FREE SOFTWARE AS COMMONS

BETWEEN INFORMATIONAL CAPITALISM AND A NEW MODE OF PRODUCTION

By
Emrah Irzık

Submitted to
Central European University
Department of Sociology and Social Anthropology

In partial fulfillment of the requirements for the degree of Doctor of Philosophy

Supervisor: Professor Jakob Rigi

Budapest, Hungary
2015

Statement

I hereby state that the thesis contains no material accepted for any other degrees in any other institutions. The thesis contains no materials previously written and/or published by another person, except where appropriate acknowledgment is made in the form of bibliographical reference.

Emrah Irzık

Budapest, November 2015

Abstract

Free Software is a particular way of organizing the production and distribution of software that offers a solid alternative to the intellectual property regime by constituting an open commons: non-proprietary, created and held in common by all. Considering that in contemporary capitalism a significant amount of wealth is created through the application of intellectual effort to existing knowledge to produce new, higher compositions of knowledge that can be privately monetized as intellectual property, the challenge that Free Software might present to capitalism is bound to have important transformational potential. This potential needs to be studied both on an empirical level, in its partial and concrete manifestations in actual projects, and investigated more theoretically, to see if Free Software can be characterized as a nascent, new mode of production.

This dissertation aims to contribute to the theorization of the relation between Free Software as a commons and the tenets of informational capitalism by means of an analytical study that is supported by an ethnography of a particular instance of Free Software as a project and a community. In the theoretical part, I analyze in what ways and to what extent Free Software is incorporated into and comes into conflict with informational capitalism. I consider the contributions as well as the limitations of the liberal progressive and classical revolutionary perspectives on Free Software, pointing out the necessity of seeking to explain it as a revolutionary mode of production on its own terms, as the 'transcendent synthesists' do. I then move on to contribute to such an explanation by analyzing three major aspects of Free Software with a claim to capture what is idiosyncratic in it as a mode of production: the forms of remuneration for labor, the emerging class position of the producers, and a specific, radical form of decision making within the sphere of production.

My empirical analysis consists of an ethnographic study of the Free Software community in Turkey with a focus on the Pardus project, the largest experiment with Free Software production undertaken yet in Turkey, under the aegis of the Scientific and Technological Research Council. It seeks to exemplify and complicate my theoretical characterizations by focusing on the community of various actors such as volunteers, employees, users, and academics within and around the project. By means of extended interviews with Pardus programmers, the observation of public conventions, online ethnography of convention archives, social media sites, blogs, and related company or university web pages, I explore how the development of the project is influenced by institutional and cultural-political concerns, the different motivations and perceptions of individuals participating in it, and the social movement aspect of the Free Software community. The Pardus project's bell-shaped historical trajectory of success and its entanglement in Turkish politics reveal important insights into the uneasy relationship of Free Software with capitalism on both micro and nation-state scales.

Table of Contents

List of Abbreviations

AKP	Justice and Development Party
A. Ş.	Incorporation
BİÇDA	Informatics Workers Solidarity Network
BSD	Berkeley Software Distribution
BYOD	Bring Your Own Device
CHP	People's Republic Party
CRL	Content Reciprocity License
CPU	Central Processing Unit
CTO	Chief Technology Officer
ÇOMU	Çanakkale Onsekiz Mart University
DIY	Do It Yourself
DNS	Domain Name Server
DPT	State Planning Organization
EU	European Union
FS	Free Software
FSF	Free Software Foundation
FOSS	Free and Open Source Software
GNU	GNU's Not Unix
GPL	General Public License
GSOC	Google Summer of Code
HDP	Peoples' Democracy Party
ICT	Information and Communications Technologies

IDE	Integrated Development Environment
IHS	Istanbul Hacker Space
IP	Intellectual Property
IT	Information Technology
LETS	Local Exchange Trading System
LKD	Linux Users Association (of Turkey)
METU	Middle East Technical University
MMS	Multimedia Messaging Service
PP	Peer Production
PPL	Peer Production License
P2P	Peer to Peer
P2P Foundation	Foundation for Peer to Peer Alternatives
R&D	Research and Development
SaaS	Software as a Service
SaaSS	Service as a Software Substitute
SME	Small and Medium Size Enterprise
SMS	Short Message Service
TMMOB	Chamber of Architects and Engineers of Turkey
TRT	Turkish Radio and Television
TÜBA	Turkish Academy of Sciences
TÜBİTAK	The Scientific and Technological Research Council of Turkey
UEKAE	National Research Institute of Electronics and Cryptology Department of TÜBİTAK
VPN	Virtual Private Network

PROLOGUE

The Digital Artifact*

The wealth of those sectors of the economy in which the production of information goods prevails, presents itself as "an immense composition of digital artifacts," its unit being a single digital artifact. Our investigation must therefore begin with the analysis of a digital artifact.

A digital artifact is, in the first place, an item represented by a series of bits on a computer, a thing that by its property of representing information, satisfies human wants of some sort or another. The nature of such wants, whether, for instance, they spring from the needs of informational-industrial production, education, or from digital recreation and entertainment, makes no difference. Neither are we here concerned to know how the item satisfies these wants, whether directly as an item of personal use, or indirectly as means of production.

Every useful thing, as iron, paper, etc., may be looked at from the two points of view of quality and quantity. It is an assemblage of many properties, and may therefore be of use in various ways. In the case of the digital artifact, the issue of quantity is only relevant socially, it is irrelevant individually. Possession of multiple copies (beyond the trivial number that may be utilized by multiple devices belonging to the individual) of the same digital artifact represents no additional wealth to the individual, while possession of a copy of the same digital artifact by multiple individuals does represent additional wealth to society. This increase of wealth ranges from the linear (individual information goods) to the exponential (network information goods). To discover the various uses of things is the work of history. So also is the establishment of socially-recognized standards of measure for the quantities of these useful digital artifacts, whether by lines of code, number of words, density of pixels or

number of frames. The diversity of these measures has its origin partly in the diverse nature of the digital artifacts to be measured, partly in convention.

The utility of a digital artifact makes it a use-value. But this utility is not a thing of air. Being limited by the digital form of the item, it has no existence apart from that form. A digital artifact, such as a software application, text file, image, or video file, is therefore, so far as it is a material arrangement of stored bits, a use-value, something useful.

This property of a digital artifact is independent of the amount of labor required to appropriate its useful qualities. When treating of use-value, we always assume to be dealing with definite quantities, such as megabytes of code, pages of text, number of pixels, or minutes of video. Use-values of digital artifacts become a reality only by use, and owing to their special nature of being information goods, do so *without* being consumed in the process. That is, when one individual makes use of the information stored in a digital artifact, the artifact is not destroyed in the process, but remains available for further reproduction and sharing. This is their d*ifferentia specifica,* setting them apart from use-value containing material objects proper. The *value* (as distinct from use-value) of digital artifacts is spread ever so thinner over each and every copy made of the same digital artifact: these artifacts also constitute the substance of all digital wealth, whatever may be the social form of that wealth. In the form of the contemporary society we are about to consider, they have been forced into the commodity-form purely by force of law, and yet they are also undergoing the process of liberation from this commodity-form. They are moving away from being, by legal fiction of the capitalist state, the forced digital depositories of exchange-value by means of licenses, which constitute a form of rent, towards being a universal commons with singular nature: use-value.

A digital artifact can be useful, and the product of human labor, without being a

commodity. Whoever creates digital artifacts to directly satisfy their informational wants with the product of their own digital labor, -"scratching their own itch"- creates, indeed, use-values, but not commodities. In order to produce the latter, they must not only produce use-values, but use-values for others, social use-values. (And not merely for others: the programmer in the university research lab produced code both for the university and for the government, but neither became commodities simply by being produced for others. In order to become a commodity, the digital artifact must be transferred to the other person, for whom it serves as a use value, through the medium of exchange, i.e. by sale of a license). Nothing can have value, without being an object of utility. If the digital artifact is useless, so is the labor contained in it; the labor does not count as labor, and therefore creates no value. Digital artifacts that are indeed objects of utility, and are produced not solely for the satisfaction of individual needs but the needs of any and all who have said needs, and are transferred to them not through the medium of exchange but by free sharing through the medium of digital reproduction, constitute an inexhaustible, and through the means of distribution that is the internet, universal digital commons.

* This text is a shortened adaptation of the first section of Chapter 1 of *Capital,* Vol.1 by Karl Marx, titled "The Commodity" (123-131).

CHAPTER 1: INTRODUCTION

It is generally agreed that since the seventies capitalism has entered into a new stage, variously described as "postindustrial", "informational", or "knowledge-based". This economy is characterized by an increasing emphasis, in terms of value-added, on the input of high-quality, scarce knowledge in the production process. Qualitatively, it is also an era where the creation of a significant amount of wealth comes about through what Manuel Castells has dubbed "knowledge acting upon knowledge" (Castells, 1996), in the sense that intellectual effort applied to existing information and previous knowledge results in a new, highly sought after, higher composition of knowledge. This knowledge, to various extents, can either be privately monetized as intellectual property and used as a means of rent-seeking, or become part of the new, digitally representable commons which is shared, immaterial and inexhaustible, distinct from the classical, exhaustible commons like land and water. Under contemporary conditions, the profit principle dominates for the most part, and the distinguishing characteristic of knowledge-based capitalism has been that knowledge and information are transformed into a restricted, commodified factor of production. The fruits of knowledge-based labor as embodied in works of science, software, literature and art are monopolized via an expansive regime of intellectual property (IP). IP maximalism is legitimated on the individual level by reference to romantic notions of authorship (which is then assumed to be alienable and therefore transferable from the author to an intermediary such as the publisher), and on the collective level by assuming that strong IP protection promotes development by offering a course for compensation.

It must be noted that once produced, knowledge and knowledge-embedded goods in turn act as the materials of labor necessary for the subsequent round of knowledge production,

as each is a partial and constantly evolving embodiment of the accumulated cultural and intellectual riches of society. This fact, which has been latent in all eras of production, is only fully expressed in the contemporary knowledge economy. It significantly erodes the distinction between what is a consumer good (Department II product or means of subsistence in Marxian economics) and what is a capital good (Department I product or means of production). The consumer product is now also a means of production; the consumer is replaced by the user (user-producer, prosumer, etc).

The strategy of profit-based enterprises so far has been to enclose knowledge on the basis of IP laws, trade secrets and employee regulations so that artifacts of the knowledge economy can be treated as if they were a rival good in the market, like material products of industry. Being a rival good means that one person's use of it necessarily bars another person from using the same, which is not the case for the artifacts of the knowledge economy. On the more ontological level, since knowledge is by its nature inalienable, in the sense that one does not part with it upon transferring it to another, it must be made a commodity by force of law, by way of restrictions placed on its reproduction (copyrights) and implementation (patents), so that knowledge can be treated as if it were alienable, in conformity with the logic of capitalist property. The commodity-form taken by these digitally representable knowledge goods, or digital artifacts, appears as *beyond fetishistic*. It does not merely substitute the appearance of relations between objects for what are in fact relations between persons. It denies, in the process, the material reality of the object (digital duplicability) in favor of legal fiction (copyright restriction). The digital artifact in commodity form not only appears to have value by virtue of its intrinsic physical properties rather than by virtue of being a product of social labor; it is furthermore only able to maintain this appearance because it comes into the hand of its buyer attached to a prohibition. In this sense, digital artifacts as commodities

should be properly seen as embodiments of a *tabooistic* economic relation.

There is however, a counter-movement that proposes and practices a different knowledge economy against this arrangement in contemporary capitalism. This is a current which is advancing the Knowledge Commons, advocating and practicing the release of human knowledge in all its creative forms as public goods, aiming to make this wealth open, shareable and accessible to all. There are diverse actors and avenues relevant to this process. In the software realm, this new intellectual commons takes the form of Free Software, as christened by Richard M. Stallman's GNU Manifesto in 1985.

Free Software (FS),[1] which offers a solid alternative to the IP regime, is a particular way of organizing the production and distribution of software. It is based on social collaboration and free sharing, carried out by the coordination of a number of individuals that form an interactive community of producers and users. In FS, the human-readable programming code, called "source code," is made open and freely available under an appropriate, freedom-guaranteeing legal license, together with the compiled binary computer software packages, which cannot be modified as is, but can only be executed by a machine. This allows anyone to develop the software by improving existing components and deriving new software, in addition to freely using it. FS thus constitutes an open commons; it is non-proprietary, in other words, held in common by all.

My research aims to contribute to the theorization of the relation between FS as a commons and the tenets of informational capitalism through an analytical study of FS that is supported by an ethnography of a particular instance of FS as a project and a community. I will analyze in what ways and to what extent FS is incorporated into and comes into conflict

1 "Free Software" (FS) is interchangeable with the term "Free/Libre Open Source Software" (F/LOSS) that is frequently used in the literature. I have settled on the term Free Software, as defined by the four freedoms articulated by the Free Software Definition provided by the Free Software Foundation. https://www.gnu.org/philosophy/free-sw.html

with informational capitalism. Considering that the action of knowledge on knowledge is central to the production of value and the increase of productivity in contemporary capitalism, the challenge that FS might represent to capitalism in this central area is bound to have important transformational potential. This transformational potential needs to be studied both on an empirical level, in its partial and concrete manifestations in actual FS projects, and investigated more theoretically, to see if FS can be characterized as a nascent, new mode of production. My empirical analysis will consist of an ethnographic study of the FS project and community behind Pardus, which is a particular GNU-Linux operating system (referred to as a "distribution" or distro) developed under the aegis of the Scientific and Technological Research Council of Turkey. My theoretical analysis will address the applicability of existing concepts and models in political economy to FS, through the discussion of the empirical findings and a critical engagement with current debates on the nature of FS and its social and political implications.

We immediately see in FS two aspects of interest that are analytically separable yet remain intimately related to each other: FS as a production process, which is a matter of organizing people in their productive capacities, and FS as a product, which organizes people in their capacities as users. In reality these categories are not mutually exclusive; producers are also users, and users also play a role in production. That is why the notion of a "FS community" is merited over the more alienated relationship between producers and consumers (not to mention sellers and buyers).

Producers are sometimes motivated to produce software as use-values (items of utility) for themselves; that is, they are driven to produce a software solution so they can use it "to scratch their own itch". The result, however, benefits everyone who may have the same or similar need, thanks to infinite and perfect digital reproduction of the solution at near zero

cost. This socialization of the product requires nothing but the faintest will to allow others practically costless, immediate and non-rivalrous access to what one produces for oneself. In practice, there are many further motivations to do so. Users on the other hand influence the development of projects either directly by giving feedback to producers through online communication channels like support forums and bug-reporting systems, or indirectly by extending network effects that add to the use-value of the product as a whole (for example, a free software office package is only as good as the number of people who use that package and can therefore collaborate on producing documents).[2] Users are therefore not mere consumers but auxiliary producers and nodes in a network that constitute the *ecosystems* of various software.

FS as a process is a form of labor organization where individual producers have the ability to relate to each other neither through the market nor through hierarchical command structures, but through a myriad of arrangements of self-organized collaboration and project leadership made possible by FS. This characteristic is captured in the term "Free as in Freedom", that is, the freedom to participate in use, development, production of derivatives, and distribution. The actors in this collaboration, composing a community made of programmers and users, can be volunteers, employees paid by companies, students, or workers employed by government agencies (among others). I will outline how the FS commons serves as a platform that harmonizes these various groups and interests while elaborating on the internal and external politics that result, the often delicate balances and enduring frictions.

FS as a product is the concrete form of de-commodified software packages, which poses a threat to the commercial nature of the software industry insofar as it competes with

2 Software interoperability and compatibility can in principle alleviate this issue; however, in practice proprietary incumbents often have little interest in adhering to standards because it would undermine vendor lock-in. See https://en.wikipedia.org/wiki/Vendor_lock-in

and displaces established proprietary solutions in the market. This characteristic is reflected in the notion of "Free as in Beer", meaning the distribution at zero-price for users, which, while not the priority goal of FS as a movement, is a practically guaranteed corollary outcome and is no less significant overall. There is nothing keeping one from attempting to sell a piece of FS for any price one likes; however, one would be hard pressed to find buyers for a product that is free to download elsewhere. Therefore, as long as "consumer knowledge of the market" is not obscured,[3] competition will drive the price of a FS to zero.

FS, like all software, may provide a foundation for the complementary commercial business of selling "support and services" around it;[4] however, for a given market size, it must shrink the total profit that can be made in this way as it does away with monopoly rents (license sales) and cannot close off competition in the services market in the manner that the supplier of a proprietary software package can. All other things being equal, a company that can sell both software licenses and support and services will make more profits than a company that releases its software freely and restricts its profit generating activity to selling support services. All other things *not* being equal, it is of course possible that a FS producing company may in the end make more profit than a proprietary competitor. This may occur in the following manner: If the FS product is more or less on par with the proprietary rival, by virtue of releasing its software for free, the FS company will have a much larger user base that may translate into more users who go on to buy support and services predominantly from the FS producing company, which, as its producer, is likely to be the company with the highest expertise on that FS (even if not a monopolist). This and similar second-order mechanisms for making a profit in no way negate the fact that in the domain of the relevant type of software

3 Fly-by-night commercial operations that take a FS, rename or otherwise superficially alter it and attempt to sell it exist, and are legal, but are frowned upon and in terms of scale are not much more than a nuisance.

4 And indeed numerous forms of remuneration and funding that do not involve any sales (commodity relations) at all. These will be discussed in Chapter 3 of this work.

per se, profits will have been reduced and the market will have shrunk in favor of the free commons. This means that the wealth of society increases dramatically (more users accessing the FS due to zero price), and as a result, the size of a separate market, the market for support and services for the software at hand, grows. The net result to society is an increase of wealth when compared to the system of selling software licenses, which is characterized less by the creation of wealth, and more by the transfer of wealth from one group in society to another, i.e. rent.

While the history of intellectual property spans centuries and goes much further back than the emergence of computer software, the historical evolution of software production in general and FS in particular within this overall capitalist IP regime has its own specificity. Software was not originally born in a setting of IP-based enclosure and commodification. In its early days in the 50's and 60's, the technology and expertise concerning software production and use emerged not in corporations and commercial enterprises, but mostly in universities, research facilities and the surrounding hacker milieu, as part of an intellectual commons. The intellectual commons, or knowledge commons in software was the commonly held open pool of knowledge that had been created by its producers who shared their intellectual products freely, rather than making property claims on them.

An important milestone toward the closing of these early times of software production was the infamous "Open Letter to Hobbyists" written in 1976 by Bill Gates. He argued against the sharing of computer software and source code, which he referred to as theft (of the work of the original programmer), and defended the commodification of computer software, claiming that the profit motive and market-based remuneration model would spur development and lead to the production of high quality software.[5] This moment, which Steven Weber describes in *The Success of Open Source* as a shock (Weber, 2004: 36-37), epitomized

5 https://en.wikipedia.org/wiki/Open_Letter_to_Hobbyists

a pro-commercial break from the commons tradition in software, a Polanyian first movement of enclosure and marketization,[6] an assertion of latent copyrights that were up to that point being customarily ignored. This rapidly marginalized the de facto commons and characterized the software industry throughout its booming years. For about a good two decades, the commons in software was to remain an undercurrent.

The challenge that Free Software (FS) presents as a Polanyian second movement against this development is apparent in the fact that it fosters a culture of cooperation, diversity and freedom by providing not only the end product (the binary executable) but also the most crucial means of production (the source code) for free (both as in freedom, and as in cost). Furthermore, the most popular contemporary FS license tied to FS is the GPL (General Public License), which not only grants all of the relevant freedoms to users and developers, but also precludes the use of GPL code in non-GPL projects in the future, explicitly safeguarding the code from future cooptation. Referred to as "the viral clause" by its detractors, this restriction on placing restrictions on code is an example of "Copyleft", the subversion of copyright law to serve ends opposite to its spirit. Copyleft is enforceable by copyright law because the original author (the holder of copyright) is setting these terms, which allow all freedoms except the freedom to restrict the freedom of others. In this sense, the software commons of the pre-Gates era was merely FS avant la lettre, while the software commons of the GPL era took the form of a conscious movement. Today there are diverse actors that participate in the creation and development of FS, some motivated by idealism, others by income. However, within its domain, the GPL guarantees that all who want to be part of the game must play by the rules originally laid out by Stallman's Free Software

6 In *The Great Transformation* (1944: 75), Karl Polanyi describes "the dynamics of modern society" being "governed by a double movement". The first movement being the expansion of the market towards becoming dis-embedded from society, the second movement being society's reaction towards limiting and re-embedding it. In a similar vein, Boyle writes: "We are in the middle of a second enclosure movement . . . the enclosure of the intangible commons of the mind" (Boyle, 2003: 37).

Foundation. They can add to but never free-ride on the software commons if they want to make use of it to produce derivative software. Any profit motivated enterprise that wants to make use of GPL licensed software legally must build its revenue model around the commons (by whatever method devised), rather than by making a proprietary derivative and selling licenses for it. For example, by force of the GPL license, all contributions to the Linux kernel, whether from volunteers or commercial companies, have necessarily been free and open source.

Not all FS is under GPL or similar copyleft licenses that have this militant safeguarding stance against proprietary derivation, however. "Permissive" FS licenses such as the BSD (Berkeley Software Distribution) license do not stipulate that future modifications to BSD licensed code must be made free and open source as well. This is similar to releasing works to the public domain. Apple's immensely successful proprietary Mac OS X operating system, for example, is heavily based on a version of the free and open source BSD operating system released under the BSD license, which does not preclude proprietary derivatives. Integration of non-copyleft FS into proprietary software does not remove the FS portion from the commons and is therefore not an enclosure per se (such a retroactive revocation of rights of the public is impossible under copyright law, even if universal consent of all original contributors to the FS portion were secured). It does, however, represent a missed opportunity to use existing FS as a carrot (through copyleft) to draw a software developing entity such as a capitalist company to contribute to the FS commons. The lack of copyleft clauses therefore represents a non-combative stance and an attitude of "peaceful coexistence" between free and proprietary factions in the realm of the digital commons.

We can summarize the basic historical dialectic of the evolution of software as follows: It started out as a commons among a somewhat elite group of computer scientists and

hackers around university milieus predominantly in USA's Silicon Valley. Sensing the potential value of such scarce expertise, most people in this milieu took the direction of commercializing software in the market through monopolization of further contributions by means of the copyrighted, closed source, proprietary model. A minority element dissented from this trend on an ethical basis and took its own independent course, albeit while being eclipsed by the proprietary commercial movement. While the proprietary software industry boomed, the use of software necessarily became more democratic, spreading across segments of the population and geographies of the world, multiplying the number of people with various degrees of programming talent in and out of formal education, perhaps creating a surplus of non-employed (hobbyists) or partially employed (professionals who contribute on their free time) skilled labor. Some time in the mid-90's, the dissenting element of the old guard met this new influx of talent from around the world in new online communities, and passed a critical threshold, perhaps through a dynamic of increased accessibility of computer hardware, software, and internet connectivity feeding into natural over-time network effects to arrive at a tipping point. This is what gave us the exploding FS phenomenon that we know and take for granted today.

This view is compatible with Eric Raymond's intuitive statement that "the gestation period of Linux coincided with the birth of the World Wide Web" (Raymond, 1999: 51), which went mainstream around the same time. A contradiction in the nature of software production seems to have been historically unfolding in this manner. There is a contradiction between the exclusive requirement of scarce expertise in the production of software and the non-exclusive, abundant character of its distribution (through digital reproduction of code). In the long term, this character of its distribution undermines the exclusivity in its production because the product is simultaneously the means of production. As the FS tools such as

compilers, libraries, and frameworks necessary for developing new software became increasingly available on the web, further FS development became increasingly inexpensive and democratized.

The FS undercurrent re-emerged into the light in 1994, when the first version of the GNU-Linux operating system was completed by combining the GNU project led by Stallman and the operating system kernel (Linux) provided by the programmer Linus Torvalds. Both ends of the project eventually relied on the voluntary social collaboration of thousands of programmers and contributors from around the world, although they followed different organizational styles, referred to as Cathedral-like vs. Bazaar-like (Raymond, 1999). Linux is a strategic piece in the FS puzzle and the Wikipedia article on Linux adoption cites estimates of Linux accounting for up to 60% usage share in internet servers, 98% in supercomputers, 80% in mobile devices (in Android-Linux form) and 2% on desktop and laptop computers. This last segment, where Linux has traditionally been the weakest and is yet to manage a breakthrough, happens to be Microsoft's traditional stronghold.

We are now in the midst of the second decade of the 21st century, and expectations from the success of FS have grown considerably. The success of the underdog narrative of the 90's, which fascinated observers, has long given way to the casual acceptance of FS as a resilient, growing and mainstream fact of life. As Stallman has stated, the question "could we possibly develop enough software and make it free?" is like asking "could airplanes really stay up?".[7] Stallman often recites his personal history of starting, in 1983, the GNU project that gave birth to the FS movement in the following terms:

> I was just one man believing in an idea that most people would have thought was ridiculously radical. I had no political skill. Not much fame - outside of the circle of editor developers. So what could I do to change this. I didn't think I could convince governments to change their laws or convince companies to change their practices. But there was one thing I was very good at and that was developing software.

7 http://www.gnu.org/philosophy/rieti.html

> Particularly operating system software. And when I put that together, I realised I could solve this problem without convincing anybody in particular by developing another operating system that would be free. And then we could all switch to it and live in freedom.[8]

And we have come a long way riding on that ethos since then. Writing a quarter of a century later however, James Leach criticized the same individual-centric ethics he perceived as pervading the FS movement: "Ethics and politics are pursued through making material objects which then in themselves, without the need for 'social' action, achieve political ends" (Leach, 2009: 65). Perhaps the criticism from the social anthropologist's perspective is merited. Yet for decades, Stallman has not really been writing software! Instead he has been thoroughly preoccupied with FS activism: writing articles, touring the world to give speeches, trying to organize and influence policy. He hopes that if one day the world will be rid of proprietary software, he can relax and go back to coding.

The question of the feasibility of FS has given way to an anticipation of and yearning for large scale societal change along Peer Production (PP) lines—the generalization of the FS mode of production to spheres beyond software production. How far can it go, or better, how far can we take it? A fresh spirit of utopianism has taken hold of radical thinkers on the left who have set their gazes on avant-garde techno-communities and seen the promised land.

The common pool of decommodified free knowledge, the knowledge commons, is constantly growing and it seems that it will keep threatening more and more pillars of the old system. Therefore we should be ready to build an analysis on the basis of an increase of conflicts in the medium and long term. Even though different actors will try to manipulate the process in the short term to their particular advantage, and try to control or limit the outcomes, the nature of competition under capitalism (the market pressure every capitalist feels to prioritize their individual profit over the total profitability of the entirety of capital) is such

8 https://fsfe.org/freesoftware/transcripts/rms-fs-2006-03-09.en.html

that these efforts will most likely not stem the rising tide of the knowledge commons in the long run. My hypothesis is that the knowledge commons represents the coming crisis of the IP regime, and in the software realm, this has taken the form of a growing eco-system of legal, GPL licensed FS alternatives and not piracy of proprietary software. What is ultimately at stake here is the realization of the liberating potential of a universally open and accessible resource in the form of FS, towards a more egalitarian world. A decommodified software industry, free from the burden of IP that is owned by corporations (predominantly based in the rich countries) could play a serious role in decreasing inequalities and would present a positive example of globalization.

My dissertation aims to contribute to and add layers to this vision with a theoretical investigation of the political economy of FS and an ethnographic study of the members of the FS community involved in a more or less failed attempt at FS implementation by the government in Turkey. The case study is meant as a small-scale illustration of how the realization of potentials depends on institutional and political circumstances and initiatives unfolding through nodes of individuals, activist associations, educational institutions, workplaces and government branches. It aims to underline the importance of the element of agency that will have to be at work at sites of production of the commons and its political defense. We cannot place all faith in the supposed technological inevitabilities of social production to lead us into an information age utopia. That means ethnographic work should illuminate what forces encourage or discourage individuals and communities to participate as producers and users in FS, or an institution to aid developing and to choose to deploy FS for its own use.

I suggest that understanding FS projects through their inbound and outbound connections in the world will highlight their significance as a force of social change, and this

can be accomplished by "following the thing" as a practice of multi-sited ethnography that George Marcus describes. "This mode of constructing the multi-sited space of research involves tracing the circulation through different contexts of a manifestly material object of study (at least as initially conceived) such as commodities, gifts, money, works of art and intellectual property" (Marcus, 1995: 106-107). It must be noted here that since "the thing" in FS is digital, its form deterritorialized and universally accessible, to follow it is to see the use and development of code in different nodes of the global network that constitutes the FS ecosystem. Locality is produced in this ecosystem not by territoriality but by local priorities coloring the common interest in collaborative production, which is nevertheless influenced by individual, institutional and cultural-political concerns that may manifest themselves in geographical concentration.

The first part of my dissertation, which includes Chapters 2 and 3, constitutes my theoretical investigation of FS. In Chapter 2, I trace the evolution of some of the existing theoretical approaches on FS. The first steps were taken by those who are themselves FS programmers, or hackers, who have reflected on FS. They are the organic intellectuals of FS with diverse political leanings. A regularization of their insights and positions, however, was undertaken by academics, scholars and radicals who were a step removed from the immediate producers of FS. I have categorized these into three groups based on their backgrounds and orientations. The first group consists of progressive scholars who have studied FS with a focus on how the FS reality will modify markets and the prevailing IP regime. Next, I move on to literature that takes a systemic approach to analyzing the relationship between FS and capitalism. This group draws on the revolutionary traditions of Marxism, autonomism and anarchism. Finally, as the third group, I present the theorists of FS and Peer Production who have embarked on the construction of a new discipline (and even praxis) with the aim of

studying FS and PP on its own terms, synthesizing and expanding many elements found in the previous categories.

Chapter 3 is my own theoretical contribution to the study of FS as a new mode of production. I describe the unique and novel features of FS with an analysis of its political economy, highlighting its idiosyncrasies. I outline the class characteristics of software programmers and their relationship to the means of production. I theorize the existing and potential forms of remuneration found in the FS world as a series of concentric circles and critique the Copyfarleft / Peer Production License proposal. Finally I present forking as a historically novel economic concept and claim that it is the highest expression of freedom humans have achieved in the realm of production.

In the second part, which includes Chapters 4 and 5, I share a multi-sited ethnographic account of Pardus developers and the larger Turkish FS community of which they are a part. Chapter 4 is a study of this community's physical and discursive convening spaces, including the example of Istanbul Hacker Space and a detailed description of the annual Free Software Days conventions that regularly bring together the entire Turkish FS community (including Pardus developers) and related groups in a condensed public space of accelerated interaction. I thus provide a sense of the social movement aspect of FS, the nature of the FS community as an "alterpublic", and its characteristics in terms of class and outlook.

Chapter 5 is a case study of the Pardus project, the largest experiment with FS production undertaken yet in Turkey, under the aegis of the Scientific and Technological Research Council of Turkey (TÜBİTAK), which is the major government funding body for scientific research and technological development. It seeks to exemplify and complicate my theoretical characterizations by focusing on the community of various actors such as volunteers, employees, users, and academics within and around the project. It explores how

the development of the project is influenced by institutional and cultural-political concerns, and the different motivations and perceptions of individuals participating in it. The Pardus project's bell-shaped historical trajectory of success and its entanglement in Turkish politics reveal important insights into the uneasy relationship of Free Software with capitalism on both micro and nation-state scales.

PART I - THEORY

CHAPTER 2: THREE MOMENTS OF THEORETICAL DEVELOPMENT

The closing years of the last millennium were eerie times during which a number of high impact studies on the state of the world were being published. With ominous titles such as *The Information Age*[9] (Castells 1996), *Empire*[10] (Hardt and Negri, 2000) and for us academics even *The End of the World as We Know It* (Wallerstein, 1999), the books emanated towards the reader a sense of reckoning and apperception. Society was fundamentally different now, the masters of the world had been transformed into something suited to rule over this new world, and social scientists had better come up with fundamentally new approaches to catch up if we hoped to understand what was happening. Everything seemed to be blowing up towards the groundbreaking, the macro and the fundamental.

During these heady years, the major protagonists of the FS world were engaged in comparatively obscure battles, which were perceived to be cordoned off from the worries of larger society in their single-issue corner. Stallman stated for example:

> I hesitate to exaggerate the importance of this little puddle of freedom... Because the more well-known and conventional areas of working for freedom and a better society are tremendously important. I wouldn't say that free software is as important as they are. It's the responsibility I undertook, because it dropped in my lap and I saw a way I could do something about it. But, for example, to end police brutality, to end the war on drugs, to end the kinds of racism we still have, to help everyone have a comfortable life, to protect the rights of people who do abortions, to protect us from theocracy, these are tremendously important issues, far more important than what I do. I just wish I knew how to do something about them. (Williams and Stallman, 2010: 73)

9 The first volume of the *Information Age* trilogy, *The Rise of Network Society* came out in 1996, and was followed by *The Power of Identity* in 1997 and *End of Millennium* in 1998.

10 This also turned out to be the first volume of a trilogy, which later included *Multitude* (2004) and *Commonwealth* (2009).

Intellectual debate that had FS (or "open source" depending on which camp they were in) as its main concern was being carried out by the active insiders themselves, the hackers. This was done on the one hand by concentrating on specifics internal to the FS production process, such as the relative merits of an emergent organization of software production compared to tight control by a core group (Raymond's respective concepts of a Bazaar vs. a Cathedral in 1999). On the other hand, it was fought on the rather narrow platform afforded by characterizations of the so-called idealism of GNU project leader Richard Stallman versus the pragmatism of the creator of the Linux kernel, Linus Torvalds. Much of these internal debates were became institutionalized, when as a challenge against Stallman's incumbent Free Software Foundation, the Open Source Initiative was founded in 1998 by Eric Raymond and Bruce Perens with a mission to make FS more "businesses friendly".[11] Perens maintained a somewhat middle-ground position between Raymond and Stallman on whether to emphasize the ethical or technical superiority of this form of software.

It was the first decade of the 2000's in which the gap between the millenarian sociology of the intellectuals and the navel-gazing interventionism of hackers would start to be bridged. As the gale of FS solidified into a spear that pierced one sphere of digitally expressible production after the other, the subversive potentials within the new knowledge-based economy and its networked humanity could be analyzed ever more concretely. The FS principle that emerged in software penetrated first into other fully digital artifacts such as encyclopedias (Wikipedia), cultural works (Creative Commons), scientific publications (Open Access), educational material (Open Educational Resources), map-making (OpenStreetMap), and then into the digitally produced components of physical products such as electronic

11 The Free Software Movement was never "anti-business" or anti-commercial. It condoned and encouraged all business models that did not interfere with the four freedoms. This did rule out the the possibility of basing the business on software being the commodity though, which was enough to upset very large interests in the software sector.

circuitry (Open Source Hardware) and machinery and manufacturing (Open Design).[12] This proliferation warranted a term to encompass the entirety of these phenomena: commons-based peer production, or simply peer production (PP). The increased visibility of peer production also led to a deluge of scholarly interest on the field, and the first efforts towards a unifying theoretical perspective on the phenomenon. This could only be done by placing peer production within the context of the knowledge-based economy of the information society and its networked individuals.

In the following three sections of this chapter, I will trace the three moments I identify in the development of theory on FS. First, the perspectives of the liberal progressives, who hold that FS and capitalism come into partial conflict but peaceful coexistence in a reformed setting. Second, the perspective of the classical revolutionaries of anarchist, Marxist, and autonomist tendencies who can be referred to in order to explain the revolutionary potential of FS as a mode of production with the terms of industrial age revolutionary theory. Third, the transcendent synthesists who seek to explain FS as a revolutionary mode of production on its own terms, with a vocabulary that presents a sublated continuity to the reference points of the classical revolutionary perspective.

2.1 Conflict and Adaptation in the Market: The Liberal Progressives

Some of the most extensive monographs on the subject of FS and PP appeared in the mid 2000's and onward, in the works of Steven Weber and Yochai Benkler. The reason for focusing on these works is that they treat FS and PP specifically and explicitly, while also studying their wider effects and consequences through an analysis of political economy and social organization. Even though they shied away from some of the radical conclusions that

12 Such as designs, blueprints and schematics which are produced as digital files on computers.

would be drawn later by the Marxist and anarchist analysts, these works were some of the earliest systematic treatments of the issue that connected it with the larger society. They also have a complementary nature: Benkler presents a transformative vision in terms of concentrating more on identifying macro level tensions and future trends, sometimes with less scrutiny of the empirical cases than necessary, which results in confounding genuine commons-based peer production with certain forms of crowdsourcing, while Weber demonstrates an example of a working ethnography of virtual communities but sometimes shies away from taking his findings to their logical conclusion on a larger scale.

Weber's book is titled *The Success of Open Source*. He is interested in studying the political organization of communities and "large-scale collective action problems—how it is that people collaborate together over long periods of time to build things or to create institutions".[13] He carries this out by giving an account of the history of prominent FS projects like Linux and Apache, describing how their open-source development has evolved through universities, tech firms and online communities. He gives detailed descriptions of the organizational structures in place and decision making processes involved in these structures.

One of Weber's guiding concerns in his inquiry is to reject the two notions of "self-organization" and "altruism" to explain the functioning of open source projects. According to Weber, these are empty concepts that do not explain anything, but act merely as place holders for a lack of understanding. He illustrates his critique of the notion of self-organization by describing the actual decision-making processes employed in open-source projects with their own forms of meritocratic leadership, voluntary division of labor, prestige incentives, sanctions and dissent. That is the *how* part. However, I find it much more interesting to study the direction pointed at by his challenge to the simplistic notion of altruism as explanation for *why* people participate in this process. It is here in the "why" part that his study takes Weber to

13 http://globetrotter.berkeley.edu/people3/Weber/weber-con3.html

his most insightful and thought-provoking conclusions.

According to Weber, rational choice theory predicts that public goods will be systematically under-provided due to the self-centered tendency of every individual towards free-riding and a lack of incentives to act otherwise (Weber, 2004). This problem is often referred to as "Tragedy of The Commons" after the title of an article by biologist Garret Hardin (Hardin, 1968). The political conclusions drawn from this premise usually come out in favor of privatization and installation of appropriate market mechanisms to create incentives, the other alternatives being stringent state regulation of resources or ruin. FS as a pure public good appears to defy this predicament. Here we have a body of public goods which are both non-exclusive, and voluntarily produced (initially completely, and to a significant extent still). In fact, FS dispenses with the whole notion of property as the right to exclude, in favor of property as the right to distribute (Weber, 16). This means anyone can take, while no one has to give. While the introduction of the concept of altruism into political economy would be one explanation for why this arrangement is sustainable, Weber is intuitively unsatisfied with it and digs further. He arrives at an interesting finding in doing so, and this brings him into dialogue with Benkler and his 2006 work titled *The Wealth of Networks*.

Both Weber and Benkler point out that products of the knowledge economy such as FS are non-rival goods. One person's use of a piece of software, for example, does not result in another person being deprived of it, because digital technology allows the infinite reproduction of the good at infinitesimal cost, rendering scarcity economics on the demand side obsolete. But this still does not explain the supply side of things; even if taking does not detract from the common pool available for others, the temptation to contribute nothing and simply free-ride on the work of others remains.

Benkler correctly identifies the underlying infrastructural pre-requisite for social

production in the movement from industrial-information society to information society proper: "The core technologically contingent fact that enables social relations to become a salient modality of production in the networked information economy is that all the inputs necessary to effective productive activity are under the control of individual users" (Benkler, 2006: 99). Benkler is basically referring here to the widespread availability of personal computers and internet connectivity to individuals at low cost. He overlays this contingency on top of the "deeply ingrained" cultural notion of sharing that humans possess pretty much inherently—which possibly diverges from the notion of altruism towards a form of Kropotkinian evolutionary mutual aid—to explain the emergence of FS and other (collaborative) social production (ibid: 120). However, in addition to his issue with the notion of altruism, Weber also points out that while the technological material basis for producing free public goods may be almost universally in place, high quality programming skills are still scarce, and that this is the crucial scarce ingredient rather than computing resources when it comes to software (Weber, 2004: 150).

Weber's key to understanding why the free-rider problem does not seem to emerge in FS is the realization that in fact every new user of a piece of FS makes the product more useful and valuable for everyone. In this fact lies the important difference between a consumer of material products and a user of software. Raymond formulates this in the following statement: "Users are wonderful things to have, and not just because they demonstrate that you're serving a need, that you've done something right. Properly cultivated, they can become co-developers" (Raymond, 2001: 26). Even if they do not program and write patches, more users mean compatibility and ubiquity, which are essential to the viability of a software platform, feedback in terms of bug reports and feature requests, and an overall increase in the sense of worth and synergy that makes the product more useful and developers

more motivated as more people use it. The free-rider turns out to be in practice not so much of a free-rider after all (ibid: 154-58). Such a phenomenon was observed in the context of certain traditional commons such as dancing festivities by Carol Rose:

> The more persons who participate in a dance, the higher its value to each participant. Each added dancer brings new opportunities to vary partners and share the excitement. . . . Activities of this sort may have value precisely because they reinforce the solidarity and fellow-feeling of the whole community; thus the more members of the community who participate, even if only as observers, the better for all. (Rose, 1986: 767-68)

Weber's analyses lead him to conclude that beyond being a public good, FS becomes a "network good" that leverages positive externalities in a self-reinforcing fashion—a process also referred to as the "network effect". Here lies the rationality of "giving away" according to Weber, because giving automatically results in coming into possession of something better. For software that leverages the network effect, which is most software today, each increase in popularity increases the potential usefulness for each user linearly, while increasing the potential usefulness of the system as a whole quasi-exponentially.[14] To quote Rose again: "We might even think that properties devoted to such noncommercial uses as recreation or speech could achieve their highest value when they are accessible to the public at large" (ibid: 723) and "this activity is exponentially enhanced by greater participation" (ibid: 769). Contra Hardin's "tragedy of the commons", Rose dubbed this the "comedy of the commons". The FS commons is thus comedic.

