

FREE AND OPEN SOURCE SOFTWARE IN TURKEY

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF SOCIAL SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

İBRAHİM İZLEM GÖZÜKELEŞ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
SCIENCE AND TECHNOLOGY POLICY STUDIES

APRIL 2006

Approval of the Graduate School of Social Sciences.

Prof. Dr. Sencer Ayata
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Erkan Erdil
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Belkız
Tarhan
Supervisor

Examining Committee Members

Assoc. Prof. Dr. Erkan Erdil (METU) _____

Assist. Prof. Dr. Belkız Tarhan (BASKENT) _____

Dr. Barış Çakmur (METU) _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: İbrahim İzlem Gözükeleş

Signature :

ABSTRACT

FREE AND OPEN SOURCE SOFTWARE IN TURKEY

Gözükeleş, İbrahim İzlem

M.S., Department of Science and Technology Policy Studies

Supervisor : Assist. Prof. Dr. Belkız Tarhan

April 2006, 159 pages

In recent years, there has been growing interest of governments and firms for Free and Open Source Software (FOSS). However, FOSS is still a puzzlement for a wide spectrum of academic disciplines. Social scientists ask why FOSS hackers participate in FOSS if they do not get any monetary rewards. On the other hand firms ask, how a firm can earn money from FOSS. In this study, motivations of FOSS hackers and firms are taken as an interrelated phenomenon. For FOSS hackers, social and technical aspects of FOSS are essential. On the contrary to FOSS hackers, firms have profit oriented goals. In this context, conflicting and coinciding interests of FOSS hackers and firms are investigated through development of FOSS in Turkey. Software industry in Turkey is not developed enough and FOSS may play a prominent role in developing its own software industry. This study discusses FOSS both from the view of hackers and the firms in Turkey. In that respect, social conditions that make hackers voluntarily contribute, discrepancies between attitudes and behaviors of the firms and FOSS licenses are explored. According to these facts, possible FOSS business strategies which may be adopted in Turkey are presented.

Keywords: Free Software, Open Source Software, Hackers, Business Models

ÖZ

TÜRKİYE'DE ÖZGÜR YAZILIM VE AÇIK KAYNAK KOD

Gözükeleş, İbrahim İzlem

Yüksek Lisans, Bilim ve Teknoloji Politikası Çalışmaları Bölümü

Tez Yöneticisi : Yrd. Doç. Dr. Belkız Tarhan

Nisan 2006, 159 sayfa

Son yıllarda, hükümetlerin ve firmaların Özgür ve Açık Kaynak Kodlu Yazılıma (ÖAKKY) ilgileri giderek artmaktadır. Fakat, ÖAKKY bir çok akademik disiplin için hala tam olarak anlaşılamayan özellikler taşımaktadır. Sosyal bilimciler, "hacker"ların, hiçbir parasal karşılığı olmadan neden katkı sunduklarını sormaktadır. Firmalar ise ÖAKKY'den nasıl para kazanılacağını sormaktadır. Bu çalışmada, "hacker"ların ve firmaların motivasyonları ilişkisel bir olgu olarak ele alınmaktadır. "Hacker"lar için ÖAKKY'nin toplumsal ve teknik özellikleri önemliken firmalar kar elde etme güdüsüyle hareket etmektedir. Bu bağlamda, "hacker"ların ve firmaların birbirleriyle çelişen ve örtüşen çıkarları, ÖAKKYnin Türkiye'deki gelişimi doğrultusunda soruşturulmaktadır. Türk yazılım endüstrisi yeterince gelişmemiştir ve ÖAKKY Türkiye'nin kendi yazılım endüstrisini geliştirmesinde önemli bir rol oynayabilir. Bu çalışmada, ÖYAKK, Türkiye'deki "hacker"ların ve firmaların bakış açılarından tartışılmaktadır. Bu amaçla, "hacker"ların gönüllü katkıları için gerekli toplumsal koşullar, firmaların eğilimleri ve davranışları arasındaki farklar ve ÖAKKY lisansları araştırılmaktadır. Bu olgular doğrultusunda, Türkiye'de uygulanabilecek olası ÖAKKY iş stratejileri sunulmaktadır.

Anahtar Kelimeler: Özgür Yazılım, Açık Kaynak Kod, Hacker'lar, İş Modelleri

To the Hackers

ACKNOWLEDGMENTS

I express sincere appreciation to Assist. Prof. Dr. Belkız Tarhan for her guidance and insight throughout the study. I could not complete the study without her help.

I would like to thank to my committee members, Assoc. Prof. Dr. Erkan Erdil and Dr. Barış Çakmur for their critical contributions. Furthermore, I am very indebted to the staff of Science and Technology Policy Studies. I have learned a lot from them.

Many software engineers and hackers contributed to the study. I would like to thank to them for their suggestions, valuable comments and sincerity. I also thank to Emre Uğur for his help about Latex.

Additionally, I would like to express my special thanks to Cahit Cesur for providing the GNU/Linux platform at work.

I decided to give up the study a few times. Thanks to my parents' encouragement to continue.

Lastly, I want to thank to my brother and wife for their endurance during the study.

