for them all to be included under a single explanation. But I dare to say that the labour perspective deserves more attention than it has been given in popular press and academic literature until now. Though there is no lack of critiques against intellectual property law, these objections tend to be formulated as a defence of consumer rights and draw on a liberal, political tradition.

There are, of course, some noteworthy exceptions. People like Eben Moglen, Slavoj Zizek and Richard Barbrook have reacted against the liberal ideology implicit in much talk about the Internet and related issues. They have done so by courting the revolutionary rhetoric of the Second International. Their ideas are original and eye-catching and often rich with insight. Nevertheless, the revolutionary rhetoric sounds oddly out of place when applied to pragmatic hackers. Advocates of free software might do better if they look for a counterweight to the hegemony of liberalism in the reformist branch of the labour movement, i.e. in trade unionism. I believe that such a strategy will make more sense the more the computer industry matures. In accordance with Harry Braverman's general line of argument, the profession of software engineering has already been deprived of much of its former status. Indeed, from the early 1960s and onwards, writers in management journals have repeatedly been calling for the subjugation of programmers under the same factory regime which had previously, and partly through the introduction of computer machinery, been imposed on blue-collar workers[71]. With this history in the back of the mind, I would like to propose that the advocacy of free software, instead of falling back on the free speech amendment in the American Constitution, could take its creed from the 'Technology Bill of Rights'. This statement was written in 1981 by the International Association of Machinists in the midst of a raging industrial conflict:

> "The new automation technologies and the sciences that underlie them are the product of a world-wide, centuries-long accumulation of knowledge. Accordingly, working people and their communities have a right to share in the decisions about, and the gains from, new technology."[72]

8.5 Acknowledgements

The author would like to thank the editor, Stian Rødven Eide, as well as Michael Widerkrantz and Don Williams, for constructive comments on earlier drafts of this paper.