Let us take Weber's finding that the free-rider is not a meaningful category in the setting of FS and accept that everyone in the network contributes something, no matter how big or small, to the whole, and see it in the context of Benkler's observation that "By lowering the capital costs required for effective individual action, these technologies have allowed

14 "The number of unique connections in a network of a number of nodes (n) can be expressed mathematically as the triangular number n(n − 1)/2, which is proportional to n2 asymptotically". https://en.wikipedia.org/wiki/Metcalfe's_law

various provisioning problems to be structured in forms amenable to decentralized production based on social relations, rather than through markets or hierarchies" (Benkler, 2006: 121). If we also keep in mind that the information product is infinitely reproducible at near-zero cost, in other words, that abundance of the finished product is guaranteed, we can seriously pose the following question: is this a political economy, operating in a context of abundance of outputs, that is approximating the principle of "from each according to ability, to each according to need"? How is this potential affected by internal factors like the relative scarcity of inputs (programming skills) and the external context of generalized capitalist market relations that in certain ways can seem to be harnessing FS?

The possible relevance of the ability-needs paradigm is not spelled out and explored in either Benkler or Weber, both of whom express that they operate within a liberal framework, and instead discuss whether the concept of the gift economy applies to FS. Weber rejects the idea that FS can be seen as a gift economy because he assumes that a gift economy would require abundance of all inputs, meaning programming skills, which are in reality scarce (Weber, 2004: 150-151). Benkler makes a reference to the notion of the gift in anthropology to point out a viable example of a non-market, non-hierarchical transaction to support his concept of social exchange (Benkler, 2006: 109). Benkler's point is legitimate, but it conceals crucial differences between the gift economy and FS.

Most cases of gift giving imply reciprocity between more or less equal counterparts, even if it is not of a one-to-one, immediate kind, and are often carried out in a ritualistic form. There is no tangible expectation of reciprocity in FS in the non-ritualistic exchange between producers and users and no obligations to give back. The notion of gift would perhaps be applicable if the phenomenon were limited to exchanges between software programmers only. Indeed, Eric Raymond puts forth this thesis in "Homesteading the Noosphere", claiming that

open source is a gift culture where participants compete for prestige by giving away time, energy and creativity (Raymond, 2001), and others seem to echo him to various extents. Bruce Perens, for example, suggests that this was true only of the early open source movement (Perens, 2005), an observation that I intuitively agree with. The fact is, however, that today there is an infinitely larger group of people who benefit from FS—that is, users, whose per-individual contribution to the creation of FS is infinitely smaller than that of the programmers. Their combined contribution, as Weber explained, is considerable, but then it makes no sense to continue using the analytical category of the gift to denote such an asymmetric and unaccountable form of collectivity. If FS programmers were mostly producing gifts for fellow FS programmers, one would expect an attitude of disdain towards free-riding lay users to be expressed in the interactions between producers and users. Nowadays, such an elitist attitude survives as a rapidly vanishing hangover, but is marginal and frowned upon by the majority of developers. In fact, a big part of FS evangelism is getting people to use it despite other seductive or convenient proprietary alternatives. Finally, in gift economies, the gift giver is parted with the scarce object of value upon giving the gift. This is not the case at all in digital reproduction. My view is that in FS, there is a process of generalized exchange, which the ability-need paradigm is much more adequate in explaining, compared to the notion of the gift.

The question then is why both Weber, who rejects the notion of the gift in regard to FS, and Benkler, who generally accepts it, feel that the political economy of the gift is at least relevant enough to be worth addressing, while they ignore the ability-need paradigm. There seem to be two factors that are at work here.

The first is the ideological baggage of "actually existing socialism". When we hear "communism", we think State control, hierarchy and central planning. These have been the

principal features of state socialism and policies of the Communist parties that dominated the communist movement in the 20th century. Needless to say, they are completely alien to the FS processes. However, the state socialist route to communism was only one among other rival theoretical paths to communism, such as libertarian (anarchist) communism, with its emphasis on decentralization, voluntary association, and adversity towards hierarchy. On the flip side, the cliches of right wing libertarianism that equates the free association of producers and lack of central command with the free market tend to confound our understanding. Nothing could be more illustrative of this point than Raymond's quick jump from quoting Kropotkin, the father of anarchist communism, on the "principle of common understanding" that leads to "the convergence of many wills" (the principle of free association of producers) to the conclusion that the Linux world is "a free market, or an ecology of selfish agents attempting to maximize utility . . . which produces a self-correcting spontaneous order more elaborate and efficient than any amount of central planning could have achieved". This erratic move comes about by the employment of the bizarre notion of altruism as ego-boosting through the back door (Raymond, 2001: 22), a favorite formulation of right-wing, cynical political thought.[15]

There is quite an amount of confused thinking and lack of depth with regard to political theory being demonstrated by actors in the field, as the following Weber quote demonstrates:

> But as open source has begun over the last several years to attract more public attention . . . it has become a kind of Internet era Rorschach test. People often see in the open source movement the politics that they would like to see—a libertarian reverie, a perfect meritocracy, a utopian gift culture . . . a political movement aimed at replacing obsolete nineteenth-century capitalist structures with new 'relations of production' more suited to the Information Age. (Weber, 2004: 7)

I propose however that we take Weber's accurate depiction of the field as a point of departure instead of as a dismissive conclusion. The proliferation of such claims should hint at

15 I take here the liberty to point out from years of personal observation of discussions around FS that take place on prominent pro-FS avenues such as Slashdot.org, that such thinking is widespread.

the inadequacy of our existing political and ideological frameworks to describe the nature of FS as an economic and organizational model, rather than being reason for dismissal (The Oekonux project, which I will present in the last section of this chapter, captures this brilliantly). We have already seen the difficulties that emerge when trying to work with concepts and categories derived from the existing theories of political economy, such as altruism, the free-rider, and the gift. On this issue, Benkler seems to point towards a useful perspective when he suggests that "freedom and justice can and should best be achieved by a combination of market action and private, voluntary (not to say charitable) non-market action, and the state is a relatively suspect actor" (Benkler, 2006: 21). Nevertheless, he is quick to insert a disclaimer against ideological commitment and "fashionable" anarchism, and suggests that he could envision a liberal state playing a constructive role "if it stopped listening to (industry) incumbents" that have so far lobbied the state towards restricting the knowledge commons tendency (ibid). Both Weber and Benkler see themselves as operating within a liberal framework, although one can add that they are pushing this framework towards the left with their positive conception of freedom, that is, the freedom that empowers one to do more. The same Weber who is critical of utopian bombast regarding FS for example, does not refrain from mentioning that "'Bit-twiddlers' are neither exactly proletariat nor bourgeoisie. They may not own the means of production in the sense that Marx argued, but they certainly do have significant control over those means, in a more profound way than the term 'knowledge workers' captures" (Weber, 2004: 247). This kind of wobbling in political analysis demonstrated by both Benkler and Weber should again be seen as a sign that the FS phenomenon is pushing the boundaries of economic structures we currently have in place, but we did not have a clear, systematically articulated theoretical grasp on what FS fully entails in the mid 2000's. This would only truly begin to materialize in the beginning of the 2010's with

the conclusion of the Oekonux Project and the maturation of the P2P Foundation, which I discuss later.

The second factor that seems to offset the relevance of the ability-need paradigm is the limited sphere of activity that FS represents in the information economy, not to say in the economy as a whole. Whether a circumscribed gift economy, an infrastructural public sector (Perens, 2005), or limited communistic sphere of relations, the knowledge commons currently operates within a sea of market-based initiatives, even as it is constantly expanding, and it interacts with the market dynamics in complex and sometimes ambivalent and confusing ways. In the absence of generalized decommodification, it is perhaps difficult to see the nascent mode of production breaking out of the space it has carved out for itself. On this point, it becomes very important to see future trends, the relationship between FS and the information industry that surrounds it,[16] how FS is influenced by it and how FS transforms it.

2.2 A New Mode of Production: The Classical Revolutionaries

Kevin Carson is one of the leading living representatives of the mutualist school of anarchism. Classical 19th century mutualism, the economic philosophy of Proudhon, advocated a non-exploitative market economy based on a combination of personal and co-operative forms of property ownership. Arguably, as large scale industry prevailed over craft based production, the grounds upon which mutualist strategy was conceived became increasingly untenable. The half workerist, half petite-bourgeois overture gave way to Bakunin's collectivism and later Kropotkin's anarchist-communism within the anarchist camp of the labor movement, not to mention being gravely wounded by Marx's critique in *The Poverty of Philosophy* (Marx, 1847/1992). The mutualist position remained relatively obscure

16 See the Appendix for information on the size of the global software market and stock of IP.

throughout the 20th century. It seems however to be making a comeback in today's world of supposed decentralization, informationalism, and peer production networks. In two major works, *Organization Theory* (2008), and *The Homebrew Industrial Revolution* (2010), Carson attempts to revitalize mutualism for the 21st century.

Echoing Castells's thesis, he argues that "in the information age", "the value of the production process is becoming increasingly embodied in the intellectual skills of the worker". He perceives contemporary economy as one which has "far lower levels of investment required to enter the market" and the era of "the second industrial revolution" with its "very asset intensive and highly vertically integrated" model that aims to "exploit economies of scale and scope" as being on the way out (Carson, 2008: 325).

Carson sees IP as "a tollgate to prevent existing technical knowledge from being built and improved on" and a barrier to progress (as it blocks the free flow of information) rather than a spur to progress (ibid: 326). An instance of the Marxist notion of relations of production becoming a "fetter" on the development of the forces of production can be felt here. He points out that an increasing portion of a product's final price is made up of tributes to owners of IP, that "more and more intellect and less and less materials" are being sold. Carson is fully on board with a switch to "an economy where software and product design were the product of peer networks, unrestricted by the 'intellectual property' of old corporate dinosaurs" as the solution. He anticipates that such a switch, combined with the elimination of marketing expenses as lean demand-pull distribution replaces mass supply-push distribution (ibid: 547), would cause a precipitous fall in prices. This fall in prices could translate to vastly increased free time for laborers (ibid: 327), who could reorganize production in "groups with extremely porous boundaries that will constantly federate and divide for specific projects" (ibid: 331).

In a nod to the autonomist emphasis on exodus, Carson cites Johan Söderberg's claim that "the current model of outsourcing and networked production makes capital vulnerable to being cut out of the production process by labor" and that "the rise of peer production in the informational realm" is reason to hope that "independent and self-managed networks of laborers can route around capital" (ibid: 328). Carson also envisages that piracy of digital artifacts will play a significant role in this, expecting that IP itself may become "unenforceable in an age of encryption and bittorrent" (ibid: 330). He identifies Linux and other open source products as providing a new paradigm for the management of labor and is extremely bullish on its future: "(it) will soon grow to become the principal model of production in the principal industry of the leading economies of the planet" (ibid: 333). I agree with Carson on the point that the emergence of FS as a model not in a marginal but in a "principal" sphere of production in developed economies is key to its potential. Likewise on his observation that "in any such industry, where the basic production equipment is affordable to all, and bottom-up networking renders management obsolete, it is likely that self-managed, cooperative production will replace the old managerial hierarchies (ibid: 333). I would add, however, that the chances of success for such self-managed and cooperative forms of production are increased in cases such as FS, where the product can be distributed for free rather than as commodities that have to compete with commodities produced by traditional capitalists on the market. This abandoning of market exchange requires, of course, alternative forms of compensation for this production to take hold, which I will thoroughly explore in the next chapter.

Söderberg takes the idea of FS-style production becoming dominant in principle industries further, claiming that not only will this be limited to certain high-stake industries like IT, it will permeate all other industries as well. Söderberg points to Marx to support his

case:

> The characteristics of the information sector will gradually encompass most of the economy. This tendency was essential in Marx's analysis. "In all forms of society there is one specific kind of production which predominates over the rest, whose relations thus assign rank and influence to the others. It is a general illumination which bathes all the other colours and modifies their particularity". (Söderberg, 2002: 8)

He then mobilizes Hardt and Negri: "Just as the processes of industrialization transformed agriculture and made it more productive, so too the informational revolution will transform industry by redefining and rejuvenating manufacturing processes"(ibid: 8).

This brings us to the question of how the wide scale transition to a new mode of production can take place. For classical Marxists, the theory of transition from capitalism to socialism has always been constructed with reference to a "scientific" analysis born out of the laboratory of history: the transition from feudalism to capitalism. The historical materialist reading of the transition from feudalism to capitalism holds that as feudal relations of production started to become fetters to the development of the forces of production, the age of bourgeois revolution began, boosted of course by massive primitive accumulation: enclosures at home and colonialism abroad. The accumulation of riches on one hand and free laborers on the other gave rise to the possibility of concentrated production (as opposed to the "pigmy-like" scattered production) based on the wage relation, which ushered in an unending cycle of self-expansion of value in production. The masters of this new relation were the bourgeoisie, who conquered first the economic kingdom, making the lord-serf relation appear hopelessly archaic and inefficient, and eventually managed to project this power on to the political realm through multiple bourgeois revolutions. One class society was replaced by another. Now, to take this historical transition as analogy for a transition from a class society to classless society (from capitalism to socialism) required one crucial modification. If the general schema of the laws of accumulation in Marx's *Capital* are true, the proletariat under capitalism, unlike

the nascent bourgeoisie under late feudalism, cannot under the normal functioning of the system build up economic hegemony prior to conquering the political structure. The proletarian position to the contrary is one of accelerating downward movement towards ever more economic polarization (relative impoverishment) vis a vis the bourgeoisie. While the proletarian economic situation tends towards relative deterioration, its numbers (through the decline of the middle strata of society, i.e. the artisanry and peasantry) grow ever larger. Provided that a consciousness of this movement arises, the proletariat can use its numbers first to conquer the political kingdom (through violent revolution or democracy), and then move to transform the economy. The proletarian revolution seems destined to follow the course of the bourgeois revolution in inverted form: political power first, economic power second.

Note that this entire theoretical-political edifice is built on the recognition that the concentrated, centralized production enabled by capitalist ownership and mass wage labor is objectively superior (despite being terribly exploitative), that is more efficient, than the distributed small-scale production of the peasant and artisan that preceded it. It is the primacy of "the development of the forces of production" in Marx's theory of history that renders the unjust, obscene and base capitalist development ultimately progressive. The next historical task is not to reverse, but to maintain (and for most Marxist-Leninists, to accelerate!) this concentration of the means of production while changing the relations of production, i.e. bringing them under social rather than private ownership.

But what if this industrial era picture is fading? What if, "the development of the forces of production" is taking on a new meaning in today's era of digital artifacts and peer production? Söderberg argues that "the universally applicable computers run on free software and connected to an open network... have in some respects leveled the playing field. Through the global communication network, hackers are matching the coordinating and logistic

capabilities of state and capital" (Söderberg, 2008: 2). Söderberg comes off here as rather modest. There are reasons to think that not only is peer based FS production matching capitalist software production, but besting it. One of these reasons is less technical and more existential. Linus Torvalds for example comments that,

> Proprietary software systems are bad because the people don't care. To a hired programmer, the code he is writing is a means to get a pay check at the end of the month. Any shortcut when getting to the end of the month will do. For a hacker, on the other hand, writing code is an end in itself. He will always pay full attention to his endeavour, or else he will be doing something else. (Quoted in Söderberg, 2008: 26)

If FS has come about as a result of a situation where the distributed availability of certain means of production and the autonomous application of novel organizational methods result in a superior way of production, to the extent that this is generalizable to society, the classical Marxist transition strategy needs to be be reconsidered. A footpath to classless society seems to have opened that is indeed based on the achievement of economic power by a new, relatively autonomous class of producers first. The political fight can now be conceived as coming second. The analogy to the transition from feudalism to capitalism can be reproduced in direct, rather than inverted order. Carson's concept of the "pseudomorph",[17] or the old anarcho-syndicalist adage of "building the new society in the shell of the old" suddenly finds new applicability. Certainly, there have always been streaks within the labor movement that focused on building cooperatives, solidarity networks and various economic counter-institutions throughout the industrial era. These have existed in both social democratic and libertarian wings of the movement, not to mention the labor unions. What is new, however, is the historic opening that shifts the terrain of class struggle in a favorable direction. If FS turns out to be the vanguard expression of a generalized paradigm shift, the transition to classless society (which is no longer socialistic, but directly communistic) may

17 "In mineralogy, a pseudomorph is a mineral or mineral compound that appears in an atypical form (crystal system), resulting from a substitution process in which the appearance and dimensions remain constant, but the original mineral is replaced by another." https://en.wikipedia.org/wiki/Pseudomorph

suddenly be much more straightforward, and therefore easier. Columbia law professor and chairman of the Software Freedom Law Center Eben Moglen summarizes this hope:

> We now find ourselves in a set of property relations that have become obsolete. The Free Software Movement is at its core a struggle for freedom in this new setting, and not merely a method for production. Running free code is a revolution, one that does not need to go through the violent phases of the guillotine, the commune, and the suppression of the commune. This revolution does not need to seize control of the means of production other than our own heads, for us to become sharing producers. The network society makes this possible. With free software, free hardware, free culture and free spectrum (communication), we can build a society of justice, equality, and liberty. (Moglen, 2004)

Hardt and Negri harbor a similar if somewhat more implicit vision in *Empire.* To bring the parallel to the fore, we need to consider their theory of the contemporary world. Their understanding of the contemporary world dubbed 'Empire' operates on a "third economic paradigm", paradigms being defined by the dominant sectors of the economy (from agriculture and extraction to industry and manufacture to services and information). The current third paradigm has been reached by the informatization of production, where productive labor tends to be increasingly immaterial, its product a service, a cultural item, knowledge, or communication. This shift is not merely quantitative (employment figures according to sectors) but qualitative, in the sense that the latter paradigm transforms and redefines the prior sectors (Hardt and Negri, 2000).

Hardt and Negri's thinking therefore lends itself to a rather diffused conceptualization of *exodus* as class struggle, which is applicable to all "immaterial labor". The producers of value, whom they refer to as the *multitude*, have merely to cast off the increasingly redundant tentacles of capital in order to reaffirm their autonomous status in production. What form this move is to take (cooperatives? migration? revolution?) is not specified.

Compared to the visibly delineated autonomy of the FS producers, other immaterial laborers who labor in symbolic analysis, problem solving and the production and

manipulation of affects (knowledge and service workers) hold a position that is more "inside and against capital" rather than outside capital. The primary production of surplus value is shifting from the mass factory labor to intellectual, immaterial, communicative labor which is immediately social. Cooperation is not organized externally—by the capitalist in the factory for example—but is immanent to the laboring activity itself. Contemporary capitalism increasingly becomes a system of producing social relations and forms of life (biopolitical production). The distinction between productive and reproductive labor is being blurred in favor of what the Italian workerists in the 70's called the *social factory,* where abstract social labor can no longer be measured by labor time, because living and producing has started to become indistinguishable (Hardt and Negri, 2004). They sum up this notion in *Commonwealth*, stating that in a contradictory manner, real subsumption of labor under capital is taken to the extreme, while this real subsumption is becoming less and less organic (Hardt and Negri, 2009). The importance of wresting constant capital away from the capitalist class fades, as variable capital is the primary factor of production.

There are, however, a few points in Hardt and Negri's trilogy where the connections to FS and peer production are rather pronounced. First, they point out that the computer increasingly appears as the universal tool, abstracting all labor homogeneously so that all concrete labor tends towards the manipulation of symbols and information (in front of a computer). Second, they point out that software and internet application production demonstrates an example of the regime of capital (copyrights) hindering the commons-based production. They hold that to further the multitude's ability to produce the common, we need an open infrastructure of information and culture in the form of open access to wired and wireless communication, open software code and protocols, and open content of cultural, intellectual and scientific works. These could counter immaterial property forms like

copyrights and patents in a post-scarcity world. Such a political program is of course another exposition of the attractiveness of the FS path to change, where not even the abolishing of IP law is necessary to effect radical change. The multitude does not need to overthrow the regime and seize the means of production; it merely needs to keep its channels of production open, improve its position of primacy over capital and continue to drift away—just as how Stallman started out by deciding to *produce* the ethical alternative to proprietary software rather than trying to change the laws or behavior of corporations.

2.3 A Revolution of Sublation: The Transcendent Synthesists

There are broadly two schools that have emerged which take as their starting point (rather than a possible conclusion) the view that FS is a new mode of production that has the potential to revolutionize society and seek to analyze it on its own terms rather than with reference to prior modes of production and theories that sought to explain them.

The first of these is the grouping around Project Oekonux,[18] a compression of the words "Oekonomie" and "linux", which was active between 1999 to 2012, and which currently exists as an archive. It brought together a lively community of unorthodox radical thinkers on the question of FS and society through conferences, its wiki, mail list and journal *Critical Studies in Peer Production*. The second school is the P2P Foundation (Foundation for Peer to Peer Alternatives)[19] which was formed in 2005. The P2P foundation web site serves as a major hub for all kinds of information on FS, practical P2P initiatives, projects and articles. It also puts out the *Journal of Peer Production*. There is considerable overlap between Oekonux and the P2P Foundation in terms of points of interest, contributors, and authors, and in most ways the P2P Foundation can be seen as continuing the work of the avant-garde

18 http://www.oekonux.org/
19 http://p2pfoundation.net/Main_Page

Oekonux project on a larger scale and in a way that is more accessible. I would like however to underline a very important difference in tendency between the two initiatives, enough to merit in fact the current (post-2012) orientation of the P2P Foundation as a *turn*. I will do this with reference to the two leading figures respectively, Stefan Meretz and Michel Bauwens.

At the tail end of the active existence of the Oekonux project, Meretz published a seminal text titled "Ten Patterns Developed by the Oekonux Project", which summarizes ten basic theses on FS as a mode of production that have been discussed and developed over the course of the project's life.

> The Oekonux project seeks to establish a new basis for analyzing a new historical phenomenon: the emergence of peer production, starting with the creation of Free Software. If the initial hypotheses of Free Software being the germ form of a new mode of production beyond capitalism is valid, it would be necessary to develop new epistemological patterns to be able to analyze it adequately. (Meretz, 2012)

My engagement with FS is very much in the same spirit, and I will present my theoretical contribution to this in the next chapter. The ten "patterns" listed by Meretz contain the following: Pattern 1, "Beyond Exchange", holds that the process of collaboration in FS is not mediated by exchange, not even a socialist exchange (based on value equivalence without exploitation) or gift exchange. Instead we see voluntary contributions towards a common goal, the ability side of the ability-needs paradigm of communism. Meretz rejects exchange on principle, which he sees as inevitably leading to capitalism. He also sees all past and possible future attempts to build socialism doomed to failure. I see this very first principle as the major point of contention between Meretz and Bauwens, whose program is to introduce the principle of mutuality to peer production as an interface between the spheres of peer production and capitalist production. I side with Meretz against Bauwens on this debate and explore the alternatives to Bauwens's position when it comes to peer production in the next chapter, while refraining from commenting on the viability of socialism in general.

Pattern 2 is "Beyond Scarcity". Here Meretz wishes to dissolve what he sees as the false dichotomy between analog and digital goods on the point of scarcity, claiming that abundance does not have to be limited to the realm of digital goods, that it can also be achieved in the realm of analog goods if the commodity form is abolished. On the flip side, he claims that even digital goods rely on the existence of material goods (computers, infrastructure etc.) which can be seen as scarce. While I see the merits and agree with the general spirit of the argument, I believe that this hasty formulation blurs actual reality by confusing two distinct senses of abundance. Instead I offer in the next chapter the concepts of relative and absolute abundance to address the point.[20]

Pattern 3, "Beyond Commodity", asserts that FS is not a commodity, and that the organization of its production is in the form of a self-management of common resources as per Elinor Ostrom's dictum of "Neither state, nor market".

Pattern 4, "Beyond Money" is a corollary of the previous "Beyond Commodity". Meretz correctly points out here that the existence of "money around free software" is a result of capitalists adapting to the fact of FS, and not the other way around: "Why did IBM put one billion dollars into Free Software? Because they were forced to do so. Economically speaking they have to devalue one business area to save the other profit-making areas".

Pattern 5 or "Beyond Labor" underlines the fact that FS production is often an expression of an individual's self-realization rather than a process of alienated labor. I hold this to be true in the ideal case of the autonomous FS *hacker*; however, the principle is muted to various extents in the case of FS programmers, who are in precisely the positions mentioned in Pattern 4—producing FS for a wage at a capitalist corporation or other

20 In any case, I hold that providing universal computing access to every last person on earth (which would guarantee abundant access to all free digital goods) is orders of magnitude easier than doing the same to every kind of physical good imaginable. In fact the prior is mostly being accomplished even under capitalism!

commanding body (State, NGO, etc.). I explore the trajectory of intrinsic and extrinsic autonomy in FS production in the next chapter in my concentric circles model of remuneration.

Pattern 6, "Beyond Classes" posits that there is no class relation among individuals collaborating in a FS project. This is true, as there is no alienable surplus production and hence no exploitation relation. The more interesting question, however, is the class position of software programmers (if there is such a unitary position) within general society, at the actually existing transitional stage in which FS operates. I will discuss this "new class" in the next chapter as well.

Pattern 7, or "Beyond Exclusion", extolls the virtue of universal inclusiveness in FS and an end to divisions constantly reproduced by capitalism like "job-owner vs. jobless, rich vs. poor, men vs. women, people of color vs. white people, bosses vs. subordinated, owners of means of production vs. non-owners, members of social security vs. non-members etc.". Here Meretz veers a little too much towards a reductionist optimism as far as the "intersectional" injustices such as gender and racial inequality under capitalism are concerned. In Chapters 4 and 5, I will briefly share my observations from my case study that revealed that the FS community I studied has achieved very little towards inclusiveness of women compared to the regular software industry *in practice*.[21] Meretz also mentions in relation to Pattern 7 that forking, which is a parting of ways for FS developers who have fundamental disagreements about the direction a piece of software should be developed, is actually a mechanism that guarantees that the effect of exclusion is minimized in FS. This is because exclusion in FS can only mean exclusion from future collaboration, and not from future development of code altogether. This is an important observation, and I will expand upon the crucial role of forking in guaranteeing liberty in production in the next chapter.

21 This unfortunately appears to be the norm almost universally.

Pattern 8, "Beyond Socialism", builds on Pattern 1 in rejecting the need for or desirability of a socialist transitional stage, and argues for a re-ordering of the revolutionary strategy in favor of economic empowerment first, political empowerment second. I have discussed the implications of this position commonly held by politicized FS enthusiasts earlier in this chapter. I hold that this is a seductive proposition but the issue is as of yet an open question. I hope that my study of the Pardus case, which is a mix of autonomous economic production and politically influenced institutional production, can inform this discussion from one specific context.

Pattern 9, "Beyond Politics" develops the theme of universal inclusiveness in pattern 7 in order to argue that the fading of partial interests in production will lead to a fading of politics as the political expression of these interests. My reservations about the "necessary yet not necessarily sufficient" nature of this claim applies.

Finally, Meretz's pattern 10, "The Germ Form" which is the only pattern expressed in affirmative grammar, charts out a trajectory of the increasing power of the FS mode of production as a growing power from the past to the future. The germ form theory promises a dialectical approach to the progression of peer production:

> When discussing peer production the debate is often dominated by two groups: those who are in favor of peer production and who try to prove peer production is anti-capitalist and those who see peer production only as a modernization of capitalism. The challenge is to think it as both. The germ form model accomplishes this by viewing the emergence and development of commons-based peer production as a process of its own contradictory unfolding in time. (Ibid)

The initial phase of the trajectory is the appearance of the germ form. At this initial stage, "commons-based peer production is not the new itself, but the qualitatively new aspect it shows is the need-oriented mediation between peers"(ibid). In the second stage, the old system enters a crisis, which allows the germ form to break out of its niche and develop further because of an enlarged space for alternatives. The third stage is "function shift", when

the old system increasingly starts to depend on the germ form for its own existence: "Peer production is usable for purposes of cost-saving and creating new environments for commercial activities, but it rests upon non-commodity development within its own activities" (ibid). This is the stage that we are currently experiencing according to Meretz and it is also the most confusing, because the same phenomenon appears to be serving opposite ends depending on one's perspective. I concur with Meretz that this is the source of much of the confusion in studies of FS. Contributing to the dispersion of this confusion is a major task I set out to achieve with this dissertation. In a future fourth phase according to Meretz, we will observe a "dominance shift", where the relative positions of peer production and capitalism shift around, and it is commodity production that is marginal rather than commons production. A final fifth stage is then expected to come about, "restructuring", when the antagonism is resolved in favor of the new form, and it will be time for currently unforeseeable new contradictions to emerge, initiating a new round of social development.

The main priority of the Oekonux project as represented by Meretz seemed to be to develop a dialectical analysis that could trace the unfolding change and future direction of the FS and peer production phenomena. While the P2P Foundation builds on the Oekonux foundation in many ways, it exhibits a more pronounced desire to seek immediate implementations of instances of peer production, and the development of theory in the P2P Foundation has veered towards reflecting this priority: "We need to move from empty and ineffective anti-capitalist rhetoric, to constructive post-capitalist construction" (Bauwens, 2007). This shift of priority can be seen in the intellectual trajectory of the leading figure of the P2P Foundation, Michel Bauwens over the years. In "The Political Economy of Peer Production", for example, he held that "P2P follows the adage: each contributes according to his capacities and willingness, and each takes according to his needs" and that "With P2P,

people voluntarily and cooperatively construct a commons according to the communist principle" (Bauwens, 2005). Much like the germ form theory put forth by Meretz, he expected that "historically, though forces of higher productivity may be temporarily embedded in the old productive system, they ultimately lead to deep upheavals and reconstitutions of the political economy" (ibid). The source of the divergence is felt, however, in what I see as a hasty desire to expand FS-like production into the sphere of material production.

The generalization of the (currently nascent, circumscribed, or proto-) FS mode of production is of course the ultimate endgame of revolution-by-FS, and the holy grail is resolution of the material production issue. Bauwens, already in 2005, presented a number of alternative directions for the transition. I have put these alternatives into three groups in order to bring to the fore their different spirits.

1. The Path of Immanence: "P2P can arise wherever the process of design may be separated from the process of physical production".
This is an important insight because it underlines the commonalities between a *component* of physical production and FS-production: the design stage of certain physical production is actually digital production. Thus it can take place on a collaborative network of personal computers in cases where final production is automated by machinery, such as CAD-CAM and 3D Printing.

2. The Path of Cooperatives: "P2P can arise... wherever there is access to distributed technology" and "wherever other forms of distributed fixed capital are available".
The problem with this proposition is that it glosses over the fact that not any distributed technology will do. The distributed technology must constitute at least equally advanced

means of production as what the capitalists can command if its products are to challenge the products of the capitalists on the market. This is a rare thing, and one of the outstanding features of software production. More on this in the next chapter.

"P2P can arise wherever financial capital can be distributed. Initiatives such as the ZOPA bank[22] point in that direction. Cooperative purchase and use of large capital goods are a possibility."

This is the least inspired, if not most backward-facing suggestion in Bauwens's list. The idea of many independent laborers pooling together their resources in order to invest in a cooperative venture, and funding it through a form of mutual banking is as old as Proudhon, if not as old as capitalism, and suffers from the same utopianism. The strategy may have its place, but it does not explain why cooperatives and mutual banks have not taken over the economy in the last two centuries yet.

3. The Path of Policy: "State support and funding of open source development"

State support and funding of open source projects is quite a different matter, because the state is in the unique position to tax capitalists to generate funds, as opposed to having to compete with them on the market. The political and institutional viability of this strategy will of course vary greatly depending on political circumstances, and the Pardus case presents one example of this.

"P2P could be expanded and sustained through the introduction of universal basic income."

22 ZOPA (Zone of Possible Agreement) is a "Peer 2 Peer Finance Assosciation" that connects lenders to borrowers. Contrary to its claims of "not being a bank", it appears to be just that. It is not even a non-profit entity. It pays interest to lenders and demands interest from borrowers on differential rates (a classic source of banking profit). It just dispenses with brick-and-mortar branches in favor of a web-based platform, and offers less risk protection to lenders than traditional banks. Its combination of less risk-averseness and leaner operation with P2P-hype based marketing is in line with neoliberalism more than anything else.

While even further tied to the constraints of the political climate than regular state funding, this is a deeply radical suggestion that I am in favor of. I will address why this should be a serious consideration for policy advocacy for FS and peer production supporters in the section on forms of remuneration in the next chapter.

Bauwens more or less reproduced this three-pronged approach to furthering peer production in his 2007 "Peer to Peer Manifesto", envisioning a non-reciprocal peer-production core for immaterial production, a reformed market economy with "true-cost accounting" (the internalization of negative externalities of production) for materials production, and "a universal subsidy to all" on the level of the state.

Bauwens published another major article titled "From the Theory of Peer Production to the Production of Peer Production Theory" in 2012, however, and it exposed a somewhat more narrowed suggestion for the way forward for peer production, which unfortunately seems to have been a result of a few important theoretical regressions. Here Bauwens presents the position of the P2P Foundation as an advanced synthesis of four perspectives: the Benklerite progressive-liberal approach, the Oekonux germ form approach, Dmytri Kleiner's "mutualist and self-styled 'venture communist'" approach, and Massimo De Angelis and George Caffentzis's anti-capitalist commons approach. Bauwens's synthesis takes the following form. First, he takes further the Benklerite position that peer production exists as mere addition to capitalist production in certain areas, to state that there is a possibility that peer production can succeed capitalist production altogether. He basically gives the same treatment to the Oekonux position but in reverse, pointing out that while peer production has the potential for revolutionary change, it is not yet fully autonomous and hence only a proto-mode of production. So far so good, but now comes the derailment. First, he imports the

Caffentzisian distinction between capitalist commons and anti-capitalist commons. This distinction is a product of Caffentzis's critique of Elinor Ostrom, and it distinguishes between common-pool resources that are utilized to produce commodities (capitalist commons) and common property regimes that are utilized for social reproduction (anti-capitalist commons).[23] This, however, is an issue that is irrelevant to non-rivalrous open access digital commons such as FS and other cases of digital peer production. At no point are capitalists who *use* FS able to subtract a certain share from this commons and sell it as a commodity. For an unexplained reason that seems nevertheless to be a product of a healthy instinct, Bauwens is able to distance his position from the irrelevant Caffentzisian issue somewhat, through sheer fortuity of temperament: "Our approach stresses that it is more productive to focus on the post-capitalist potentialities of peer production, and make them real and concrete, than to fight against commons that are compatible with capitalism" (Bauwens, 20012). The poison has entered the system, however, and its deleterious effect is felt in the form of a misguided search for the cure: Bauwens embraces Dmytri Kleiner's "proposal for a peer-based counter-economy" that revolves around the introduction of "a semi-free non-commercial license", or CopyFarLeft, in place of the copyleft free licenses such as the GPL, which have served the movement quite well so far. These free licenses guarantee the founding four freedoms outlined by the Free Software Foundation, which explicitly prohibit any restrictions on use, commercial or otherwise, while also prohibiting commodification of derivative works. Instead, Bauwens regresses back to vague notions such as "while value is created in the commons, it is captured outside the commons" to describe the position of FS and peer production in the context of the larger capitalist economy, brushing aside the entire theoretical accomplishment of the Oekonux project as well as other Marxian analyses of FS in favor of the superficial and tired notion that "capitalists make money off of FS". The later works of

23 http://p2pfoundation.net/Antagonistic_Usage_of_the_Commons_Concept

Bauwens in collaboration with Vasilis Kostakis have unfortunately been dedicated to fixing what is not broken, dragging the P2P Foundation towards a dead end of voluntaristic activism aiming to implement a misplaced mutualism, instead of building on the clarifying mission of Oekonux regarding the (communistic) "real movement" of FS "which abolishes the present state of things" (Marx, 1845/1998: 57).

Now that I have presented the main moments in the development in FS theorizing and commented on what I see to have been the strengths and weaknesses in these three approaches, I will move on to Chapter 3, where I make my own contributions in an effort to build on the synthesist project, in the tradition of the theoretical achievements of the Oekonux Project, which I believe has been challenged but has not been surpassed yet.

CHAPTER 3: THE IDIOSYNCRASY OF THE FS MODE OF PRODUCTION

The most important achievement of the Oekonux project is that it foregrounded the need for recognizing the idiosyncratic in the FS mode of production—the "beyond"ness of its patterns. It is natural that when we attempt to explain any kind of authentically novel phenomenon, we cannot avoid making analogies and referrals to concepts we already claim to understand. But to the degree that the phenomenon is indeed novel, these analogies and referrals will necessarily be partial and contrastive as much as they are demonstrative. We cannot reduce to the old what is irreducibly new. The new will have to be explained in the first instance by utilizing past concepts, but the past concepts will appear in sublated[24] form. The old principles can only be preserved while being transcended. Ideally, a new vocabulary will be invented that captures what is idiosyncratic and historically specific in FS, which we can eventually switch to as studies of FS are formalized.[25]

As a step in this direction, I analyze in this chapter three major aspects of FS as a mode of production with a claim to capture what is idiosyncratic in it: the forms of remuneration for FS labor as a series of concentric circles (section 2.1), the emerging class position of FS producers as the basis of the mode of production (section 2.2), and the radical implications of forking on collective decision making within the sphere of production (section 2.3).

3.1 Concentric Circles of Remuneration

I see the remuneration mechanisms of FS as a series of concentric circles. The

24 The process is captured by the German concept of *Aufhebung*.

25 Here is a precedent: While having something in common with the slave, the "wage slave" is not a slave in the historically specific sense. Once we define a terminology that is proper to the historical phenomenon we are analyzing, we stop referring to the "wage slave" in favor of the "proletarian" (unless we are trying specifically to underline the continuity).

innermost circle, Circle 0, is defined by a single FS producer producing FS for their own use. Circle 1 operates on the level of the community of all FS producers who benefit from each other's work. Circle 2 functions at the point of interface of the FS mode of production and the larger capitalist economy with Copyleft playing an important role. Circle 3 consists of rising direct compensation mechanisms between FS producers and the public that uses FS. With the analysis of these already functioning circles complete, I critique the Copyfarleft proposal which has emerged as another potential compensation mechanism and instead argue in favor of basic income as a new, potential Circle 4 mechanism which can further accelerate the generalization of the FS mode of production.

The expansion of the circles will provide a mental image to aid our understanding of the expanding economy of FS. But it is not a solely analytical tool. It also follows the empirical, historical path of FS as its remuneration models diversified. Often, the same trajectory may also be observed over the lifetime of an individual programmer as the programmer moves between academic settings, from FS volunteering, to employment by FS producing corporations, to becoming an independent producer of various forms such as freelancer (e-lancer), donation recipient or public patronage beneficiary, or even a FS start-up entrepreneur.

New outer circles in the concentric circle model do not negate or constrain the operation of the inner circle mechanisms, but include and supplement them. As the number of individuals participating in FS grows by the inclusion of new people into outer circles, the inner circles also expand, strengthening the system as a whole. Therefore do not think of the movement as a mere tacking on of outer *rings.* Each quantitative (more individuals in given circle) and qualitative (formation of new circle) expansion furthers the displacement of the market in favor of the commons.

The succession of the circles is towards increasing *extrinsic autonomy* of the FS producing laborers within the historical *interim* between capitalism and the FS mode of production, while the system moves in the same direction as a whole: achievement of economic self-reliance. In other words, the movement is towards the realization of self-reproduction within the FS mode of production; it is the movement from proto- to full mode of production. This is *not* accompanied in the interim by a linear increase of *intrinsic autonomy* from the perspective of the individual FS producer, in the sense of choosing what to work on within which form of governance and with what regime of regularity. I rather claim that intrinsic autonomy follows a v-shaped curve where in between the two cases of maximal intrinsic autonomy which are the first and final circles, extrinsic autonomy comes mostly at the expense of the intrinsic.

The addition of each new circle of remuneration to the system increases both the mass of use-values produced (amount of useful software), and thus the non-capitalistically satisfied needs of society, and the number of individuals (communities of software producers and also users) with a stake in the life of the system.[26] As the number of individuals engaged in the sphere of FS in various capacities increases, the *cultural* influence of FS increases as well, giving it the character of a social movement. This is reflected in both the explosion of academic interest in FS, PP, open-source and the digital commons, as well as the interest of left-wing political movements and mainstream media commentators, whether business-minded, critical or utopian. The cultural influence of FS is sowing the seeds of a mass

26 It is important to note that this constantly increasing production of new software code must continuously compensate for "bit-rot"; the deprecation and degradation of old software code due to the constantly co-evolving software ecosystem. Thus we are looking at a field of work that is not simply cumulative, but which is always chasing moving targets in order to stay relevant. "Finished" and "complete" software projects are rare things. Constant improvements, iterations and maintenance requirements characterize the field. There is however an element of *decadence* in the world of proprietary software which increases the chances of the FS competition catching up: The inclusion of "anti-features" such as DRM and tracking mechanisms which make the software *less* useful to the user while more profitable to capital, planned obsolescence, as well as prematurely pushing out buggy, half-finished releases due to cut-throat competition among proprietary vendors.

political consciousness of FS and PP, which is already starting to be reflected in the programs of progressive parties, the Pirate Party phenomenon, and FS activism. I will comment on the significance of this trend in the second section of this chapter, where I discuss class.