TABLE OF CONTENTS

PLAGIARISM	iii
ABSTRACT	iv
ÖZ	v
DEDICATON	vii
ACKNOWLEDGMENTS.....	viii
TABLE OF CONTENTS.....	ix
LIST OF TABLES.....	xii
LIST OF FIGURES.....	xiii
CHAPTER	
1 Introduction	1
2 What is Free and Open Source Software?	9
2.1 A Brief Inquiry into the Literature on FOSS.....	9
2.1.1 Social Movement and Community.....	9
2.1.2 Gift Economy	14
2.2 FOSS in the Informational Economy	18
2.2.1 FOSS as a Social Relation.....	18
2.2.2 Hackers	23
2.2.3 Firms	25
2.2.4 FOSS Licenses	30
2.3 FOSS In The World.....	33
2.3.1 Unconscious FOSS	33
2.3.2 Hackers and Free Software Movement	38
2.3.3 Internet and Linux	42
2.3.4 New Economy and Open Source Initiative	45
2.3.5 Governments and Intellectual Property Rights.....	52

3	General Outlook of FOSS In Turkey	58
3.1	Research Questions and Methods	58
3.1.1	Questions	58
3.1.2	Methods	59
3.2	Main Actors and Organizations	63
3.2.1	LKD.....	64
3.2.2	Debian Turkey.....	67
3.2.3	EnderUNIX	69
3.2.4	Other Organizations	70
3.3	Works, Festivals and Activities.....	70
3.3.1	Works	70
3.3.1.1	Localization	70
3.3.1.2	Local GNU/Linux Distributions.....	72
3.3.1.3	Local FOSS Projects	75
3.3.2	Festivals and Activities.....	76
3.3.2.1	Free Software and Open Source Days.....	76
3.3.2.2	Free Software and Linux Festival.....	77
3.3.2.3	METU Computer Club Programming Con- test.....	77
4	Hackers in Turkey.....	79
4.1	Hackers	79
4.2	Hackers, Licenses and Commerce in Turkey.....	92
4.3	Software Engineers	98
4.4	Main Findings of The Research About Hackers.....	100
5	FOSS Business in Turkey	109
5.1	FOSS Firms.....	109
5.2	Firms and Licenses in Turkey	123
5.3	Uludağ Project	130
5.4	Main Findings of the Research About FOSS Firms	133
6	Conclusion	138
	REFERENCES.....	144
	APPENDICES	151
A	Glossary	151

B	Research.....	155
B.1	Hackers	155
B.2	Firms	156
B.3	Software Engineers	158
B.4	Activities	159

LIST OF TABLES

TABLES

4.1 Motivations of FOSS Developers in Europe	106
4.2 Motivations of FOSS Hackers in Turkey	106
5.1 Categorization of the FOSS firms in the study.....	114

LIST OF FIGURES

FIGURES

2.1 Machine Code.....	35
2.2 Programming Language Code.....	35

CHAPTER 1

Introduction

Governments all over the world are promoting Free/Open Source Software (FOSS¹) as a matter of policy. European Parliament (EP) called on member nations to use FOSS whenever it is practical (Hahn, 2002). Germany and France are the most active countries in Europe which had a number of pilot projects for the introduction and use of FOSS (Ghosh et al., 2002). Brazil considers FOSS as a strategic decision for a developing country (Technology, 2003). Among these countries, China is the most interesting one with its population which can effect the future of software industry. Chinese Government placed official support behind the domestic software industry using FOSS (Xiaonan, 2003).

Although these countries have different socio-economic backgrounds, they agree on FOSS, but with different priorities. For example, while cost reduction and independence from proprietary software vendors are decisive factors for German and French governments, developing countries give importance to developing their own software industry. On the other hand, China is mostly concerned with national security. Localization for countries, where English is not the common language, is also playing prominent role for promotion of FOSS. By localization, software products, developed for a specific nation and culture², are adapted for other nations and cultures. These benefits (cost reduction, security, localization etc.) are natural consequence of source code's availability in FOSS (Wong, 2003).

¹ FOSS is used as a general term for both Free Software and Open Source Software, except where a specific distinction between the terms is explicitly made.

² Localization is not limited to translation from one language into another. For example, in localization for Turkey, one must also consider different usage of percent sign: while it is 75% in US, it is %75 in Turkey.

Availability of source code in FOSS is defined with its three essential features³.

- 1- Source code must be distributed with the software or otherwise made available for no more than the cost of distribution.
- 2- Anyone may redistribute the software for free, without royalties or licensing fees to the author.
- 3- Anyone may modify the software or derive other software from it, and then distribute the modified software under the same terms (Weber, 2004: 5).

However, FOSS is still a puzzlement for a wide spectrum of academic disciplines: software engineering, industrial engineering, economics, sociology, psychology, etc. Each academic discipline asks its own questions to understand the FOSS phenomenon and try to explain it in the boundaries of its own discipline. For example, recent literature focuses on two questions and seeks their answers in different disciplines. The first question is, 'what are the motivations of individuals for engagement in FOSS projects?' (von Krogh et al., 2003; Bitzer et al., 2004; Raymond, 1999b, 2000; Zeitlyn, 2003) and the latter question is about FOSS's relationship with commercial world, 'in what level will FOSS have impact on the competitive strategy and organizations of firms?' (Baldwin and Clark, 2003; Lindman, 2004). Consequently, answer of the first question is mostly sought in the context of either social psychology or anthropology and latter is studied in the context of microeconomics. These studies give essential insights either about the motivations behind individuals or about FOSS business models. However, they grasp the FOSS by an approach that makes parts of the whole as static and independent of one another.

Making of FOSS has been an active process which owes to agency of its members and to conditioning of technological innovations and market since 1980s. Even though needs of the market have been an important factor in emergence and development of the FOSS, FOSS firms such as Cygnus Solutions and Red Hat showed that there was something as a FOSS business model. Yet, what's more surprising was to see the world's major companies such as IBM and Oracle turn their attention to this new model as a new business opportunity (DiBona et al., 1999). First of all, it was against the conventional intellectual property law. However, it was not a marginal phenomenon. Instead, it was a widespread revolutionary way of doing

³ Actually, there are differences between Free Software and Open Source Software. Yet, these are their common features.

and distributing software in the new environment, both technically and legally by inverting the logic of IPR (Intellectual Property Rights) (Valimaki, 2005). By natural, the questions about the hackers' motivation and firms' new business strategies were unavoidable.

For this reason, the widespread attitude in FOSS literature became investigating the answer of the motivation behind hackers' work and the viability of firms' new business models. Consequently, conceptualizations of the FOSS literature (e.g. FOSS as a social movement, FOSS Community and Gift economy) based on these attitudes. As discussed in the first part of the **second chapter**, although these studies give important insights about the hackers, they are ahistorical. In social movement and community theories, evidences in specific period of the history are generalized. Increasing involvement of firms and increasing commercialization of FOSS make these theories more questionable.