In the tradition of the Free Software Foundation's (FSF) Four Freedoms Definition of FS, which begins with Freedom 0, I denote the innermost circle in my model as Circle 0. This circle consists of FS produced by an individual coder for their own personal need. This is what Eric Raymond has referred to as "scratching your own itch". The crucial point is that upon completion, code which is produced in this manner gets shared with all others who may have the same need. This is the simplest form of FS production but at the same time it is where the *immanent principle* operating at the core of the entire FS edifice is visible in its purest form. It is a radical phenomenon brought about by the nature of the digital artifact, which I presented in the opening page of my thesis: an individual producer of a digital use-value, by mere willingness to share which comes at no additional cost to themselves, by the trivial means of digital copying and distribution over the Internet, automatically provides the use-values for potentially everyone possessing the same need for such a use-value.[27] This positive externality is the main driver behind the entire system, and grasping it is essential to understand its resilience and sustained expansion. Also, despite the partial parallel, this already sets apart the FS producer from the traditional subsistence producer who produces on their own what they will *individually consume.* Contrary to the material nature of commodified analog goods, in the material nature of the digital artifact, there is no

27 They have to be able to actually find out about it though! This discovery process may be commodified. Here are two examples: Certain sneaky, small parties sometimes venture to re-brand and sell FS items to customers who are unaware that they can acquire the genuine product for free online elsewhere. While unethical and frowned upon, this practice is technically legal and is in observance of FS licenses, because FS licenses allow charging for distribution. The scenario in mind however was FS CD sales and not this type of scam. Another case is the App Store model that has come to dominate the smartphone and tablet computing platforms, where software installation is mediated by a gatekeeper (Apple's iTunes, Google's PlayStore etc.) who may collect fees and/or commissions from app makers and/or users of the App Store, including for FS. These are cases of consumers' lack of information regarding alternatives in the market leading to the realization of rents (a market inefficiency). See the concept of the "Attention Economy" (https://en.wikipedia.org/wiki/Attention_economy) for further inquiry into such issues.

contradiction and dialectical conversion between use value and exchange value. A contradiction between the individual and the collective does not arise on this point.[28]

Circle 1 in the concentric circles model of FS is the so-called gift economy relation among FS coders. I refer to this as the "so-called" gift relation because the nature of the digital artifact was unaccounted for or under-appreciated by those who asserted the identification. Gift exchange economies by necessity operate on the basis of an item gifted to someone specific equaling said item being removed from possession in the process of exchange—whether immediately or after a delay—in favor of another. Digital copying, as is the case in FS, means any and all parties, who are mostly anonymous, maintain possession of the exchanged artifact simultaneously. Still, let us ignore this theoretical shortcoming for the moment because flawed as it is, the gift economy perspective is based on an important insight. Early theorizations of FS arrived at the gift economy model because they observed a community of vocational programmers freely sharing code predominantly among themselves. This implied a degree of reciprocity, even if it was implicit. The producer base of software was more or less identical to the user base. This of course was to change dramatically with the PC revolution, leading to a situation where the vast majority of users are not programmers, although their use itself partially contributes to the production of software through the indirect mechanisms of network effects and providing feedback.

Nevertheless an essential positive externality was in effect in this "gift exchange" among programmers. As a thought experiment, let us imagine now the first act of code exchange between coder Alice and coder Bob, a hypothetical *ursprünglich* moment in the software commons. Let us assume that through their own labor, Alice and Bob have

28 The realization of the use-value of a digital artifact does not take the form of consumption, but instead the form of using a copy. Replication of the digital artifact among its users is just one more instance of the mundane operation of copying which computing as a whole is based upon. In its technical functioning, copying data over the network, i.e. between users, is not categorically different than copying data from the hard drive to the memory of a system that belongs to the same user.

respectively produced codes Foo and Bar. Let us further assume that Alice and Bob are both producers of average skill and it took the same amount of labor time to respectively produce Foo and Bar, making them of equal value. Both interested in the use-value of each other's piece of code, Alice and Bob now engage in exchange, i.e. they provide copies for each other of Foo and Bar. Both are now in possession of both Foo and Bar. Concerning fairness in remuneration, both are now fully compensated for their efforts in producing their respective pieces of code. But something extraordinary happens here due to the nature of the digital artifact, which sets Alice and Bob apart from two simple commodity producers engaged in a barter. The "exchange" is not a private affair as in the case of market exchange. The exchange occurs over a public network, and there have been no copy protections placed on Foo and Bar of either a technical or legal kind that would limit the exchange to Alice and Bob as the only authorized parties. The result is that now not only Alice and Bob, but in principle every potential user of Foo and Bar have also come into possession of the software. This is the secret behind what Eric Raymond identified as the "Magic Cauldron" (Raymond, 2001: 113-67) of open-source: a common stew, to which each contributes a small bit, yet is able to receive as much stew as personally needed in return, simultaneously and non-subtractively.

In a hypothetical assessment of fairness defined as equal exchange for groups larger than the most basic symmetric schema of Alice and Bob, we could make the following calculation: On the one hand, we must know the labor-time Alice contributed to coding FS (we ignore non-code forms of contributions for the moment). Then we make an inventory of every piece of FS that Alice uses which has been coded by other FS producers. For each item in the inventory, we assume we know the labor-time that was expended in producing the item and we also know how many copies of the item are globally in use. We divide the first by the latter to arrive at a per-copy value of the item.[29] We repeat this for every item in Alice's

29 The more general purpose the software, the more users, hence the per-copy value of the software tends

inventory of FS used, and add up the values and arrive at a sum. If the labor-time Alice contributed to FS is equal to this figure, we can reach the conclusion that the relationship of Alice to the community is fair on the basis of the law of value within a system of generalized reciprocity. With each new fair participant in the commons, the positive externality born out of the individual's socialistic "exchange" with the collective spreads throughout the system, as in the previous case of the one to one exchange between Alice and Bob.

This calculation of what goes on between Alice and the collective is all well and good, except for one problem: putting it into practice would be insanity. Not only would it prove utterly unfeasible to implement, it would hardly be desirable. We would need to install a draconian surveillance mechanism on each and every person's computer that would track every piece of installed FS on their system, as well as a mechanism that would track how much time they spend towards FS production. These would then have to be aggregated and constantly updated in real-time, accounting for the millions of hours worked and millions of installations of FS made every day. The end result would be a major disenchantment in the form of a number spit out on each individual's screen, stating their balance of account towards the commons, a quantified amount of credit or debt. And then what? Presumably after a certain period, a check or an invoice, followed up by enforcement, with all the nastiness that would go along with it.

The point of course is not to actually account for and guarantee such fairness in practice but to transcend it, in the sense that over time and across a large number of individuals, the principle roughly holds without conscious intervention. In fact the power of the communistic FS mode of production is demonstrated precisely in the tolerance for

towards zero. For certain niche software that takes a large amount of labor to develop however, the per-copy value will remain non-trivial. This could possibly explain certain holes in the currently existing gamut of FS solutions, as well as very high prices on the proprietary counterparts. Examples that come to mind are game engines, professional CAD-CAM software, and Non-Linear Video Editors.

individual cases of "unfairness": Unlike traditional material commons, those who maintain a relatively one-sided relation to the commons do not have a subtractive effect—in fact, recalling the comedy of the commons, they still have a relatively small but additive effect. The FS ethos therefore prioritizes the maximum satisfaction of needs instead of obsessing over equality of contributions. We settle for a subjective fairness that leaves it to the operation of the moral urge to reciprocate that springs from the individual's conscience and sense of appreciation, which proves to be enough to sustain the system. External motivations for contributing are not essential for the system to function. At the end of this section I will discuss whether a modified mechanism of calculation could be beneficial, by critiquing a proposed socialistic Copyfarleft license intended to replace copyleft licenses.

The real limitation in Circle 1 remuneration *in this historical phase of transition* is not the issue of fairness, but the in-kind nature of remuneration that takes place.[30] Only software needs can be met within this circle (or in the case of PP in general, only needs for digital artifacts). Although the spending of individuals who benefit from PP is reduced by the amount that they would have otherwise spent on digital artifacts (software, e-books, digital music files etc.), they cannot pay for "food and rent" through the operation of Circle 1. They would have to either do something outside of FS to earn money as well and limit the time they spend on FS production, or engage in the higher circles of FS remuneration.

What has truly propelled FS from a small "gift economy" among programmers towards the cyber-communism (Barbrook, 2000) we now observe was the Personal Computer revolution. The PC revolution started in the late 70's, and exploded in the early 90's with the advent of the home internet connection. PC's meant that non-programmers would own their individual computers and use them for tasks other than programming. In fact the general

30 If we were to imagine a future society where most or all production is the production of digital artifacts, all remuneration could be in-kind, and the issue would not arise.

purpose personal computer became ubiquitous in production, used in every sector of the economy. The software accompanied the hardware, creating a vast market for what used to be called "packaged software": binary-only software that does not include accompanying source code. The emergence of Microsoft, which focused on serving this personal computing software market, was part of the same trend. The PC revolution has been a massive democratization of computing, which created swathes of computer users that vastly outnumbered the number of user-programmers. The role of programmers shifted from serving each other in academic research facilities where a form of avant-la-lettre FS had emerged, towards serving mere users. The dominant form this service took has been the market-based proprietary software model, i.e. selling licenses for usage of binary software that came without accompanying source code. However, FS adapted to this new terrain, where a massive discrepancy exists between the number of FS coders contributing code to the software commons and the number of users benefiting from it. It is this new terrain that provided steam to what I call the Circle 2 of FS remuneration.

Circle 2 of FS consists of the contributions made by coders who are employed by an entity such as a corporation to produce FS because it furthers the commercial success of the corporation in a related field, or by a government which employs FS producers with a variety of economic or political motivations. A FS worker operating within Circle 2 is not concerned with what mechanism the capitalist, the government or other entity has devised to benefit from its spending on FS development; the relation of the entity to the FS producer is wage labor. This means that a category of FS programmers are compensated with money rather than the in-kind exchanges of the previous circles.

Let me point out that this does not reduce the size of Circle 1. Even when individual contributors in Circle 1 are recruited by entities into Circle 2, they are naturally replaced by

other newcomers. Furthermore, the volunteer phase of an individual FS coder may in fact have been motivated by expectation of future employment in Circle 2 as a result of volunteer work to begin with, a process analogous to the role internship plays. With the Circle 2 mechanism, individuals who at some point might otherwise have to abandon FS or greatly limit their contributions can be sustained. Also, there is the likelihood that the recruiters will hire volunteers to continue working on their existing project (where they have demonstrated their competence), rather than assigning them to different work.

There is a widespread tendency to see capitalist sponsorship of FS as proof of its capitalistic character, but this is a superficial conclusion which is ultimately fallacious. Capitalist investment in FS production does not turn FS into commodity. Neither are the use-values seized by capitalists in any other way. As I have explained in Chapter 2, capitalist contribution to the FS commons can be a result of the contradiction between the interests of a particular capital and capital as a whole. Not to mention the fact that the particular capitalist may be dragged into FS production by the desire to build upon already existing Copyleft'ed code (produce a derivative work).

The Circle 2 model allows economic independence to the FS contributor so that they may pay rent and buy food while contributing to the commons which benefits everyone. This is a good thing. Corporate or state direction of FS projects does mean, however, that some freedom in organizing the productive activity and defining its goals (the question of what to work on and how) must be surrendered to the corporate or state managers. The economic independence of the FS developer in this circle comes with managerial strings attached.

Personal economic sustainability in Circle 2 may thus come at the cost of a degree of *alienation* in the work setting, which is the characteristic feature of all wage labor. The mere fact that a laborer is paid a wage in order to produce FS as opposed to proprietary software

cannot negate alienation in the production process, when it is managerially organized by corporate, governmental or other non-self-organized entities. Even within the wage relation, FS does, however, have a tendency to *reduce* alienation compared to analogous proprietary software production. There are two factors that effect this amelioration in the condition of alienation in *FS production under external management:* The first is that the product by definition remains a commons, so producers are not alienated from the fruit of their own labor in exchange for the wage. The wage is received in addition to access to the product that is produced. The second is that FS production has to be open to some degree to the collaboration and contributions of a larger community. To have it otherwise would defeat the entire purpose of engaging in FS production for the entity—the whole idea for the entity is to benefit from free external inputs instead of developing a proprietary solution where the entire costs would have to be internalized. Therefore, any FS project must engage to some degree in dialogue and partnership with the community in the way it organizes production, taking into account the needs and wishes of this community, of which the waged producers will comprise a (major or minor) subset. If this interaction is deemed to be dysfunctional by the community as a whole, the project's success will be jeopardized and may be threatened by the appearance of a fork. This ever-present pressure of community opinion and the threat that the community will vote with their feet in FS production provides a check on the amount of managerial fiat that a FS producing capitalist or state entity can exert on their waged FS producers (the same applies to any kind of leadership in FS projects).

The interests of the sponsoring entity may overlap with the interests of the public fully or to a partial extent. This is a matter of the use-value of the produced FS. I intuit that as a general rule, the public good will be furthered by capitalist sponsored FS in similar fashion to how public goods produced by the capitalist state (such as roads etc.) benefit not only the

capitalists but the population as a whole (assessing exactly who benefits how much can be a complex matter). In cases of government sponsorship (whether at the local, national or international level), the harmonization of public and governmental benefit will be dictated by politics (which no doubt is influenced by economics) rather than direct economics. Chapter 5 of this work is dedicated to the study of the Pardus project, which is such a case.

Circle 3 contains FS remuneration schemes where FS producers are funded by their users voluntarily and directly, without formalized procedures such as contracts or reviews of work performed. This remuneration is essentially in the form of donations and it can be seen as an example of collective patronage. Donations are made to FS coders either prior to or after/during (as software is rarely "complete", there is no clear "after" but rather continued development) the initial work of development takes place. The simplest form of donation is usually facilitated by posting a bank account number, PayPal button or other electronic currency id on a FS project website. Donors may sometimes receive notices of appreciation such as appearing on a hierarchically ranked list of donors on a webpage. Those that donate over a certain amount may also receive tokens of gratitude like t-shirts or mugs.

As the donation model has gained traction and with the general proliferation of freelance work and the start-up phenomenon in the larger economy, innovations have taken place in the facilitation of donations. There are micro-tipping systems such as Flattr where the donor pre-allocates a certain amount of monthly donations in their Flattr account, from which donations are drawn in proportion to how many times the user clicks the Flattr buttons on multiple recipients' webpages during the month. The pay-what-you-want model pioneered by "The Humble Bundle" game sales is another variation on the donation model, where a purchase action is required, but the amount paid can be as low as a single cent or dollar (the requirement forces the user to break donor-inertia).

Crowdfunding has emerged as a systematic method of pooling donations for projects that are in initial or continuing development. There are a few variations of crowdfunding. In the Kickstarter model, work is premised on the prior promise of donations. The project often presents an introductory video and page explaining what the project aims to be, and a certain target sum of money and a duration limit to gather the donation pledges is stated. If the targeted sum for donation pledges is reached within the given time frame, the project is undertaken. When compared to the usual circulation of commodities in the market, this form of crowd-funding reverses the production first, sales second approach with a seek funding first, delivery second approach. It also shifts some of the risk of enterprise on to the backers because they cannot evaluate the finished product before purchase. In the case of crowdfunding of already existing FS projects, however, the risks are much reduced because trust has already been established and distribution is instant upon completion. In the Patreon model of crowdfunding, "patrons" pledge recurring donations to projects instead of the one-off model of Kickstarter. This increases certainty and regularity of income for the producer compared to impromptu donations.

The beauty of donations when coupled to FS is that each individual decides how much to give themselves, taking into account their own ability to pay. This is much nicer for the user than the proprietary alternative of a one-size-fits-all price tag which will be set at a revenue-maximizing level, shutting out those who cannot afford it. One drawback of donations for users is that it may not be easy to figure out just how much to give to what, which could lead to donor-fatigue / donor-cluelessness. Nurturing the sense of community between producers and users, as well as accounting transparency in FS operations and further systematizations of donation mechanisms can go a long way towards solving this. Another issue may be the relative difficulty of generating donor interest for non user-visible FS

projects. Donation sharing and kickback schemes between upstream and downstream FS projects are being put in place to alleviate this issue. A sore spot in these advanced donation systems so far is that the platforms are capitalist intermediaries which take a cut out of the donations. An obvious solution would be for the FS community to produce non-profit alternatives to these existing platforms. This is a young field which is still seeing major innovations and the best models will be settled on with time.

The significance of the improvements in donation schemes and the development of a culture of patronage among the public for FS is that it provides a foundation for FS programmers to take a major step towards becoming a class of independent producers without relying on the mechanism of selling their products as commodities on the market. This means the coupling of extrinsic autonomy with an upswing in terms of intrinsic autonomy because FS producers who can fund their work through collective patronage can self-manage their own organization of production, without the bosses and managers in Circle 2.

This sums up the circles of remuneration for FS that have emerged up to now within the constraints of existing society. It is important to keep in mind that Circle 1 (which contains within it Circle 0) is the defining form of remuneration of FS as a mode of production because it is unmistakably stamped with its own internal logic. If the progress of FS and PP is not blocked by reactionary forces, the ultimate long-term historical trajectory will be towards the complete dominance of Circle 1, which is an idiosyncratic form of what Marx called the higher stage of communism (Rigi, 2013; 2014). Circle 2 is a product of the interaction of the emerging new mode of production with the old capitalist mode of production. While Circle 3 is a step towards breaking out of the capitalist mode of production, it is still of a transitional nature. The more human labor in production as a whole moves exclusively into the realm of producing digital artifacts through increased automation, the more relevant the FS mode of

production will become and the more acute its contradiction with capitalism as a historical system. This will take a while, though. In the meantime, we need to keep thinking about the transition. One idea for accelerating this transition has appeared in the form of a proposed Copyfarleft license. I now present my critique of this proposal, before putting forward what I see as a better alternative, Circle 4.

3.1.1 What About Copyfarleft?

Before going into Copyfarleft, we need to review the role of what it proposes to replace, which is Copyleft. The Copyleft principle (exemplified by the GPL for software and the Creative Commons SA—Share Alike—attribute for creative works) provides an extra gravitational pull that helps to enlarge the sphere of FS, especially by luring a special stream of contributions into Circle 2. Copyleft licenses offer a social contract to potential contributors, encouraging them to make use of and partake in the existing commons while mandating them to contribute their improvements to the commons if they distribute their derivative products. Contribution to the commons due to the Copyleft incentive is at once reciprocative towards the producers of the original work and "plays it forward" towards the rest of the public. Copyleft accelerates the materialization of new contributions to the commons of digital production.

All of the qualitative principles of FS would still hold if Copyleft had never emerged as a legally defined social principle (as indeed Copyleft is not a defining feature of FS); however, its size would have been limited to a smaller sphere of the economy. Without the combative stance of Copyleft, FS would have continued to exist, but the logic of interface between the FS commons and the domain of capitalist relations of production would be different. The contribution of capitalists to the commons would be much smaller. Without

Copyleft, or more accurately, to the extent that non-Copyleft FS production takes place (permissive licenses such as the MIT or BSD licenses are also quite popular), the self-interested contribution of commercial companies to FS is limited by a logic of an *embedded commons*. Where Copyleft is not in place, this embedded commons, that is embedded within the logic of capitalism, is narrowed down to instances where otherwise competing capitalist enterprises need to collaborate partially. The purpose of such collaboration is to build a common non-profit economic *infrastructure* that supports competing for-profit activity taking place on top of it. The creation of such commons augments the for-profit operations of every capital relative to a state of non-existence of said commons. Hence it is a mere rationalization, analogous to the role public works undertaken by the State and State-owned enterprises play in capitalist economies.

There is still a public good derived from such a capitalistically motivated FS commons because it is open to the public (in the same way public works and public enterprises also serve the public good to an extent) but this is a second-order and constrained outcome. The militant character of Copyleft, however, tempts individual capital to engage in commons production when it is in its self-interest, even if it goes against the common interest of capital as a whole. In FS, the capitalist enterprise does this with a view towards increasing its own share of profit elsewhere while shrinking the market of commodity-form proprietary software. This is why, unlike what takes place through the mediation of the capitalist state, which looks after the collective interest of all capital when it comes to traditional public works, the FS producing capitalist engages in the production of the public good directly, by individual initiative. This means that Copyleft succeeds in achieving from the bottom-up what is usually only achieved through the use of state power under the influence of mass social struggle:[31] capitalist contribution to the public sector that benefits the population universally.

31 I am referring here to taxes paid by capitalists to fund the welfare state.

Hence, Copyleft FS "extracts commons from capitalism" (Rigi, 2014). While its inventor Richard Stallman probably never thought of it in these terms, Copyleft dealt capitalism a blow at the level of the *commodity*—by creating an incentive for decommodification. This legal "hack" (Stallman) or "dialectical negation" (Rigi, 2013: 399) of copyright used copyright law to serve ends opposite to its intention.

Copyleft does not, however, bar capitalists from *using* FS for any purpose (as it does not discriminate against *any* use). FS gets used in all kinds of commercial enterprises. In The Telekommunist Manifesto, Dmytri Kleiner proposes to replace licenses such as the GPL, which embody the Copyleft principle, with what he sees as an improvement: the Peer Production License (PPL), which instead embodies the "Copyfarleft" principle, which is incompatible with the GPL. Contrary to the principles of FS, the proposed Copyfarleft license would discriminate among the type of users of a licensed item. Copyfarleft would not ban all commercial activity though (unlike the Creative Commons -NC- Noncommercial option). It is more specific: worker-owned cooperatives would be allowed free use of a PPL licensed item, while capitalistically organized corporations that employ wage-labor would be excluded.[32]

While Kleiner advocated the PPL for creative works, as an alternative to the Creative Commons licenses which are popular in this area, the concept has since been picked up and modified by Michel Bauwens and Vasilis Kostakis (2014), who argue that they could be applied to all PP. The modification that Bauwens and Kostakis make to Kleiner's Copyfarleft PPL is a form of dual-licensing; instead of banning capitalist use of peer produced digital artifacts outright, they suggest that those capitalists who do not contribute to PP can only use them in exchange for paying a license fee. Capitalists who contribute to PP, as well as worker cooperatives are not affected. This is intended to provide a flow of income from the capitalist

32 For a description of why this does not get to the root of the problem (commodity production), see Meretz, 2014 and Rigi, 2014.

sphere of the economy to the PP sphere and bootstrap the PP mode of production. A further modification to the PPL is Vieira and Filippi's Commons Reciprocity License (CRL), which is basically a small revision on Bauwens & Kostakis's version of the PPL (Vieira and Filippi, 2014): Instead of a binary category of "capitalists who do or do not contribute to FS", it aims to measure the level of contributions and demand equal reciprocity, possibly through a token system similar to Local Exchange Trading System (LETS) credits used in certain local and solidarity economies. Both the modified PPL and the CRL licenses posit a Circle 2 mechanism of remuneration for the FS mode of production that is an alternative to Copyleft. They are, however, deeply problematic.

To begin with, this is based on a strategy of social transformation where, because of their internal distributive structure, cooperative enterprises are defined as revolutionary economic institutions that are to outcompete and extinguish capitalistic enterprises in the market. This revolutionary strategy known as cooperativism is not new. It dates back to the 19th century workers' movement in the shape of Owenism (after Robert Owen) and is also expressed in the economic thought of Pierre-Joseph Proudhon. The PPL is thus a suggestion to rekindle the cooperativist strategy on the new terrain of possibilities that has opened up with the appearance of the digital artifact.

I hold the Achilles' heel of the cooperativist strategy to be that even when successful, in its minimal definition, it does not address the issue of economic inequality that arises between nominally defined workers that belong to different cooperatives. With their prescriptive approach to the law of value, cooperativists advocate maintaining the market system of exchange for the distribution of goods in the economy. But self-managed by the workers as they are, cooperative enterprises still can and will be in possession of means of production which are of quite diverse values, due to the different amounts of capital (and

these means of production remain capital, for the reason I will soon explain) the cooperatives will employ. Some cooperatives will employ machines that are more productive than others, some cooperatives will have access to more valuable resources than others, some cooperatives will reside in more advantageous locations than others, etc.

In the absence of a higher organ of coordination that functions according to non-market principles, the diversely valued products of these worker-owned cooperatives will be exchanged according to their market prices, and such differences will result in various sorts of rents to accrue (technological rent, ground rent, etc.) away from the worker-owners in the disadvantaged cooperatives, towards the worker-owners in the advantaged cooperatives. Therefore the means of production will maintain their character as capital, entailing the transfer of surplus value from one set of worker-owners towards another set of worker-owners, who now indeed face each other as two classes. This situation can perhaps be held in favor as nevertheless an improvement over the capitalist-proper situation that exists in regular firms, if the level of inequalities (indeed the level of exploitation) could be held in check over time. According to the laws of accumulation of capital, however, the antagonism would be aggravated with time, increasing polarization of value generating value on one hand, and proletarianization on the other. There is nothing intrinsic to the system that would prevent successful cooperatives from eventually buying out the assets of unsuccessful ones and employing the ex-worker-owners as wage-labor. Unless, of course, there is some sort of *law* against this, which would defeat the whole purpose of trying to construct an *emergent* alternative in the first place. In its minimal form, cooperativism can therefore only present an unstable, half-measure solution to class conflict as long as an unregulated market remains the mode of exchange. Owenism and Proudhonian anarchist socialism predate Marx and lack many concepts he developed. One such concept is the significance of divergences from the

average rate of profit and how surplus value is captured through the market mechanism from workers under capitalists who employ means of production bestowing lower productivity to labor by capitalists who employ means of production that bestow higher productivity to labor. (Where both enterprises produce identical goods, the one with higher productivity can produce the same good cheaper, retaining a larger profit from each sale.)

Yet this critique applies to market-cooperativism in general and is not specific to Copyfarleft. There are also many features that could be cited in favor of cooperatives, and it would not be fair to list only their shortcomings, if our point were to make an assessment of the role of cooperatives in general.[33] Cooperatives have their place in the progressive arsenal developed in the period of classical industry. My real point is that there are better alternatives for the sphere of informational production, and the FS mode of production has already moved beyond the cooperativist logic, and it should not look back.

The specific problem with the PPL scheme is that it ignores the insights we gained into the economic nature of the digital artifact. The fact that decommodification allows a capitalist to use the decommodified article as a factor of production without paying for it is not a sufficient argument against its decommodification because such use is not rivalrous when the commons is digital. At its worst, it merely points to a rationalization of capitalist interests in that the individual capitals have recognized that their collective interests are better served through this decommodification, cutting total costs. In this case one can see this as a neutral development in terms of class struggle. However the effect of FS is in general much more than that. To the extent that there is also a use value of the software in question for non-capitalists in society (i.e. all laboring classes, whether worker, petite-bourgeois, or member of worker-owned cooperatives), the net effect of the decommodification of software is

33 See, for example, J. K. Gibson-Graham, 2006, Ch. 5 for an excellent study on the biggest cooperative structure in the world, the Mondragon Corporation in Spain.

progressive. Copyfarleft supporters are mounting arguments against the decommmodification of software through the GPL as if such software were used merely by a consortium of capitalists. In fact there already exist such forms of exclusively capitalist "commons": shared-source licenses, patent-pools and other IP held by industrial consortia. Copyfarlefters are attacking an imaginary case of FS, where it would turn out upon inspection that a FS is actually only nominally a public good, while in practice a shared-good, shared among capitalists, that is. If such cases could be identified and demonstrated, the critique would be valid. However, this begs the question: Why would any volunteer FS producer choose to make their contributions to a FS project that seems to be used only by a capitalist corporation rather than by the producer themselves and the public?[34] A dose of common sense against gullibility on the level of the individual would be adequate to the challenge here. If corporations find themselves in the position of having to produce such software as FS due to Copyleft, let them hire programmers to do it.

As it stands, Copyfarleft embodies a misguided form of purism, where the mere fact that something which benefits the working class also benefits the capitalists appears as sufficient grounds to oppose it. This would be a form of class struggle more reactionary than Ludditism. I sense, however, that beneath the misguided purism lies a possibly valid concern. While I hold that capitalists are not in a position to *exploit* FS producers ("capture value" etc.), I sense that Copyfarlefters' concern could be a valid one in the sense of fairness defined as equal exchange, which I demonstrated with the thought experiment above involving Alice and the collective. In short, we may suspect that the capitalists may not be contributing to the commons as much as they are receiving.

34 One could of course choose to participate in such a project for learning or because one thinks they can get a job at the company etc. in which case there is a personal benefit. Furthermore, if such phenomena is objectionable, the solution lies elsewhere (such as regulations against unpaid internship etc.) than in the licensing.

But this concern is hardly backed by empirical reality. The annual Linux Kernel Development report, for example, demonstrates not a case of capitalists free-riding on the labor of the unpaid volunteers, but quite the opposite. For the Linux kernel which is under the GPL license, in recent years up to 80% of the work is being done by coders paid by corporations (Corbet, et. al., 2015)! In this major example, and no doubt in many others, the shoe is on the other foot. To the extent that the Linux kernel benefits everybody, which is undoubtedly the case, this is a win. Recall that the Linux kernel was originally produced entirely by volunteers, starting with Linus Torvalds himself in 1991, but over the years more and more corporate contributors have been drawn in (the largest single block of contributions are still from volunteers however, at 12%). FS under Copyleft has created such a gravitational attraction that it has successfully interfaced with the capitalist sector, converting some of their activity to the new logic of the commons. Had the Linux kernel been placed under a Copyfarleft license, the capitalists would have either no incentive to contribute because they would not be able to use it (the original PPL) or they would have a reduced incentive (modified PPL's) because they would have to pay for using it. Not only does this strategy dispense with the benefit of the network effect that derives from capitalist use of FS,[35] the Copyfarleft licenses would in practice amount to an either complete or partial self-imposed embargo. It would also damage the communistic ethos of FS in favor of a regression to a socialistic ethos, not to mention that it would have to introduce elements of the insanity of accounting into the FS mode of production whose idiosyncratically nurtured communistic culture of reciprocity defies it.

Despite their error, the driving motive behind Copyfarleft proposals is the laudable goal of securing economic independence for PP producers and expanding the mode of production. The issue of how to provide for the means of subsistence of FS producers is no

35 Indeed a large part of the energy of FS activists is spent to encourage adoption of FS!

doubt vital. As demonstrated by circles 0 to 3, this provision is partially already observed. Remuneration is important, but one does not need to fixate on it to the point of risking obliteration of the idiosyncrasy of the mode of production in the process. It might even be better to acknowledge and clarify the historical tendency within it, rather than muddying it with an ill-conceived intervention due to impatience.

Alternatives for the provision of the needs of laborers who produce use values defines the nature of classless society under discussion. If this is done through compensation of labor, maintaining the principle of the law of value, i.e. the exchange of labor-time of equal magnitude for products of equal value, it is a form of socialism. This remains the case, whether this is carried out through measurement and calculation (planning) or through allowing the expression of value through exchange-value (markets). There are competing Marxist and anarchist wings within both styles of socialism. The typical form of Marxist collectivism has been authoritarian state-socialism, while Bakunin's collectivist anarchism advocates a more libertarian socialism through the principle of federation. Within the cooperativists, the Marxist camp is known by the market-socialist school, which maintains some role for a state, whereas the state-abolitionist Proudhonians known as mutualists represent the anarchist wing of market-socialism, to which the line of the Copyfarlefters is closest. The struggles between these alternative approaches have been at the core of the 19th century workers movement, and overall the Marxist camp has won out, although 20th century actually existing socialism has been mostly a disappointment in creating classlessness. Thus we have had a wing that has failed to succeed, and a wing that has successfully failed. But there have also been those in the past who have called for a direct transition to communism on both sides of the divide and they have been the most marginalized, whether by real conditions of historical impossibility or ideological repression. The Marxists among them such as Anton

Pannekoek are known as left-wing communists and the anarchist communists are exemplified by Peter Kropotkin.

The great opportunity presented by the emergence of an economy where production of digital artifacts has a rising importance, if not a looming hegemony, is that this time the position of the silenced "utopians" of old is looking like the most realistic, while the "realists" are looking more and more utopian. The position that is most in tune with the existing conditions of production is in this case the communistic position. This is a vision of classlessness born out of the informational economy, rather than the industrial economy. The whole raison d'etre of socialism as a transitional period to develop the means of production in order to produce abundance is becoming a fait accompli under informational development. Within the field of digitally representable forms of wealth, absolute abundance is becoming an abundantly clear fact. Even the notion of communism itself is being sublated: Contrast the nature of absolute abundance represented by digital goods in the information era with the industrial era communist vision of abundance, which was abundance relative to socially determined needs (which must have some sort of upper-boundary, whether implicit or explicit).

Thus the proper organizing logic of the informational sector must be communist, and the wider the economic significance of this sector within the economy as a whole, the bigger will be its potential progressive impact in the world. In fact I see this as the entire pillar upon which the Negrian political project is founded. In contrast to the wide definition of Hardt and Negri's concept of immaterial labor, I am taking a more cautious approach, reserving my case to the production of digital artifacts only. I believe Hardt and Negri's thesis itself applies much better to the production of digital artifacts like FS than most of the cases of immaterial production cited in their trilogy. Rigi's formulation of this thesis captures my outlook:

> Yet capitalism prevents the free flow of knowledge in all directions in the net. It is true that the capitalist mode of production, adapting itself to ITP, has become global, and has increasingly adopted a network form. However, the sum of all potential links in the net exceeds dramatically the sum of links of the global networks of capital. Hence, the potential of the net, as a paradigmatic productive force of our time, exceeds the capitalist mode of production. (Rigi, 2012)

In this spirit, I propose one final remuneration mechanism that would introduce a Circle 4 to the concentric circles model as a better alternative than the regressive Copyfarleft concept. This is, a universal basic income.

3.1.2 Moving Towards Circle 4: Basic Income

The function of remuneration is to allow the satisfaction of needs. To recap, the previous circles of remuneration shaped this pattern for FS producers: Circle 0 is about software needs one can satisfy oneself. Circle 1 is about software needs that the universal community of FS producers can satisfy. These circles were alienation-free. Circle 2 is about the satisfaction of all of one's needs at the cost of alienation under the wage relation. This was also the impact point of Copyleft and the grounds for the challenge of Copyfarleft. Circle 3 is about the satisfaction of all of one's needs with minimal alienation.

I believe, the next step can be the introduction of a basic income. "A basic income is an income paid by a political community to all its members on an individual basis, without means test or work requirement"(Parijs, 2004: 8). There is a virtuously circular logic behind my argument. The already existing phenomenon of FS and PP proves that individuals are producing use-values for society as a whole without necessarily being motivated by money; they tend to do it as a matter of self-realization. As a corollary, they deserve being accommodated by society in a way that does not reintroduce alienation.[36] On the flip side, by

36 The timescale of Circle 1 world domination will be epochal. Not every deserving contributer to the digital commons can currently make a Circle 2 or Circle 3 mechanism work for them.

easing the provision of their basic livelihood, the introduction of a basic income will allow more individuals to cross the threshold of security in order to be able to participate in FS and PP. Under a regime of basic income, which is an elegantly simple demand, all kinds of PP would be boosted by new participants, and as of yet unforeseeable new instances of PP may appear.

Another beauty of introducing basic income to a world in which FS and PP is ascendant would be that it would not jeopardize the operation of any of the lower circles, although it would strenghten the position of the laborer in Circle 2 and Circle 3 by increasing their options. "Give all citizens a modest, yet unconditional income, and let them top it up at will with income from other sources" (Parijs, 2004: 7).

There are already instances of support for the idea of basic income in the FS and PP literature.[37] In fact while basic income seems to have dropped off the radar for Bauwens following the *mutualist turn,* he had espoused the idea in his earlier works (Bauwens, 2005). Perhaps even more significantly, however, software programmers themselves seem to be arriving at the idea. In a recent essay on his blog titled "Funding FOSS", software programmer Noah Kantrowitz points to the "non-capitalist system" of basic income as "an end game solution". "Provide a basic standard of living so people that want to dedicate themselves to enriching society can do so without putting their own needs in jeopardy".[38] While Kantrowitz approached the issue from the perspective of funding FS, I find it equally exciting that other software programmers are pointing to basic income as a logical solution to the contemporary global problem of structural unemployment.[39] Acknowledging the role software plays in inducing technological unemployment, software developer "Jason" of the blog "Practical Elegance" published a post titled "Confessions of a Job Destroyer" addressing the

37 Cosma Orsi, 2009; Jakob Rigi 2014. See also http://p2pfoundation.net/Basic_Income
38 https://coderanger.net/funding-foss/
39 http://blog.p2pfoundation.net/why-the-tech-elite-is-getting-behind-universal-basic-income/2015/02/24

issue:

> We (programmers) all are, on some level or another; we're taking mundane repetitive tasks and automating them with code. In a perfect world, we would be hailed as heroes, freeing the toiling masses from their humdrum routines to engage in more ennobling pursuits… but there's that pesky issue of needing an income. (...) This, gentle reader, is where I make the argument for a basic income. It's just common sense as the amount of socially necessary labor decreases with each passing year. (...) I'm a job destroyer, and I love what I do. Now if only we had a rational economy, I could stop having mixed feelings about the net effect of my work."[40]

I expect that the demands for a basic income and reflections on the expansion of the FS mode of production will increasingly coincide. This hypothetical Circle 4 remuneration mechanism could prove to define the penultimate phase of information society on its path towards "fully automated luxury communism".[41] Its realization, however, will require mobilizing the cultural influence of FS in order to express it in the sphere of politics. Its realization could take a while. But it would even provide a forward-looking solution to the Copyfarlefters' concern with making sure that the capitalists contribute their fair part to the digital commons, because progressive taxation would provide at least part of the funding for basic income. Even more importantly, basic income would leave the communistic cultural experience in the FS mode of production undisturbed; no regressive introduction of the wage or equal value exchange. Furthermore, by looking out for the workers who lose their jobs due to the march of software based automation (not to mention the software developers employed by proprietary vendors who could lose their jobs due to FS competition), basic income could be the quintessential "non-reformist reform" (Gorz, 1968) demand that unites the bit-twiddlers with the rest of the proletariat in a long march towards post-class society.

This naturally brings us to the question of class. It may come off as unorthodox that I have gone into so many of the details of the new mode of production without explicitly

40 http://decomplecting.org/blog/2013/03/11/confessions-of-a-job-destroyer/

41 See http://luxurycommunism.tumblr.com/ and http://www.theguardian.com/sustainable-business/2015/mar/18/fully-automated-luxury-communism-robots-employment for this humorously serious proposition.

referring to its subjects as a class. This was necessary because the mode of production itself is only appearing as a historical tendency, as a germ form expressed in partial human activities of fluid subjects. The subjects that have been engaged in FS have therefore only been a class in the making. It is only with the recent systematization of the Circle 3 forms of remuneration that they have begun to be somewhat legible as a class in-itself that consists of independent digital artifact producers. Perhaps with the articulation of a political movement towards Circle 4 they will start to exhibit signs of a class for-itself.

3.2 A New Class?

It is manifest that the internal organization of FS production is classless. Whatever forms of seniority, meritocratic hierarchy or charismatic leadership are found to be operating within FS projects, these cannot solidify into class relationships. The reasons for this are in the very DNA of FS. There is universal access to the means of production, that is, the computers and networks (on the basis of personal property) and the previously existing FS code (on the basis of a universally open commons) that the current project builds upon. Second, there is no alienable surplus production that can be the basis of a one-sided accumulation. By definition the fresh product immediately becomes a commons.[42] We can therefore hold that the spread of FS and PP is at the same time the spread of relations of classlessness.

But what is the class position of a developer engaged in FS production within the capitalist society in general? Is there a class of FS developers? The answer to these questions

42 This guarantees that even in cases of FS production where remuneration takes the form of collective monetary compensation for independent production (e.g. contracts or donations), a perception of injustice in the internal distribution of income among the co-producers of a FS project will almost immediately lead dissenters to withdraw their labor and reorganize in a new project by means of forking and to compete against the first project. A just distribution of income among a group of FS developers does not have to mean absolute equality. This will depend on the norm freely decided upon by the developing community: it could be based on seniority, output, effort expended, needs or a combination of these.

is not straightforward because "FS developer" is usually not a stable category. Some who develop FS may in fact do it exclusively, that is, without developing any proprietary software, but perhaps not as their only vocation. Some do it on their free time while working regular IT jobs (programming or otherwise). Others contribute to FS as a hobby. Some are employed by companies or governments to work on FS projects but can be transferred to working on a proprietary piece of software at any time. Then there are hackers, those motivated by self-actualization or even ethics and politics, who may or may not be independently financially secure. Some FS producers have devised ways of earning income independently, through donations, private or public patronage or crowd-funding schemes. Still others sell services around FS as wage-earners in capitalist firms or as self-employed persons. The FS commons functions as a platform upon which all of these collaborate. To approach the question of class in the context of FS, we first need to take a step back and try to figure out the class position of software programmers in general.

Vocation does not define class, the relation to the means of production does. And there is something peculiar going on in terms of the relation of programmers to the means of production. The instruments of labor (PC's and network access[43]) are for all intents and purposes under personal property. The subject of labor, the already existing source code, is either the private property of the capitalist (IP), which is the case of the employed programmer developing the proprietary software of a capitalist, or it is under public property, which is the case of the FS programmer. Let us take a close look at the means of production utilized to produce software (and many other digital artifacts) and the relation of the laborer to these means.