On the other hand, theory of the gift economy goes back to roots of the hacker culture, academia, to explain the hackers' motivation. This approach explicitly removes the paradoxes of hackers' motivation in FOSS projects. It states that hackers do not get direct monetary rewards, but gain reputation as a consequence of their contribution to FOSS projects (Raymond, 1999b, 2000; Kelty, 2004). Additionally, the gained reputation may transform into monetary rewards. Indeed, experiences of the hackers affirm the process of contribution-reputation-money. However, theory of the gift economy has the same problems with the conceptualizations of hackers as a social movement or a community: First, it does not consider firms' interest in FOSS and second it focuses on the hackers, but undervalues their relationship with their work, hacking. Additionally, the gift economy, implicitly, presumes that one prefers leisure to work. But more importantly, none of these theories conceptualize FOSS as a phenomenon in the informational capitalism, historically specific form of capitalism, "which information generation, processing, and transmission become the fundamental sources of productivity and power" (Castells, 2001b: 21).

Second part of the second chapter is devoted to analyzing FOSS from the perspective of hackers' relation with their work, firms' business strategies and the relationship between hackers and firms in the informational capitalism.

Despite the popular view in media, hackers are not criminals who break computers, web sites and phone systems. On the contrary, hackers build things instead of breaking them:

Hackers built the Internet. Hackers made the Unix operating system what it is today. Hackers run Usenet. Hackers make the World Wide Web work (Raymond, 2001).

In this context, hackerdom is defined with "technical adeptness and a delight in solving problems and overcoming limits" (Ibid.). On the other hand, for Raymond, someone is really a hacker when people appreciate one's work and start to call one as hacker. However, in this study, the term hacker is used in a wider context that represents the passionate workers of FOSS. Thus, a newbie who asks in a GNU⁴/Linux user mailing list:

Hi! I am new to GNU/Linux. What should I do to learn using it?
What do you advice?

is considered as a hacker as well as a master programmer whom Raymond calls as a hacker. Furthermore, between these two extreme points, there are also hackers who test software, report bugs, offer new features, contribute to localization and give seminars. Each adds value to FOSS by their activity without getting any direct monetary reward.

This does not mean that there are not any differences between them and their relation to FOSS. Indeed, there are essential qualitative differences. For example, for programmers, sometimes their activity is "just for fun" (Torvalds and Diamond, 2001) as Linus Torvalds⁵. On the other hand, hackers who translate software manuals from English to Turkish are more interested in the product of their activity, making FOSS for more people rather than the activity itself. Moreover, a hacker, who has been contributing to FOSS for a long time, has more tight relation with FOSS.

Furthermore, this study diverges from most of the studies in the FOSS literature, because of its two premises about hackers. First, rather than presuming one always escapes from work, it presumes that under certain social circumstances one may also

⁴ Recursive acronym for the phrase 'GNU is Not UNIX'.

⁵ He is the creator of Linux.

prefer work. Second, societies act and react to informational capitalism differently because of their cultural and historical diversity. For this reason, emergence and propagating of hackerdom is not the same in the all countries.

On the other hand, firms' cause of existence has always been making profit. Their relation with FOSS is established in this respect. However, one of the arguments of this study is FOSS business models are not counter to intuitive logic of the informational capitalism. On the contrary, it is a major part of the mainstream informational economy.

Business is a relationship between the firms and customers that is firms providing value for customers and getting money for this provision. In this context, selling proprietary software and FOSS business are different ways of this relationship. However, they consider software distinctively. In proprietary software business, source code of the software is crown-jewels of the firm. Its revenue mostly comes from selling right to use licenses of the software. Source code, that it has and its competitors do not have, makes it more advantageous in the market against its competitors. From stand point of the FOSS business models, software is continuously accumulated social knowledge. FOSS firms mostly earn money from selling complementary products (especially services) for FOSS, complementing their commercial products by FOSS or selling integration services for FOSS.

Furthermore, emergence of infoware, web sites as software applications, also triggered the rise of FOSS business. Demand for software increased and software became more complex. In this context, FOSS appeared as one of the solutions of the software industry against the complexity problem of software. As Weber writes, software was different from any informational good:

Complex software is not like a book, even the longest and most complex books ever written. It is more like a living organism that must continually adapt and adjust to the different environments and tasks that the world puts in front of it (Weber, 2004: 10).

However, the solution, FOSS, presumes the continuity of the hackers' work. For this reason, in the FOSS business, supporting the developers aiming at this continuity is essential. Most of the firms are aware that while making money from FOSS, they have to give back something to the hackers. On the other hand, possible con-

flicts of interests between individuals and firms are usually undervalued in the FOSS literature despite the studies that underline the key role of hackers' work in FOSS business.

Whereas hackers and firms have different sets of motivations that pull FOSS in the different directions. In one of the rare studies that discusses this conflict, Bonaccorsi and Rossi (2004) point out "altruistic individuals" and "selfish firms" from the perspective of relationship between their different sets of motivations. Bonaccorsi and Rossi (2004) conducted interviews with 146 firms in Italy that deals with firms' business model and their attitudes towards FOSS hackers. They compare behaviors and attitudes of the firms towards the hackers and show the discrepancy between them. Bonaccorsi and Rossi (2004) research mostly deals with if the firms really comply with the rules of the hackers.

Another important fact about the relationship between hackers and firms is the software licenses which is the objectification of their relationship in law. Each of the sides establish their relation with FOSS in terms of their interest. For the hackers, availability of the source code is essential, since it is necessary condition for the realization of their productive activity. In this context, GPL (General Public License), the most widespread FOSS license, is the most appropriate license that defines and defends the rights of the hackers. GPL does not merely define the common features of FOSS such as freely using, distributing and modifying the software, but it also guarantees the permanence of these features. By the reciprocity rule, hackers are obliged to distribute the software with the same license whether they modify the source code or not. In other words, it would be an illegal act to proprietarize the source code of a software application with GPL and sell it without the essential features of the FOSS.

On the contrary, BSD (Berkeley Software Distribution) style licenses do not have any obligation about the reciprocity. Anyone can make the source of a FOSS application private. For this reason, it seems attractive for firms. By a FOSS license without reciprocity obligation, a firm can benefit from the contribution of the hackers and by making its contributions private, it may gain advantage against its competitors in the market. However, as Fink (2003) emphasizes, communal development process of FOSS projects may be lost with BSD license. In that case, it results a discontinuity in

accumulation of social knowledge.

Last part of the second chapter is devoted to emergence, development and likely future of FOSS in the world. In this context, records and artifacts that shape or are produced by these processes are investigated in order to find tendencies of the FOSS. In this chapter, historical movement of FOSS in the world is studied through its five moments which have been conditioned by two contradicting processes contained in development of software: tendency to commodify and to be free. In this context, first moment of FOSS is named as **Unconscious FOSS** (Fogel, 2005: 12) that implies software which was free in itself before the emergence of commodification process of software named as proprietary software. Only after the reaction of hackers to commodification process, FOSS became conscious act of developers (**Hackers and Free Software Movement**). Decreasing number of computer prices, and increasing number of Internet users point out the third moment (**Internet and Linux**) in the history of FOSS. This changed the relationships within the FOSS world and integration of FOSS with the market started by different business strategies. In the second half of 1990s, those business strategies became more visible which resulted a shift from Free Software to Open Source Software(**New Economy and Open Source Initiative**). After 2000, some of those business enterprises failed, but governments entered in the FOSS world(**Governments and Intellectual Property Rights**); business strategies became national software policies. This also stimulated the international discussion of FOSS in the context of software patents.

In the first part of the **third chapter**, research questions and methods are presented. As stated in this part, the brittle relationship between the hackers and the firms in Turkey is the focus of this study. The research's main questions are 'To what extent does the conflicting interests of the hackers and the firms affect the development of FOSS business models in Turkey?' and 'what should a firm do in order to benefit from voluntary contribution of the hackers?'. In order to answer these questions, qualitative research methods, involving interviews and participant observation, were adopted.

In the second and thirds parts of the chapter, general outlook of Turkey is presented. As discussed in this part, FOSS followed a different path in Turkey be-

cause of Turkey's different socio-economic conditions. Data about the development of FOSS in Turkey is gathered from the mailing list archives of LKD (Linux Users' Association-Linux Kullanıcıları Derneği) and from the interviews with the past actors of the FOSS in Turkey.

Focus of the **fourth chapter** is the hackers in Turkey. This chapter begins with a general discussion of hackers in Turkey in the context of their relation with their work. In the second part, hackers' perception of FOSS licenses and their attitudes towards commercialization are investigated. Despite the high number of European software engineers participating in FOSS projects, there are only a few Turkish software engineers. In this respect, hackers and software engineers are compared in the third part. Main findings of the research about the hackers in Turkey are presented in the last part from different vantage points: types of work that hackers prefer, comparisons between *hacking and non-hacking activities*, *hackers and non-hackers*, *Turkish hackers and European hackers*.

In the **fifth chapter**, first, FOSS firms in Turkey and their business models are explored. In this context, viable business models in Turkey are stated and FOSS businessmen in Turkey are introduced. These businessmen are categorized according to their past experience and their activity at work. After this, relationship between the firms and the hackers are studied from two perspectives: Discrepancies between attitudes and behaviors of the firms and FOSS licenses. FOSS business depends on the voluntary contribution of the hackers. Although there was a governmental institution behind the Uludağ Project, the project gives essential hints about how a firm can make hackers participate in its project. In the last part of the chapter, there are main findings of the study about the firms.

Lastly, in the **conclusion chapter**, possible FOSS business models and alternative strategies of the FOSS firms in Turkey are discussed.

CHAPTER 2

What is Free and Open Source Software?

2.1 A Brief Inquiry into the Literature on FOSS

2.1.1 Social Movement and Community

A Microsoft memorandum, which was about GNU/Linux and FOSS, leaked to Eric Raymond in the last week of October 1998. A second leak followed this, which was particularly on GNU/Linux. Microsoft publicly acknowledged that these documents were authentic. However, they claimed that those were just an engineering study which did not tell anything about Microsoft's policy towards FOSS. Furthermore, it was speculated that these documents had been deliberately leaked in order to strengthen Microsoft's defense in the court. According to these speculations, it was just a tactic to claim that Microsoft was not a monopoly and had a serious competitor: GNU/Linux (Raymond, 2004). Actually, Microsoft's lawyers used GNU/Linux in their defense (Wayner, 2000).

Whether these documents were leaked intentionally as a tactic for the court or there was really a leak, Microsoft's policy towards FOSS in the last years reveals that they have been considering it as a serious threat for a long time. In this context, documents show the early signs of their combat strategy against FOSS.

It was obvious for Microsoft that their new rival in the market was not a commercial company as IBM, Apple or Netscape. In other words, there was not a battle between Windows NT and GNU/Linux or Microsoft and major FOSS companies, Red Hat/SuSe/Caldera. As implied by the writers of leaked documents, money was not often the primary motivation behind FOSS and understanding it required a deep

comprehension of the process and motivation of FOSS development teams. For this reason, writers offered that Microsoft should target a process rather than a company to compete against FOSS (Raymond, 2004).

In this aspect, it must not be surprising that Microsoft's strategy in the last years directed against the process and culture of FOSS instead of Red Hat or Apache. For example, Bill Gates condemn FOSS hackers supporting communism¹.