For software, we are talking about owning a PC, or Personal Computer, the very name

43 Network access is rather under de-facto personal property, as each individual buys it as a subscription service.

of which is suggestive.[44] An average PC, costing somewhere between a few hundred to a few thousand dollars is enough for a programmer to run all of their developer software such as Operating System, IDE (Integrated Developer Environment), compiler and any additional software needed.[45]

A waged programmer working for a capitalist company will be working on a similar machine, a PC, and not for example on a "terminal" connected to a centralized "mainframe" on the back-end as they used to do before the 80's.[46] The recent BYOD (Bring Your Own Device) trend, where employees bring their own computing devices to work as a matter of their own preference[47] is the logical outcome of this parity—the company is seen by the employees to have nothing worthwhile to offer them on this count. Barring requirements for capital-intensive computing machines like supercomputers or server farms (to which the P2P and FS community has in fact produced certain answers as well), the independent programmer is on a level playing field with their waged counterpart. Naturally, one can always get a little bit more productivity by adding say, multiple screens, an even more expensive CPU with 8 cores instead of 2, or by doubling the amount of SSD storage, etc., thereby shooting up the price of the PC, but these only provide diminishing productivity gains on the productive activity, that is, the labor which goes on in the mind and is realized by the keystrokes and mouse clicks of the programmer. The BYOD trend is a sign that the increased customization, familiarity and consistency of one's own device trumps such marginal advantages any way.

44 Note here the difference between Personal Property and Private Property

45 There is ofcourse also the issue of the software used to produce software. Except for edge cases where an individual programmer can't afford the software he uses at work, these too will be either under pesonal (purchased license) or public property (FS).

46 The PC as a means of production is distinguished from other means of production in the house that function as "labor-saving devices" like the oven, washing machine etc. by its universality and its nature as an asset of the most advanced and profitable industry of informational capitalism.. A PC is pretty much an "industry-grade" machine as opposed to any other machine in the household.

47 This is not a case of companies externalizing costs. Usually, BYOD requires extra work on the part of the company to ensure computing security because of increased diversity of devices on the network. But companies have embraced it because increased productivity outweighed such concerns.

In addition to owning a PC, the moment the software programmer wants to attempt a project that requires the coordination of other programmers, they will need network connectivity, in other words, the Internet. This commodity too is now readily accessible in the form of an inexpensive monthly subscription. Again, what matters is not the absolute megabits per second of bandwidth, but that any basic Internet access is sufficient for the coordination of software production. As in the case of PC hardware, paying for higher tiers of bandwidth will only yield marginal advantages in terms of software developer productivity.

The key to understanding how this state of affairs has come about under capitalism is to realize that these means of production, that is PC's and the Internet, are merely what *consumer goods* become when put to productive use.[48] In the early ages of computing, computers were indeed conceived as capital goods, with high prices prohibitive to the general worker, marketed to business owners for the exclusive purpose of increasing productivity in capitalist enterprises where all kinds of labor were employed. The "PC revolution" that occurred in the early 80's and which has only accelerated since, has instead marketed computers as personal consumer goods intended for the general public. It is now being historically and culturally established that all households need to have PC's and Internet connections. PC ownership and network access is becoming, in other words, part of the "labor fund" of society, part of the means of subsistence of the workers, i.e. for labor power's reproductive consumption. With the world population with internet access growing from 16% to 40% between 2005-2014,[49] the "digital divide" is shifting from a question of absolute access to a question of relative literacy. Whatever their current price, with the ordinary development of capitalist society, these items are becoming cheaper and already accounted for in the determination of the average value of labor power. Once amortized for the citizen as

48 Department I 1/2!
49 https://en.wikipedia.org/wiki/Global_Internet_usage

part of their consumption spending, they do not represent an additional entrepreneurial investment. The average citizen today is increasingly assumed already to need a PC and Internet connection for day-to-day information, entertainment, etc., as well as interaction with commercial entities like banks and e-tailers and with e-government services.

This suggests that such ownership is and will continue to be a mundane fact under contemporary capitalism and is not some vestige of a previous social arrangement or exceptional transitory phenomenon. The real limiting factor to the expansion of the class of professional digital artifact producers is not ownership of hardware but acquiring the relevant skill sets through education, whether through formal institutions or online resources for self-learning. I will briefly comment on the importance of educational opportunity for the expansion of the FS mode of production in the conclusion of this thesis.

This situation, which is favorable to FS production, is also a result of the fact that PC's have been "General Purpose Computers", that is to say, they can be reprogrammed through software to theoretically accomplish all computable tasks. They are not limited to mere consumption of what is put out by the "content industries"; they are also a means of producing such content itself.[50] This allows me to assume that programmers' access to their means of production is secure for the foreseeable future. While computers and networking are not the only means useful for software production (corporate office space still has its uses), they have been sufficient to serve as the basis for a culture of start-ups, SME's, self-employment, work-from-home and freelancing to develop around working with computers. The major barriers that have prevented programmers and other keyboard laborers from making a complete

50 If, by cultural influence, the general populace is convinced to purchase non-General Purpose devices geared exclusively towards content consumption (mobile devices as PC substitutes come to mind) instead of PC's, or machines that are so restricted by the vendor so as not to allow general purpose computing, things will be different. There is no trend to suggest, however, that there is a movement towards prohibitive pricing of general purpose computers in favor of mere consumption devices. Neither is there any sign so far of the would-be software or digital artifact producer not being able to procure a general purpose machine due to such machines being crowded out in the market by competition from restricted devices.

exodus from corporations are mostly the unproductive functions fulfilled by sales and marketing departments, lawyers and tax-filers, as well as the already established brand name recognition and other monopolistic advantages of big software companies.

We also know that software developers have a very positive occupational outlook. The median average wage for a US software developer was pushing 100K per year in 2012, which is at the top end of engineers and just below the first ring of managers. With one million employed and the number of jobs increasing at a break-neck pace of 20% per year, Silicon Valley visionaries have already reached the conclusion that "software is eating the world".[51] Software developers are also well educated, with at least a university level degree being standard.[52]

Combining these facts with their relation to the means of production, we can suggest that software programmers constitute a layer among laborers who are laterally mobile among the following class positions: When employed by major capitalist software companies, they are among the higher strata of white-collar workers. Considering that the income of capitalist software companies (especially in developed countries) is in the form of rent, programmer wages may be taking a share of that rent, which would render programmers in such positions part of the labor aristocracy. When programmers move towards self-employment and freelancing, they make a trade off, foregoing regularity of work in favor of self-management, which brings them closer to the class position of a modern petite-bourgeois.

> These digital artisans form the so-called 'virtual class': '…the techno-intelligentsia of cognitive scientists, engineers, computer scientists, video-game developers, and all the other communications specialists…' Unable to subject them to the discipline of the assembly-line or replace them by machines, managers have organised such skilled workers through fixed-term contracts. Like the labour aristocracy of the last century, core personnel in the media, computing and telecoms industries experience the rewards and insecurities of the marketplace. On the one hand, these digital artisans not only tend to be well-paid, but also have considerable autonomy over their pace of

51 http://www.wsj.com/articles/SB10001424053111903480904576512250915629460
52 http://www.bls.gov/ooh/computer-and-information-technology/software-developers.htm

> work and place of employment. As a result, the cultural divide between the hippie and the organisation man has now become rather fuzzy. Yet, on the other hand, these skilled workers are tied by the terms of their contracts and have no guarantee of continued employment. Lacking the free time of the hippies, work itself has become the main route to self-fulfillment for much of the virtual class. (Barbrook and Cameron, 1996: 47)

How then can individuals destined towards such a middle-rung position in society play a historic role towards classlessness? The answer is that under the right circumstances, they become independent producers who produce not commodities but free public goods for all. The set of circumstances that brings this about is FS production; when they produce FS instead of software in commodity form. In this case, independent FS producers are compensated not by license sales on the market, but by the several alternative mechanisms I described in the previous section.

There is an element of "class suicide" in this, as those who turn to live by independent FS production forgo the opportunity to receive high wages out of rent income. Stallman, himself a programmer who for a time earned his living through the FS remuneration mechanisms mentioned, is very clear on what this entails: "programmers will not starve... it will still be possible for them to get paid for programming; just not paid as much as now... that is not an argument against the change" (Stallman, 2010: 33). On the up side, FS hackers enjoy more intrinsic satisfaction in their work, as well as increased control over the organization of production and the better socialization experience that results from working in a collaborative community. The programmer who has "converted" to FS is therefore a person who has gone through a *class reincarnation.* Rather than an absolute renunciation of privilege, the alienated money privilege gives way to the experiential privilege of self-actualization and a glimpse of life under a post-class productive collectivity. As suggested by Richard Barbrook and Pit Schultz in "The Digital Artisans Manifesto":

> For those of us who want to be truly creative in hypermedia and computing, the only

> practical solution is to become digital artisans. The rapid spread of personal computing and now the Net are the technological expressions of this desire for autonomous work. Escaping from the petty controls of the shopfloor and the office, we can rediscover the individual independence enjoyed by craftspeople during proto-industrialism.[53]

In FS, the socialization of production and distribution takes unprecedentedly advanced forms. The socialization of production, that is, the organization of the collective effort, is achieved by individual initiatives mobilizing to carry out modularized tasks (Benkler, 2006). Any contribution, however large or small, takes the form of a patch upon the entire body of pre-existing code, that is submitted for inclusion. While each contributor may not have a grasp of the entire immense architecture of a software project (as used to be the privilege of an artisan), they do have an immediate grasp of the effect their patch has on the functioning of the software (unlike the assembly line worker). This is because the individual immediately applies their patch and runs the resulting improved code on their own system, thus verifying the result before submitting. This is a type of *kaizen*[54] inherent in FS that puts to shame lean production teams, quality control units and other simulated versions of autogestion in capitalist production.

Instead of producing IP for the capitalists, the FS programmer produces commons for society as a whole on the global level, coordinating their work in cross-national communities. This introduces a practical universalism into the FS programmer's lifeworld that reaches out towards the dialectical, potential universalism of the proletariat. The position of the FS producer that can live on producing commons for a global public serves as a manifest exemplar for all workers.

In a nutshell then, we seem to have in the subjectivity of the FS producer, an amorphous revolutionary element that has emerged from within the class of software

53 http://www.imaginaryfutures.net/2007/04/16/the-digital-artisans-manifesto-by-richard-barbrook-and-pit-schultz/

54 Japanese term meaning "continuous improvement" pioneered by Toyota. See https://en.wikipedia.org/wiki/Kaizen for related toyotaisms that have superseded fordisms.

programmers in particular, and intellectual producers in general. With autonomous ownership of its own means of production and its cutting-edge forms of coordination, this sublated neo-artisan element appears as capable, in its sphere of production, of rendering its capitalist competition obsolete. As opposed to the small peasant proprietor or the individual artisan, with their archaic instruments of labor and modest purchasing power for the materials of labor, who could not but be descendent, and even moribund, in the face of the ever-concentrating and technically developed forces of production utilized by the industrial capitalist, this element has been increasingly carving out a space for itself from within a domain which was previously under capitalist control. As "The Digital Artisans Manifesto" puts it,

> We create virtual artifacts for money and for fun. We work both in the money-commodity economy and in the gift economy of the Net. When we take a contract, we are happy to earn enough to pay for our necessities and luxuries through our labours as digital artisans. At the same time, we also enjoy exercising our abilities for our own amusement and for the wider community. Whether working for money or for fun, we always take pride in our craft skills. We take pleasure in pushing the cultural and technical limits as far forward as possible. We are the pioneers of the modern.[55]

"The social prophecy" is that the innovative model workforce "announces a new economic and social paradigm" of all future work. "Their mode of being and, in particular, of producing, is set to become hegemonic. No matter how numerically limited at present, the way they live and work today is the way everyone else will live and work tomorrow" (Barbrook, 2006: 7). It is important though to keep in mind that the class reincarnationists will probably remain a minority element within the middle strata that they come from for quite some time. Barbrook warns us against the uncritical, utopian technological determinism of "The Californian Ideology" of Richard Florida's "Creative Class" (Barbrook and Cameron, 1996). The same Barbrook, however, maintains a streak of optimism in his book *The Class of*

55 http://www.imaginaryfutures.net/2007/04/16/the-digital-artisans-manifesto-by-richard-barbrook-and-pit-schultz/

The New (2006). The experimental style of the book makes quoting difficult (as it almost entirely consists of quotations). Its gist is that the elusive "interim class" has historically been depicted as revolutionary, progressive and creative as well as reactionary, oppressive and bureaucratic, by both the left and the right. One of the ways to separate the wheat from the chaff according to Barbrook is to look at their attitudes towards IP. While the "entrepreneurial creative class" is intent on hoarding and is strongly in favor of copyright, the self-organizing and sharing "precarious digital proletariat" comes out against it.

Quite some time before the onset of the "information age", Walter Benjamin had shared certain interesting remarks about the relationship of the intellectual producer and the working class at the Institute for the Study of Fascism in Paris. Quoting from his address titled "Author as Producer" (comments in brackets mine):

> You may have noticed that the reflections whose conclusions we are now nearing make only one demand on the writer [the hacker]: the demand to think, to reflect upon his position in the production process [Stallman's GNU project]. We can be sure that such thinking, in the writers who matter—that is to say the best technicians in their particular branches of the trade [Torvalds's Linux kernel]—will sooner or later lead them to confirm very soberly their solidarity with the proletariat.
>
> The revolutionary intellectual appears first of all and above everything else as a traitor to his class of origin. [Class reincarnation] In a writer this betrayal consists in an attitude which transforms him, from a supplier of the production apparatus, into an engineer who sees his task in adapting that apparatus to the ends of the proletarian revolution [The Free Software Movement]
>
> Will he succeed in furthering the unification of the means of intellectual production? [Free Software] (...) Does he see ways of organizing the intellectual workers within their actual production process? [The GPL] Has he suggestions for changing the function of the novel, of drama, of poetry? [Of software] (...) The more completely he can address himself to these tasks, the more correct his thinking will be and, necessarily, the higher will be the technical quality of his work. [The technical superiority of Open Source development] (Benjamin, 1998: 101-103)

Therein lies the historical continuity from Benjamin to Barbrook, and the revolutionary element in the FS producing class-in-the-making. On this note, I turn to an in-depth analysis of what I identify as the single most revolutionary practice of these new

"intellectual workers within their actual production process" that underpins the entire operation of FS. The most revolutionary idiosyncrasy of the FS mode of production: Forking.

3.3 Forking: The Highest Expression of Productive Liberty

> In software engineering, a project fork happens when developers take a copy of source code from one software package and start independent development on it, creating a distinct and separate piece of software. The term often implies not merely a development branch, but a split in the developer community, a form of schism. Free and open-source software is that which, by definition, may be forked from the original development team without prior permission without violating copyright law.[56]

In FS, and in principle other digital production, forking represents a most idiosyncratic resolution of dissent and disagreement in the community of producers. As a last resort option, forking replaces what in material production organized under *any mode of production* would be the playing out of a contestation over use of machinery, materials of labor, and other items which comprise the means of production of the productive enterprise. That this contestation does not arise allows in the FS mode of production a radically liberated space for experimentation, the pursuit of alternatives, and the expression of differences of opinion in the productive act itself rather than in deliberation.

The beautiful thing about producing digital artifacts such as software rather than physical goods is that alternative directions for projects can be explored without the cost of dividing up and possibly wasting the means of production on bad decisions. One could not have forked a factory or an office in the same manner, because either the machinery and office space would have to be divided up between the original collective and the dissenting group, or either one of the groups would be left with nothing and have to start from scratch or yield.

A fork simply means that a dissenting group of producers remove from the collective production effort their own future labor in favor of developing the software on a different

56 https://en.wikipedia.org/wiki/Fork_%28software_development%29

path. They do this by simply copying the code that has been developed up to that point, and reorganizing their break-away production elsewhere online. The costs associated with a fork is an amount of organizational overhead as the old and new teams reorganize their production management and deal with potential temporary confusion over which project they should continue to contribute to (if any), potential user fragmentation, possible incompatibilities arising, and some brand name confusion.

Forking does not have to actually occur, however, for its benefits to be felt. The simple ever-looming *threat* of the appearance of a fork disciplines any FS production leadership, whether volunteer or profit seeking, to behave as responsively as possible to the mood of the community, including its overlapping producers and users.

> The potential to fork a program has been called 'the indispensable ingredient that binds developers together', as since a fork is bad for everyone, the more serious the threat of a fork, the more willing people are to compromise in order to avoid it. Thus, the potential to fork is also a significant element in the governance of open source programs, as no one has a 'magical hold' over any project since anyone can fork any project at any time. (Nyman, 2014)

Thanks to the open availability of source code and abundance of personal computers and cheap networking, in one stroke the entire industrial era political-economic set of alternatives on how to manage decision making in production can therefore be transcended.

In private capitalist firms, decision making power over production resides in the capitalist (often delegated to managers) who own all the means of production. In state enterprises, the property—with considerable abstraction—belongs to the public through the state, and decision making power is concentrated in the state appointed managers or bureaucrats. The drawbacks of these class society forms of management of workplaces in terms of producing inequality and alienation are well known. Radical schools of the labor movement such as council communism and anarcho-syndicalism sought to provide a solution. They defended the principle that the collective body of workers should make decisions over

production in individual workplaces through general assemblies, employing some form of egalitarian self-management mechanism ranging from consensus to majority vote. These workplaces would then be associated in a bottom-up or federated structure to engage in larger scale productive operations, perhaps with some sort of democratic or participatory planning.

Yet even the most ultra-leftist, anarchist and other avant-garde answers to self-management in collective industrial production were constrained by the physical nature of production, which dictated that once decided upon by whatever process, any particular machine or resource could only be put to use in an exclusive manner, as the materials of labor and a certain portion of the life of the machine gets productively consumed in the process of production. Whether by managerial fiat or by democratic means, one general, binding decision had to be reached on how to productively consume the finite resources at hand, in exclusion of all the other potential uses of said resources. A general will (if not the will of the general) over production had to emerge, and dissenting views had to give way at some point. In commons-based digital production exemplified by FS, this entire restriction is obsolete. Any dissenting group of producers can withdraw in order to pursue an alternative path of production, without demanding that a share of the machinery or resource assets of the collective be transferred to them.

With its internalization of forking, the FS mode of production supersedes all prior radical arrangements and appears as truly post-democratic in the sense of achieving a sublated version of democracy. As there is no scarcity of means of production in FS due to the inexpensiveness of computers and network access, and no scarcity of outputs once a product is yielded due to digital reproduction, the political character of production is confined only to the question of how *living labor* will self-organize. Unlike its physical cousin, *digital dead labor* cannot weigh upon living labor. The nightmarish hold upon the living of the *tradition* of

past generations is broken.

What determines the success of a fork and the fate of the original project is roughly how many developers and users choose which. First the developers and then users in effect vote with their feet, and the outcome arises organically. There are roughly four possible eventual outcomes of a fork: 1. the death of the fork. 2. a re-merging of the fork. 3. the death of the original. 4. permanent branching (ibid). Only forks which turn out to correspond to real needs for different use-values will eventually survive (whether as a separate branch or by being re-merged into the original), as opposed to frivolous forks whether born of character clashing immaturity or commercial interest idiocy (ibid).

Let us be sober about the meaning of this. Forking is not a happy affair by any means. It is the last resort for settling disputes within a FS project. What is notable is the non-destructive, non-violent nature of even this "nuclear option", compared to the headaches associated with serious dissent that arises in analog production.

Studies of attitudes among hackers on forking have provided conclusive empirical confirmation of this principle. "Interestingly, while all programmers noted potential downsides to the right to fork, it was seen by all as an integral component of open source software, and a right that must not be infringed regardless of circumstance or outcome" (ibid). "The practice of forking has been both complimented and condemned, described as a 'Good Thing', a 'Bad Thing', open source software's 'cardinal sin', and a vital part of open source software. Code forking appears to be both loved and feared" (ibid). These attitudes arise because forking is at once both a regrettable admission of non-unity, of the unreachability of a viable compromise, and at the same time its least-damaging resolution (compared for example to debilitating endless quarreling). Neither is every instance of forking rational, as no mode of production that still involves humans can transcend human character: "Many times [a fork is]

about people colliding and not features colliding, and that shouldn't be, but... we are human" (ibid). It is still a good idea "to think before you fork" (Robles and Gonzalez-Barahona, 2012: 4).

According to Gregorio Robles and Jesus Gonzalez-Barahona, the following are situations that have merited forks in FS: disagreement over technical functionality, disagreement over fundamental direction, discontinuation of the original project, conforming to diverse laws, desire for more community-driven development, implementation of a commercial strategy (ibid: 5-6). The last two items in this list are usually related, as it reflects the tension within FS between the inclinations of communities as a whole and the commercially motivated elements within those communities. Forking also acts as a means of articulating such conflicts into productive practice. Robles and Gonzalez-Barahona also found that forking was pervasive on all levels of FS development, from kernels to end-user software, and that communities are more eager to resort to forking in the tug of wars between community interest and commercial interests, with community-driven forking representing 13% of cases in their survey against 9% of cases being commercial strategy forks.

PART II - ETHNOGRAPHY

In Chapter 4 and Chapter 5, I will share a multi-sited ethnographic account of Pardus developers and the larger Turkish FS community of which they are a part. Accordingly my ethnographic study has two loci.

With respect to the latter, I have attended a series of conventions that take place regularly and bring together individuals from all walks of the Turkish FS community. The data has been collected through my attendance to these events between 2011 and 2015 in Istanbul, as well as video recordings[57] and presentation notes[58] of various sessions that date back to 2000. The main large scale events where these sessions were hosted were the annual Free Software Days and Academic Informatics Conference. Additionally, there were also Free Web Technologies Days and Document Freedom Day, which functioned as smaller thematic versions of the Free Software Days, with mostly overlapping speakers and audiences. Almost all of the presentations made at these events were archived by the Linux Users Association's Seminar Working Group.[59] This archive includes textual and video documentation of more than 500 seminars, which I used in addition to the nearly 40 sessions that I was able to observe in person. Of these, I filtered out technical presentations on specific pieces of FS and other free technologies, which constituted the majority, to focus on the occasions where individuals of the FS community spoke about and discussed the political, social and economic aspects of FS as they see them. This has narrowed the pool of material that I draw on in this section to about 30 sessions. I have supplemented this material with information on the individuals gathered from their personal blogs, social media sites and company or university

57 https://seminer.linux.org.tr/seminer-videolari/
58 https://seminer.linux.org.tr/seminer-notlari/
59 https://seminer.linux.org.tr/

web pages.

In addition, I attended Istanbul Hacker Space (IHS) events as a participant observer over the course of six months in 2012-2013. IHS is a physical space founded under the aegis of the Alternative Informatics Association, which serves as "a workplace to have fun with the technical facilities gathered together by the contributors and a social environment to work together and create events for the people who are interested in computers, electronics, technology, Internet, DIY and such subjects."[60] I also took part in a project to create a digital inventory of books and gadgets available at IHS using FS and contributed to the translation of written material for the Document Freedom Day festive event hosted there.

With respect to the Pardus project, which is the largest experiment with FS production undertaken in Turkey by the Scientific and Technological Research Council (TÜBİTAK), my ethnographic work included semi-structured in-depth interviews with 14 developers[61]. The Pardus project, which lasted effectively from 2003 to 2011, employed developers whose numbers varied from 4 to its maximum of 35. My interviewees consist of full-time developers one of whom was a community manager, part-time paid developers, interns, and volunteer developers one of whom was a university professor of computer science who functioned both as a community spokesperson and as a key connection between Pardus and his students, some of whom eventually became paid Pardus developers. In addition to interviews, I also conducted an online ethnography of Pardus developers' blogs and social media sites to gather data on their views and reflections concerning the Pardus project and details of their life stories. I also visited the TÜBİTAK campus located at Gebze, Istanbul, which contained the work site of the paid Pardus developers.

My general approach to the ethnographic method was inspired by George Marcus. In

60 https://istanbulhs.org/wiki/
61 The names of the interviewees have been changed for reasons of anonimity.

an article revisiting his 1995 essay on multi-sited ethnography, Marcus proposes an ethnography which will go beyond "description for an archive, or reportage for an academic audience, to the performance of mediations of found perspectives in multi-sited space," where "the engaged reflexive subject . . . cannot be a mere informant or subject of research, but in some sense, must become involved in its intellectual work and scope" (Marcus, 2007: 1133). Such a call is in keeping with the new modes of anthropological engagement in which, according to Boyer, "ethnographers have come to be interested in professionals, intellectuals and other experts as interlocutors, in knowledge practices, networks and institutions as sites of ethnographic inquiry" (Boyer, unpub: 3). Ulf Hannerz shares this interest in calling for "studying sideways," treating ethnographic interlocutors not as data-delivering "informants" but more as collaborative "allies" or "epistemic partners" (Hannerz, 1998). It is perhaps not a coincidence that one of Marcus's principal examples for such anthropological work with interlocutors as equal partners is Chris Kelty's work on the open source movement (Marcus, 2007: 1140). The different actors in the open source community participate in a reflexive culture of expertise and political engagement, which makes it particularly necessary and fruitful to define the terms of the research in dialogue with them and to "blur the boundaries between the field site and the academic conference or seminar room" as suggested in the website of Marcus's Center for Ethnography Initiative at the University of California-Irvine.[62] In this spirit, I suggest to dub my particular approach "peer ethnography", in the sense that the subjects can be broadly seen as peers of the ethnographer.

62 http://www.socsci.uci.edu/~ethnog/

CHAPTER 4: SPACES OF CONVENING

4.1 Physical Spaces of Convening

According to Gabriella Coleman, recurring events bound to physical spaces are important for showing how hackers constitute publics that go beyond the cliché of a virtual community, allowing an in-person celebration of their virtual translocality: "More than ever, hackers participate in and rely on a physical space common to many types of social groups (such as academics, professional groups, hobbyists, activists, and consumer groups): the conference, which in hacker lingo is usually designated by its shorthand, the 'con'" (Coleman, 2010: 49). The social experience at the physical space then acts back upon online space: "If immediacy and immersion set the tone of the con experience, as soon as one leaves, a new experiential metabolism takes its place: one of heightened reflexivity" (ibid: 63). In the following two sections I will explore the dynamics of these in-person gatherings and interactions in physical space through the description of the annual Free Software Days convention and a local hackerspace in Istanbul..

4.1.1 The Annual *Free Software Days*

The annual Free Software Days event is organized by the Linux Users Association. This festive conference event constitutes a wide tent that brings together FS, Linux, Internet freedom/privacy activists and others who can express their aims and positions fully and freely, side by side with large and small commercial interests around FS, as well as academics and the occasional governmental presence related to Pardus. The reflexivity that Coleman refers to was observable following the Free Software Days, when participants often duly reported their

thoughts and experiences on their personal blogs, especially with regard to issues around Pardus.

The majority of the panels and presentations at these events are technical and apolitical (related to use and development of pieces of FS). These organized knowledge-sharing events can be considered part of the self-activity of the FS-oriented members of the digital artifact-producing class. A typical example would be something semi-accessible like "Designing an Event Poster with Free Software" or the more specific "Processing SVG Vectoral Graphics under the QT Framework". These are educational sessions, as Coleman mentions in "The Hacker Conference":

> Cons offer ample opportunity for individuals to present their own work or interests to a larger audience. After laboring either in isolation or with others but only online, developers feel a rush of pride and honor in presenting their work to a roomful of collaborators and peers who are keen and interested to learn more or lend a helping hand. (Ibid: 57)

Coleman points out, however, that talks also "usually span multiple topics: legality, politics, cooperative sociality, and even the anthropology of their project" (ibid). At the Free Software Days, there is a minority of sessions that are of an explicitly political or evangelizing nature and are balanced by usually business sponsored sessions that focus on a specific piece of Free Software or open technology that is developed or pushed by a corporation in the IT field.[63] Examples of evangelism are panels such as the practical "Transition to Linux on Institutional Desktops" or the more ambitious "GNU/Linux: The Operating System that Changed the World". Examples of a panel categorized as commercial would include both a directly sponsored one like "Markafoni: An E-commerce Firm's Trial by Free Software" or an independently organized panel that is specific to a commercial enterprise

63 Free Software Days sessions according to categorization between 2011-2014. (Panels spanning multiple sessions are counted multiple times) E: Evangelist/Activist T: Technical, C: Commercial G: Government. 2011: E=9, T=33, C=8, G=1. 2012: E=10, T=50, C=7, G=0. 2013: E=4, T=44, C=8, G=1. 2014: E=7, T=19, C=3, G=0.

such as "Facebook App Development Workshop". Academic-oriented sessions that either present supportive philosophical and sociological insights on the FS phenomenon (e.g. "Free as in Freedom: A brief History of the World in terms of Freedom") or that are about specific FS tools useful to the academic community (e.g. "Linux for Academics") are also in the mix. These too ought to be considered as either part of the activist-evangelistic effort, or knowledge and experience sharing self-activity of knowledge producers.

In these conventions, Linux and FS activists have the leading role as organizers. The main organizer, the Linux Users Association, started as a mail-list in 1993, with its first face-to-face meeting taking place in 1995. In 1997, the association published the first book on Linux in the Turkish language. Over the following years, the association expanded its yearly convention from a single conference room that was part of the Academic Informatics Conference, to its first own yearly conference in 2002. The association runs the linux.org.tr website, an introductory resource about Linux and Free Software, and hosts Linux Planet, a blog portal that consists of posts by a large number of individuals on issues related to FS and Linux. The membership number is stated to be "in the hundreds".[64]

The Free Software Days events in recent years take place on Bilgi University campuses, which are either close to or have free shuttle access from city centers. Bilgi University is a private university ("foundation university" in official Turkish parlance), that serves a mix of higher income group students (majority) and lower income students with scholarships (minority). It is known for its liberal leaning institutional culture. If as Gabriella Coleman states, "the cultural ethos and class of a group is inscribed in where they are willing to meet" (Coleman, 2010: 67), the Bilgi University venue suggests a mid-way point between the counter-cultural and the posh. The key person in Bilgi University for organizing the hosting of the event has been the socialist figure Chris Stephenson, then head of the Computer

64 The Linux Users Assosciation. http://www.lkd.org.tr/hakkimizda/tarihce/

Science Department.[65] Stephenson was demoted from head of the department to lecturer in 2012, after a disciplinary inquiry related to an internal email he sent in support of a conflictual unionization drive of workers at the university. He is also an editor of the left-wing marx-21 website.[66] Stephenson has not only been instrumental in securing the venue for the activities, but also regularly delivers his own keynote presentations related to FS at the opening panel of the conferences.

During my attendance to the Free Software Days events between 2011 and 2015, I routinely sighted familiar faces known to me from interviews as Pardus coders. Some others I had come to know by name through the Turkish FS-related online milieu. There is considerable overlap between Pardus coders and contributors and the attendees of these events in either organizing or participating capacities. Attendance to these events has always been free of charge and open to the public. Event websites that advertise the schedule and participants of the events always pop-up about a month ahead of time each year, where both commercial sponsors and NGO's and activist initiatives at the convention are featured.[67]

From personal observation I would estimate that about a thousand people attended the Free Software and Linux Days event every year, with the most popular panels drawing up to many hundreds, while the smaller events draw a few dozens to about a hundred people. As per the geek stereotype, the male dominance was overwhelming. A ten to one or even twenty to one ratio of men to women was common occurrence. Long haired, beard & belly sporting men typifying the old school hacker aesthetic intermingled with professionally dressed suit-wearing types (gender representation was much less skewed among these professionals) manning the corporate stalls that advertise their commercial FS product or seek to recruit

65 http://www.bilgi.edu.tr/en/directory/people/chris-stephenson

66 http://marx-21.net/ The website appears to be internationally affiliated with the International Socialist Tendency, a Trotskyist organization. Domestically it seems to be politically close to the HDP, the Pro-Kurdish and minority rights umbrella Party that is the 4th largest political party in Turkey. HDP received 13% electoral support in the June 2015 general elections.

67 http://www.ozguryazilimgunleri.org.tr

talent to their FS based businesses.

The physical event space would usually be mostly taken up by such corporate stalls. In between, one could always come across the left-wing BİÇDA (Informatics Workers Solidarity Network) booth that seeks to organize white-collar workers in the sector - "IT Workers of the World, Unite".[68] This juxtaposition was an embodiment of the political-economic conflict between IT workers and their employers, who otherwise may find some common ground in the production of the software commons that is FS. For capitalists who produce FS, FS is merely a rational investment decision based on their specific business models I mentioned in previous parts of this dissertation. In general, the workers they employ are still exploited as workers and the workers experience (albeit reduced) alienation. In this context, it is only natural for the digitally laboring workers to want to organize themselves against the corporate pole of interest, FS-based business model or not. Thus the corporate stalls and the workers' organizations' booths spring up next to each other at the Free Software Days convention. Furthermore, that workers' rights organizations like BİÇDA find a welcoming platform at these conventions suggests an implicit *affinity of values* between FS enthusiasts and the left, very much in line with Coleman's observations on how "Italian anarchists work alongside US liberal democrats" at hacker conventions (Coleman, 2010: 62).

Another usual suspect worth mentioning at these events is the Alternative Informatics Association.[69] This is an independent organization, funded by its individual members and donations, that focuses on Internet freedoms and privacy. It has published several important books such as *Brave New Media*, *An Informatics Guide for NGO's*, *Digital Surveillance in Turkey*, and *Hack Culture and Hacktivism*.[70] These are available free of charge in e-book formats under Creative Commons licenses, and I have often observed them being distributed

68 https://twitter.com/bicda
69 https://www.alternatifbilisim.org/wiki/English
70 http://ekitap.alternatifbilisim.org/

in paperback form in exchange for "name your own price" donations as well. The Alternative Informatics Association was among the leading organizers of the massively successful May 2011 "Hands off my Internet" protests in Turkey against online censorship and the planned introduction of internet "filters" (legally required opt-in services to be offered by ISP's with predefined black-lists of censored content) that fortunately never ended up being mandatory. The protest gathered a couple hundred thousands of very loud and very colorful protesters to Taksim, Istanbul as well as 15 other city centers across Turkey. The Facebook "attending" list numbered in the 600.000 range. As a participant, I can attest in retrospect that the mood and composition of the protest clearly and directly foreshadowed the 2013 Gezi Protests against creeping authoritarianism, that shook Turkey for two weeks.

During the actual Gezi Protests, the Alternative Informatics Association put out a flurry of press releases, analyses and supporting statements. People were being picked up from their homes for tweeting, and thus was the lesson of the importance of online anonymity learned by a large public. The relevance of the Alternative Informatics Association only increased with the countrywide block on YouTube and Twitter imposed by the government during the late 2013 Graft Scandal that implicated the Justice and Development Party (AKP) government. This in turn served as a crash-course for half the country on how to circumvent internet censorship, with DNS servers, VPN's, proxies and TOR becoming household concepts. And who better to disseminate to the public reliable know-how around these issues in Erdoğan's "New Turkey" in a post-Snowden world than the same beardos manning the Alternative Informatics Association booths and ex-Pardus geekfolk that are occasionally the same individuals and generally their fellow travelers? The relevance of the activism conducted by the Alternative Informatics Association reached its maximum during Turkey's turbulent year of 2013 and is still going strong.

4.1.2 Istanbul Hacker Space

Istanbul Hacker Space (IHS),[71] which is founded under the aegis of the same Alternative Informatics Association, resides in the outer edge of Kadıköy, the Asian center of Istanbul, in a less-than integrated, gray commercial periphery near the terminal Metrobus station, where rents are cheaper than the bustling Kadıköy center. As "Maxigas" points out in an insightful comparative article on the history of hacklabs and hackerspaces, one of the fundamental distinguishers between the overtly radical hacklabs and the more moderate hackerspaces has been "the legal status of the spaces: while hacklabs are often located in squatted buildings, hackerspaces are generally rented."[72] On this point, the IHS is definitely true to its name and on the hackerspace side of the divide. However, the somewhat marginalized urban location of the rented space betrays a tightness of funding; this is not a boutique part of the district. As a modernizationist middle class stronghold, Kadıköy politics is defined by a progressive blend of a Kemalist (secular, cultural-nationalist) majority that tolerates and is home to various small groupings of the radical left and the occasional secular ultra-nationalist oddball. The Kemalist center-left Republican People's Party (CHP), the main opposition party of Turkey, enjoys overwhelming electoral support against the ruling religious conservative, neoliberal AKP.[73] Furthermore, Kadıköy was one of the hotbeds and main center of anti-government mobilization on the Asian part of Istanbul during the 2013 Gezi Park protests. Second only to the European city center Taksim, Kadıköy has always been one of the most dynamic loci of urban subcultural activity, and it is no coincidence that the Hacker Space has chosen to set up shop there.

71 https://istanbulhs.org/english/
72 Maxigas - 2012 - Hacklabs and Hackerspaces p.1
73 %58 (CHP) vs. %25 (AKP) in the 2011 General Elections. %73 (CHP) vs. %20 (AKP) in the 2014 Local Elections. %72 (CHP) vs. %22 (AKP) in the 2014 Presidential Elections.

The IHS is at the end of a small dead-end street. Upon walking towards the edge of the small, gritty dead-end adjacent to the IHS, you are greeted by a wall covered with subcultural geek graffiti depicting retro computer game graphics, and a banner denouncing online censorship. The "tradition of setting up artistic installations in various places in and around the building, the most striking example being (...) graffiti" (Maxigas, 2012: 4) is alive and well. In the corner is a makeshift cardboard home constructed for the benefit of neighborhood cats. The small street is surrounded by some small residential buildings, a few stories high, with their ground floors serving as various commercial enterprises, usually storehouses or small business headquarters. With no storefronts or window shopping opportunities, the adjacent streets are almost always empty: This is not the kind of space you might run into during a weekend stroll; you have to know it exists to get there (probably through the istanbulhs.org website or mail-list). I suspect that this contributes to the rather introverted nature of the activities that the space is generally used for.

The individuals who frequent this place are almost always of the social strata of struggling white-collar workers looking to improve their skill sets through workshops, always on the lookout for freelance work opportunities, or even the occasional copyright-infringing hustle. Some of these are the more engaged Alternative Informatics Association activists sporting their proud beards as they put in work for the cause. One such (unbearded) permanent is a young ex-Pardus hacker. I ended up collaborating with him on a project to build an inventory of items available for lending, mostly books and some computer hardware and hand tools using the web-library building features of Calibre, which needless to mention, is FS. In the process, I was schooled a little bit on installing and updating software on his Debian Linux powered laptop.

The physical space consists of two stories of about 100m2 per flat. Most of the public

activities and general hanging out happens on the top floor which is at ground level, while the "dungeon" is one level below ground, under-utilized as storage space at the times of my frequenting in late 2012 and early 2013. Upon immediate entrance there is a lounge area for rest and socialization, and to one side you are greeted by the quintessential WW2 era "We Can Do it" poster depicting a young woman flexing her muscles. Below the poster lies a small number of hand held mechanized tools such as electric screwdrivers, a soldering rig and drilling machine.[74] In an other corner is a fledgling attempt at building a library out of donated books that are relevant to the issues of coding, hacker spaces, maker culture, etc. The Creative Commons licensed Alternative Informatics Association books I mentioned earlier are among those in the best condition. At the deep end of the top floor is the conference table and a projector. This is where the workshops and other events take place. A curtain screen separating that area from the casual space at the front was later added. Curiosities in the form of a hand-built pacman-clone cabinet and a desktop robot round up the scenery. Turkishness prevails in the permanently operational kettle brewing Turkish black tea. I observed the mood of the regulars to be between bored and playful in off-hours, but lively and focused during events and while working on improvements.

I participated in some of the day-to-day activities of the IHS as a volunteer and attended the open public events that were accessible to non-coders. The nature of the events that took place at the Istabul Hacker Space mostly mirrored the culture of the Free Software Days conventions. Activities directed towards skill-sharing among coders in the form of workshops such as "Basic Git Versioning System", "Peripheral Interface Controller Programming", "Android Programming Workshop" were in the majority. However, more accessible and intriguing initiatives that were open to the public were in no short supply,

74 Looked down upon as an inept social scientists with lower back problems who would have difficulties with any kind of manual labor, on one instance I was encouraged to take a shot at drilling a hole in a high spot on the wall with the drilling machine in order to hang up a board. I believe I succeeded in the ordeal.

either. These include screenings of films on hackers, a show and tell of a device that measures tear-gas concentration in the air (think Gezi), "The 3D Hubs 3D Printing Summit", "What to Do When They Fully Cut Out the Internet—A Meshnet Workshop" (think Graft Scandal era censorship in extremo), "Document Freedom Day Party", an introductory teach-in on the merits of open standards for document formats over proprietary ones, a "CryptoParty", another invaluable event that introduced the attendees to the basic principles behind secure end-to-end encrypted digital communications over untrusted channels such as the Internet, which was delivered by none other than two brilliant ex-Pardus developers. Also worth mentioning is the recurring Ada Lovelace[75] series of workshops oriented towards attracting more women to programming.