Microsoft's extraordinary response to FOSS might seem as absurd as General Motors were to complain about a collective of hippies producing free bicycles against the cars that pollute the world (Wayner, 2000). This announcement would sound strange. But Microsoft's relation with FOSS is similar:

A bunch of pseudo-communist nerds, living in their weird technoutopia where all the software was free, competing against the world's richest company. (Wayner, 2000: 5).

Furthermore, they are not producing bicycles against cars as in the example of General Motors. GNU/Linux is being taken seriously in some parts of the world. Many web servers are running on GNU/Linux or FreeBSD. A free web serving tool, Apache, is used by more than 50 percent of the web servers². Moreover, Google, the most popular search engine, is working on 100000 GNU/Linux servers (O'Reilly, 2004).

In this sense, FOSS is extraordinary and unprecedented. From the vantage point of competition, FOSS is a threat for Microsoft (Raymond, 2004) which is different from its old competitors. On the other hand, FOSS is also an interesting phenomenon for social scientists from the vantage point of relations of production which it contains, its work organizations and its culture. Indeed, what is problematic for Microsoft is also problematic for social scientists: Money is not the primary motivation behind FOSS.

Despite there is a huge literature about FOSS, most of the studies are about "individual projects or internal phenomena of the field" (Lehman, 2004) and FOSS as a social whole "has been largely overlooked or was treated only superficially" (Ibid).

¹ Retrieved December 5, 2005, from <http://news.com.com/Gates+taking+a+seat+in+your+den/2008-1041.3-5514121.html>

² Retrieved April 18, 2006, from http://news.netcraft.com/archives/2006/04/06/april_2006_web_server_survey.html

Additionally, many studies repeat words *social movement* and *community* to describe FOSS phenomenon, but they have clear and distinct meanings in social sciences which may not correspond to FOSS phenomenon. For Lehman (2004), a more holistic approach is needed for comprehension of FOSS which includes both analysis of the field as a whole and interactions in it.

In social sciences, definition of *social movement* is controversial. For Blumer (1969), there is a relation between the dissatisfaction of enterprise and the emergence of movement. In this aspect, a social movement aims to establish a new order of life. Eyerman and Jamison's (1991) emphasis is collective creation of ideas, identities and ideals. According to Tarrow (1998), there is also a confrontation with elites, authorities and opponents³. Social movements are important, because they

problematize the ways in which we live our lives, call for changes in our habits of thought, action and interpretation (Crossley, 2002:8).

In this sense, FSM (Free Software Movement) can be considered as a social movement since it problematize the ways in which people develop and use software. Its emergence and development is strongly related with the emergence of proprietary software and collapse of hacker community as its consequence. There is a collective creation of ideas, ideals and symbols (e.g. a manifesto⁴, a song⁵) There has always been a confrontation with proprietary software vendors, especially with Microsoft. FSM played an important role in the transformation of FOSS from its unconscious form to a conscious form, in other words to a goal oriented activity.

In European and the US literature, social movements are explained with different theories. In Europe, it is called as *new social movements* theory and based on social movements emerged after 1960s (e.g environmentalists, second wave feminists, the peace movement, gay rights movement, anti-globalization movement). This theory has essential role in the writings of post-marxists (Crossley, 2002). In FOSS literature, this theory is held mostly by The Oekonux Project. In the stated project, focus of

³ Crossley(2002) discusses the problems of these definitions in detail.

⁴ The GNU Manifesto, Retrieved December 5, 2005, from <http://www.gnu.org/gnu/manifesto.html>

⁵ Free Software Song, Retrieved December 5, 2005, from <http://www.gnu.org/music/free-software-song.html>

debate is "whether the principles of the development of free software may be the foundation of a new economy which may be the base for a new society."⁶. Indeed, at the very start of GNU project Stallman's goal was not merely better software:

The principal goal of GNU was to be Free Software. Even if GNU had no technical advantage over UNIX, it would have a social advantage, allowing users cooperate, and an ethical advantage, respecting the user's freedom (Stallman, 2002).

It is obvious that FSM has some implications for a new society and these suggestions are contradicting with the capitalism (Pestimalcioglu, 2003). On the other hand, FSM is a moment of FOSS and as it is discussed in the latter part, considering FOSS as a social movement has the possibility of underscoring its contemporary relations with the business world.

On the other hand, in the US literature, *resource mobilization theory* is widely used to explain social movements. In this theory, main concern is "balance of costs, rewards and incentives that provided agents with the motivation to become involved in the struggle" (Crossley, 2002: 12) For instance, Krogh et al. (2003), study on the rewards for individuals in FOSS projects. These rewards are reputation, control over technology and learning opportunities. Although this theory gives certain insights about the FOSS developers, it is criticized because of its base, rational actor model in "social movements" debate (Crossley, 2002). Additionally, its ahistorical approach does not help to understand the tendencies of FOSS. For example, it is presumed that people prefer leisure rather than working. However, as it is discussed in the second part of this chapter, man's attitude toward work may change under certain social circumstances.

Another frequently used term in FOSS literature is *community*. Tönnies (1958), who was one of the leading figures to study communities, used community (*Gemeinschaft*) to distinguish it from society (*Gesellschaft*). Community describes both similarity and difference. Members of community have something common within themselves and different from others (Cohen, 2000) and fulfill their needs by belonging to a community (McMillan and George, 1986). However,

The idea of community (in the sense of *Gemeinschaft* as described by Ferdinand Tönnies) emphasizes the non-rational sense of belonging among

⁶ <http://www.oekonux.org/>

human beings, which is not bound to any specific purpose. This bond between individuals is based on sympathy, adaption or familiarization and memory, depending on the exact nature of the relationship(Lehman, 2004).