Maxigas stresses the importance of policies and practices of inclusion and exclusion for hacklabs and hackerspaces (Maxigas, 2012: 7). While Maxigas's focus was on the level of tolerance towards undesirable attitudes such as sexism and racism, the problem of access in the IHS example arose out of the economic issue. An ever-present problem during my period of engagement with the IHS was the shortage of funding. Since it had opened, the IHS had been relying on donations and on payments the core group of regulars were making out of their own pocket. The hope was that as the IHS got more popular, the donations would rise to a level sufficient to sustain its operation. This failed to happen month after month, and as the balance sheets turned deeper red, a structural change was implemented. Instead of the open access and donations system, the decision was made to introduce membership dues and limit public access to the space to weekly public events that lasted for set amounts of time. Events such as workshops and teach-ins also started to be advertised with a "recommended donation"

75 "Augusta Ada King, Countess of Lovelace (10 December 1815 – 27 November 1852), born Augusta Ada Byron and now commonly known as Ada Lovelace, was an English mathematician and writer chiefly known for her work on Charles Babbage's early mechanical general-purpose computer, the Analytical Engine. Her notes on the engine include what is recognised as the first algorithm intended to be carried out by a machine. Because of this, she is often described as the world's first computer programmer." http://en.wikipedia.org/wiki/Ada_Lovelace

notice attached. The retreat from the ideal of open access and community space towards a more club-like basis was an unhappy turn of events, which serves as a reminder of the cold realities the hacker ethic faces in the face of capitalism. For proponents of the FS movement, compared to the unique advantages of the realm of digital production, it is much harder to persevere in defense of the values of openness and freedom in meatspace.

4.2 The Discursive Space of Convening

The many conventions and conferences in the field of IT where proponents of FS find venue present a concentrated overview of the Turkish FS community as a self-constituted public as well as the major discourses on FS. The most important of these conventions is the annual Free Software Days, of which Pardus (TÜBİTAK) was among the sponsors. These conventions bring together the following types of actors, with individuals often moving between these subject positions over time: paid and volunteer Pardus developers, TÜBİTAK management representatives and government bureaucrats, small and medium-sized FS business entrepreneurs, organic FS intellectuals, hacker-activists with various political colors ranging from Kemalism and economic nationalism to libertarianism, leftism, Marxism and feminism, academics theorizing on FS, computer science professors and interning students, and finally Pardus and FS evangelists.

In my interpretation of and commenting on the first-hand and archival data on the presentations, I have not shied away from violating the conventional epistemological distance of the anthropologist from its sacred subject, in order to engage with my subjects as intellectual equals. I believe this "peer ethnography" is warranted because what I have done is not a study of my subjects in their lifeworlds, but of their own public engagements where they present their own reflective analyses, advice, and criticisms to open public audiences. This

blurring of subject of study and intellectual equal is both fruitful and honest. On the opposite end, I have limited the scope of my study to the circulation of ideas within the public sphere enabled by conferences and conventions rather than expanding into a review of the full theoretical literature that some of these individuals have produced elsewhere. The aim of this approach is to inform my theoretical study of FS as a political and economic phenomenon that I presented in previous chapters with empirical observations on the (often fluid) economic positions and political tendencies of actors within the general Turkish FS setting.

4.2.1 FS and Business: Entrepreneurs, Interns and SME's

Hakan Uygun is the founder of Uygun Teknoloji (Uygun Technology) and later a co-founder of Özgür Yazılım A.Ş. (Free Software Inc.). He manages the consultancy services of the company and is responsible for solutions in software development services. He has headed the Linux Users Association from 2010 to 2012.[76] Özgür Yazılım A.Ş. "Produces products and services using primarily free and open source software... It develops and integrates software and provides support and hosting services. It tries to publish and share its accumulated knowledge and the software it develops in a way they can be accessed with freedom".[77] The company develops a number of FS. There is an accounting and commercial automation software called Tekir, for which it sells support services,[78] and a project management software called Jonglr, for which users can buy tiered hosting, installation, training and consulting services. It also sells system administration, software development, training and consulting services that focus on FS. Among the Özgür Yazılım A.Ş. management team is co-founder Doruk Fişek, a prominent figure and past head of the Linux Users Association, who has also been a voluntary Pardus contributor. Finally on board is Onur Küçük, who has been a high

76 http://www.ozguryazilim.com.tr/yonetim
77 http://www.ozguryazilim.com.tr/neler-yapiyoruz
78 http://www.tekir.com.tr/

profile Pardus developer, working for TÜBİTAK as expert researcher, consultant, project manager and team leader.[79] Özgür Yazılım A.Ş. has a strong emphasis on encouraging internships, stating that interning students will be preferred for hiring.[80] They took in 11 interns for the summer of 2014, all of whom were undergraduate students of computer and software engineering departments.[81]

Uygun made a presentation in 2010 titled "Free Software Business Models and Students". He started out by stating that "there is no shame in making money off FS" and that "FS only lives if there is economic support". These two statements must be immediately scrutinized. First, while there indeed may be no shame in making money "off FS" depending on how this money is made, and while indeed FS is not necessarily anti-commercial (a point repeated many times by Stallman and the FSF), this expression is too vague to describe what exactly we have at hand here. Since FS is in essence not sold per se, the question arises: Is money to be made as compensation for the labor of producing FS through a mechanism other than a sale, or is it to be made through income earned by linked activities such as selling support services? Furthermore, will these linked services be produced by waged laborers generating profit for the owners of the company? (It will soon be revealed that Uygun has this latter case in mind). These are defining differences of the relations of production in effect and cannot be captured by the phrase "making money off of". This pervasive lack of lucidity appears in even critical literature on FS and is a major source of confusion that I wish to clear up with my study. Second, it is manifest that FS does not necessarily "only live if there is economic support" the way Uygun means it. What Uygun means and his company exemplifies involves the alienated means that I have touched on in my discussion on Circle 2 remuneration mechanisms. We know, however, that FS can be produced directly for the

79 http://www.ozguryazilim.com.tr/yonetim
80 http://www.ozguryazilim.com.tr/biz/kariyer
81 http://www.ozguryazilim.com.tr/2014-yaz-stajyerlerimiz-belli-oldu

producer's need (Circle 0) and also numerous all-volunteer FS projects have existed and continue to exist in models that in some ways resemble a gift economy (Circle 1). Furthermore, Circle 1 type remuneration (a variation on remuneration in kind, in the form of common access to software produced by others), which should also be seen as a form of "economic support", is always operational, even when subsumed under other circles.

Uygun went on to list several ways of making money around FS such as selling installation, training and management services, consultancy, support and custom solutions. Having established the promise for gainful engagement with FS on these terms, he delivered his main point. He wants to encourage students to start contributing to FS projects as a form of alternative internship. This "interning at the FS commons" as I call it, offers several benefits over conventional internship according to Uygun: choosing what to contribute to instead of being assigned a task, earning experience on real-world software problems, facing user demands directly, and a convenient proof of public work that can be more easily advertised in a CV compared to a record of opaque internal company work done at a standard internship. Uygun suggested that aspiring programmers can contribute to FS through testing, bug reporting, helping other users, writing documentation, bug fixing and coding. It was not made clear in the presentation whether Uygun had in mind students contributing to any odd FS, or to specific FS that his company leads in producing. Considering Özgür Yazılım A.Ş.'s appetite for student interns, it is reasonable to assume that he would at least hope for the latter. In any case, a particular intersection of business interests with the production of FS is revealed in this proposal. Aspiring students can intern at the FS commons, thus providing, for a limited time of their early career (the time deemed appropriate for increasing their future employability), a certain amount of labor input to FS production that enlarges the FS commons.

The connection between FS and internship is not limited to the local scale. To the contrary, some massive initiatives to mobilize intern labor for the production of FS are operating on a global scale. Onur Küçük, who is now on the management team at Özgür Yazılım A.Ş., was previously one of the top developers of Pardus employed by TÜBİTAK for 7 years.[82] He described in a 2009 presentation the ethos of open source code development as "sharing, communicating, team work and the pleasure derived from what you produce getting used". In his talk titled "How to Become a Pardus Developer" (not to be confused with Murat Eren's talk from 2007 with the identical title) he advised individuals to start with any kind of contribution: translations, helping new users, creating documentation, theming, bug reporting and finally package maintenance and coding. He also recommended interning to work on Pardus at either TÜBİTAK or as part of the Google Summer of Code (GSOC) program. GSOC offers students stipends to work on various open source projects every summer:[83]

> Since its inception in 2005, the program has brought together over 8,500 successful student participants from 101 countries and over 8,300 mentors from over 109 countries worldwide to produce over 50 million lines of code. Through Google Summer of Code, accepted student applicants are paired with a mentor or mentors from the participating projects, thus gaining exposure to real-world software development scenarios and the opportunity for employment in areas related to their academic pursuits. In turn, the participating projects are able to more easily identify and bring in new developers. Best of all, more source code is created and released for the use and benefit of all.[84]

Independent of the particular interests behind it, interning at the FS commons contributes to the shrinking of the market for the capitalistically produced proprietary competitor of the FS in question, replacing it with a FS commons. Although this provides a boost to the business position of particular capitalistic companies such as Özgür Yazılım A.Ş. that sell services around this FS commons, looking at the total picture from the perspective of society as a whole, the software market is shrunk in favor of commons, while the total size of

82 https://www.ozguryazilim.com.tr/yonetim
83 https://www.google-melange.com/gsoc/homepage/google/gsoc2015
84 https://www.google-melange.com/gsoc/document/show/gsoc_program/google/gsoc2015/about_page

the support market remains the same. From the perspective of the FS-based company, the move to encourage the production of FS is a bid to get a larger slice of the support market pie to the detriment of a support seller for the proprietary competitor, while foregoing any attempt to capture the software market pie, and in fact reducing the size of that pie. Thus, the total amount of profit-making economic activity in society is reduced, while some profit-making activity in this smaller total shifts from one capitalist to another. This is a net benefit for society in terms of increased common wealth, and a reduction of the sphere of capitalist profit making.

From the interns' perspective, several scenarios may play out. At worst, by interning at the FS commons, they will be contributing to the commons just as much unpaid or underpaid labor as they would otherwise contribute to a capitalist enterprise, working on what the enterprise assigned them. Thus they will be no worse off. At best, they will contribute to a FS niche of their own choosing, in accordance with their intellectual passion, in other words with reduced alienation. (They were forced to work on *something* for their career, but they were able to choose *what* to work on.) Perhaps they will also be using the software that they are producing, thus receiving directly the fruits of their own labor (Circle 0), which would not be the case for an internship at a proprietary software company, where the software they produced would become the intellectual property of the company.

This form of labor supply to the FS commons is arguably marginal. We must recognize that the very institution of internship in capitalist societies is open to criticism, and if this form of unpaid exploitation is eventually abandoned in the economy in general, it will probably take away along with it the phenomenon of interning at the FS commons. Nevertheless, the temporary career-building FS production activities of young programmers hoping for eventual employment by a capitalist company is a real, if marginal source of labor flowing into the FS

commons.

Interestingly, two years later in 2012, Serdar Dalgıç delivered a talk that was practically identical to Uygun's presentation, based on the same FS-as-internship point, under the slightly different title "Free Software and Making a Career in Professional Life". This turned out to be a fruit not of plagiarism, but of professional collaboration. Dalgıç was a paid Pardus developer at TÜBİTAK until he quit in mid-2012, citing in his personal blog, "21st Century Schizoid Man", "TÜBİTAK's inconsistent approach to Pardus Linux".[85] He mentions that while finishing his master's degree on Software Engineering, he attended the 2012 Linux Users Association Summer Camp, where he wrote "an integration suite to Tekir General Financial Accounting program, developed by Özgür Yazılım Inc",[86] and adds: "Hakan Uygun, who is the CTO of Tekir and one of the managers of Özgür Yazılım Inc., helped me a lot while selecting this project".[87] The educational summer camp is revealed to have played a similar role to internships, where FS that was particularly beneficial to Özgür Yazılım Inc. was produced without pay. It also seems that this rubbing of shoulders explains the identical nature of the presentations by Uygun and Dalgıç. Dalgıç later worked briefly as a freelance software developer before getting a job in late 2012 at the shopping site Markafoni, where the "IT crew is interested in open source software, especially Python and Django".[88] Dalgıç's career changes in 2012 are particularly worthy of notice because they exemplify how the highly employable and horizontally mobile class position of software programmers is further diversified by FS. Within the span of a single year, Dalgıç moved from being employed at the most prestigious government FS development job in Turkey to producing FS without pay for a small private FS company, to independent freelancing, to working for a large commerce

85 http://www.serdardalgic.org/2012/07/briefly-in-my-life/
86 ibid.
87 ibid.
88 http://www.serdardalgic.org/2012/12/shortly-at-the-end-of-2012/

company that made use of FS internally.

Mahir Aşut holds a business administration degree and is the founder and company manager of EkoPC ICT Services, which provides "end-to-end ICT solutions".[89] The services provided by EkoPC include Internet Services (web design, e-commerce, etc.), Operational Services (computer and POS maintenance and repair), Authorized Computer Repairs for Fujitsu, Corporate File Servers which "depend on Open Source Software",[90] Electronic Card Repair, as well as Outsourcing and Recruitment.[91] Aşut's CV cites expertise on several FS such as "GNU/Linux operating system management" and "Oracle and MySQL Development and Management" and membership in the Linux Users Association Turkey (LKD).[92] Aşut's company is an example of the FS business model of coupling FS with a hardware solution (the corporate file servers) and for selling services and support for FS. Therein lies EkoPC's business interest in FS.

Aşut appeared at the 2011 Free Software Days with a presentation titled "An Introduction to the Use of Free Software at Large Institutions" (2011). While he felt the need to start off with a feel-good statement that "the philosophy of FS goes beyond the money aspect", his main theme was indeed how FS fits into a particular model of generating income. His argument was that cheap bandwidth and ubiquitous internet access enabled by a myriad of devices beyond the traditional client PC led to the market share of internet-based Software as a Service (SaaS) solutions to increase. According to Aşut, the market share of FS is growing along with SaaS, because a majority of SaaS solutions are based on FS. However, SaaS, or more accurately Service as a Software Substitute (SaaSS), is one of the rather questionable ways by which capitalists are trying to shift the terrain of the class relations in the realm of

89 http://www.ekopc.com.tr/index.cgi?sayfa=hakkimizda&lang=EN#
90 http://www.ekopc.com.tr/index.cgi?sayfa=hizmetdetay&id=4&lang=EN
91 http://www.ekopc.com.tr/index.cgi?sayfa=hizmet&lang=EN#all
92 http://mahir.asut.net/resume-en/

computing back in their favor. That this entrepreneur's contribution to the Free Software Days space revolved around SaaSS is a manifestation of the complex relationship between FS and capitalism. It is also an example of the diverse interests represented by Free Software Days participants that converge in a common space which is implicitly and sometimes only vaguely politicized (There is after all *something* that makes Aşut feel the need to open with the line "the philosophy of FS goes beyond the money aspect"). While the entire phenomenon of SaaSS, enthusiastically presented by Aşut, technically pushes for FS, it is in fact predicated on undermining most of the social goals and spirit of FS as a movement: securing user freedom, encouraging sharing, and establishing individual and social control over software, among others.

Another proponent of commercial FS business is Bora Güngören of Portakal Teknoloji (Orange Technology). The company provides support for Pardus Linux and the Ministry of Education through the Public Sector Linux Center.[93] Güngören is an electrical and electronics engineer who has served in the management council of the Ankara Chamber of Electrical Engineers, and in the management council as bookkeeper for the Linux Users Association.[94] He has also worked for TÜBİTAK on two separate occasions in 2000 and 2004, the latter case being under the UEKAE, although not on Pardus.[95] In a 2006 presentation titled "The Software Economy and Free Software", he shared the important insight that the non-free software market is artificially created by licenses (i.e. the state) and that the free market theory of liberal economics is an abstraction. He pointed out that the non-free software economy had been shaped by certain "smart" actions of some large firms led by exceptionally intelligent people who foresaw back in its days of infancy how widespread software would eventually become, i.e. by monopolization from early on (Bill Gates's famous "letter to

93 ibid.
94 http://www.boragungoren.com/
95 http://www.boragungoren.com/hakkimda.html

enthusiasts" comes to mind). Another insight he offered was that software is legally defined neither as a product, nor as a service or a commodity, but as "a work of art" similar to a book or a painting. This meant that users (consumers) could not legally return software as defective, as the quality of art is a matter of subjective taste, which shields its producer from accountability. As opposed to other works of art, however, in most cases it is not possible to view software before purchase, creating knowledge asymmetry in its marketing. This "worst of both worlds" designation of the status of proprietary software is then compounded by further monopolistic activities such as price hikes, planned obsolescence through dropping backwards compatibility, updates or platform support. Monopolization of one software product is also used as the springboard for the monopolization of another (e.g. the monopolization of the operating system market by Microsoft Windows was used to push for the monopolization of the web browser market by Microsoft Internet Explorer).

Keen to contrast monopoly with competition, Güngören juxtaposed the FS economy as being based on "the protection of the rights of all" (as opposed to only the rights of the monopolist) such as second sale, independent development, the right to sell services and support on an equal footing, which "encourages entrepreneurship", embodying the "FS maxim" of "if you are in possession of it you own it". He claims that monopolization is impossible under FS, and even if a piece of FS would become a "monopoly" in the sense that it becomes the universally used solution, this will have no negative consequences because it will come free of price and cannot erect barriers to competitors. This, according to Güngören, forms "a real free market from the neoclassical point of view". His presentation then takes an interesting turn, as he attempts to approach the FS economy from other economic schools of thought. He claims that "from the NeoMarxist point of view, the FS economy establishes an ideal balance in terms of its productive process", from the NeoRicardian point of view,

revenue sharing is ideal, and from a NeoSchumpeterian point of view, barriers to innovation are removed". On their own, these one-liner formulations leave much to be elaborated upon. What I find significant in Güngören's listing of these diverse perspectives is that it is a reflection of the idiosyncrasy of FS as a mode of production. Comprehending FS fully as an economic phenomenon requires concepts and explanations derived from its own unique features. Absent these, semi-accurate piece-meal mobilizations of prior economic schools of thought are resorted to. This is not a result of mere lack of theoretical prowess. As discussed in Chapters 2 and 3, there is a tendency towards transcendence in FS in the sense that it includes and improves upon favored economic features espoused by radically different industrial era economic schools of thought which can only partially (and often misleadingly) begin to describe it.

One year later, in 2007, Güngören was back on stage with a presentation titled "How to Make Money with Linux". Güngören described the well known selling services model, along with a model of custom development. The custom development model assumes that a customer requires a software with a certain functionality that does not yet exist. However, the case may be that this functionality can be brought about by making a relatively small addition to an existing piece of FS. The developer then charges for making the specific improvement to an existing FS (bespoke or contract FS coding).[96] Güngören cited the advantages of such a business model as providing a solution quicker than it would take to develop from scratch, and lower costs of development. He also mentions that the mere fact that FS comes free of charge means that institutions stand to make savings by switching to them. He adds, however, that switching and deployment may be non-trivial and require a planned transition process.

96 In such cases, the improved end product, in other words the derivative work may become FS commons in only certain cases. If the original FS was under a copyleft license, that it does is mandatory. If it was under a permissive license, it is up to the producer of the new code to designate the derivative work under any license they choose, including FS (permissive or copyleft), or a proprietary license. Naturally, only cases where custom coding results in the derivative work being FS can be considered as part of FS production.

Güngören suggests that a smart way to make money in this case is to sell consultancy to the institution on how to manage this transition most efficiently. Güngören's Portakal Teknoloji itself practices this business model as a "training and consultancy firm that will help institutions in their technology needs".[97]

Following up on some of his 2006 themes of economic theory, Güngören's 2007 presentation also reflected theoretically on the economy that FS engenders. Stating that "Just as anarchy does not mean the lack of order, free software does not mean arbitrariness", he claimed that "Free software puts forth a true free economy based on the mutual trust between persons who are honest about some basic issues". Building on an anarcho-capitalistic notion of ordered anarchy, he then analyzed the FS economy as he sees it through the lens of liberal political economy. "From a liberal perspective, this economy brings a free market to life". What Güngören is referring to here is that in the proprietary software market, monopolization of the associated support and services market by the company that produces the proprietary software is the norm due to advantages derived from exclusive access to source code. With FS, however, the support and services market is relatively decoupled from who wrote the FS, because source code is freely available. The initial software producer's competitive advantage in the support and services market is reduced to their relatively more intimate knowledge of the code rather than total opaqueness of said code to outsiders.

This analysis is accurate; however, its caveat is that it provides an explanation of the FS economy *exclusively* in terms of the associated support and services economy. A free market in support and services may indeed be brought about by FS, but the FS economy as a whole is much larger than that coupling mechanism, as I have described through the remuneration mechanisms of the concentric circles in Chapter 3. Güngören is aware of the significance of the cost of reproducing (copying) software being near-zero. With a nod to the

97 http://www.portakalteknoloji.com/?lang=en

theory of marginal cost, he alludes to the rationality of software as a commodity arriving at zero price. However, he then quickly inserts the associated commodity of services and support to redefine "software" as a package commodity that includes software code plus support services, and jettisons his previous analysis based on marginal cost. "There are many side costs to transitioning to even the simplest piece of software". Furthermore, he also confounds a fresh market for software per se, with a market already served for by competing products, and where therefore the commodity specifically at hand becomes one of software code plus support service plus transition service. What Güngören does here is a case of what Marx refers to as "bourgeois economics" at work: he extrapolates from his own business model around FS the theoretical explanation of the FS economy as a whole, despite the diverse mechanisms (the many Circles) that feed into the creation of the FS commons, only some of which are even properly economic. Correctly observing that the realization of the use value of FS (which is not a commodity) is for some users dependent on the realization of the use value of services and support (which are indeed produced capitalistically as commodities), he falls into the trap of collapsing the two phenomena into one. This analysis fails on two accounts. First, the attachment of services to software is not particular to FS but instead applies to all software. Hence, the specific emergence of FS cannot be explained by the existence of a related services market. Second, not all users require such support services for all software, whether free or proprietary.

4.2.2 The Big Civic Tent: The Linux Users Association

In a seminar in 2010 titled "The Linux Users Association Uses Its Own Medicine", Adil Akbaş and Emre Eryılmaz introduced tenets of the Linux Users Association (LKD). They emphasized the decentralized structure of the LKD, pointing out that it had no central

office or branches. The members live in different cities and almost all of the activities carried out by the association are planned and organized on the Internet. Their main, most noteworthy point however was that "free and open source applications are used exclusively in the directing of association activities on the Internet". To expound, there is a threefold relationship between FS and the LKD. First, there are all the user-facing FS applications, such as Trac, Wiki, SVN, Mailman, Tuxweet, Planet and Wordpress. Second, there are the backend FS technologies utilized like MySQL, PHP, Python and Apache. Third, there is FS developed by the LKD itself, such as the Membership Software. This assortment of applications and technologies are needed for running a decentralized organization online. The LKD's case is an example of both utilizing and producing the FS commons while also making a proud display of integrity out of such utilization. The subtlety worth noticing here is that even the mere *use* of FS is seen as a display of commitment in the sense Charles Tilly ascribes to social movements (Tilly, 2004). Along with displays of worthiness, unity and numbers, commitment is expected to engender sympathy to the cause of a social movement according to Tilly. This is a prevalent fact of the FS world, where even mere usage without development contribution is celebrated as an essential part of supporting FS as opposed to being shunned as "freeriding". The reasons for this are well understood not only ethically but also economically, with reference to the *network effect* I discussed earlier, which is integral to the success of most software projects (specifically, any software that involves communication and exchange of data between users—which is arguably most software). Another way to phrase more use-value for every user is more wealth as a whole, irrespective of the *value* of software which in aggregate remains constant (copying does not create more value, only more use-value). In its social movement aspect, use of FS is taken as an important sign of integrity. *Dogfooding*[98] is

98 In computing slang, "To use (programs developed by oneself or by one's own company) as end users" (Wiktionary.org)

encouraged, while admitting using non-free software can become a source of embarrassment, especially for organizations.

4.2.3 Big P Politics at the FS Conventions

Uygun made two almost identical presentations with the title "Freeriding on the Work of Others when Developing Software" in 2014. One was at the Academic Informatics Conference, and the other was at the 2014 Free Software Days. The main content, describing priorities on how to choose a piece of FS upon which to build a derivative work, is not of particular interest. What's interesting is that the Free Software Days version had one extra slide depicting a quote from the founding leader of the Turkish Republic, Mustafa Kemal: "Peace at home, Peace in the World". How this quote relates to the topic of the presentation is unclear. The interesting bit is that this small addition does not appear in the Academic Informatics Conference version, which indicates that Uygun felt that the Free Software Days setting was more conductive to expressing a symbolic statement of political identity than the Academic Informatics Conference. It must be noted that formerly all-catching and apolitical, national symbolism around Mustafa Kemal has in recent years been increasingly identified with a centrist secular opposition to the ruling religious right-wing AKP government. However mild, it seems that Uygun found the Free Software Days a safer and/or more receptive space for such a gesture.[99]

The same dynamic was revealed to be at play even more forcefully when again two near-identical presentations made in 2014 by Adil Akbaş of the LKD are compared. The common title "Everywhere is Linux, Everywhere is Free Software" explicitly pays homage to the 2013 Gezi Protests and its massively popular slogan "Everywhere is Taksim, Everywhere

99 I remind the reader that the Free Software Days convention itself was originally part of the Academic Informatics Conference in the early 2000's.

is resistance". A further reference to Gezi Protests was a play on the instant cliché "It's not a matter of a few trees" that was tweeted by an actor that was sympathetic to the protests at the time. In Akbaş's presentations, it appeared as "It's not a matter of a few presentation slides" in self-reference to his highly politicized presentation. The version at the Free Software Days, however, included three additional slides: The first is an iconic sketch depicting seven of the Gezi martyrs below the words "Do not forget the names of the dead, the folk songs, and the squares", which is a line from a song of Ezginin Günlüğü (Melody's Diary), a left-wing folk band. The second is an assortment of portraits of four of the children killed by Turkish Security Forces in recent years, below the words "Let the children not be killed, so that they may eat candy as well", which are the last lines of famous communist poet Nazım Hikmet's poem "Dead Little Girl / Hiroshima". The third is a colorful photo from the 2013 Istanbul Gay Pride parade, under the words "I have learned how to share, that's why my name is Sevcan", which are lyrics from a song by Özgürlük Türküsü (Folk Song of Freedom), another left-wing folk band. Once again, the open inclusion of general left-wing political dissent that is not directly related to FS took place at the Free Software Days, but was omitted at the Academic Informatics Conference.

Akbaş's claim is that FS is not merely a technical issue, but a political issue as such. It is a social issue, and a matter of ethics. It should interest not only people in the IT field, but all of society. It is directly related to freedom of thought and expression and the privacy of personal information. This position is very much in line with the perspective of Stallman and the FSF. Akbaş, however, takes things further. He explores the question "Free Software = Socialism?", starting out with the connection between the two movements revealed through the uncanny similarity of the beard styles of Karl Marx and Stallman. He continues this tongue-in-cheek style by making an analogy between the propensity of leftist organizations to

splinter into numerous small sects and the pervasive phenomenon of forking in the FS world which has resulted in hundreds of different Linux distros. These are nicely illustrated by graphical representations of a "Family Tree of the Turkish Radical Left" and the "Yet Another Linux Distribution Timeline" graphic. Following up with a more serious note, Akbaş claims that with FS we have "something of a different sort" (another reference to a Yeni Türkü song, which is also another left-wing folk band) in terms of "property, commodification, freedom in production, equality in consumption, and work and human nature". Using the language of Marxian political economy, he asserts that "source code is a means of production". He also claims that the motivation for voluntary contributions should be looked for in the propensity of humans to share, echoing Stallman's emphasis on sharing and "helping your neighbor".[100] Akbaş attributes a significance to sharing that goes beyond the sharing of software, expressing his wish to extend it to the realms of "knowledge, beer, hardware, art/culture, democracy, and money" in the name of "freedom, solidarity, art and penguins!".

Enver Altın's personal blog "The truth about my life" reveals that he founded a start-up called Construia in 2008 with himself as "the sole investor and managing director".[101] Construia "innovates in many areas mainly related to mobile devices and networks, including but not limited to mobile messaging (SMS, MMS, Instant Messaging), software development for Mobile Handset Platforms, Voice/Video over IP and other value-adding services" (ibid). His successful one-man start-up operation was acquired in 2010 by an Istanbul-based multinational communications company called Telenity which became Altın's new employer.[102] Altın, although a LKD member and one who "bases some of his commercial solutions on FS",[103] stated in 2012 in a post titled "Pardus postmortem" that he "has not been

100 http://www.gnu.org/philosophy/free-sw.en.html
101 http://enveraltin.com/blog/construia/0707091417.html
102 http://enveraltin.com/blog/construia/1202100135.html
103 http://enveraltin.com/blog/lkd/2303121058.html

and is not directly in the world of FS" (ibid). Nevertheless, he describes his relationship with FS as having sentimental ties to the Pardus project due to past engagements in common projects and co-working with individuals from the LKD and other free software communities (ibid). Altın's interest in FS appears to be not directly related to his personal business practices, but to politics of economic independence. At the 2009 Free Software Days, he talked about "How Not To Defend Linux", where he claimed that the main selling point for Linux and FS in Turkey is that Turkey is currently dependent on foreign producers in software, which contributes to the country's trade deficit. FS could decrease this dependency while increasing productivity in the Turkish software sector. According to Altın, the Turkish software sector is trapped in an unproductive cycle of duplicating effort, where, for example, 100+ proprietary accountancy programs are developed and sold on the market. A single collaborative FS project could replace all of these by building on existing code instead of constantly writing new programs from scratch, resulting in excellent software and freeing up hundreds of software engineers for more useful tasks.

Altın's perspective on Pardus, Linux and FS in general is in line with his broader politics of economic independentism. He lamented how the new Turkish R&D support law presented by the minister of transport and communications Binali Yıldırım that went into effect in 2008 appeared to be providing tax-breaks for operations of foreign investors, including Microsoft, who have little to do with conducting true R&D in Turkey. Altın's stated preference against this neoliberal approach would be for local and individual entrepreneurship to be supported, instead of "selling off the country in yet another way, for cheap".[104] He has also posted comments in praise of the nationalization of energy firms in Bolivia under Evo Morales.[105] Altın's views coincide with an anti-imperialistic strain in pro-FS thinking in

104 http://enveraltin.com/blog/politics/0505081307.html
105 http://enveraltin.com/blog/politics/0205081015.html

periphery countries, which overlaps with part of the official reasoning behind the initial decision of the Turkish government to go ahead with the Pardus project.

Cihan Okyay, a software developer who could be seen on his Google+ profile photo in 2011 fuming up a hookah while sporting a t-shirt with "LINUX" printed on, where the "X" was replaced with a hammer and sickle,[106] made a presentation in 2012 titled "Socialization with Free Software".

Before going on into the content of his presentation, I want to make a cultural interpretation of the hammer and sickle and hookah imagery associated in this case with Linux. It is obvious that this connotes less of a serious commitment to Leninism and more of another tongue-in-cheek expression of leftist subversion in similar vein to Akbaş's parallels between Stallman's and Marx's beards. It is a symbolic expression of what led even the profit-oriented commenter on FS, Aşut, feel the need to say "the philosophy of FS goes beyond the money aspect". It is a symptom of a dissident attitude that recognizes that something out of the ordinary is going on here, an alternative to capitalist business as usual. But the alternative is not yet donned in the vocabulary of self-awareness its advocates need to express it idiosyncratically. Thus, proponents of FS as an alternative to the mainstream, mobilize the now largely out of fashion symbolism of capitalism's great other, 20th century state socialism to express the radicalism within FS. This symbolism is necessarily employed with a dose of tongue-in-cheek, self-humoring irony because those that resort to it are aware that while FS may represent a new great other to capitalism, perhaps embodying many of the stated ideals of old socialism (in sublated form), it is also very different in content and form than for example the Soviet system. Therefore, the old imagery is used, but always self-consciously, with an element of ironic distancing as a corrective. We must not fall into the error of mistaking this humorous distancing for complete frivolity, however. There is a kernel of

106 https://plus.google.com/117916730930196217048/posts

seriousness in all of this. This is not the playing out of an adolescent attitude of shock value, as indeed I have never come across an "X" of "Linux" being replaced with a swastika, or the "S" of "Open Source" made into a "$" symbol. Such fooling around would be completely out of place in the FS milieu, and not for a lack of humorous potential. The near-autistic, sometimes reality-defying uncompromising conviction of Stallman (which deserves great respect) when it comes to FS principles could easily lend itself to his being seen as a FS-Nazi (or perhaps a FS Stalin, 'Stallman' and 'Stalin' sounding strikingly similar). Some of the brutal honesty and often offensive harshness in how Linus Torvalds conducts himself in his Linux kernel development leadership position (his bonafide "Swedish" style), could also lend to such humor. The "first billion dollar open-source company", Red Hat and many other profitable FS enterprises could easily receive the clichéd "Micro$oft" treatment. Yet they do not, which is significant.

4.2.4 Small p politics: Socialization and Gender

Okyay's approach in his presentation eschewed the technical and economical aspects of FS in favor of focusing on how working on FS could be seen as part of the social self-actualization of individuals. With this quite original perspective, he claimed that "being social" means belonging to, being interested in and being attached to society. Participating in FS is thus a more authentic form of socialization than "going wild tonight at the bar" or "having 500 friends on Facebook" according to Okyay. He claimed that FS development, in its universalism, did not care about race, faith, language or culture—a bold claim that could be scrutinized but is nevertheless significant for revealing his ideals with regard to FS. According to Okyay, socialization itself is to be seen as a learning process that improves things like communication skills.

The hacker subculture worldwide is notorious for being male dominated, online and offline.[107] The Turkish FS community that I observed is no different in this regard. There have been, however, some conscious efforts to address the gender issue. Pınar Yanardağ gave a presentation at the 2006 Free Software Days titled "Women and Linux" on the topic. She sought to start providing answers to common questions like "How can I get my female friend to use Linux?", "There are almost no women in my Linux User Group, how can I encourage them to join?" and "Why aren't there more women in the world of open source?". The list of problems identified by Yanardağ should be familiar to anyone who has been exposed to feminist consciousness-raising: discrimination against women, negative male attitudes, women's internalized sense of false incompetence or women's false modesty, the idea that computers are associated with being asocial, gendered media representations, lack of female role models, sexist jokes, the social invisibility of women in tech, stereotypes related to child rearing, real child rearing responsibilities, etc. My observations in the field were completely in accord with this laundry list. FS events easily had a ten to one or worse male to female ratio of attendants; various levels of awkwardness and male guilt-based overcompensation could casually be recorded in the interactions between males and females, and there was never more than one woman employed at any time by TÜBİTAK to work on Pardus. Yanardağ listed several efforts to rectify the lack of female participation to FS projects such as GNOMEWomen, KDEWomen, DebianWomen and LinuxChix as examples; however, there hasn't been a PardusWomen or equivalent effort in the context of Pardus. The stereotypically large gender imbalance in the programmer community remains intact and unaddressed in this context, despite good intentions and a lack of overt sexism. FS vis a vis proprietary software on its own seems to have made no difference on this count.

107 Coleman mentions her own observations of "the handful of women attending" (Coleman, 2010: 58).

4.2.5 The University Connection: Schoolwork and The Academic Ethos

Necdet Yücel has been an exceptional figure in the Turkish FS and Pardus community. While teaching as a professor at the Computer Engineering department of Çanakkale Onsekiz Mart University (ÇOMU), he has simultaneously managed to become one of the top ten contributors to the Pardus project in terms of code commits.[108] That makes him the individual with the highest level of contributions to the Pardus project among those not employed by TÜBİTAK. This prolific contributor to the FS commons did not limit his work to Pardus or only code, either; he has also led multiple initiatives to translate many essential FS pieces into Turkish, namely desktop environments like KDE and Gnome, and office suites like OpenOffice and its successor fork LibreOffice.[109] Indeed, in recognition of his efforts by the community, he has earned the 2006 LKD (of which he is a member) "most hardworking penguin" award for "personally contributing the most to the development and spreading of the phenomenon of free software in Turkey".[110]

Yücel shared his insights derived from contributing to FS as an academic in two presentations, the first being "How to Support Free Software Projects" at the Free Software Days in 2010, and the second being "The Place of Universities in the Free Software Ecosystem" in 2011. He reported on how the 64-bit version of Pardus and the multiple desktop installation system for Pardus called ÇoMak was realized in 2009 and 2010 by contributors from the ÇOMU. Yücel's 64-bit Pardus team consisted of 8,[111] and his ÇoMak crew consisted of 13 individuals from ÇOMU, 12 of which were undergraduates at the time.[112] Toward the end of 2010, 3 among those that had graduated in the meantime went on to work on Pardus professionally, employed by TÜBİTAK.[113] The ÇOMU FS community

108 https://www.openhub.net/p/pardus-linux/contributors?query=&sort=commits
109 http://www.nyucel.com/2015/06/hangi-masaustu-ne-kadar-turkce.html
110 http://akademik.comu.edu.tr/onizle.php?cvno=A-1004
111 http://2uzeri6.blogspot.com.tr/2010/04/ekibin-elemanlar.html
112 http://www.nyucel.com/2010/10/pardus-coklu-masaustu-kurulumu-comak.html
113 http://2uzeri6.blogspot.com.tr/2010/10/ucuncu-m-de-pardusta.html

demonstrates another form of software engineer contributions flowing into FS production in the pre-employment phase of their careers. In this case we have students participating in FS production under guidance of a FS-enthusiastic professor without leaving their alma mater for an internship or summer camp. In return, in a few years they may enjoy preferential hiring when entering the job market.

As in internship, undergraduate participation in FS production in this case also serves as a means of getting ahead in the competition for jobs. Regarding the caveat I mentioned for interning at the commons, namely that it may be collectively disadvantageous for these undergraduate programmers to be raising the bar against each other for getting jobs in this fashion, the contradiction seems weaker here. To the extent that this production is part of regular coursework and substitutes for final projects, etc. (as is the case with Yücel's students according to my interview) as opposed to being extracurricular loads on students, I see no issue. Furthermore, as in the case of interning at the commons, even if this undergraduate contribution to FS represented an additional workload on students that could potentially detract from what should be their primary concern, i.e. getting an education, this would still be counterbalanced by the fact that as opposed to most other alternatives, society benefits from the public good that is being produced. The implicit competition of young skilled workers for future jobs is carried out in a form that serves the social good.

Yücel's commitment to FS goes well beyond his own contributions and practical guidance role that has established a young programmer conveyor belt between the university campus and the TÜBİTAK campus. He is also interested in the institutional role of the university with regard to software. In his 2011 presentation, he underlined the fact that the university is at once an institution that develops software, rears software developers, creates software users, rears educators that will use software, and uses software itself. "Whatever

software the university teaches gets used". While the argument could be made with equal validity that whatever gets used (especially in business) gets taught by the university, Yücel nevertheless draws our attention to the question of the social duty of the university when it comes to choices in software; what software should be taught, used, and produced? Citing that important FS projects like the Slackware Linux distro and OpenSSL, which is a core internet security technology, were developed by universities, he makes the case for universities to actively prefer FS for teaching, use and production. As an academic myself, I couldn't agree more. Why should a university, especially if it is publicly funded, be made to serve private interests such as proprietary software companies through requiring that their solutions be used or taught? Software produced at such admittedly *universalist* institutions ought also to be publicly licensed as FS. Yücel's practice and thinking on the issue represents an enlightened attitude to the phenomenon in all its aspects.

The academic overlap with FS as a philosophy and movement has not been limited to grooming undergraduates for FS jobs or even limited to the field of software engineering. Professor of Computer and Information Science Özlem Özgöbek, for example, delivered a presentation in 2012 at the Free Software Days on "Open Moves in the Academic World" that introduced various academic open access initiatives to the Free Software Days public. The affinity between FS and the idea of open access to digital textbooks and other educational material comes naturally, especially to a teaching software engineer. That this presentation was right at home at the Free Software Days goes to show that the community is interested in the greater cause of providing free digital commons and not just software.

Özgöbek cited the examples of the MIT OpenCourseWare project, open access initiatives at MIT, Yale, Berkeley, and Tufts, as well as initiatives like the Open Education Consortium and Open Educational Resources Commons. She also promoted the Elsevier

Boycott campaign led by "The Cost of Knowledge" website which had reached 10000 pledges as of 2012 and 15000 as of 2015 to boycott publishing, refereeing or editing for Elsevier's academic journals because of Elsevier's monopolistic price gouging.

Within the Turkish context, she informed the listeners about the Open Course Materials Project of the Turkish Academy of Sciences (TÜBA) that had been initiated in 2007 with 8 universities, both public and private. As of today, the project appears stalled. Sections of the website have not been updated since either 2011 or 2012, and the collection of available material is quite modest.[114] TÜBA went through a controversial reorganization in 2011 in a fashion similar to what happened at TÜBİTAK-UEKAE. Whereas up until late 2011 TÜBA members were elected from among academic ranks autonomously, a new law in that year gave TÜBİTAK and the Higher Council of Education each the authority to appoint one third of the new members. This caused a big kerfuffle which led to an exodus (not of the Negrian kind) from TÜBA where some members regrouped around the newly founded alternative called Science Academy Association.[115] To what degree the fizzling of the TÜBA Open Course Materials project had to do with this ordeal is an open question awaiting research.