Lehmann (2004) remarks that Tönnies's community theory focuses on organic groups (e.g. family) and "not fit FOSS developers well". According to Lehman (2004), on one hand, some of the basic assumptions of community theory (e.g. "close non-rational ties in a homogeneous, spatially delimited environment") does not fit to FOSS developers well. On the other hand, there are also community formation processes (e.g. "the writing of an endogenous history"). Studying FOSS developers as a virtual community is common in FOSS literature. Furthermore, some of these studies (e.g., Margret and Scacchi, 2003; Laurent, 2004) are related with the resource mobilization theory. Yet, the problems about change and tendencies of FOSS remain intact. Thus, hackers themselves also started to ask if the community still existed in FOSS:

The nature of the open source community is changing. I'm not exactly sure what "open source community" means anymore. When I first got involved with open source in 1998/99, the community was distinct: It was Eric Raymond, Bruce Perens, Robin Miller, and others like them. Developers. Gear heads. Hackers. Today, it's unclear whether that community still exists in any separate, discernible form.

Look at their former haunts (Apache and Linux mailing lists, among others) and you may still find them, but many have been hired by Novell, IBM, Hewlett-Packard, or some corporation, including a range of startups interested in cashing in on the open source gold rush. In addition to these hard-core developers, there is a significant, growing crowd of "newbies": casual developers, corporate developers (those who work for Morgan Stanley, Unilever, and the like), and a range of other, interested parties (Asay, 2005).

Both community and social movement based theories are too narrow abstractions to understand FOSS. First of all, they study only a short period of FOSS and generalize its results. Secondly, their analyses include only a few relations. For instance, most of the studies focus on a rational individual software developer. But for a long time, firms and governments are parts of the FOSS world. Young hackers transformed into venture capitalists in the second half of 1990s (Castells, 2001a) and additionally, the founder of the FSM, Stallman, accepts IBM as a part of the FOSS

world, although he does not approve IBM's IT policy entirely⁷. Moreover, China, EU, Third World have to be involved in FOSS. Thirdly, the conditions for existence of FOSS are part of what it is. In other words, the contemporary world is contained in FOSS. Yet, this fact is not fully discussed in the FOSS literature. For this reason, instead of merely asking 'What is FOSS?', additional question must also be asked 'In what kind of social and economic relations does FOSS exist?'

2.1.2 Gift Economy

For Hardin (1968), a tragedy exists between individuals' interests and common goods. The conflict between them arises since the individuals do not consider the entire cost of their actions when they use public goods. Hardin gives an example of pasture open to all herdsman. A herdsman, as rational being, would think to add one more animal to his herd. Adding an animal into the pasture would have two results. First, only one of the herdsmen would maximize his gain: the herdsman who added one more animal. Second, by adding one more animal, he consumed the pasture which was scarce. But this time negative effect of his behavior would be shared by the other herdsmen. So when he compared the positive and negative results, he would add one more animal. Furthermore, as rational beings, the other herdsmen would follow him which would result in destruction of the pasture and themselves.

In the article, Hardin proposes solutions for the end of tragedy. Although both the tragedy itself and Hardin's solutions are controversial, premises of tragedy makes it interesting in discussion of property relations within FOSS. Hardin's starting point is scarcity in physical world. If there were not any scarcity, tragedy would end without any agency of man. Additionally, the tragedy would totally reverse, if herdsman's animal addition to herd improved the pasture instead of destroying it. Indeed, these assumptions describe the domain of FOSS in capitalism.

Because of its different character from other goods in capitalism, Smith and Kollock (1999) described Linux as "the impossible public good". FOSS is nonrival and nonexcludable. It is nonexcludable, since contrary to the tragedy of Hardin's com-

⁷ Retrieved December 5, 2005, from <http://www.soldergisi.com/yazi.php?yazigoster=2068>
Stallman did not explain his disapproval in the interview. However, it is a known fact that hackers and IBM are on the opposite poles in the patent and the trusted computing debate

mons, there can be unlimited numbers of users who can not restrict the use of others. For example, anyone can download the GNU/Linux freely. In addition by having its source code, anyone can add new facilities and distribute it with another name, such as Ali-x, Ayse-x myOS etc. It is nonrival, since it can be replicated with nearly infinite cost. In other words, there is not any significant marginal cost in its replication. Proprietary software is also nonrival but it is excludable. In this context, being nonrival and nonexcludable FOSS stays in the realm of public goods and proprietary software in the realm of private goods (it is excludable) with its tendency towards public goods (since it is nonrival) (Weber, 2004).

In this respect, being nonrival and nonexcludable, FOSS represents an economy of abundance that give rise to describe it as a form of gift economy. Raymond (2000) compares three different types of organizations, command hierarchy, exchange economy and gift economy from the perspective of social status and power. In the command hierarchies, social status is related with access to coercive power. In an exchange economy, essence of social status is "having control of things (not necessarily material things) to use or trade" (Ibid.). On the contrary, gift economy (or gift culture) is not about holding power or controlling. Rather, one's social status is determined by what and how much one gives away.

For example, communities of scientists act through the principles of gift economy. In science, it is not important how much one knows, but how much one contributes to his/her field is essential among scientists. Moreover, if one had great knowledge and does not share with other scientists, it would be seen as waste of talent (Pinchot, 1999).

Furthermore, as Castells (2001a) remarks hacker culture inherited from the academic culture and hacker culture also bears the traits of the gift economy. Raymond (2000) defines hacker culture in the context of competence for prestige. According to Raymond, there is no serious shortage of bandwidth, disk space or computing power. Consequently, exchange economy became a pointless game without any serious scarcity. Reputation replaces monetary awards of exchange economy in three ways. Firstly, "good reputation among peers is a primary reward" (Ibid.). Secondly, it enables you to attract attention and cooperate with other people. For example, if

one is known as a hard-working person, it will be easy to persuade people to associate with him/her. Lastly, one's reputation spills over exchange economy and get job offers as a consequence of reputation.