4.2.6 Politically Conscious FS: The Organic Intellectual

If I had to point out one individual within the Turkish FS community as its leading organic intellectual, it would have to be İzlem Gözükeleş. Gözükeleş is a software engineer by education, and holds an MS degree on Science and Technology Policy Studies with his 2006 thesis titled "Free and Open Source Software in Turkey". He was a founding board member of the Chamber of Computer Engineers (part of TMMOB) and is a LKD member.[116] He is a Free Software Days and Document Freedom Days regular. He has co-translated Stallman's seminal

114 http://www.acikders.org.tr/
115 http://bilimakademisi.org/
116 http://bmo.org.tr/2012/07/06/bilgisayar-muhendisleri-odasi-kurucu-yonetim-kurulu-atandi/

"Free Software, Free Society" into Turkish and has reviewed Julian Assange's book "Cypherpunks". He has written numerous articles and essays on FS, intellectual property, online privacy and digital labor which have been published on technical, progressive and left-wing media such as Bilim ve Gelecek (Science and Future), Politeknik.org.tr ("The People's Engineers, Architects, Urban Planners") and Sendika.org (Union.org "The agenda of the labor movement"). He has delivered a speech at the 2015 Laborcomm conference titled "A crack in Capitalism: Free Software". Since 2012, he has been employed at the Head Department of Information Technologies of the state-owned broadcasting company TRT (Turkish Radio and Television).

I will unfortunately limit my exposé of Gözükeleş in this section to two of his Free Software Days presentations as I am more concerned with the circulation of ideas within the public sphere enabled by such FS related conferences than I am with presenting the full scope of his theoretical reflections in his thesis and articles. Nevertheless, I must mention that within the Turkish FS community, he has gone the furthest in developing a Marxian analysis of FS, addressing core questions that overlap with my own efforts.

At the 2006 Free Software Days, Gözükeleş made a presentation titled "Free Software, Open Source, Hackers and Business Models". He began by pointing out the limitations of what he identified as the two main questions that FS literature had been focusing on: The issue of motivations, and the issue of how to make money with FS. He suggests that these inquiries are based on presuppositions that need to be questioned. According to Gözükeleş, "under certain historical circumstances, humans run away from work like it's the plague, while under others they may be inclined towards it". This of course is a reference to the notion of *alienated work* under capitalist employment versus self-actualizing work (or species-being activity) envisioned in a classless society (Gözükeleş explicitly cites Marx's *1844*

Manuscripts). Gözükeleş's charge is that the very search for extrinsic motivations (including the monetary kind) to explain why some individuals code FS is flawed, as for hackers the activity is itself intrinsically rewarding.

His second criticism of the approaches in FS studies is how the object of study has been defined: As social movement, as community, and as gift economy. FS as a social movement led by such forces as Stallman, the FSF and the Oekenux Project has indeed emerged as a reaction to proprietary software. It has had a great deal of success as a movement in pushing people to question and change their perceptions of software. The community approach has defined the object at hand as "a virtual community of hackers", as indeed software coders form virtual communities every time they collaborate on a common software project. While according to Gözükeleş both of these designations, movement and community, are merited to an extent, they are nevertheless too limited because they are "very narrow abstractions based on generalizing facts that belong to a very short period of history", whereas "FS has taken different forms at different destinations of its historical development". He explains that these designations concentrate only on hackers (people who code because they enjoy it) while ignoring the role of firms and governments in FS, as well as hackers who have become business people. The gift economy theory may explain the relationship among charitable and sharing hackers, but it fails to account for the collaboration and conflict between hackers and non-hacker coders, in other words mere "software engineers" (an analysis I have concurred with in Chapter 3, with the added emphasis on the divergence of the categories of coders and users). Furthermore, Gözükeleş pointed out that software is not the same as any other commodity; it defies Hardin's "tragedy of the commons" where "gains are individualized while losses are shared". He defined GNU-Linux as "an impossible public good" that is inescapably ownerless and non-rival: everyone can use it without it being

consumed and there is no significant cost to its reproduction. This, according to Gözükeleş, annuls the scarcity premise of classical economics. Hackers (note his earlier distinction from software engineers) obtain prestige instead of money by giving away instead of possessing, in line with the scientific ethos and academic work which is the milieu from which FS and open source have emerged. Finally, Gözükeleş balanced his presentation by pointing out that "we still live in a capitalist society" where business models around FS such as integration, education, support and consultancy services remain relevant, in addition to donations (Circle 3).

At the 2011 Free Software Days, Gözükeleş expanded upon his 2006 themes with a talk titled "GNU-Linux, The Operating System that changed the World". This time he expounded the materialist idea that unlike god, who remains the same god before and after creating nature, when humans change nature through technology, they also change themselves. Therefore, the development of GNU-Linux not only resulted in a specific kind of software, but also a specific kind of person and culture. He also made explicit that when the production process is analyzed, FS as a mode of production is distinct from the dominant mode of production. Specifically, FS replaces private property over the means of production with social property, mandatory division of labor with voluntary division of labor, and alienated labor with free (as in freedom) social labor. He asserted that FS replaced cut-throat competition with solidarity, without however destroying individuality, and embellished his assertion with the famous lines from Nazım Hikmet: "Like a tree alone and free, like a forest in brotherhood". Finally, he concluded by saying that IT is a field of struggle, and not just a given mass of devices shaped only by the powers that be, once again channeling Hikmet: "Either we'll take life to the dead stars, or death will descend on our world". I was able to pose a question to Gözükeleş at this session, and basically directed to him the main question of my

own doctoral thesis: "The ideal-typical schema of FS as a mode of production based on hackers performing non-alienated labor is crowded by the empirical situation of many other actors and motivations. Is FS a window on to a new world, or is it a mere parenthetical phenomenon circumscribed by capitalism?" His answer was that this remained an open question within the field of struggle, and while FS production is molded by many changes (away from pure hacker-activity), the core principle within the productive relationship persists. This is in line with my position in this dissertation.

***

The overview I gave in this chapter of the physical and discursive spaces where a variety of actors within and surrounding the FS community come together for interaction and expression yields a number of key findings.

First, the characteristics of the venues and physical spaces themselves reflect the nature of the community as an "alterpublic". In between a regular public and a counter-public, the alterpublic combines alternative provisioning for acquiring new skills (skill sharing workshops, business models) that are useful in the mainstream IT world, with instances of expressing political dissent and suggestions of radically different ways of doing things (peer production, counter-technologies, etc.). The complex relationship between the operation and values of FS and the larger capitalist economy in the realm of digital production is reproduced in the sometimes parallel and at other times intersecting presences in the physical spaces of convening. As opposed to the ease of spontaneous formation of FS publics online, thanks to the combination of the personal ownership of the physical and the public ownership of the digital means of production and communication, sustaining FS publics in the offline world

requires greater effort and even sacrifice. Key individuals in the FS community who can facilitate the cooperation of public-good-motivated institutions like universities play an important role here. Without the support of already established institutions, it proves harder to sustain certain ideals of open, free access and commonness in the face of funding pressures, as in the case of the hackerspace.

The FS community is part and parcel of the strata of white collar IT workers in terms of class and outlook. Software coders are highly employable and their class position allows much horizontal mobility, which FS enhances. The Conventions, NGO's like the Linux Users Association, and FS-engaged computer science departments serve as recruiting grounds for the Pardus project, local FS SME's, and even large IT companies with stakes in FS. Furthermore, even prior to official employment, there is a significant amount of labor going into FS through what I referred to as "marginal sources" such as students working on projects, interns "interning at the commons" and production geared towards learning taking place in education camps and training initiatives

The question of how to make a living while being part of FS is a constant point of concern, inquiry and innovation. This is of course an inevitable result of the standard market society default of commodity sales not applying to FS. Despite the constant reflections on the subject within the community both in practice and in the literature, the lack of lucidity in theorizing the relationship between FS and income generation, and how this relates to capitalism remains pervasive. Captured in the phrase "making money off of FS", this important question remains a major source of confusion for both speakers and audiences. This indeterminacy also causes certain capitalist business initiatives which are in-letter FS-based, such as the SaaSS model, who in fact subvert the spirit of FS in order to establish a rent relation to go unquestioned. This suggests that the spaces of convening are only implicitly and

often vaguely politicized, compared, for example, to the clear lines of the Free Software Foundation.

Despite the difficulties of articulation, proponents of FS have an instinctive understanding that FS constitutes an alternative to the mainstream economy, and they often culturally express this through socialistic symbolism tempered by a dose of ironic self-distancing. Furthermore, FS is often embraced as a process of self-actualization of individuals, beyond its formal technical and politico-economic role.

Interjecting on to the level of the nation-state and the issue of dependence and global inequalities in IP, the desire for economic autonomy articulated through various concepts of independence and economic nationalism informs the FS community. This is in line with economic factors cited in the original reasoning of the government for green-lighting Pardus as a government sponsored project. There is, however, much disagreement in the FS community about the precise content of the espoused value of autonomy that ranges from personal libertarianism to universalist "love of freedom" to national pride.

The validity of the FS mode of production for fields outside of software is confirmed by the overlap of interests in other PP forms such as map making, open access journals and Creative Commons licensed publishing (Suber, 2012; Siefkes, 2012). The aura of FS extends even beyond expressions in productive activity per se. Many academics, for example, have understood the compatibility of the values of FS with the academic ethos and the public good. They advocate the teaching, use and production of FS in schools and universities in a shared spirit of the scientific commons.

A prevalent attitude in the FS world, which is that even the mere usage of FS without contribution to development is celebrated as an essential part of supporting FS as opposed to being shunned as "freeriding", is reproduced in the Turkish context. The positive "network

effect" and the importance of promoting adoption is well internalized by the FS communities.

CHAPTER 5: THE PARDUS PROJECT

> "The promise... is that software innovations can and should come from everywhere. Emerging markets are not implicitly stuck relying on commoditized, hand-me-down innovation from the developed world. They can have their own lead users who pull technology development toward applications that fit specifically the indigenous needs and demands of emerging markets." (Weber, 2004: 252)

5.1 Context and Background: The Story of Pardus

The Pardus project is the largest experiment with FS production undertaken yet in Turkey. It is carried out by the Scientific and Technological Research Council of Turkey (TÜBİTAK), which is the country's major government funding body for scientific research and technological development. As a government-led project that attracted a diverse community consisting of employees, volunteers, academics and users, it provides a rich illustration of the uneasy relationship of FS with capitalism at the level of individual, communal and institutional dynamics, against the background of economy and politics on a nation-state scale.

Along with the implications of FS for national security, part of the impetus behind the implementation of this project is related to the promise FS holds as a means of countering international inequality and dependency in the software sector. FS can be seen as a form of instant technology and wealth transfer, removing the need for "hand me down technology" to arrive at underdeveloped parts of the world with significant disadvantages in terms of time lag and costs. "Better access to knowledge and the emergence of less capital-dependent forms of productive social organization offer the possibility that the emergence of the networked information economy will open up opportunities for improvement in economic justice, on scales both global and local" (Benkler, 2006: 131).

It is certainly optimistic to assume that the democratization of the computing and communication infrastructure will automatically alleviate a significant portion of the inequalities in the global software arena. The availability of free and open code is an incredible boon for software development, but the relative scarcity of skilled labor in different parts of the world may significantly limit the realization of the positive potential of FS. For example, while the source code is available to all, Linux code contributors in 2000 were overwhelmingly from developed countries, with Scandinavian countries in the lead, followed by Western European, Benelux and Anglo-Saxon countries (Weber, 2004: 68). 70% of FS developers seem to be based in EU countries (Reding, 2007). Furthermore, many key FS projects are led by core developers working in the U.S.

We therefore face a picture where, while FS is a global commons and FS development a global collaboration, FS development is spearheaded by developed regions, with a strong Euro-flavor. This resonates with Appadurai's early observation that "Thus far, access to these virtual (electronic) neighborhoods tends to be confined to members of the transnational intelligentsia, who, through their access to computer technologies at universities, labs, and libraries can base social and political projects on technologies constructed to solve information flow problems." (Appadurai 1996:195). While the facilities that Appadurai refers to have become considerably cheaper and democratized over the 15 years since the publication of these words, on this higher level of knowledge production the circles that Appadurai mention still seem to have disproportionate weight on the processes. The Euro-flavor in FS also underlines the diversion of interests between competing centers in the core, especially considering the unique position of USA's Silicon Valley based dominance in software IP.

At this point, however, it is useful to remember the distinction I made in the

introduction chapter between FS as a production process and FS as a product. While FS development (production) may remain a predominantly core/Eurocentric activity, as a free and open product FS deployment is attractive to non-core countries because it constitutes a technology and wealth transfer. Whereas commodity chains in the world system usually reveal an unequal process of production and consumption that disadvantages the periphery and advantages the core, FS is a public good that seems to be flowing in the reverse direction, from the core to the periphery, and therefore has the potential to have a reverse effect within the world system. As Weber puts it, "one of the advantages of the GPL is that it then prevents a dysfunctional enclosure of mobilized 'southern' resources into 'northern' properties protected by patents that are offered for resale to the 'south' at exploitative prices, a depressingly common pattern for knowledge-intensive products as diverse as music, plant varieties and pharmaceuticals" (Weber, 2004: 251). Perhaps FS can therefore be seen as an "anti-commodity" chain in the world system. The current domination of the software market by large US companies which make huge profits is a prime reason why many countries are experimenting with funding FS alternatives as a matter of national policy.

A striking case in point is Brazil, which is often pointed out as the most enthusiastic supporter of FS as government policy on the national level. The Lula government, especially in its first term (before counter-lobbying accelerated) approached FS as a serious development issue and nation-wide policy, "turning Brazil into a tropical outpost of the free software movement" (Benson, 2005). An adviser to the Lula government from MIT has stated that for Brazil "free software provides a basis for more widespread access, more powerful uses and a much stronger platform for long-term growth and development" (ibid.). "The government's top technology official wants to create what he calls a 'continent of open source', leading a trend that is 'little short of a war between Microsoft and Linux'" (Ashurst 2004). Brazil's

country-wide moves have included ordering ministries and state firms to switch to FS, requiring companies and research institutes that receive government financing to develop software to license the resulting software as open-source (Benson, 2005), granting tax-exemption to computers sold with Linux and FS installed by default,[117] and deploying FS on computing centers targeted at providing computing and internet access to persons without private access to such.

FS is also seen as an issue of independence and democracy by the Brazilian government. Ashurts quotes Professor Arnando Mandel of Sao Paulo University: "The main thing is to be in control of your own data . . . In a democracy it's important that a free government shouldn't be subject to the whims of a company" (Ashurst, 2004). The president of Brazil's National Institute of Information Technology has stated, "We're not going to spend taxpayers' money on a program so that Microsoft can further consolidate its monopoly. It's the government's responsibility to ensure that there is competition, and that means giving alternative software platforms a chance to prosper" (Benson, 2005). In the Brazilian case we see a mix of concerns with a strong emphasis on development and independence, as well as cost-saving motives and popular-democratic overtones. The commitment to FS in Brazil appears to be entrenched.

The Turkish government's implementation of the Pardus project in 2003 can be seen as a similar attempt on a smaller scale and with less pronounced political goals. According to TÜBİTAK's website, the project was undertaken following research on "similar applications all over the world, existing conditions of the software industry, . . . the human resources of the country in the information technology area, the capabilities of the local software industry and their competitive aspects".[118] The website describes the aims of developing Pardus as follows:

117 See http://br-linux.org/brazil/
118 See http://www.pardus.org.tr/eng/about/

> The following observations are revealed when we examine the place of the operating systems in the area of information technology in our country, the sectors they indirectly affect and their relations with the development motion in general: An operating system is required on which critical applications can work for the purpose of national defense, security and savings, which supports an open and standard data structure, whose source code is open as to allow for security monitoring and which can be deployed without suffering any financial burden. Redirecting the activities in Turkey about the information technology toward the value added projects and focusing on the high technology based on research and development activities are required. It should be preferred to determine the direction of the technological development, to change the weight of differing areas and thus to be dominant on the roadmap of the said operating system depending on the requirements of the country. The developers shall make original contributions to the Linux and open source society on a global scale by bringing applicable and innovative solutions to the existing technological problems. (Ibid)

The Turkish governmental initiative on FS thus seems to be concerned primarily with issues of national security, defense and cost-saving. Concerns with influencing the global development of FS to better satisfy Turkish interests, and issues of national development are also present. With such objectives, at least a limited challenge to the global software giants by means of FS is possible for Turkey (or Brazil) because what enables such challenges is the same reason underlying the sky-high profit margins of the leading software companies: the whole edifice is built on rents accruing to IP that is produced by skilled labor utilizing minimal fixed capital. In other words, having well trained, skilled programmers is the main requirement to challenge the masters of the software industry—a requirement that shouldn't be beyond the capacity of the educational infrastructure of a middle tier nation state like Turkey.

The GDP of Turkey in 2011 was 731 Billion USD. The size of the Turkish software market (including services) in 2011 was 700 Million USD, in other words about 0.1% of its GDP. This percentage is lower than the global 0.5% and a chasm lies between it and USA's 5.2% and the EU's 2.6% (*BAKA* 2012: 8). Part of the explanation for this appears to be widespread copyright infringement (so-called "software piracy"). A 2015 report by the US

Chamber of Commerce Global Intellectual Property Center titled *Unlimited Potential* rated Turkey 2 out of 6 points in terms of its copyright regime with a blunt 0 out of 1 point for "availability of frameworks that promote cooperative action against online piracy" (132). The same report gave a decent 0.5 out of 1 point, however, for "clear implementation of policies requiring proprietary software used on government ICT systems to be licensed software" (ibid). This second assessment is important because it underlines the compliance with software copyrights in the public sector, and explains the motivation for the effort to position the FS Pardus project as an attempt to escape the dependence on foreign proprietary software vendors. The success rate of said transition attempt, however, has been a whole other story.

Salman & Yapıcı wrote an article titled "The Public Sector is Failing to Transition to Pardus" in 2009 for the journal of the Chamber of Electrical Engineers, which has a progressive, left-leaning institutional character. Their research revealed that despite a 2006 decision by the High Council of Planning which was put together by the Ministry of Development to act on "Use of Open Source Software in the Public Sector" as part of its Information Society Strategy, "public sector interest in the Pardus operating system remained limited" (Salman and Yapıcı, 2009: 21). The transition to Pardus was not made mandatory; instead the decision to take action was left to the freedom of the individual public agencies (ibid: 23). The State Planning Organization revealed that the use of Pardus in the public sector was confined to the Department of Military Drafting (attached to the Ministry of National Defense) and the Radio and Television Supreme Council (ibid).[119] Despite operating under the civilian Ministry of Defense, these offices are staffed by a mixture of civilian and military personnel and are headed by a ranked military officer. It is not a surprise that this unit of government was at the helm in the switch to Pardus, considering that the Military had always

119 I was able to anecdotally confirm the use of Pardus at the local Kadıköy Draft Office in 2012 when I visited it for a purpose unrelated to research, and it came to me at the time as a neat discovery.

demonstrated high interest in a "national operating system" for security considerations, beyond mere cost-saving considerations.

The lackluster pace of migration towards Pardus in the public sector is further demonstrated by the spendings. A parliamentary question by a Republican People's Party (main opposition, center-left) MP on "public spending on operating systems between 2003-2007 and efforts to assure use of Pardus" led to the unearthing of the following: Among the 64 units of government, 47 provided full and 9 provided a partial cost record to the parliament. The payments made as operating system licenses (Windows) added up to a 28 Million Turkish liras (TRY) receipt, or 7 Million TRY per year (ibid). Costs were often unidentifiable due to bundling of operating system purchases with hardware purchases, as well as purchases being made through grants and "packaged purchases", all of which are business practices that promote vendor lock-in and dependency (ibid: 22). To put the (lower than actual) 7 Million TRY per year spent on licenses into perspective, the State Planning Agency revealed that only 200 Thousand TRY per year were spent for the Pardus Project between 2008-2010 (ibid: 23) although a larger amount was pre-allocated for future spending. This was a very small figure compared to the 1.5 Million TRY per year that TÜBİTAK was spending on Pardus out of its own income throughout its existence.[120]

I must remind the reader here that TÜBİTAK is a scientific research institute whose main purpose is to engage in high-end research and development. It's core mission should not be to serve as a procurer for software or any other mass good for the government. A project like Pardus would have been merited for the know-how and strategic national security benefit alone, even if it were to serve no economic cost-saving purpose for other public agencies. Pardus was organized under the National Research Institute of Electronics and Cryptology

120 http://blog.erkantekman.org/2012/07/pardusun-makus-tarihi-2007-2011/

(UEKAE)[121] department of TÜBİTAK, which "produces solutions for military and civil needs of our country, . . . develops information security and electronic system projects vital to strategic government institutions, making significant contributions to Turkey's ability in information security."[122] Therefore, the real political attitude of the government towards Pardus as an economic rather than a security good should be gauged by the willingness to provide funding from the treasury rather than from within TÜBİTAK.

The government's large gap between monetary commitment towards purchasing software licenses and paying for FS development suggests that Pardus funding could have been massively boosted by a more aggressive governmental transition policy, but this path was not taken. A shift of spending away from license purchases in favor of Pardus development could have gone a long way towards speeding up Pardus development, which could have addressed whatever shortcomings Pardus had at the time (shortage of workforce was a chronic issue) that caused the government units to refrain from voluntarily switching to it. The at times accusatory tone of Salman & Yapıcı in their article therefore comes off as understandable, and also their frustration at the mostly wasted opportunity around Pardus and FS in passages such as: "a major factor in software development is that it does not require large infrastructural spending as in hardware production . . . brain power is important", and "GNU/Linux carries great importance because it is open to users seeing the code and shaping it according to their needs, has a structure that can break commercial dependency and provides forms of local distribution that can serve in various countries"(Salman and Yapıcı, 2009: 20).

The Pardus funding tragedy continued in the years after Salman & Yapıcı lambasted the government's tepid approach to transition in 2009. Pardus project manager Erkan Tekman

121 BİLGEM-UEKAE (Informatics and Information Security Research Center - National Research Institute of Electronics and Cryptology) after 2012.
122 http://uekae.bilgem.tubitak.gov.tr/en/kurumsal/national-research-institute-electronics-and-cryptology

has shared the figures for the years before and after 2009 on his personal blog. First, let us get a sense of the neglect in comparison to the semi-autonomous achievement at hand as of March 2009:

> According to information taken from Ohloh.net,[123] the lines of code and annotation for software developed specifically for the Pardus project has reached 2.1 Million. Had this much software been developed by proprietary software development methods, it would have required 600 person-years. That would have corresponded to the continuous employment of a team of 150 developers since October 2004. With average labor costs, such a team would have cost 53 Million TRY. The Pardus project, however, began in September 2003 with a team of 4 and has steadily grown to a size of 18 in total. Within this team were also certain persons like project managers, graphics designers, project developers, project supporters, community managers and system managers who operated in functions that do not directly create value. Even including these, the total labor expenditure of the Pardus project as of the end of 2008 was only 56 person-years.[124]

As Tekman goes on to mention, the source of this "open source miracle" is, of course, external contributors who are volunteers. The total number of contributors for Pardus at the time was 112, only 28 of whom were at any point employed by TÜBİTAK, meaning there were 3 external contributors for every 1 person employed. Tekman points out that such outcomes are common in successful FS projects.

Despite this healthy state of Pardus at the time, funding remained modest and the process of trying to secure greater funding from government agencies other than TÜBİTAK proved frustrating. Tekman presented this history in a July 2012 blog post titled "The unfortunate history of Pardus":[125] In 2006, a proposal was created to be submitted to the State Planning Agency to budget "a few million USD" for Pardus in 2007. The year 2007, however, started out with bad news. The proposal had not made it through the TÜBİTAK evaluation process and therefore was not submitted to the State Planning Agency. Tekman then turned begrudgingly to seek private venture investment for Pardus in Turkey and abroad. After his

123 Currently OpenHub. The website tracks code contribution and other statistics on open source software projects.
124 http://blog.erkantekman.org/2009/03/bilisim-dergisi-ozgur-yazilim-ile-katlanan-verimlilik/
125 http://blog.erkantekman.org/2012/07/pardusun-makus-tarihi-2007-2011/

pitching of the project to 150 companies, institutions and individuals, as of mid-2007, only 1 company and 1 NGO had expressed concrete interest in investing, which was wholly insignificant. In the following year, the 2006 proposal was updated and resubmitted in 2009, which this time made it through TÜBİTAK. The goal was to bring up the number of employed developers from 15 to 50, at a time when the original "dream team" of programmers were starting to move on from the project for pursuing their careers elsewhere. The State Planning Agency would only eventually greenlight the funding in 2010, after stalling in 2009 due to "bureaucratic disorder" (ibid). Finally, by August 2010, the number of developers employed at TÜBİTAK was increased by 10. At its height in June 2011, the number of paid developers reached 33, 2 of whom were part-timers. Plans were in place to bring that number up to 48 till the end of 2011 and 58 till the end of 2012, with 14.3 Million TRY funding reserved at the State Planning Agency.[126]

Unfortunately, later in 2011 the whole Pardus project would implode, with the vast majority of the developers leaving the project before most of the funding could be utilized. Just as the funds for new recruits were supposedly materializing, a "roadmap accident" resulted in "losing our (volunteer) developer community almost entirely", "a detail which we ignored at the time—a big mistake".[127] Tekman here is recognizing that the optimism due to finally getting the big funding secured from the state played a role in overlooking the deterioration of the relation between the paid developers and the outside volunteers—"the community". The collapse of the Pardus Project as a whole thus occurred within a short time frame of about one year, with the first step being the alienation of the very community that Tekman was pointing at as the source of "the miracle of open source" only two years before—a failure from the bottom.

126 http://blog.erkantekman.org/2011/06/pardus-yn-ana-sozlesme-2-0/
127 http://blog.erkantekman.org/2012/07/pardusun-makus-tarihi-2007-2011/

The second step was to be the arrival of new management at TÜBİTAK, which did not share Tekman's vision, resulting in the near-total shedding of the paid developers in late 2011 —a failure from the top. The much-requested and needed funds that seemed to be materializing in the very year this happened paradoxically may have contributed to the project's demise. The promise of large funding first seems to have contributed to an under-appreciation of the unpaid community due to an expectation of an imminent salvation through massive new paid recruitment, and then fallen short of fulfilling this promise due to reshuffling of the project management that ended in the dismissal of the paid developers. This bitter course of events provides an important part of the explanation of why, after the official project faltered, the once dynamic volunteer community did not pick up the work with the enthusiasm that it could have, which was technically entirely possible due to all code being FS and thus open for resumed (and if necessary forked) development. The infusion or expected infusion of big money into a FS project can in some ways play an unintended, detrimental role that is corrosive of community.

Before going on to consider how this story of the rise and fall of Pardus between 2003-2011 played itself out in the eyes of the Pardus developers and the Turkish FS community, a further contextualization in relation to the software sector in Turkey and the different interests involved in plans about the implementation of FS will be useful. The direct government spending on operating system licenses and other proprietary software is just one piece of the whole when it comes to a nation's spending on software IP rents. With private sector spending accounted for, the balance of trade in software in Turkey in 2009, the same year the Pardus project was trying desperately to secure "a few million dollars" of funding from the government, looked as follows: Turkish software exports were estimated to be at 13 Million USD, while imports stood at 110 Million USD (*BAKA* 2012: 11-13). Such a trade deficit in

software goods suggests plenty of room for a national import-substitution policy based on FS to operate in. When it comes to strategic targets, however, the Turkish government appears radically optimistic, if not cavalier. Despite the fact that "export of software is a relatively new phenomenon that has not so far satisfied the high expectations", Turkey has set for 2023[128] the goal of 50% of software used nationally to be domestically produced, and software export goals are set at a massive 10 Billion USD (Canlı, 2012: 23).

The capitalist Turkish software business interests have expressed various suggestions on how to improve the Turkish software industry through various industry reports prepared by sector think tanks. The Institute of Strategic Thinking, in a report titled "The Software Sector in Turkey" makes the following observations: "The biggest advantage of the software sector is that fixed capital requirements are low. There is no need for factories or huge R&D centers. . . . Barriers to entry are low and the global market is not saturated yet. Reaching customers is easy (through online distribution)" (ibid: 24). The reader will recognize at this point that the opportunities cited by local software sector capitalists are exactly the same conditions as what allows, under a different mode of organization, the forms of remuneration and the ethos for FS to develop as a communistic mode of production. The same centrifugal opportunity inherent in the production of software (and plausibly other digital artifacts) is here mentioned only in so far as it can serve to carve out a market share for capitalists of a smaller nation against larger ones.

The same is true for their approach to open source.[129] Open Source is mentioned as "a different business model" that has "important potential for Turkey". The report points out that instead of paying foreigners for licenses, one can spend that money inside the country, buying services (Note how the alternative of directly paying the FS laborers is avoided in favor of the

128 The centennial anniversary year of the republic that features prominently in the AKP's ideological symbolism around the concept of a "New Turkey".

129 Note the preference for "open source", the more business-friendly term for the same thing as FS

classic services model, which implies a role for capitalist mediation in FS production). Furthermore, the report suggests that services for Turkish-developed open source software can be sold to foreigners (ibid: 28-29).

Another important theme that the report captures is that a student can start coding before getting a degree, earning experience which "is an opportunity for developing human resources" (ibid: 28). And here is what this think thank has to say about the lot of this "human resource": "The IT sector is known for workers working overtime. Concerns exist about whether they are paid properly for it. Wages and benefits are not standardized. . . . Project-based hiring creates uncertainty" (ibid: 49). In opposition to this recognition of working conditions issues of the IT worker, however, is a riposte. The commonplace perception of "doing all the work while others end up with the earnings" among IT workers is described as "a falsehood and information pollution. . . All the profitability is expected to be linearly reflected on to the full operation and on to all the personnel, but things usually don't work that way" (ibid: 48). The report also recognizes the pressure on IT workers to constantly improve themselves, overtime pressure, and the anxiety-inducing pressure to acquire new skills in their free time (which is squeezed by overtime). This latter pressure under the regime of "management by stress" (Carson, 2008) is described as "an opportunity" for the precarious and flexible IT worker.

Finally, when it comes to policy suggestions, the sector demands the familiar neoliberal laundry list of pro-business incentives: tax-breaks, R&D support, public-private partnerships, and stronger IP protection. It is claimed that this package will result in the desirable outcome of improved IP exports. In a similar vein, the Informatics Association of Turkey lists problems around R&D incentives, insufficient IP rights, insufficient university-industry cooperation, high taxes, and "unfair competition between the public and private

sectors"[130] as industry grievances (TBD, 2012). Thus is represented the perspective of small to medium scale capitalists within a nation-state seeking to compete in the world market when it comes to software in general, and FS in particular. (According to a 2010 report by the Software Industrialists Association, 1600 companies operate in Turkey in the software sector, 87% of which are small and medium scale enterprises) (Canlı, 2012: 23).

Keeping this general socio-economic environment in mind, let us retrace the defining moments of the Pardus story in the community milieu I described in the previous chapter: Free Software Days.

5.2 Pardus Town Hall Meetings at the Conventions

Free Software Days 2006 and 2007: The Years of Enthusiasm and Cooperation

The 2006 Free Software Days opening session at the Middle East Technical University in Ankara featured a diverse assortment of individuals speaking about FS, Pardus, and Turkish IT policy, which resulted in a condensed microcosm of actors and perspectives on FS. It also started the tradition of holding one major session that served as a collective update on "The State of Pardus" every year at the Free Software Days.

Professor of Metallurgical and Material Engineering at METU, Bilgehan Ogel, presented a FS policy vision for Turkey with the following highlights that I am paraphrasing:

> FS is important for savings, security, IT development, employment and competitiveness. The EU encourages open source in the public sector, and open source has become government policy in many places such as Munich. The very birth of Pardus, which started out as project Uludağ,[131] is due to the security needs of the military. Commercial FS is possible through schemes such as dual-licensing.[132]

130 Whether this alludes to Pardus or potentially other government developed software is not made explicit

131 Uludağ is both the name of a mountain in Anatolia and a condensation of Ulusal Dağıtım which means National Distro

132 Dual-licensing or Multi-licensing is a scheme for the monetization of copyleft software based on the practice of authors of copyleft licensed code unanimously agreeing to sell exceptions to the enforcement of copyleft. In other words, copyleft FS is changed into permissive FS for payers. See https://en.wikipedia.org/wiki/Multi-licensing. While dual-licensing can be somewhat confusing, fork-prone

> Schools, SME's and the military should be given incentives to use FS. Pardus should not be mandated by decree, but should be evaluated by objective scientific criteria in a spirit of full competition. Domestic software production should be encouraged. Basic IT education should be conceptual (general) rather than brand and product specific. FS should be taught in classes and students should get their hands dirty with FS. The FS "way of producing" is distributive, pluralistic, democratic and based on producing together, sharing and creating a heritage common to all humanity. The same principle is at work with the creative commons, open access and open biology movements. Thus not only IT folks but political scientists and sociologists need to take it seriously. Mental labor is overshadowing manual labor. The production of the first digital copy is expensive but the following copies are free. IP should be limited and excluded from basic goods. Access to information is finding its way into constitutions as a basic human right. IP should be protected but a new balance must be struck. If it comes down to the masses against a monopoly the masses should prevail. Patients should prevail over drug companies. Software patents should be rejected. We propose not piracy but Free Software.

Arda Durukan, who led IBM's University Relationship team and worked on Linux Centers in Turkey was also present.[133] He emphasized how the "FS world view" was compatible with and similar to the mission of universities in society: the aim of FS is not profit but the benefit of society and science. Just as in a university, FS is open to research, creativity and sharing. Information increases the more it gets shared.

Next up was Recep Çakal of the State Planning Organization (DPT). The DPT was established in Turkey in 1960 following the "progressive military coup" at a time when government planning, a mixed economy and state-led development was seen in a favorable light. The organization was directly attached to the office of the prime minister. Over the years, the prominence of the DPT faded as liberalization became the dominant trend in the 80's, until in 2011 the AKP government restructured it in favor of the independent Ministry of Development. Çakal's appearance at the Free Software Days betrayed an air of bureaucratic detachment from the issue at hand. His speech, with nary an insight to offer, boiled down to a tired praise of competition: "Closed source or open, both are beautiful things that should be

and controversial, RMS has condoned the practice as acceptable, if not necessarily desirable. The ethical deficit is on the licensee who pays for the right to make the proprietary variant. See https://www.fsf.org/blogs/rms/selling-exceptions

133 https://tr.linkedin.com/pub/arda-durukan/28/362/895

competing. That's what matters," declared the bureaucrat at the 2006 opening session.

More of interest was Osman Coşkunoğlu, a professor of Systems Engineering. He has headed the Association for Social Democracy and was at this time in 2006 a Republican People's Party (CHP) MP and member of the Turkey-EU Mixed Parliamentary Commission. He praised the FS efforts in Turkey, denoting those working for FS as "the new forces fighting against the old forces", who display the healthy characteristics of "naughtiness and a love of freedom" in favor of distributed systems against tendencies of monopolization. Constructing a progressive narrative, he drew parallels between what he saw was happening with the early years of the industrial revolution.

The key speaker at this opening session was Erkan Tekman. An academic with a physics PhD, he had started The National Academic Network and Information Center (ULAKBİM) and The National Academic Network (ULAKNET) under the auspices of TÜBİTAK in 1996. In 2003, he started Pardus and managed the project until 2011,[134] the year in which the project transitioned into its ongoing twilight life. While Pardus was born out of an inquiry of the office of the prime ministry into the viability of "a national operating system" upon demands from the military, it materialized as Tekman's brain child. The eventual demise of Pardus had enough of an effect on Tekman that he abandoned his 15 year career at TÜBİTAK in 2013, in favor of establishing Linuxera, an "open source solutions and services start-up" (ibid):

> I gave the Pardus project an important portion of my life, perhaps the most productive and energetic period of it. My hand, tongue and head have been involved in the product and many of its components, including their names. This is where I learned what Free Software is and what it needs to be. Perhaps this is also where I taught the same. While I gave shape to the Pardus project, it also gave shape to me. The relationship and connection between me and Pardus is hard to describe, and I won't even try. [135]

134 https://tr.linkedin.com/pub/erkan-tekman/4/2bb/9a7
135 http://blog.erkantekman.org/2011/12/benzin-bitti-pasam/

Tekman remains the CEO of Linuxera, and two other ex-Pardus developers comprise his management team. The company sells installation, maintenance, support, consultancy, optimization, virtualization, system design, training and cloud computing services around Linux and FS, and is an official Red Hat Advanced Business Partner (Red Hat is the largest commercial Linux company in the world).[136]

Many years before Tekman's exit to small private sector entrepreneurship, his enthusiasm at the 2006 FSD opening session was unmistakable. He spoke with an aura of conviction and anticipation, emphasizing that while Pardus was officially a TÜBİTAK project, "its true owner is the Turkish FS community, where it all came from". He balanced this enthusiasm for community with the hard-headed emphasis on the need for providing "painless migration" for institutional customers who are always seeking savings and who are interested in Total Cost of Ownership (TCO), a business term that refers to transition, training and support costs of software on top of any license costs. Tekman pointed to the importance of these business opportunities and embraced them. He advised young people to found companies and enter this market, foreshadowing his own eventual future.

Tekman's interest in FS was never limited to his career, though. His personal blog is full of support for FS related causes such as promoting the Free Software Foundation and calling for donations.[137] He has made warnings against vendor lock-in[138] and closed document formats.[139] He also opposes Digital Rights Management (DRM) in e-books[140], an anti-copying measure derided by the FSF as Digital Restrictions Management, and has published a polemic against equating "piracy" with theft.[141] He protests against the "Microsoft Tax" on computers,

136 http://www.linuxera.com/hizmetler/destek-hizmetleri/
137 http://blog.erkantekman.org/2007/12/join-the-fsf/
138 http://blog.erkantekman.org/2008/07/bilisim-dergisi-bir-hapsolma-masali/
139 http://blog.erkantekman.org/2008/06/bilisim-dergisi-tehlikenin-farkinda-misiniz/
140 http://blog.erkantekman.org/2015/01/kitabin-elektronigi-ile-imtihanimiz/
141 http://blog.erkantekman.org/2015/01/caldim-ama-sor-bakalim-neden-caldim/

i.e. the bundling of Microsoft Windows with new computer sales.[142] On a more general political level, Tekman has protested the ban on May Day celebrations in Istanbul,[143] and has expressed his negative opinion on the AKP government: "I am not content with the existing government of Turkey. . . . Turkey is not rising, it is falling fast. The AKP has taken back in recent years some of the positive things they realized in the beginning of their rule and this negative turn will continue accelerating".[144] It is not hard to imagine that the fate of Pardus post 2011 only served to reinforce his position on the AKP.

At the 2007 FSD, Pardus developer Murat Eren shared some insights into the way things worked while he was employed at TÜBİTAK-UEKAE. His talk titled "How to Become a Pardus Developer" described the Pardus project as being run in a way "transparent to the public and open to the help of contributors, in accordance with free software philosophy" (There are differing opinions on the accuracy of this claim, as will be shown later in this chapter). He stated that the project was managed completely in the open with almost all internal communication done on open email lists. As for the developer culture, he said that no one should be assumed to be necessarily "responsible" for particular pieces of software, so individuals should not be blamed for what they have or have not done with regard to such specifics. He advised that the best way to find an answer to "When will feature X be added?" is to add it yourself: an aspiring developer should eye one bug and get to working on it. He also advised, however, to keep communication alive and not start off on tasks without notifying others. New contributors should "respect the e-mail ethics and act in a manner that is appropriate to the list atmosphere". An attitude that combines being open to criticism with the resolve to persevere in the face of the same is required.

Unfortunately 5 years later, Eren would be forced by circumstance to shift his interest

142 http://blog.erkantekman.org/2007/05/microsoft-vergisi/
143 http://blog.erkantekman.org/2007/05/bir-ayip-daha/
144 http://blog.erkantekman.org/2015/02/soner-cagaptay-yukselen-turkiye/

in the development culture and form of management in Pardus in an entirely different direction. In contrast to the positive, hopeful and ambitious spirit of this 2007 talk, in 2012 he would embark upon conducting a postmortem study on how things went at Pardus. He conducted and released a retrospective survey on the demise of Pardus called "Criticisms of Pardus by its users and developers",[145] which is an important historical record that is pretty unique because the prevailing attitude of most key Pardus figures after 2011 towards their recent past seems to have been despair and disillusionment rather than one of critical inquiry. The document contains comments on some of the points that have emerged in my interviews, which I will present later in this chapter.

Free Software Days 2011 and 2012: Conflict and Disillusionment

The tone of the Pardus meetings at the 2011 and 2012 Free Software Days conventions presented a stark contrast to the hopes and positive expectations expressed in the same meetings in 2006 and 2007. Generally speaking, during the successful course of Pardus development between 2005-2011, volunteer contributors, except for the ones who were eventually recruited as full time developers, had increasingly been feeling that they were being alienated as a result of the paid workers' snobbery and insularity. The accumulated tension between the employed developers and the volunteer contributors, which is a concrete illustration of the distance between potentiality and actualization with reference to the new forms of collaboration enabled by FS, broke out as open conflict at the Pardus meetings during the 2011 convention. I will have occasion to comment on this development in greater detail in the next section while presenting my interviewees' perceptions and interpretations of the issue.

In 2012, while the clash between the volunteer contributors and paid developers

145 http://meren.org/tmp/PardusElestirileri.pdf

remained unresolved, the Pardus meetings reflected the even more damaging conflict between the employed developers, many of whom had already resigned or were about to do so, and the new Pardus management. The key management positions in TÜBİTAK and the UEKAE had changed hands as a result of changing balances of power and policies at the level of national politics. A number of issues related to the new managing team's different vision and bureaucratic style, as well as their ignorance or skepticism about Pardus as a project established by a prior team, quickly led to further deterioration of the project. The final blow seems to have been the decision to allow the option of using Windows alongside or instead of Pardus in project FATİH (Movement of Enhancing Opportunities and Improving Technology), a huge Ministry of National Education "smart class" project by means of which "42.000 schools and 570.000 classes will be equipped with the latest information technologies and will be transformed into computerized classes".[146] The negative repercussions of this decision on the sense of mission and belonging of the Pardus developer team will also become clearer in the next section.