Veale (2003) elaborates and extends Raymond's theory and explains gift economy on three principles. His first principle is the role of exchange in the Internet as a facilitator of reciprocity. Second principle is again the intangible rewards. This may be in many ways. Reputation plays prominent role as an intangible reward. Being appreciated by well-known hackers or national feelings, in other words feeling of working for one's own country may motivate people as an intangible reward. An intangible reward may also be just a self-esteem. Veale's last principle is leading of intangible rewards to tangible rewards. In this principle, Veale explains the process of contribution-reputation-money as Raymond, but also adds voluntary payment mechanisms. In a voluntary payment mechanism, gift-giver's (hacker) intangible reward transforms into a tangible reward by the donation of gift-receiver. In this exchange gift-receiver gets an intangible reward such as reputation of being generous.

Theory of gift economy explicitly removes the paradoxes of hackers' motivation in FOSS projects. In a commodity-based economy, people are paid for their work. In contrast, hackers voluntarily participate in FOSS projects and unremunerated. By voluntary participation, first of all they lose time that can be spent in different ways. First of all, they may develop software for commercial companies. Second, instead of focusing their real tasks, such as daily work, academic career or education, they prefer FOSS as an activity (Lerner and Tirole, 2004). It is not uncommon that student's progress slow down as Linus's (Torvalds and Diamond, 2001). Thus, voluntary production and free distribution of economically valuable goods is considered as behavioral anomaly by economists (Dalle et al., 2004). What the gift economy does is returning to early days of FOSS and studying its roots in academia to explain FOSS, so some researchers give up asking for monetary rewards or, at least, direct monetary rewards. Rather, their answer to the question becomes what the hackers get from FOSS is reputation instead of money (Kelty, 2004).

However, theory of the gift economy has some problems and as Weber emphasizes that there is a only surface resemblance between the FOSS culture and the gift

economy. According to him, it is true that FOSS

seems to bind people together, it encourages diffuse reciprocity, and it supports emotive feelings of stewardship for the gift that is both taken and given in return (Weber: 150).

Yet, Weber questions the extent of abundance. For him, despite the abundance of bandwidth and disk space, there is still some sense of scarcity:

Meaning and value depend on human mind space and the commitment of time and energy by very smart people to a creative enterprise. And the time, energy, and brain power of smart, creative people are not abundant. These are the things that are scarce, and in some sense they become scarcer as the demand for these talents increases in proportion to the amount of abundant computing power available (Weber: 151).

Thus, Weber is not convinced with the explanation of the gift economy. Even though it theorizes the practice of hackers, it is a contingent process and has the same problems with the conceptualizations of FOSS as a community and as a social movement. First of all, process of contribution-reputation-money may explain some hackers' practice, but it is not enough for a sustainable economy. Secondly, the gift economy, undervalues the social and economic conditions which it exists in. As Ghosh (1998) implies one should earn money to live in the capitalist economy.

The main thing people like Torvalds get in exchange for their work is an enhanced reputation. So there are, in fact, lots of Net gods. Net gods get hungry, though, and reputation doesn't buy pizzas. So what does Torvalds do? (Ghosh, 1998)

Consequently, Ghosh (1998) avoids from describing FOSS economy from the vantage point of reputation. Instead, he uses an image of vast tribal cooking pot process which people put into something (e.g meat, onion, carrots etc.) and take out a bowl of stew:

The cooking-pot model provides a rational explanation for people's motivations to produce and trade in goods and services, where a monetary incentive is lacking. It suggests that people do not only - or even largely - produce in order to improve their reputation, but as a more-than-fair payment for other goods - "ideas" - that they receive from the cooking-pot. The cooking-pot market is not barter, as it does not require individual transactions. It is based on the assumption that on the Net, you don't lose when you duplicate, so every contributor gets much more than a fair return in the form of combined contributions of others.

The cooking-pot model shows the possibility of immense value being generated through the continuous interaction of people at a numbing

speed, with an unprecedented flexibility and aptitude towards intangible, ambiguously defined goods and services. The cooking-pot market already exists, it is an image of what the Internet has already evolved into, calmly and almost surreptitiously, over the past couple of decades (Ibid.)

In this regard, Ghosh (1998) suggests a more rational framework than the contingent process of the gift economy. By describing FOSS, as a continuously accumulated social knowledge, he makes an important step to comprehending the existence of FOSS in the capitalist economy.

2.2 FOSS in the Informational Economy

2.2.1 FOSS as a Social Relation

According to Halverson (1998), the world in the last forty years is no longer broken up into stones, plants, and animals but into the unholy trinity of hardware, software and wetware. As it is discussed in the following paragraphs, new wares emerged as a consequence of mutual interaction between hardware, software and wetware, e.g. *Googleware, Infoware*.

The phrases such as, *digital economy, information society, information technology revolution* and *new economy* are used to describe this continuing change in the world. Their common argument is that the arousal of new world is conditioned by ICTs (Information and Communication Technologies). However, in this study, Castell's terms "informational economy" and "informational society" are rather preferred than the above terms. Castells (2001b) uses the term informational instead of information because of their different significations. Information society denotes the role of information in society. In all societies information was critical. But in this stage of society, information conditioned by ICT became profound component of social relations. For this reason, the term informational indicates:

the attribute of a specific form of social organization in which information generation, processing, and transmission become the fundamental sources of productivity and power, because of new technological conditions emerging in this historical period (Castells, 2001b: 21).

In this study, FOSS is conceived as a social relation which exists in informational society and determined by *wares*. Thus, instead of narrow (e.g FOSS Community)

and ahistoric (e.g Gift Economy) conceptualizations, FOSS is discussed in the context of its relationship with wares and it is studied as a process which contains its history and possible futures.

In its social context, hardware, the physical equipment of computer, defines the surrounding conditions for wetware and software, but it is also determined by both of them. In its historical movement, two distinct processes shaped the hardware: sophistications in semiconductor technology and emergence of Internet which transformed computers into a big whole computer. Each process had essential impacts on informational society; both quantitatively and qualitatively. First, decreasing computer prices and increasing computing power, increased both number of users and use of computers in different⁸ areas. As new users participate in the informational society, various cultures started to coexist. Furthermore, the Internet, interconnected networks, increased the complexity of interaction among computer users and altered the forms of work and leisure. On the other hand, new demands from these new participators of the information society also conditioned hardware.