In 2012, in the aftermath of this management crisis, it was now the former paid developers who were angry at the volunteer contributors for entertaining the idea of regrouping around the sham ULAKBİM neo-Pardus project. I have field notes from a Pardus community forum style meeting where this 3-way tension was played out almost theatrically. It roughly boils down to the two camps alternately breaking solidarity with each other in response to the changes in the attitude of ULAKBİM towards Pardus. The forum was conducted after a workshop was held at TÜBİTAK and the decision was made to transfer the project from UEKAE to ULAKBİM and to move the project headquarters from Gebze, Istanbul to Ankara. Here is a snatch of heated debate from the forum:

TÜBİTAK developer 1: Pardus is Dead. Linux Users Association (LKD) should take

146 http://fatihprojesi.meb.gov.tr/tr/english.php

over Pardus and fork it as Zardus or something. The nationalist bombast is silly, what matters is spreading Linux and FS use.
Citizen 1: We should have gone with .deb or .rpm instead of Pisi.
Citizen 2: We're sick of the Pisi and package format debate.
Necdet Yücel: The current situation is not that bad. It looked like we would be down to zero Pardus developers. Now at least we still have a few.
TÜBİTAK developer 2: We were not able to participate in TÜBITAK decision making. Ahmet (referring to Ahmet Kaplan, ULAKBİM director and FATİH coordinator) was well-intentioned but he was unserious.
TÜBİTAK developer 3: The number of paid developers went from 30 to 24 to 10. 9 resignations in a week. Also as of now we have zero volunteer developers.
Necdet Yücel: The situation wasn't good before this latest turn of events, either. (At this point tensions rose visibly between Yücel, who was speaking for the volunteer developers, and the group of (ex-)TÜBİTAK developers)
TÜBİTAK developer Ozan Çağlayan: I had to resign after being assigned to Ankara, which I take as amounting to exiling me.
Citizen 3: Pardus is orphaned, bounced between UEKAE, BTE, ULAKBİM, there is no institutional memory in Pardus.
TÜBİTAK developer 5: That is not the case. Everything is well documented.
Citizen 3: The management doesn't care about Pardus. The Workshop was not taken seriously by the management. The Workshop presentation was awful. Ahmet means well but he is not competent. The community is weak. TÜBİTAK is needed.
Citizen 4: My expectations are very low at this point but there may be hope. The new TÜBİTAK management is incompetent. The community should take steps. The TÜBİTAK institutional connection became a liability rather than an asset because of the trademark issue.
Citizen 5: (old man with a lamenting tone) Pardus had inspired morale in the Turkish people...
Citizen 6: What's going to happen now? I don't want to move to Windows.
Citizen 7: The military will continue using it. What's going to become of them? (The group of TÜBİTAK Pardus developers break out in laughter)
TÜBİTAK developer 6: "Non-disclosure agreements (NDA) bar us from speaking" is a myth. Cowards are hiding behind this excuse. The NDA's only cover technical matters.
(At this point Necdet Yücel walks out of the meeting. The atmosphere is tense upon observation of this theatric.)
Citizen 8: It seems to have become like a corporate project rather than a community project anyway, with zero external developers.
TÜBİTAK developer 7: There are only a few developers who say they'll keep on developing without pay. Ahmet was good for a while, he messed up when he was assigned as a director over us. I also disagree with the mentality of "showing to the world the power of the Turk". FS should be independent of TÜBİTAK.
TÜBİTAK developer 8: TÜBİTAK sucks, I'd be better off developing for Fedora.
Doruk Fişek: (to the TÜBİTAK developers) So is it only now that you've come to your senses? Where were your minds when volunteer contributors hit zero? Besides, when the TÜBİTAK developers resigned, they didn't switch to volunteering. What's up with that?
TÜBİTAK developer 9: (answering Fişek) TÜBİTAK has changed at the roots, from

the developer at the very top to the very bottom.
TÜBİTAK developer 3: (to Fişek) We are not hypocrites. We weren't rejoicing when you guys fell out. What will the LKD do?
Hakan Uygun: (Accusing tone) You participated in the community with the cold attitude of a large institution.
Doruk Fişek: Within the group of 30 TÜBİTAK developers there weren't more than two or three people who communicated and shared things with us
TÜBİTAK developer 9: Our aim was for Turkey to have a FS policy. The FATİH project showed that everything is decided at the top. The most important question is whether Pardus will be in FATİH.

Free Software Days 2013: A Post-Pardus Protest Incident

The 2013 Free Software Days event was held in April at the central downtown campus of Bilgi University. During the TÜBİTAK sponsored panel session titled "The TÜBİTAK-ULAKBİM Mission: The Spreading of Free Software" delivered by ULAKBİM director Ahmet Kaplan, Ozan Çağlayan, the top coder of the Pardus project,[147] stood up from his seat among the audience to interrupt the presentation and delivered the following message: "It's been a year and a half since the team was reduced to zero. We, by the way, are of the old team. A year and a half later, you put out a product which is not even on par with the product from 4-5 years ago in terms of quality. It's obvious that you are producing an Operating System specially tailored for the ministry, and you say that due to their request you cannot open up the project. However, you use the Pardus name that we built over the years with our labor. We are bothered by this. You claim that you have advanced the Turkish localization. We looked about and inquired; it is nowhere to be seen. We read the scripts that you prepared for the distribution; it is a comedy and has already been subject to comic strips. You disregarded the entire accumulated knowledge and know-how. We had already accomplished everything that you just talked about in the [upcoming] enterprise version. Smart cards were working, we supported smart cards and all the rest of it one thing at a time. We had published the enterprise version in 2011. In the end, we are indignant, and we will keep following this issue. Thank

147 https://www.openhub.net/p/pardus-linux/contributors?query=&sort=commits

you." Thunderous applause followed in the conference room.

To this day this remains a unique incident, the only protest event to take place in the history of these conferences. It is a surfacing of latent conflict between the idealistic ethos and the drive towards self-realization of the coders and a bureaucratic government-managed project organization subject to changing political whims, that had been up to then concealed within the harmonizing platform of an—in principle—open, commons-based organization of production.

Free Software Days 2015: A Militarized Approach to FS Evangelism

The most recent FSD convention in 2015 hosted yet another heated disagreement about the philosophy of Pardus and FS, this time in the form of a strong audience reaction to "Pardus community team leader" Ümit Dereliler's presentation on "The Yesterday, Today and Future of Pardus". The discordance among the different conceptions and discourses that emerged during the discussion was powerful enough to merit being described as a small scale "clash of civilizations".

Derelier's emphasis was on firmly defined organization-building: "a team that believes" which will address "our chronic weakness as Turks to join together" in order to "build a strong foundation upon which the floors will be built". His body language and demeanor easily betrayed his personal history as a military officer. The way certain phrases rolled off of his tongue as a rapid succession of words *sans* punctuation such as "Team leader so and so will report on this and that" almost provoked a reflex in one to yell out "awaiting orders and opinions commander!". His presentation opened with the rather strong claim that "the freedom of a country starts with its Operating System", and he described the state of Turkish computing as being "under the invasion of foreign-origin software".

The talk proceeded in an atmosphere of heavy audience hostility. Piercing questions, negative and ironic crowd reactions, frequently interjected sarcastic comments mixed with giggles and a body language depicting incredulity. Some community figures shouted out objections to even factual claims, contradicting Derelier's narrative of events. The effort was accused of being "incompatible with the spirit of FS". Someone asked whether this militaristic model was developed by Derelier himself or whether it was copied from somewhere. The way the question was phrased insinuated that something this crazy could only have been cooked up by Derelier himself. The irony was lost on Derelier, however, who gave the wrong defense: "It is a completely authentic model. It is not copycat work." He later revealed that this organizational structure, complete with a strict hierarchy based on team leaders, assistants to the team leaders and even senior assistants to the team leaders, as well as the system of points-based reward and punishment was modeled on the structure of the Turkish Armed Forces. This was not met with enthusiasm by the audience. The abuse was never ending. Another member of the audience asked if the new Pardus would be licensed under the GPLv2 or the GPLv3. It was immediately obvious that Derelier was not aware of the distinction, and he kept repeating that "all of our stuff is open source". The question was posed several more times by others in an effort to humiliate him.

Derelier described the 2011 implosion of Pardus as "the intervention of dark hands". According to him, Pardus was the target of "a defamation campaign" and the fate of Pardus had been "much like the Devrim automobile"[148]. He accused the new management of Pardus for "turning Pardus into a house cat", a tame creature that never wanders outside, with shut-in development, no advertisement and no outreach. Having managed to score a number of

148 The "Devrim" (Turkish word for Revolution) was a prototype personal automobile developed and produced domestically at a state factory in the 60's. It never entered mass production. It became a nostalgic relic of the Import Substitute Industrialization era, a folk symbol of unrealized potential for national industrial independence.

sponsorship partnerships with small local businesses, he was quick to reaffirm the commitment to core values when cooperating with companies: "our rules will be in force, no compromises on freedom and security". He also mentioned plans for a distro specially tailored for "ladies", creating embarrassment among many in the audience.

Derelier's organizational enterprise for a Pardus revival was actually focused on the promotion of Pardus across the country and not on code development. This was not really taken into consideration by the audience. I tried to point this out to reconcile the two cultures (FS development vs. Pardus promotion) and soften the atmosphere, but it had little effect; passions were too strong. Derelier's "Pardusers Win" effort would include the following: creating Pardus tech support centers, creating hardware and software compatibility certification programs, encouraging Pardus communities and volunteers, founding remote education initiatives, a "You are not obliged" campaign aimed at being able to buy computers without bundled (Microsoft) operating systems, and "social responsibility" efforts. A "Parduscan" effort was designed to promote Pardus use in primary, secondary and university education. He had a chart showing an army of team leaders, assistants and representatives assigned to this massive on-paper effort, who would be selected on "career merit".

I think that the significance of Derelier's presentation was beyond its entertainment value. His militaristic idiosyncrasies aside, the position he finds himself in is a direct result of the gaping hole left behind due to the disappearance of the organic community around Pardus in 2011-2012. This must be seen as the new Pardus management's choice to outsource "community building" to an artificial cadre of "community volunteers" who are culturally completely alien to the FS tradition. The old Pardus community was vicious towards Derelier because they saw in his project a bad substitute for what they used to be.

5.3 Pardus Narratives by Pardus Developers

I have grouped my findings from interviewing Pardus developers in the following structure which roughly corresponds to the pre-Pardus, Pardus, and post-Pardus eras of their personal trajectories: 1. The motivations and ideals which guided them towards the world of FS in general and Pardus in particular, 2. Their work experience, with a focus on relations between the paid and volunteer Pardus developers, 3. How they frame Pardus in their discourses regarding serving the interests of the Turkish nation and the world public, 4. Their perspective on why and how the demise of Pardus-as-we-know-it came about, and what comes next.

Motivations and Ideals:

The most elementary reason for producing FS is personal use (Circle 0). This type of production seems to have pervaded the activity of my subjects as a small but constant undertone. Ahmet, for example, contributed a patch to add a feature to a FS IRC client he uses that would better integrate the use of the client on Pardus. Sometimes producing for self-use requires forking; Hasan had to update and fork a non-working tool used to compress javascript files to make it work in a web interface that had to run under very tight memory constraints. Mustafa was collaborating with film makers at a visual studies program and was excited by the prospect of using a FS video editor. This motivated him to help create a Kdenlive package that would work on the Pardus system, making Pardus one of the earliest distros to run Kdenlive without errors. Not all FS production for personal use was connected to Pardus. Serdar contributed a script to parse image files out of zip files to facilitate easier viewing of manga comics on Kindle devices. He says the fact that this is made possible because of FS is enough of a reason to contribute to FS. Kerem describes the experiential

excitement involved in such direct production:

> It's so phenomenal. Let's say you downloaded the Linux kernel in the morning, then you browsed the bug lists and set your eyes on one. You delved into the code for 4-5 hours and resolved it. The second you resolved it you sent the patch to the relevant developer, who happened to be at their computer and saw it. They checked it and confirmed your fix. They placed it in the main code repo and closed the bug. Let's say three days later a new version of the kernel is released. (Holding up his smartphone) On the fourth day something you solved ends up in the Android[149] phone you're using. It's a very interesting experience.[150]

The prospect of using what FS one creates is reinforced by several other intrinsic motivations. In many cases FS coders find the productive activity enjoyable and satisfying in its own right. Hasan describes writing FS as "a joy for someone who has embraced it". Serdar also defined the experience as "without a doubt enjoyable" as long as he was able to avoid "the boring things" while he was employed at Pardus, such as dealing with management. Another perk according to Serdar is that FS allows one to see other people's code, which is very satisfying for one's technical curiosity. He even described certain tasks such as identifying which version of one software works with which version of another—which sounded terribly boring to me—as "lots of fun". There is something to say about the diversity of human personality when it comes to self-expression through productive work in there.

FS also holds the advantage over proprietary software development in terms of nurturing a sense of community belonging and this can be very rewarding. This more positive socialization experience thus serves as another intrinsic benefit to working on FS. Ali: "Other developer communities did not appeal to me. Proprietary software created by commercial firms don't really have communities anyway. FS ones do, and they have lots of interaction among them. I liked that". Emir stated that he gained entry into groups and social settings thanks to his interest in FS that he could never have accessed otherwise. Ahmet pointed out

149 Google's Android mobile Operating System is mostly based on the open source Android Open Source Project, but includes proprietary components by Google.

150 For a discussion of how "user-centered innovation" contributes to the democratization of innovation processes, see Von Hippel, 2005.

that belonging to a community was enjoyable and led to friendships in many areas, beyond just interactions around the software. According to Serdar, the fraternizing is fueled by shared political and ideological perspectives:

> For me it all began when I started to understand the seriousness of this business by reading on its ideology. I read all kinds of documents on FS on Wikipedia. That's when I said "this is it". I was unemployed at the time, so I decided to make contributions. It started with embracing the ideology, and then getting to know others who do. They are all fantastic people, both personally, and technically. What enables this is FS, which is very important. That's why I like it.

FS also appears as a natural fit for those driven by the desire to learn for its own sake.[151] This is often reflected in the earliest periods of the life stories of my subjects. For example, Hasan:

> When asked since I was 5, I would tell people I would become a computer engineer. I looked up to my elder brother who was into computers. I always had the curiosity since early age so it was never a means to look for a job for me. I started being interested in Linux when I was attending the university and that turned to interest in Pardus when I came to Istanbul.

Hasan had even tried his hand at creating a distro of his own for experimenting before his involvement with Pardus. Mustafa stresses how being in a lively community feeds back into awakening the passion for learning new things: "The team around me was so technical that I started learning through a process of osmosis. I would observe how certain problems were being solved and that awoke the feeling in me that I could solve them too."

Learning is without a doubt also important for the benefit of future employment. Kerem attributed Google's contacting him for a job interview to all the things he had learned and the experience he had gained while working on Pardus. While Kerem didn't get the job, he was selected for the Google Summer of Code program, which provided him an income for a while. Ahmet stated that he noticed many individuals with email addresses from large

151 Schweik and English also cite the learning that is gained by reading and editing other people's code as an important part of the enjoyment and motivation behind participating in FS development (Schweik and English, 2013).

companies like Intel, Sony and Texas Instruments on the OpenEmbedded project, and that a good programmer who contributes to FS is likely to be recognized and land a good job later. According to Hasan, learning the ins and outs of FS used in niches where FS is dominant can be a good career investment. He points out that FS allows one to learn immediately through practice, without waiting to graduate, and the open development culture (open email lists etc.) that accompanies FS is very useful for young programmers to learn the intricacies of the trade such as up to date coding styles, which cannot be found in textbooks. Hasan also believed learning through FS was advantageous because practicing software programmers were better coders than university instructors and hence had more to offer students.

Even in the absence of specifically FS job opportunities, experience with FS production is experience with software production in general and plays a role in networking. Serdar mentioned that having worked on Pardus was an important factor in his getting his current job, which is not FS-based. In fact, he has two other coworkers from Pardus at this post-Pardus job. Not everyone's experience with job searches reflect the benefits of having contributions to FS projects on their CV's, however. For example, Ali claimed that unless the prospective employers are themselves part of the FS project, it won't help much. The potential employer might not even have heard about the FS project that appears on one's CV. Contrary to Ali, Emir believes that if he ever looks for a job in the private sector, he won't have to face a "who is this guy?" moment, thanks to his FS record.

Within the context of a single FS project, the path from voluntary work to employment can be quite transparent, as in the case of Pardus. According to Mustafa, up to 80% of the paid Pardus developers started out with small contributions like translations and bug reporting and moved up to submitting patches, seeking internships at Pardus, and eventually qualifying for the actual jobs. Mustafa pointed out that the internship experience at Pardus was much better

than the usual private sector internship because interns in the private sector are often given menial tasks instead of actual learning opportunities. Mustafa also added that TÜBİTAK was a good first employer for new graduates because it is public sector. Although the private sector offers higher wages in the long term, starting wages at TÜBİTAK were higher than starting wages in the private sector (Ahmet corroborated this). Not to mention that the public sector offers better status (researcher vs. junior programmer according to Mustafa) and unionization as well.

Most FS developers naturally tend to use FS on their own personal devices. This is sometimes a matter of principle, other times of technical preference. Ali used Pardus on his personal system for 3 years and was content with its basic infrastructure which allowed him to create packages for things he needed and rapidly present them for others to use. He also refuses to use any non-free software, as does Hasan, who is now interested in Debian Linux. Others take a more pragmatic approach to personal software use. Mustafa, for example, has bailed on the Linux operating system in favor of OSX on a Mac, stating he has lost his motivation to use it, but still prefers FS for applications. An interesting case is that while Kerem uses Fedora Linux on his primary machine, which is his laptop, he also keeps around a Windows desktop machine for use at home so that when he comes home he can relax instead of being tempted to get back to coding by his Linux system, which makes doing so very accessible (a psychologized reflection of the blurring of the work-leisure divide). Ahmet's main machine also runs Linux (the popular Ubuntu distro) and his phone is Android. Serdar runs varieties of Linux (Ubuntu, Red Hat) on "nearly all of his devices" including his computer, media player and phone (Android).

While personal use of FS is in a small way itself a contribution to the success of FS as per the network effect, the asymmetry still stirs the urge to reciprocate in appreciation (Cirlce

1). Mustafa mentions creating packages he wasn't really interested in but thought somebody could need, suggesting that he acted on a feeling of duty. With self-modesty, he belittled such contributions by saying he thought that someone else would probably have made them anyway. For many of my subjects, the Circle 1 mechanism has become ingrained in their lifestyles. Emir mundanely contributes code and translation work to numerous FS applications. Hasan, who complains about not having the free time to contribute to FS as he used to do because of his regular job and family responsibilities, stated that he does not plan to drop out without a fight: "I plan to contribute if I can make time for it. After all, it becomes a habit. One feels the desire to poke into everything." He hopes his current boss will be understanding enough to allow him to contribute to the TANGO network access control software for a few hours a week on company time.

This element of desire and curiosity, the enjoyment of the activity as an end in itself, came up repeatedly during my interviews as a motivation stronger than devising solutions for personal use.[152] All of my subjects recited incidents of working extra of their own accord, in general without overtime pay, either from home or at the office. Erhan: "Many would stay for overtime. I even know people who would sleep over at the office because their homes were far away. Actually it was quite a sacrifice." Serdar: "One of our friends created a graphic showing our work hours based on when commits were made to the system. They were almost always after 5 PM.[153] But nobody was forcing us, not Tekman, nor anyone else. We worked because we liked it." Mustafa points out that there is a dark side to this:

> The first 4 years I was supposed to be a part-time worker working 2 days but instead I was working 7 days a week. The extra 5 days can be considered volunteering. And the full-timers were no different. This was something we didn't dwell on but it was

152 This finding agrees with the conclusion of a large-scale survey of FS developers from a variety of projects: "Enjoyment-based intrinsic motivation, namely, how creative a person feels when working on the project is the strongest and most pervasive driver" (Lakhani and Wolf, 2005: 3).

153 He is referring to Gökmen Göksel's work at http://blog.gokmengoksel.com/2010/03/ne-zaman-gelistiriyoruz/ The stats show that around half of commits are made outside of work hours rather than "almost always after". Serdar is exaggerating a little to make his basically valid point.

> actually very exhausting. Just because we were doing what we loved most and they were paying us for it we saw no problem in devoting all our lives to it.

Mustafa is pointing at something important. An overzealous volunteer attitude, if unchecked or under-rewarded, is susceptible to burnout. Of all my interview subjects, Mustafa was one of the most consciously committed individuals to FS as a political issue, and has a history of volunteering for other progressive initiatives. He is also among those who came out of the experience with the greatest amount of disappointment, if not disillusionment.

The attitude of devotion to Pardus was not confined to the paid workers. Serdar mentioned that during the time he was a volunteer, he would sometimes work up to 8 hours. He partially attributed this to being a novice at the time, sharing that he could take up to 3 hours to do something he can now do in 10 minutes. Emir, who was a prolific volunteer contributor while he kept a separate full-time job, says they would sometimes work till midnight as a group on Pardus in his after hours. He recalled that once after reaching a major milestone, he emerged out of the ordeal with "a sense of victory" at 2 AM. He would also sneak in some Pardus work during lulls in his day job. Ahmet, who started as a volunteer while he was still in high school, shared that he would often rejoice when "opportunities" arose for him to pull all-nighters working on Pardus. A few years later when he became a paid worker, he would keep on working after break time and wouldn't even register his overtime to demand compensation. As if that were not enough, if something was stuck on his mind, he would also work on it from home, on top of the after-hours he had already put in. Especially in the earlier years of the Pardus project, the enthusiasm seems to have been unbounded. Ahmet recalls that around release time in 2007, he once walked out of a cram session at 4 AM along with a handful of others, including the project manager Erkan Tekman. Mustafa cited a similar story, where he had to sleep over for two nights in a row with a group that included Tekman in order to see something through. He praised the sincerity of his immediate project

manager, while lambasting higher TÜBİTAK management as "putting us through an experience of accelerated exploitation".

But what were the beliefs and ideals that fueled such devotion? For Salih, FS points to something bigger than itself: "This is about life in general. FS can be explained in terms of something more general. Software is just one part of it and that's what I got interested in. But it can be applied everywhere. Perhaps that was what attracted me." Ali: "I believe FS has an ethical value. I'm not sure if the freedom in FS can be a matter of politics but it definitely is a matter of ethics. I believe it will lead to beautiful things by distributing society's accumulated property in a more egalitarian way." Mustafa was very articulate about the implications of FS:

> From the start, I believed that FS was the most meaningful model. I saw that the way you form your relationship to what you use in FS suited the nature of information very well. I also found it very meaningful to defend the freedom of people to use something that they own against corporate monopolies. The experience itself is very important.

For Mustafa this perspective fit into his attitude towards working in the public sector:

> The only thing the public sector offers in exchange for your labor is enough money to lead a decent life, not more. Well, since I always wished the world would be like that in terms of political economy, I was contented. I never felt the need to compare it to how much I could have made in the private sector. . . . I never calculated how it would look on my CV. I don't believe the value of being part of a FS project can be measured that way.

The Work Experience

The work experience of paid Pardus developers is revealed to have been shaped by a mixture of complementary and contradictory dynamics. On the one hand, a relative amount of autonomy and flexibility of the workforce to determine their own activity, which is complemented by bursts of voluntary overworking fueled by an enthusiastic belief in the importance of Pardus (almost as a *cause*). On the other hand, the FS culture of development characterized by openness and loosely meritocratic decision-making clashing with a rather old fashioned institutional culture of a state institute managed from the top down.

The work day at Pardus was scheduled around a standard 8 hour day at TÜBİTAK's Gebze Campus. There was also always the possibility of working from home: according to Kerem, 90% of the tasks could be accomplished by working from home on a machine running Pardus just as successfully as by working at the office. Despite this, working from home was never officially recognized as counting towards the 8 hour day. However, office time was not strictly monitored, especially before 2009, which allowed the workers to work from home some of the time, without a reduction of their pay. Hours of entry and leave would be entered into the system retroactively at the end of the month, and the worker would simply declare to have averaged 40 hours per week. From 2009 on, the system was tightened by higher-up TÜBİTAK management after a few visits to the office from management were met with only a handful of the team present on site. From then on the office was almost always fully staffed. Even a more neoliberal, performance based system was considered by management at one point, but was not implemented.

I read these as signs of conservatism in TÜBİTAK management who cling to their old ways in the face of contemporary developments like the possibility of working from home and the FS work ethic that cannot be reduced to hour-counting or individualized measurable outcomes. In this sense, a FS project like Pardus run under government management can be likened to an organ transplanted into a foreign body. It was basically up to the project manager, Erkan Tekman, to negotiate this friction and secure a space of relative autonomy for the FS-minded Pardus developers to operate in.

Even the particular building that housed the Pardus office stuck out from among the others at the Gebze campus, with its fewer controls than the rest of the high-security buildings. This was beneficial to the FS work culture. Ali describes the setting: "If someone wanted to, they could play a Rock n Roll song or a piece by the jazz musician Jehan Barbur at

any time. It wasn't a stiff environment. No dress code. It was quite relaxed." Serdar attributed this to the role of Tekman, who organized things below him in a way to make it "a lot of fun", while shielding the Pardus team from interference from higher management.

It was not just about a different understanding of order and professionalism though. According to Eren, Tekman limited his authority over the project to defining a general vision and medium term goals, and stayed away from trying to micro-manage everything. Doing otherwise could clash with the FS development culture. Mustafa viewed Tekman as a "visionary project manager . . . who directed the project by making all the necessary compromises, negotiations and office politics . . . to create a system that rested on delicate balances. He tried to clear the way for us the best he could". Mustafa mentioned that higher management made insinuations that Tekman was too lax and that this may have been part of the reason for his eventual dismissal. He states that in a sense, "Pardus existed *despite* TÜBİTAK".

Significantly, the internal distribution of tasks was left to be decided by individual initiative. Eren: "Once I showed interest in doing security updates and working with other distros and learned these things while I was volunteering, I stuck to them when I got the job. I just popped up and said 'I'll do this one' and continued. I never heard anyone being told 'You're going to work on this and you have one week to finish it".

The project was run along meritocratic lines within the group of programmers. According to Serdar, "It might not have been spelled out, and it might have appeared democratic, but in reality it was a meritocracy. Nothing was decided by majority vote. Whoever was most knowledgeable on a particular thing was the one who decided." Osman: "The ones who made the most commits did not get to decide alone, but their opinions commanded a certain extra respect." Seniority also appears to have played a part: "If you can't

come up with something more sensible, someone who has been there since the beginning would have more weight". Serdar considers this a just arrangement in the context of FS production. Osman also reinforced Serdar's account by pointing out that more experienced, senior figures would be briefly consulted even on items they did not particularly have expertise on.

Although these mildly hierarchical practices that the developers framed in terms of a meritocracy do not seem to have presented a problem in the general working environment, a real point of friction was between the culture of FS and the governmental institutional culture of TÜBİTAK. According to Mustafa, large-scale institutionalization always runs the risk of overshadowing the qualities that emerge from producing something endogenously in the network structures of FS. Eren shared an anecdote that illustrates the culture clash:

> I had to fill in a form before starting to work at TÜBİTAK which was to be submitted to the National Intelligence Organization. Some of the offices required fingerprints and eye scans before entering. I declared to the security chief there that I'm going to start at my new job. He responded with Ahhh, one of the free kids, eh? You're on the Pardus team?"[154] This was my first interaction with him.

When I asked Eren whether this was just a sign of cluelessness, he insisted that this was part of a general attitude of belittling, symbolized by the free dress code of the Pardus team versus the suit-wearers. "They probably never liked us. When our office was relocated, I heard that they said they were happy to be rid of us".

On this point, Ali gives the example of how the government's insistence on holding the trademarks over the brand name and logo of Pardus ruled out the possibility of enthusiasts "brandishing something that they feel they are part of" such as wearing Pardus themed t-shirts. Such items could easily be procured by ordering from textile companies if it did not

154 The term "free kids" was popularized by a mobile phone services commercial. The protagonists were labeled "free kids" as a part sympathetic and part ironic commentary on their persona as nontraditional young travelers. The connotation in the context of Pardus is analogous to the unappreciative view middle-aged "squares" had for hippies in the US.

infringe upon TÜBİTAK's trademark. He considered this exclusionary IP to be a contradiction between the notion of Pardus as a free and public good and TÜBİTAK's view of it as belonging to the institution. While this may be a trivial example, the trademark issue came back to bite the community after 2011, when the new management was able to rebrand a version of Debian Linux as Pardus 2013. Had the Pardus trademark belonged to the community rather than TÜBİTAK, the sleight of hand could have been precluded.[155]

Part of the problem was sheer bureaucratic red tape. Mustafa cited a few such cases, with a tone of embarrassment and frustration in his voice. He wanted to set up access to the Pardus package depots for volunteer contributors, but this would be subject to a certain law that would require the contributor to disclose for record their identity down to their national ID number. This came into conflict with the pseudonym-friendly culture of FS. In order to facilitate the volunteer contributor who might want to submit their patches under a pseudonym such as "fluffy bunny", which Mustafa held to be their most natural right, he tried to put in place a workaround by getting TÜBİTAK to donate a server for this use to the Linux Users Association, and moving this usage to that server. It took one year to get this ball rolling, and Mustafa had eventually resigned before he could see the end of it. On another occasion, Mustafa complained that he wanted to placate the volunteer community every now and then by means of social events sponsored by TÜBİTAK, where there would be free food and beverages. This kind of spending, however, would require a complete formal public tendering process. Trying to offer freebies like USB sticks with the Pardus logo on them ran into the trademark issue and was subject to a lengthy authorization process. The TÜBİTAK red tape was further compounded by the high security environment of the Gebze Campus,

155 This is occasion to pay homage to another well thought-out position of Stallman. Stallman points out that trademarks "enable buyers to know what they are buying", and supports trademarks when they are used to protect consumers from fraud. Stallman opposes lumping together diverse things such as patents, copyrights, trade secrets and trademarks into a single category called IP. See http://www.gnu.org/philosophy/not-ipr.en.html

with everything from trying out wireless connectivity requiring permits to the ban on bringing in personal electronic devices to the facilities.

Beyond the bureaucratic frustrations, however, perhaps the biggest damage the state institutional culture inflicted on Pardus was at the level of the relations between the paid and volunteer producers, in other words at the level of community, as already seen in the previous section's discussion of the Pardus meetings in Free Software Days. In fact, it can be argued that the much bemoaned divide was not so much between the paid workers and the volunteers, as it was between Pardus management and the community (which consisted of paid and volunteer wings). According to Emir, the issue was not that some were getting paid while others were not. The issue, in his view, was the management, including Tekman, who, along with the paid developers at the time, was of the opinion that volunteers were only doing 10% of the work and could therefore be dispensed with. As I have cited previously in this chapter, Tekman himself eventually came to regret not considering the needs and desires of the volunteers to the extent that he should have. I must also note that this 10% estimate was not shared by some of the paid workers I interviewed. Eren, for example, estimated that it was closer to a 70%-30% division. Part of the reason for the subjective assessment is that neither counting commits nor counting lines of code is necessarily an accurate reflections of the work done.

The paid workers were often torn by contradictory demands coming from the two sides. Emir, coming from the volunteer perspective, complained that the paid workers would sometimes hold meetings among themselves to decide on things and then communicate the decision to the volunteer community in a fait accompli fashion. Emir's opinion on this was: "you might not reach the best decision by including us in the decision making process, but you will certainly not reach the best decision by excluding us". Hasan gives us the perspective

of the paid programmers and a sense of their predicament: "A lot of conflict emerged. We just couldn't get the notion 'this is the way FS works' through to the higher up managers. When we turned to our friends on the outside and said 'this is the way TÜBİTAK works', some understood and some didn't."

Further complicating things were the demands voiced by the "customers" of Pardus: the offices of government such as the Ministry of Defense, Energy Market Regulatory Authority and the Radio and Television Supreme Council, all which had started using Pardus. According to Hasan, satisfying their needs often had to take priority over what the volunteer community preferred to focus on. This rather out of place "customer demand" pressure on Pardus was felt because of a curious market arrangement existing between one government institution, TÜBİTAK, and the others: TÜBİTAK was actually being paid by the Pardus using arms of the government in exchange for providing them the Pardus product and service—recall here the troubles Tekman went through trying to secure funding from the State Planning Agency as an alternative.

Some of the tension between the paid developers and the volunteer developers seems to have been a result of the social need for recognition, and even distinction. This manifested itself in the proliferation of labels: According to Serdar, who didn't care much for the whole issue, where once everyone was referred to as a developer, later distinctions were made between developer versus contributor, and then to distinguish coders from translators, contributor-developer vs. contributor-translator. A nomenclature of waged developers versus unwaged developers, and internal developers versus external developers was also in use. Those who committed more effort to Pardus (predominantly the paid developers) wanted recognition for it. But volunteers also desired recognition and respect in their own right. With his experience of doing volunteer work in the past for Greenpeace, Mustafa points out that all

a volunteer expects is a little bit of appreciation. He admits that they alienated and scared off very valuable individuals who voluntarily contributed pieces of code to core parts of the operating system that would never directly benefit themselves in their use of Pardus but could "change the life of someone else". On the other, hand Mustafa was not sympathetic towards all the volunteers as a group: "80% of those complaining the loudest were those who only wanted to have their way, without necessarily playing their part." He continued:

> FS doesn't mean that you get to tell other developers what to do. They have demands, and when these are not met, they try to call others to account. This entitlement is then further amplified by the crazy notion that they are paying our wages through their taxes. "I recommended Pardus to all of my friends, what else do you want me to contribute?" they say. Well what is it to me? Pardus is *yours* to begin with. My wage doesn't go up with the spread of Pardus. When you took a step to spread Pardus and to use a better product, you did this for yourself. We were never able to communicate this.

Another problem was the relatively insular conduct and harsh communication style of the paid developers. This is very clearly accounted by my subjects when recalling the times before they landed the jobs, back when they were volunteers. Eren:

> I also had difficulties communicating with the first team. I got beat up a lot on the developer mail lists. Even by persons I really get along with well nowadays. I wasn't particularly unhappy but I often wondered "what in God's name is going on?". You write an email and you naturally get angry when you see the response. You're thinking you did a beautiful job and someone responds by saying "I didn't ask you to take a shit on the code". Then when some people couldn't handle this stuff, and rightly so, they were turned off. Some tried to contribute in their own ways, others dropped out.

There wasn't necessarily a lack of ideas on how things could have been improved. Mustafa stated that he would ideally prefer a more participatory and democratic Debian-like decision making structure for better collaboration between the paid workers and volunteers. Instead, he at times found himself making choices at odds with what he would personally condone if he were not a TÜBİTAK employee. For example, he believed that release directors (an individual temporarily put in charge of coordinating the release of a new version) did not necessarily have to be drawn from among the paid coders. TÜBİTAK, however, demanded

control, citing the importance of the project for the state and the investment made by the government. Mustafa considered this to be less than ideal, but not bad enough that he would consider resigning over it. Such is the experience of (ameliorated) alienation in cases of "FS under external management" that comes along with Circle 2 remuneration. Values and priorities may need to be compromised in order to sustain the support of the funding source.

What happened instead of such rectifications was a decline in volunteer contributions and eventually a complete fallout between the two wings of the community, which managed to ossify in two poles despite the permeability mechanism that permitted volunteers to eventually become paid developers. An interesting development along the way was the 2007 "1≠4" affair. Serdar, who was a volunteer at the time and the one the credit goes to for coming up with the name Sudrap (Pardus spelled backwards), gives the following account of the affair:

> The email list fights turned into an avalanche, so we decided to hold a face to face meeting at the Computer Engineering Students Congress. Some people started grouping around the idea of Sudrap (as an alternative test repository). It was very difficult to get contributions accepted into the (official Pardus) repository. Two different individuals would check them and they were very meticulous. People lose their enthusiasm. Then start the fights about how people get beat up and humiliated on the email lists. At that point Kerem makes an inflammatory comment: "When you look at the stats it turns out 1 foreign contributor does the work of 4 Turkish contributors. We really ought to be looking for more foreign contributors." We had foreign contributors at the time who did good work... I thought there was no way 1 foreigner could do the work of 4 and that this kind of discrimination didn't make sense. So we made "1≠4" our logo for Sudrap.org and placed it there that day. It was a protest. I thought at the time that Kerem was a terrible person. When I later found out he wasn't like that I suggested we remove that logo but they didn't accept it.

Interestingly, Mustafa had a slightly different recollection of the affair, where the question was not about foreign versus Turkish volunteers, but volunteers versus paid developers:

> In the period where a lot of fights broke out because volunteer contributors were not paid enough attention to, one of the waged developers, Gökmen Görsel prepared an infographic that classified all commits according to paid workers and volunteers and published it on his blog.[156] It turned out that 4 contributors were equal to 1 paid

156 The infographic in question can be found at http://blog.gokmengoksel.com/2010/03/ne-zaman-gelistiriyoruz/

> developer. The volunteers were so offended by this... As a reaction the volunteers decided to found a portal where they could share their development work among themselves. They founded sudrap.org by spelling Pardus backwards. They also had a manifesto there that stated "1≠4. You need to understand that this non-equivalence is not only an arithmetic one".

The Sudrap repository functioned as an unofficial Pardus repository where volunteers could test out packages before submitting them to Pardus. Serdar calls it "the minor league" where most of the volunteer contributors grouped. The "1≠4" affair was resolved in a productive manner with Sudrap and things seemed to calm down for a while. According to Serdar, the second shock came with the "Jira affair" in 2010:

> A project management software is used for tracking who is doing what and who is free. Erkan Tekman and the management thought this was needed to keep track of work processes, recruitment and termination. There were two alternatives, Jira and Redmine. Redmine is FS while Jira is proprietary but free of charge for FS producers. Jira had more features than Redmine. We all voted in favor of Redmine. The FS was good enough for our needs plus we weren't that many people. They didn't like it and insisted on Jira, so Jira it would be. This broke off the already strained ties with the volunteer community. They questioned why the hell we were using Jira when there was the Redmine alternative, and what kind of decision making process this was. Amidst the storm Doruk Fişek[157] quit volunteering. When Fişek left, it turned out a bunch of others were leaning towards quitting and things broke off. There were only a few contributors left and they didn't speak up and faded away towards the end of 2010. It was only the foreign contributors who were left, who had no clue what was going on because they didn't understand Turkish.

A Turkish FS: the National and the Global

Ever since its inception as 'Uludağ' (National Distro), a synthesis of the national and the global has existed within Pardus. On the one hand, the entirety of the project can be seen as but a local branch upon the gigantic global tree that is FS. The labor that has gone into creating Pardus as a specific distro pales in comparison to the labor that went into the creation of all the upstream components that Pardus draws upon. On the other hand, the novel

While the post includes statistics on TÜBİTAK workers vs. volunteers in terms of commits which reflect between a 1:3 or 1:4 proportion, the title ("At what times do we develop") and main focus of the post is about the hours in which commits are made. The post does not come off as inflammatory or particularly targeted at underlining the difference.

157 A leading figure of the Linux Users Association, see previous chapter.

additions distinguish Pardus from the other hundreds of Linux distros enough to make it a separate, coherent whole. It therefore simultaneously appears as a tangible, genuine achievement of the Turkish FS producers. The government sponsorship and the connection to concerns of military security further reinforce this aspect. However, as per the copyleft terms of the GPL, all the contributions of Pardus to FS become immediately part of the global commons of FS. The local then is defined purely by the local priorities in contributing to the global common. This is a sublated expression of what is local, because it is expressed within the mode of the global, which is an important idiosyncrasy of commons-based digital production.

FS harmonizes the interests of localities with the interests of global humanity in a unique and non-conflictual manner. Because of the novelty of this sublation, it is not always clearly conceptualized, and public discourses in the case of Pardus may sometimes diverge towards false extremes of the local versus the global, as exemplified in the previous section. In my interviews, however, I found that my subjects had an advanced grasp on this issue.

Let us start by acknowledging the direct use-value created by the Pardus team for the international pubic, which in turn was met by acts of reciprocation. Hasan pointed out that the Pardus project often contributed straight to upstream, in every field including hardware drivers and desktop applications. Furthermore, Kerem explains:

> We added local customization to the Firefox we shipped in Pardus. For example, if someone installed it in the Dutch locality, it would appear with bookmarks to Dutch news sites and banks, etc. Foreign users used this feature a lot and whenever something broke they would open bug reports about it. People really like small things like these. . . . Because of particularities and the originality of the project, we had high use abroad, and in some countries it had many fans. For example, Russia. Even while people were resigning here, we were getting daily mails from Russians offering help. The French had made a really beautiful web portal, much better than ours in Turkey. Good news, with screenshots and a forum. The Dutch had a Pardus World Forum. So we had a lot of international users.

This was a result of Pardus's being a genuinely good product by the standards of the

international Linux using public. Kerem: "We received many positive reviews, full of praise. We would hover around the 40th spot on Distrowatch.[158] Around times of new releases, we would climb up as high as the 32nd spot."

Let us now explore the local angle. The first and most obvious expression of the local concern is guaranteeing Turkish language support in software applications. Eren gives us an example of how Pardus contributed to this, not just for Pardus users but for all Turkish speaking Linux users:

> Pardus made a big contribution to all users of Linux who use Turkish localization. Imagine you run a Python module, and it works in English. Then you switch to Turkish, and it fails to run. What kind of a bug is that? It's because there are two Turkish letters, "i" and "ı" and these get messed up.[159] Our friends working at Pardus fixed this for good and sent it to upstream. If it weren't for Pardus, you would report this upstream and it would take years for it to be fixed.

Kerem offers an important insight on how localization is achieved in FS:

> The localization process works like this: Let's say someone creates a software but overlooks localization support. It only works in English. Next, someone comes along and makes it localizable. It's still English but at this point it becomes easy to integrate translations. Not only did we add Turkish support to many projects that were at this stage, but we also added localization support itself to many of them and then added Turkish. Translations were never our priority but we did as much as we could.

And therein lies the interesting part. First, FS universally allows localization contributions to be made, thanks to open source. Then, the very effort to add localization support by one local party opens the doors for all other local parties to add their translations easily without reproducing that effort. Kerem found it regrettable that many people didn't understand the nuance in this and belittled their efforts by claiming all they did was to add Turkish translations.

Operating with the native tongue is important not only in terms of using software, however, but also for communication access to the development process, especially for

158 Distrowatch.com ranks Linux distros according to popularity.
159 See http://mattryall.net/blog/2009/02/the-infamous-turkish-locale-bug for "The infamous Turkish 'i' bug"

newcomers. Eren recalls: "I tried Mandrake Linux back when I didn't have much Linux knowledge. I couldn't use it very well. When I tried Pardus I was able to use it better and I preferred it because I had heard in the press and on the web that it was developed by TÜBİTAK and that most of the developers' native language was Turkish so I could communicate with them in Turkish."

Local development facilitated by local language use also mobilizes the national identity as a marker of social proximity. As Emir describes:

> The fact that Pardus was a FS project being developed in Turkey was a big source of motivation for students. The kids could go to Pardus for interning during summer where they could see in person the developers of the operating system they are using. This was very appealing to them. They could observe that those people on the team were standing in their shoes just two years prior. Pardus greatly strengthened the feeling of "I can do it too" for them.

The legitimation for national investment in a high aiming FS project like Pardus always rested on two pillars of national interest: the economic interest and the security interest. As Serdar pointed out, "part of the vision of Pardus was to create a FS sector in Turkey." Mustafa underlined the developmental aspect of this concern and its connection to national independence:

> Paying engineers instead of paying for licenses would go beyond cost-saving benefits. It would be a good investment to train qualified workers, which would amortize the costs of the investment. Plus we would be master of the project's road map because it is our product. That's independence. . . . It is possible to unearth potentials in Turkey to produce for the country's needs, without technological dependence on abroad. Pardus has always been a distro that provided good solutions to certain technological peculiarities of the Turkish market. . . . It doesn't matter what Pardus technically created. For arguments sake, to even find out how stupid the Pisi package manager is, people need to first create something like Pisi. It is the experience and know-how that matters.

Eren explains the security aspect:

> You might tell me I've seen too many Hollywood movies but that's not the case. Routers are running closed source software and we don't know what's going on with them. Are we being monitored? Where is all our data going? All the users in the country, including politicians, members of security forces, and most importantly you

> and I, don't know where our data is going. We need FS to be able to know. That's the political side of this.

To conclude the discussion on how the local and the global plays out in the case of Pardus, I point to the transforming effect of the experience of producing a national product in sublated global form. Recall how in the previous chapter Gözükeleş formulated the effect production has on the human subject: "Unlike god, who remains the same god before and after creating nature, when humans change nature through technology, they also change themselves". Note the evolution of Eren's perspective: "My point of view was narrower in the past. It was more about doing something for Turkey. I later understood that while we are doing this for Turkey, we are using FS tools, we are contributing to upstream and therefore we automatically end up working for the world, for something global." Similarly, Emir, has shifted further toward acknowledging the importance of making the most of the already existing global commons for a "local" project to succeed:

> In the beginning I thought that the fact that such a large project began in Turkey really changed the vision of engineers and students. Nowadays, I don't think Turkey needs to engage FS at the level of the operating system by creating a new distro from scratch. Even China nowadays bases its distro on Ubuntu. It would be more productive to take a Debian variant and work to develop the applications we need on top of it.

There were of course also those who went into the project with an already formed consciousness of the issue, such as Mustafa:

> There was always a conflict between the notion of "For Freedom" and "For Turkey", and the definition and perception of a "national operating system". I never accepted that definition. . . . Yes, it was true that Pardus was a national project. But my only concern in terms of the nation was that it was better for the country's funds to be used this way in government institutions. . . . There is a lot of nationalist fuss about building our own tanks, etc. While over here we have the chance to really use humanity's accumulated knowledge and include certain solutions particular to this land to make a wonderful product. I don't understand why it is so hard for people to situate this model in their heads.

The Demise of Pardus

Several factors culminated in the effective death of Pardus in late 2011. Some sources

of its fragility as a project were there from the beginning, such as precarious funding and a chronic shortage of personpower. As Mustafa puts it succinctly, Pardus was always "the size of a startup, going against established institutions (referring to other big-name Linux distros like Ubuntu, Red Hat, etc.)". But these were not the real factors that led to the downfall. If anything, around 2011 both the funding situation became promising with the DPT funding looking like it would be coming around, and the constant trickle of new recruits led to a labor force of up to 35 persons, which was the all-time high. The real causes were first the bottom-up failure that was the loss of the volunteer community which I already described above, and second what appears to be the top-down sabotaging of Pardus after the change in TÜBİTAK management and the devising of the FATİH project. I now focus on this second factor.

Despite nominally being a government project, Pardus never had serious support from the government as a whole, neither economically nor politically. The disconnect was on two levels. The first was between the project management under Tekman and TÜBİTAK as a whole. Emir illustrated this disconnect by pointing out that TÜBİTAK never used Pardus on its own systems and that "it was crazy to expect others to use something they didn't use themselves; naturally, it didn't work out". The bigger disconnect, however, was between TÜBİTAK and the state. As we have seen, for a long time Pardus had to sustain its existence by creating a product that TÜBİTAK could then sell to other government units in a strange intra-governmental market relation. Even with the DPT coming in at the latest hour, Mustafa informed me that Pardus worker wages were never paid by the treasury. They were always paid out of TÜBİTAK's budget, including income from Pardus. The DPT money was only spent on infrastructure. Mustafa's dream for wages to be paid at the state level from the national budget, and the state to go even further by sponsoring certain LKD projects related to Pardus never materialized. This entire situation had its root in what Kerem describes as an

indirect justification of Pardus for the greater part of the state. According to Kerem, most countries act on FS with the primary motive of cost-saving in the public sector. But in the case of Pardus, the process started with the military's demand. The product was conceived for the military first and then started looking around for users in the other state units. This lack of real state-wide support for the project rendered Pardus highly susceptible to changes in the whims of TÜBİTAK. And such a change came about when the entire management of TÜBİTAK was replaced in August 2011.[160]

My subjects all agreed that this change made the future of the project uncertain. Kerem underlined how the institutionally insecure position of Pardus that I just described made them vulnerable to the downturn: "I believe the new management had it in their minds to get rid of Pardus. Right from the start it was only with the initiative of the original management back then that some things had been secured. It never got the expected support at the level of the prime minister or the head of TÜBİTAK. It was just our institute that put some heart in it." Emir linked the change to bigger changes in public sector policy, alluding to a neoliberal spirit:

> A change of mentality at TÜBİTAK... Not just TÜBİTAK but all public institutions. They prefer to outsource things instead of doing things themselves over 3 or 5 years. This is used for politics. TÜBİTAK viewed Pardus as any odd job. They didn't care for the accumulation of know-how or how Pardus expanded the horizons of many. TÜBİTAK is not even running their own mail server despite having excellent personnel to do so.

Hasan, who lost trust in TÜBİTAK completely after "the ugly way they ended Pardus and switched to the Debian-based 'Pardus 2013'" blamed the new management of using the quarrel between the paid and volunteer developers as an excuse to shut down the project. He described how the volunteers for a while thought they could get along with the new

160 One of my subjects perceived the change as the result of two politico-religious sects fighting over control, with the Gülen sect winning in 2011. Interestingly, this TÜBİTAK management was once again removed in 2015, with accusations that the old management was linked to the (now criminalized) Gülenist sect. (The AKP government and the Gülen movement had a major fallout in 2013).

management but then realized the management's intention to kill Pardus and lost interest. This general lack of trust and strong resentment toward the new management becomes particularly vivid in Serdar's narrative:

> With the change of management a conservative management mentality set in, kind of like sharia. All kinds of arbitrary personal decisions that are not in accordance with the laws and rules, mobbing, every kind of negative thing you can imagine. I'm telling these in a very objective fashion and you should put all of it in your thesis. . . . For months there was a tense waiting. Then a meeting took place. Tekman gathered all the developers and spoke to us. He said that he would not be offended or angry if we were to quit right then. . . He was implying we should leave for our own sake. We didn't, we stood around for another few months. We were expecting an explanation. Then there was some sort of panel at the institute. Everyone was there from the UEKAE institute as well as us. By the way, they placed us under a different institute, we kept getting shifted around to different units. The new head of the institute spoke on the mic, stalled and smiled. He said his voice sounded really funny on the mic and laughed. . . . He then started telling us about his graduate degrees from the US and such. . . . Why would you talk about this stuff? We could figure he was a petty man. Then he mentioned his children, and asked if the Pardus team was present. He smiled. He had such an ugly smile too. He broke the news that we weren't going to be used in the FATİH project. He said that he heard Microsoft would be making a 5 bucks discount per product and laughed. I mean... this is a project attached to your institution that your team is developing. You humiliate all of them in front of everybody with that smug attitude and you're laughing. That's unacceptable. Meanwhile he keeps talking about Microsoft and work he's done related to Microsoft. He keeps using English words in sentences, he's not even able to speak proper Turkish. We were so enraged. I wish we had walked out of the auditorium right then. We later heard he complained about how grouchy the Pardus team is. You can make out his qualifications from this. He's unqualified.

Contemporaneous with the management shuffle at TÜBİTAK, another major development in the IT field in Turkey was unfolding. In November 2010, Prime Minister Erdoğan had launched the FATİH project to coincide with his election campaign at the time. As he toured the country promising every student a tablet computer, the question naturally arose of what operating system would be used on the FATİH project devices, namely the tablets, smart boards, servers and teachers' computers. Pardus vied against solutions from Microsoft, Apple and Google. Lobbying intensified. Representatives from the companies visited Turkey, and invited government figures to tour their facilities. Many of my subjects

believe that this competition was what dealt the death blow to Pardus. Ali says: "My opinion of FATİH is that it's a fiasco... FATİH was going to be all Pardus. But Bill Gates came for a visit and things changed. Then ideas popped up like 'let's offer Windows for free' or 'It would be fine if we had both'". Hasan adds, "they're dual-booting Windows and Pardus with Windows the default. We heard they even sealed up some usb ports with silicon (so people couldn't change this)." Eren believes the political climate weighs heavily on the affair:

> I don't think highly of that FATİH thing. Big money involved. My opinion is that it was designed to allow some people to make a lot of money. (When asked whether FATİH was what killed Pardus) I don't know. Seems like we twisted somebody's tail. This is only half a conspiracy theory. We can't be certain. Perhaps one day the political climate will change and our managers and others involved will be able to speak out and then we'll be able to learn. It's completely political. People are scared to speak up in case something happens to them. If one day things change and someone decides to make a book about the Pardus story people might speak.

Mustafa provides the most detailed narrative based on the same theory:

> I believe what brought upon us our demise was the FATİH project. A decision was reached that Windows and Pardus would be used simultaneously in FATİH. We all knew this meant only Windows would be used in practice. This was the biggest injustice ever done against Pardus. I prepared my letter of resignation the day I heard Windows would also be included in FATİH. Because that meant we were being sidelined on a project so important despite having worked so hard. They openly let us know that Pardus would never be used for an actually serious informatics infrastructure project in this country. That means it's not part of the state's policy, it's not on the country's agenda. I thought to myself we'll never be useful for anything, why should I devote my life to Pardus. . . . I ended up being removed from my position on the project. They didn't even find a replacement for two months. No project manager, either. We sit around the office, purposeless. . . . With no one knowing what to do we all fall like dominoes. We waited around for management to tell us something like "alright new version before March, let's get these fixes in till then". We could work till the morning lights but someone needs to tell us what's going to happen. . . . TÜBİTAK should be demanding something from me. It's funny because our feeling of responsibility was stronger than TÜBİTAK's. TÜBİTAK had no idea. Turns out they never really understood the project to begin with.
>
> We knew the ministry of transportation (Binali "Binary" Yıldırım) was in the lap of Microsoft. He is crazy about Microsoft and sucks up to capital. We mobilized everyone we knew in the Ministry of Education, the Treasury Department, etc. and they sided with Pardus. It ended in a compromise, and they decided to have both Windows and Pardus. We later learned Windows was the default.

> I have no positive proof, because proof never emerged and probably never will emerge, but here's my take. From what we gathered, we made so much lobbying to get Pardus in exclusively and to turn FATİH into a national informatics technology development project that we stepped on some people's toes. So they had to get rid of us one by one.
>
> The Pardus project also took upon itself the job to transition the Energy Market Regulatory Authority (EPDK) over to FS. A 10 Million dollar project. All of it was to be paid to TÜBİTAK and spent on FS. With a pie that big, a dog eat dog situation emerged at TÜBİTAK. Some people wanted to push our project manager aside and get hold of the EPDK deal. They started to make political moves to sideline the Pardus project manager and tie Pardus to themselves. But our office was also strong at the time. We had already scored 2 Million dollars from the ministry of defense. We also had the funds allocated from the State Planning Agency. Our project manager held the key to the brand name power of Pardus for both Turkey and the world. We weren't an easy chunk to swallow. But over the years the political balance of forces changed. They managed to put a new management in.

According to several of my interviewees, Pardus was reduced to being used as a bargaining chip in the dealings with Microsoft. Mustafa:

> The new management perceives Pardus only as a bargaining chip to be used against Microsoft. They can threaten them with Pardus and get a discount. I don't have proof, this was said orally, not in writing... From what we heard, the ministry of education's assistant said this on the phone: Microsoft opened at no lower than $55 per device. When they threatened to go Pardus only, they were willing to give the government $5 per device instead. Had Microsoft missed the boat on FATİH the position of Windows in Turkey would be seriously jeopardized. Since they became the de facto in FATİH their way is open, they can charge as much as they want for the next version.

But not merely a bargaining chip, Pardus after 2011 would continue in some zombie form to serve as a name-holder for previously arranged bureaucratic proceedings. Hasan: "The official name of the project was changed to 'Pardus For FATİH' before the end. I have suspicions that the only reason the Debian-based Pardus exists is that the Pardus name went on paper on some tender bids."

CHAPTER 6: CONCLUSION

My central claim in this dissertation has been that FS is a new mode of production unlike any other in history, with its own idiosyncratic characteristics. Although this mode of production is only emerging as a historical tendency, it promises to be more than a niche phenomenon since it is the discovered and practiced organizational and distributional form of producing economically vital information goods, which in contemporary capitalism is not a peripheral but a core-like process. Embodied as open, public digital artifacts, these are the same information goods that would otherwise play a high value-capturing role at the core of the world system which relies heavily on informational capitalism. Hence, FS is a challenge to the system at the core, in a leading sphere of production, abolishing the precise ground whereon industrial capitalism has been giving way to a neo-rentier system based on IP ownership. Preconditioned by the contradiction between the nature of digital artifacts and the commodity form, FS has emerged at or near the top of the sectoral hierarchy in the modern world economy.

Furthermore, FS is not an idealistic endeavor or utopian experiment although it includes a healthy dose of both. It is based on the utilization of the state of the art means of production consisting of IT technology and a unique organizational form that enables productive freedom, the highest form of which is the possibility of forking. It is technically supremely efficient on the basis of inputs to outputs, and furthermore, the production process embodies an element of self-realization as opposed to alienation. It is this combination that makes FS and FS-like processes not merely *alternative*, but radically *progressive*. While alienated *software engineers* organized under competing capitalist companies are constantly wasting labor due to duplication of effort behind closed doors, self-organized communities of

hackers are collaborating openly to produce excellence with minimal wasted effort. Compared to the IP-centered proprietary software model, FS is the superior mode of production that is better in tune with the contemporary forces of production.

Since FS is a nascent form of production realized by the combined yet unequal productive activities of FS developers and users who are fluid subjects, I have argued that FS developers are best seen as forerunners of a class in the making. To the extent that the knowledge producers orient towards FS-style digital commons production, they can contribute to a historic movement towards a classless future as organic intellectuals. The current stage of development of this proto-class is expressed in the search for systematized remuneration mechanisms that are not based on a sale. This would allow FS producers to stand as an independent class that produces commons for the public. A further forward step could be taken if certain contingencies are met, such as the implementation of universal basic income which would increase the economic security of the independent producers, and expansion of higher education to universalize mental labor, which would swell the ranks of the class in the making. Naturally, this requires the articulation of a political movement that builds on the already impressive spontaneous achievements.

In the interim, the FS mode of production continues to interface with the existing system and its economic and political power structures. While the GPL plays an important role in shaping the grounds of interaction in favor of the commons, actual outcomes are still highly dependent on conjuncture, actor choices, and political forces that are beyond the direct influence of the FS communities. My fieldwork shows both the convergences of diverse interests which jointly engage in FS on the level of activity and discourse, and the temporally insecure nature of their collaboration.

The Pardus experience leads us to several conclusions. The first is that with the

mobilization of the right kind of interests, such as the computing security and cost-saving motivations of a nation-state, and an open-minded leadership with a vision, even a small FS community of volunteers and workers can achieve much greater impact than they could exert on their own. This is revealed by the fact that to the extent that Pardus was allowed to develop, it was successful both as a product and as a cultivation of domestic software expertise. However, this very success comes at the price of a surrendering of a portion of their autonomy to an external management structure, which went hand in hand with the most dangerous of compromises: damage to the bonds of community among FS producers, volunteer and paid. The change of TÜBİTAK management and furthermore the effect of the FATİH project as a force majeur influencing the fate of Pardus showed both the vulnerability of the Pardus imitative and how big of a threat it posed to the system at the nexus of national political interests and global capitalist forces such as Microsoft.

The key takeaway from the Pardus experience as revealed by the interviews is that the FS-producer subjects, while indignant and frustrated, have not been disillusioned by FS per se. They continue to contribute to the global commons and look for opportunities to reconcile their personal needs with producing FS. While a shadow of its former glory, even a fork of the original Pardus project lives on today as a volunteer developed distro called Pisi Linux. The transformative potential of Pardus for the Turkish software sector has been foreclosed, but the accumulated labor is not lost. It lives on in projects like Pisi, as well as in the upstream contributions of the Pardus team to Linux and the FS commons.

More importantly, potential lives on as a cultural force in the experience of Pardus producers and users. Almost all of the subjects continue to be proud of what they accomplished and have experienced the satisfaction of a form of work that is partly an activity of self-realization. This is reflected in their reactions to losing their jobs, which were quite

different from the reaction of the average alienated worker; the disappointment was focused on the lost opportunity to make Pardus a vehicle for progressive social change through exclusive selection for the FATİH project.

Some have reached a new reconciliation in their consciousness between doing something for their country and doing something for the world public without compromising on either, which can be seen as an example of counter-globalization through an experience of being part of a global phenomenon while carrying up the local priorities. This cultural force will most likely find new expression in unforeseeable future avenues.

The FS community in Turkey has demonstrated through its Free Software Days and Hacker Space events that it has the character of a social movement. The social movement aspect of FS sheds light on what could have been, or what could be, if the FS dynamic is allowed to develop on its own terms, shaping the parts of society that it comes into contact with in its own image, instead of the other way around. Such discrepancy is what gives birth to social movements who seek societal reform. The FS as a cause rubs shoulders with other progressive causes, from freedom of expression to IT workers' rights. It has also shown an attitude that is dynamic enough to embrace protest, both towards external forces deemed hostile, and internally towards practices deemed inappropriate. However modest, glimpses of a political awakening in larger society can also now be seen; for example in the Peoples' Democracy Party's 2015 electoral program that included the bullet point "Free open source software will be given incentives" under the section "Freedom of Press, Communication and Informatics".

The critics of IP are numerous and are raising their voices against the prevailing regime from many directions. Communizers and liberal marginalists, hackers and crypto-anarchists, neo-mutualists and common-ers are converging in their critique, and are beginning

to debate programs for alternative futures. This is happening at the same time as the importance of digital artifact production grows in the world economy in unequal and unjust fashion. The FS mode of production is not only already undermining the old regime by competing with it, it is providing vital ammunition to the critics by allowing them to point at an already functioning, superior alternative. The point is not simply that the FS mode of production is eating away at informational capitalism, but that we now know what to put in place to continue production in the eventuality that one day the IP regime is abolished wholesale politically. John Maynard Keynes had famously said: "The decadent international but individualistic capitalism, in the hands of which we found ourselves after the war, is not a success. It is not intelligent, it is not beautiful, it is not just, it is not virtuous and it doesn't deliver the goods. In short, we dislike it, and we are beginning to despise it. But when we wonder what to put in its place, we are extremely perplexed."

I am confident that Keynes's famous "perplexation" in the face of what to replace a despised capitalism with, is giving way to fresh utopias.

Appendix: The Global Market Size of Software and Intellectual Property

In this Appendix I present some data first about the global market size of software including licenses and services, second about "intangible assets" in the USA, and finally about the total IP in the world. These data are intended to give a sense of the large scale significance and the potential of FS as an alternative mode of production.

According to CIA World Factbook figures, the world GDP in 2013 was 74 Trillion USD.[161] In the same year, American information technology research and advisory firm Gartner estimated the size of the global software licenses market to be 407 Billion USD.[162] This suggests that software sales accounted for about 0.5% of world GDP in 2013. This is the traditional, packaged software for which users buy licenses and is the category of software that FS is positioned to directly replace as the price-free, collaboratively produced open alternative.

The software industries constitute only one piece of the total value of IP in the world economy. A much bigger and decisive question is how much the entirety of IP including all copyrights, patents, design rights, etc. amount to. If FS demonstrates that IP in software can be transcended by a superior mode of production, it is plausible that similar mechanisms can do the same for other IP segments. Unfortunately, precisely figuring out this total size of IP in the world economy is an unsolved problem. The reason for this is that outside of pure knowledge goods such as software, IP mostly gets embedded in material products rather than being traded as standalone items on the market, and therefore its accounting is veiled behind the opaque internal operations of corporations. According to a report on IP published by the World Intellectual Property Organization (WIPO) in 2003, "one reason the new economy has

161 http://www.indexmundi.com/world/economy_profile.html
162 http://www.gartner.com/newsroom/id/2696317

been called invisible is that old accounting methods have trouble monitoring it...Experts have not found a robust method that could fully satisfy firms in different sectors of industry and this may also be another reason which prevents many firms from systematically assessing the value of IP" (Idris, 2003: 58-59).

Indeed, the literature on the size of world IP at the moment only provides a mixture of murky statistics on certain "intangible assets" combined with qualitative assessments. An example of the confusing nature of the category of "intangible assets" is in the following:

> In the United States of America, the Omnibus Budget Reconciliation Act of 1993 introduced the definition of several classes of intangible assets (for example, goodwill, going concern value, lists of customers, patents, copyrights, formulas, processes, designs, patterns, know-how, and licenses) and allowed firms to amortize the cost of such assets. (Ibid: 58)

Patents, copyrights, designs and patterns qualify as being the commodity form of real wealth; whereas goodwill and going concern value are entirely a matter of market pricing of a company which involves an assessment of the value of all assets, IP and otherwise, as well as expected future performance which is speculative. List of customers and know-how are not matters of value but of the circulation of value and the qualifications of the labor force, both of which constitute some of the preconditions of production. The status of formulas is likely a non-value commons being used as a factor of production, as formulas are generally not patentable. The item "licenses" probably refers to rent payments by the company on the IP of another proprietor. The lumping together of these diverse set of items is a frustrating way of accounting for researchers interested in finding out definite sums of value as expressed by prices.

Shapiro and Hassett provide a degree of clarification to the intangible assets comprising the 70% of US corporate assets. It turns out that roughly half of this cocktail of intangibles comes down to actual IP:

> Using the ratio of market value to book value as a guide, and assuming for simplicity that all forms of intangible investment have the same productivity and depreciation rate, and that the patterns in the latest data reflect historical patterns, we find the value of U.S. firms' intellectual capital comprises 47 percent of their intangible value. (Shapiro and Hassett, 2005: 13).

They are then able to reach the following estimate on the size of USA's stock of IP as of 2005:

> Since intangibles are 70 percent of market value, it follows that the intellectual capital of U.S. firms is equal to roughly 0.33 [0.47*0.70] of their market value. According to Bloomberg, the market value of all U.S. equities on September 6, 2005 was $15.2 trillion. Therefore, the value of intellectual capital on that date would be roughly $5 trillion. (Ibid)

> The current value of the intellectual property which embodies those ideas -from computer software and musical recordings to patented pharmaceuticals and information technologies- is enormous. We estimate that U.S. intellectual property today is... equivalent to about 45 percent of U.S. GDP and greater than the GDP of any other nation in the world. (Ibid: 2)

GLOSSARY

Cloud computing. A term that refers to services provided over a network by a collection of remote servers.

Copyleft. A form of licensing which enables people the right to freely distribute copies and modified versions of a work on the condition that the same rights be preserved in derivative works. It thus subverts the copyright law to serve ends opposite to its general spirit.

Copyfarleft. A proposal for a license that would discriminate according to the category of user. Under a copyfarleft license, a worker-owned cooperative would be allowed free use of a peer production license item, but capitalistically organized corporations would be prevented from having free access.

Creative Commons License. A public copyright license that enables the free distribution of an otherwise copyrighted work.

Derivative Work. In copyright law, a derivative work is an expressive creation that includes major copyright-protected elements of an original, previously created first work.

Distro. Short for distribution. A term used to describe a specific software package that uses the Linux operating system.

Domain Name Server. A hierarchical distributed naming system for computers, services, or any resource connected to the Internet or a private network.

Forking. A dissenting group of producers' removal from the collective production effort their own future labor in favor of developing the software on a different path.

Foundation for Peer to Peer Alternatives (P2P Foundation). An international organization focused on studying, researching, documenting and promoting peer to peer practices in a very broad sense.

Free Software (FS). Computer software whose source code is made open and freely available under an appropriate, freedom-guaranteeing legal license. This allows anyone to develop the software by improving existing components and deriving new software, in addition to freely using it. FS thus constitutes an open commons; it is non-proprietary, in other words, held in common by all.

Free and Open Source Software. See Free Software.

Free Software Foundation (FSF). A non-profit foundation that aims to promote the production and use of free software and the General Public License

General Public License (GPL). The most popular contemporary FS license tied to FS. It not

only grants all of the relevant freedoms to users and developers, but also precludes the use of GPL code in non-GPL projects in the future.

GNU. Shorthand for "GNU's Not Unix". It is a family of multitasking, multiuser computer operating systems that use free software. It is developed as an alternative to Unix whose software is copyrighted.

GNU/Linux Operating System. A free software based alternative operating system for computers. The Linux operating system is licensed under the GNU GPL and uses most of the GNU programs as well as the Linux Kernel.

Linux Kernel. The **c**entral module of Linux operating system. It is the part of the operating system that loads first, and it remains in main memory.

Linux Operating System. See GNU/Linux Operating System.

MySQL. An open source relational database management system.

Open Source Software. Software whose source code is freely available for modification or enhancement by anyone.

Peer Production (PP). The generalization of the FS mode of production to spheres beyond software production.

Peer Production Licence (PPL). An example of the copyfarleft type of license.

Permissive Licences. A class of free and open source software licenses with minimal requirements about how the software can be redistributed.

Proprietary Software. Closed-source software. Any computer software that has restrictions on any combination of the usage, modification, copying or distributing modified versions of the software.

Project Oekonux. A school of unorthodox radical thinkers who believe that FS is a new mode of production that has the potential to revolutionize society and seek to analyze it on its own terms rather than with reference to prior modes of production and theories that sought to explain them. The word "Oekonux" is a compression of the words "Oekonomie" and "linux".

Software as a Service (SaaS). Software that is provided as a service instead of a product.

Source Code. Program instructions in original form, written using some human-readable computer language.

Tor. A free software for enabling online anonymity.

Virtual Private Network (**VPN**). A network that is constructed by using public wires, such as the Internet, to connect to a private network.

BIBLIOGRAPHY

Appadurai, Arjun (1996). "The Production of Locality". *Modernity at Large: Cultural Dimensions of Globalization. E*d. Arjun Appadurai. Minneapolis: University of Minnesota Press, 178-200.

Ashurst, Mark (2004). "Brazil falls in love with Linux". BBC News, February 1. http://news.bbc.co.uk/2/hi/business/3445805.stm.

BAKA (2012). *Yazılım Sektörü Raporu* (West Mediterranean Development Agency of Turkey. Report on the Software Sector). Ankara.

Benson, Todd. (2005). "Brazil: Free Software's Biggest and Best Friend". *The New York Times*, March 29.

Barbrook, Richard and Andy Cameron (1996). "Californian Ideology". *Science as Culture* 6 (1): 44-72.

Barbrook, Richard (2000). "Cyber-Communism: How the Americans are Superseding Capitalism in Cyber Space". *Science as Culture*. **9** (1): 5-40.

Barbrook, Richard (2006). *The Class of the New.* Open Mute.

Bauwens, Michel and Vasilis Kostakis (2014). "From the Communism of Capital to Capital for the
Commons: Towards an Open Co-operativism." *tripleC* 12 (1): 356-361.

Bauwens, Michel (2005). "The Political Economy of Peer Production". *100 Days of Theory.* www.ctheory.net/articles.aspx?id=499.

Bauwens, Michel (2007). "The Peer to Peer Manifesto: The Emergence of P2P Civilization and Political Economy".
http://www.masternewmedia.org/news/2007/11/03/the_peer_to_peer_manifesto.htm.

Bauwens, Michel (2012). "From the Theory of Peer Production to the Production of peer Production Theory. *Journal of Peer Production*, Issue #1.
http://peerproduction.net/issues/issue-1/invited-comments/from-the-theory-of-peer-production-to-the-production-of-peer-production-theory/

Benkler, Yochai (2006). *The Wealth of Networks: How Social Production Transforms Markets and Freedom.* New Haven: Yale University Press.

Benjamin, Walter (1998). "The Author as Producer" in *Understanding Brecht*. London: Verso, 85-103.

Boyer, Dominic (forthcoming). "Reflexivity Reloaded: from anthropology of intellectuals to critique of method to studying sideways". *Essays in Honor of Ulf Hannerz.* Ed. C. Garsten. New York: Berghahn.

Boyle, James (2003). "The Second Enclosure Movement and the Construction of the Public Domain". *Law and Contemporary Problems* 66: 33-74.

Boyle, James (2008). *The Public Domain: Enclosing the Commons of the Mind.* New Haven, Conn.: Yale University Press.

Canlı, Mustafa (2012). "Yazılım İhracatı ve Fırsatlar" (Software Export and Opportunities). *Türkiye'de Yazılım Sektörü.* Ed. Murad Tiryakioğlu. Stratejik Düşünce Enstitüsü. Ankara.

Carson, Kevin (2008). *Organization Theory.* BookSurge Publishing.

Carson, Kevin (2010). *The Homebrew Industrial Revolution*. BookSurge Publishing.

Castells, Manuel (1996). *The Rise of the Network Society. The Information Age: Economy, Society and Culture.* Vol. I. Oxford: Blackwell Publishers.

Castells, Manuel (1997). *The Power of Identity. The Information Age: Economy, Society and Culture.* Vol. II. Oxford: Blackwell Publishers.

Castells, Manuel (1998). *The End of Millennium. The Information Age: Economy, Society and Culture.* Vol. III. Oxford: Blackwell Publishers.

Coleman, Gabrilla (2010). "The Hacker Conference: A Ritual Condensation and Celebration of a Lifeworld". *Anthropological Quarterly* 83 (1): 47-92.

Corbet, Jonathan, Greg Kroah-Hartman, and Amanda McPherson (2015). "Linux Kernel Development: How Fast is it Going, Who is Doing it, What ate They Doing and Who is Sponsoring the Work". Linux Foundation Development Publication. http://www.linuxfoundation.org/sites/main/files/publications/whowriteslinux.pdf

Dobusch, Leonhard (2008). "Migration Discourse Structures: Escaping Microsoft's Desktop Path," *Open Source Development, Communities and Quality.* Eds. B. Russo, E. Damiani, et. al. Series: IFIP Internationl Federation for Information Processing vol. 275, 223-235.

Druey, Jean (2004). "Information Cannot be Owned." Research Publication No. 2004-05 4/2004. The Berkman Center for Internet & Society Research Publication Series: http://cyber.law.harvard.edu/publications.

Gates, Bill. "An Open Letter to Hobbyists" http://www.digibarn.com/collections/newsletters/homebrew/V2_01/gatesletter.html

Gibson-Graham, J. K. (2006). *A Postcapitalist Politics.* Minneapolis: University of Minnesota Press.

Gorz, Andre (1968). *Strategy for Labor: A Radical Proposal*. Boston: Beacon Press.

Hannerz, Ulf. (1998). "Other Transnationals: Perspectives Gained from Studying Sideways." *Paideuma* 44: 109-123.

Hardin, Garrett (1968). "The Tragedy of the Commons". *Science* 162: 1243-1248.

Hardt, Michael and Antonio Negri (2000). *Empire*. Cambridge, Mass.: Harvard University Press.

Hardt, Michael and Antonio Negri (2004). *Multitude*. Cambridge, Mass.: Penguin Press.

Hardt, Michael and Antonio Negri (2009). *Commonwealth*. Cambridge, Mass.: Harvard University Press.

Idris, Kamil (2003). *Intellectual Property: A Power Tool for Economic Growth.* World Intellectual Property Organization.

Kreisler, Harry. "Conversation with Steven Weber". http://globetrotter.berkeley.edu/people3/Weber/weber-con3.html

Lakhani, Karim R. and Robert Wolf (2005). "Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects". *Perspectives on Free and Open Source Software.* Eds. Joseph Feller, Brian Fitzgerald, Scott Hissam and Karim Lakhani. Cambridge: MIT Press, 3-22.

Leach, James (2009), "Freedom Imagined: Morality and Aesthetics in Open Source Software Design", *Ethnos* 74: 51-71.

Lessig, Lawrence (2004), *Free Culture: How Big Media Uses Technology and the Law to Lock Down Culture and Control Creativity*. New York: Penguin.

Marcus, George (1995). "Ethnography in/of the World System: The emergence of Multi-Sited Ethnography." *Annual Review of Anthropology*, 24: 95-117.

Marcus, George (2007). "Ethnography Two Decades After *Writing Culture*: From the Experimental to the Baroque". *Anthropological Quarterly*, 80: 1127-1145.

Marx, Karl (1845/1998). *The German Ideology*. Amherst: Prometheus Books.

Marx, Karl (1847/1992). *The Poverty of Philosophy*. Moscow: International Publishers.

Marx, Karl (1867/1990). *Capital* Vol 1. London: Penguin Classics.

Maxigas (2012). "Hacklabs and hackerspaces – tracing two genealogies." *Journal of Peer Production*, July Issue #2. http://peerproduction.net/issues/issue-2/.

McPherson, Amanda, et al. (2008). *Estimating The Total Development Cost of a Linux*

Distribution.
http://www.linux.com/learn/whitepapers/doc/4/raw.

Meretz, Stefan (2012). "Ten Patterns Developed by the Oekonux project". *Journal of Peer Production*, Issue #4. http://peerproduction.net/issues/issue-1/debate-societal-transformation/ten-patterns-developed-by-the-oekonux-project/

Meretz, Stefan (2014). Socialist Licenses? A Rejoinder to Michel Bauwens and Vasilis Kostakis. *tripleC* 12 (1): 362-365.

Moglen, Eben (1999). "Anarchism Triumphant: Free Software and the Death of Copyright" *First Monday*, 4 (8).
http://firstmonday.org/htbin/cgiwrap/bin/ojs/index.php/fm/article/view/684/594.

Moglen, Eben (2003). "The dotCommunist Manifesto"
http://emoglen.law.columbia.edu/my_pubs/dcm.html

Moglen, Eben (2004). "'Die Gedanken Sind Frei': Free Software and the Struggle for Free Thought. http://emoglen.law.columbia.edu/publications/berlin-keynote.html

Moody, Glyn (2010). "Why No Billion-Dollar Open Source Companies?"
http://www.computerworlduk.com/toolbox/open-source/blogs/index.cfm?entryid=3010&blogid=14.

Nyman, Linus (2014). "Hackers on Forking". *Proceedings of Open Sym'14 The International Symposium on Open Collaboration*. ACM Digital Library. Doi 10.1145/2641580.2641590.

Orsi, Cosma (2009). Knowledge-based Society, Peer Production and the Common Good". *Capital & Class* 97: 31-51.

Parijs, Philippe (2004). "Basic Income". *Politics & Society* 32 (1): 7-39.

Perens, Bruce (2005). "The Emerging Economic Paradigm of Open Source,"
http://perens.com/works/articles/Economic.html.

Perens, Bruce (2009). "Is Open Source Capitalist or Communist?"
http://itmanagement.earthweb.com/osrc/article.php/3809221/Bruce-Perens-Is-Open-Source-Capitalist-or-Communist.htm.

Polanyi, Karl (2001). *The Great Transformation.* Boston: Beacon Press.

Raymond, Eric (2001). *The Cathedral And The Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary.* Sebastopol: O'Reilly and Associates, Inc.

Reding, V. (2007). "Truffle 100: Towards a European Software Strategy". European Commission on Information Society and Media.
http://ec.europa.eu/commission_barroso/reding/docs/speeches/brussels_20071119.pdf

Rigi, Jacob (2012). "Peer to Peer Production as the Alternative to Capitalism: A New Communist Horizon." *Journal of Peer Production*, July Issue #1. http://peerproduction.net/issues/issue-1/

Rigi, Jacob (2013). "Peer Production and Marxian Communism: Contours of a New Emerging Mode of Production". *Capital & Class* 37 (3): 397-416.

Rigi, Jacob (2014). "The Coming Revolution of Peer Production and Revolutionary Cooperatives. A Response to Michel Bauwens, Vasilis Kostakis and Stefan Meretz". *tripleC* 12 (1): 390-404.

Robles, Gregorio and Jesús M. González-Barahona (2012). "A Comprehensive Study of Software Forks: Dates, Reasons and Outcomes". *Open Source Systems: Long-Term Sustainability.* Eds. Imed Hammouda, Bjourn Lundell, Tommi Mikkonen, and Walt Scacchi. Heidelberg: Springer, 1-14.

Rose, Carol (1986). The Comedy of the Commons: Commerce, Custom, Inherently Public Property. *The University of Chicago Law Review* 53: 711-781.

Salman, Banu and Kahraman Yapıcı (2009). "Kamu Pardus'a Göçemiyor". *Elektrik Mühendisliği Dergisi*. 435: 19-28.

Samuelson, Pamela (1999). "Copyright, Commodification, Censorship". http://tc.eserver.org/out/27135/http://www.sims.berkeley.edu/~pam/papers/haifa_priv_cens.pdf

Scweik, Charles M. and Robert English (2013). "Preliminary Steps toward a General Theory of Internet-based Collective Action in Digital Information Commons: Findings from a Study of Open Source Software Projects". *International Journal of the Commons* 7 (2): 234-54.

Shapiro, Robert and Kevin Hassett (2005). *The Economic Value of Intellectual Property.* Sonecon and American Enterprise Institute.

Siefkes, Christian (2012). "Beyond Digital Plenty: Building Blocks for Physical Peer Production". *Journal of Peer Production* Issue #1. http://peerproduction.net/issues/issue-1/

Söderberg, Johan (2002). "Copyleft vs. Copyright". *First Monday* 7 (3-4): 1-29. http://firstmonday.org/article/view/938/860.

Söderberg, Johan (2008). *Hacking Capitalism: The Free and Open Source Software*. New York: Routledge.

Stallman, Richard (1985). "The GNU Manifesto", *Dr. Dobb's Journal* 10 (3): 30.

Stallman, Richard (2010). *Free Software, Free Society.* 2nd ed. Boston: GNU Press.

Suber, Peter (2012). *Open Access*. Cambridge: MIT Press.

TBD (2012). *Değerlendirme Raporu* (Informatics Association of Turkey. Evaluation Report. Ankara.

Tilly, Charles (2004). *Social Movements, 1768-2004*. New York: Routledge.

Unlimited Potential: US Chamber of Commerce Global Intellectual Center Report (2015). 3rd Ed.

Vieira, Miguel and Primavera De Filippi (2014). "Between Copyleft and Copyfarleft: Advanced Reciprocity for the Commons". *Journal of Peer Production*, January Issue #4. http://peerproduction.net/issues/issue-4.

Von Hippel, F. (2005). "Democratizing Innovation". *Perspectives on Free and Open Source Software.* Eds. Joseph Feller, Brian Fitzgerald, Scott Hissam and Karim Lakhani. Cambridge: MIT Press, 267-78.

Wallerstein, Immanuel (1999). *The End of the World as We Know It.* Minneapolis: Minneapolis University Press.

Weber, Steven (2004). *The Success of Open Source.* Cambridge: University of Harvard Press.

Williams, Sam (2002). *Free as in Freedom.* Boston: The Free Software Foundation.

Williams, Sam (2012). *Free as in Freedom* 2nd ed. revisions by Richard M. Stallman. Boston: The Free Software Foundation.