In computer jargon, wetware is used for human beings attached to computer systems, e.g. programmers, operators, testers, system administrators. In this study, wetware is used in a wider context. In this aspect, wetware also denotes the attitude of governments and firms as well as the culture of prosumers⁹ in their historical movement. Wetware transformed ICT, "by appropriating it, by modifying it, by experimenting with it"(Castells, 2001a: 4). For instance, Internet, which started as a military project, was shaped by techno-meritocratic culture and the hacker culture until 1990s and had essential role in the process of restructuring capitalism after 1980s. Tehcno-meritocratic culture based is on the belief that scientific and technological development was the essence of progress of humankind.

Thus, the culture of the Internet is rooted in the scholarly tradition of the shared pursuit of science, of reputation by academic excellence, of peer review, and of openness in all research findings, with due credit to the authors of each discovery. Historically, the Internet was produced in

⁸ e.g health, education, business etc.

⁹ The term prosumer is composed of first three letters of producer and last five letters of consumer. It is used for emphasizing while users consume software, they also add value to it, in other words, reproduce it (Laurent, 2004).

academic circles, and in their ancillary research units, both in the heights of professional ranks and in the trenches of graduate student work, from where the values, the habits and the knowledge diffused in to the hacker culture (ibid: 39-40).

Furthermore, as Castells remarks (ibid: 41), it was the hacker culture which interacted on the Internet and shaped it. Moreover, hacker culture was also a bridge between the techno-meritocratic culture and the money driven culture of entrepreneurs in 1990s.

Software is defined as "sets of data and instructions capable of causing information technology hardware to perform particular actions" (OECD, 1985:10). In this study, it represents the practice of wetware on hardware. Software emerged and developed in sophistication and commodification process of hardware (Valimaki, 2005).

Software is a type of information. ICT reduced information to binary codes which is known as digitalization of information. As a consequence, digitalization eased and increased capacity for transmission and storage of information (Burnett and Marshall, 2003). Any information can be copied on cheap CD-ROMs or downloaded through Internet without any cost. For netizens¹⁰, especially for members of hacker culture, saying 'information wants to be free' was also their everyday practice on the Internet. Boyle (1996) explains this situation as decontextualization of information from any form or location:

The phrase "information wants to be free" has part of its force because it antropomorphizes this liquidity, this conceptually, technologically, and economically driven tendency to float free of some particular form and context. One might even call it the universalizing logic of the information relation. The tendency is toward the economic and conceptual separation of the informational message from the medium -cells, diskettes, telephone directories, or whatever- and of the progressive devaluation (literally, the diminishing marginal cost) of the medium as compared with the message (Boyle, 1996: 7)

However, saying 'software is a type information' is not enough to comprehend its specific role in the informational economy. Since as Castells remarks,

In the new, informational mode of development the source of productivity lies in the technology of knowledge generation, information processing and symbol communication. To be sure, knowledge and information

¹⁰ It is concatenation of substring 'net' in Internet and substring 'izen' in citizen.

are critical elements in all modes of development, since the process of production is always based on some level of knowledge and in the processing of information. However, what is specific to the informational mode of development is the action of knowledge upon knowledge itself as the main source of productivity (Castells 2001b: 17).

In this respect, software is more inclined to be defined as knowledge which is a subset of information:

Knowledge: a set of organized statements of facts or ideas, presenting a reasoned judgment or an experimental result, which is transmitted to others through some communication medium in systematic form. Thus, I distinguish knowledge from news and entertainment (Bell, 1976: 175).

Additionally, its role in processing of information makes software essential in the informational economy. In this sense, software is a means of production for both itself and other information goods.

Thus, *free* may not have the same meaning in either *free information* or *free software*. In its first meaning, it evokes the meaning of one does not have to pay any money. One may download (or copy from friends) audio files, e-books or a whole operating system, GNU/Linux without paying any money. However, for software programmers, *free* means as it is in *freedom* or *free speech*. While first *free* is more related with consumption, latter is about production. Being free in its second meaning differentiates software from other information goods. There are also collaborative production of information goods other than software in which everybody can freely contribute and benefit from the contribution of others. However, this is more common in software development.

Software may act as database, editor or calculator for performing needs of its user. In this sense, FOSS may be free in its first meaning. On the other hand, if you have the source code of software as in FOSS, it will be potential source code for a more functional software (Laurent, 2004). For instance, if you have the source code of a calculator application which performs only basic operations (addition, subtraction, multiplication, division), you may develop a more functional calculator in less time than building a calculator from scratch. In other words, FOSS as tendency of software has twofold aspects, use-value and means of production. One can use any FOSS according to his/her needs. Additionally, one can take the accumulated contribution of

programmers and by adding value one can create more functional software. In other words, if a programmer develops a basic calculator in five days, another programmer can take its source code and develop a more functional calculator in three days. Yet, if he does not have chance to take the code of basic calculator, he can develop it in eight days.

However, software which has tendency to be free, also exists in capitalism as proprietary software. In this aspect, it has also tendency to commodify which made Bill Gates one of the richest people in the world. For this reason, freedom of software contradicts with capitalism in which profit maximization is profound. According to Gates:

a competitor who is free to review Microsoft's source code... will see the architecture, data structures, algorithms and other key aspects of the relevant Microsoft product. That will make it much easier to copy Microsoft's innovations which is why commercial software vendors generally do not provide source code to rivals (Gates, 2002)

Furhermore, globalization of information harmonizes toward more protection (Nayyer, 2001) and "building fences to keep people out" (Boyle, 1996: 18) by copyright laws. In this context, while FOSS developers want software to be free as means of production, proprietary software firms want software to act as a commodity as anything in capitalism. As a result, while proprietary software build fences by copyright licenses, FOSS licenses try to remove these fences by their licenses.

In this context, FOSS contradicts with the characteristics of capitalism, profit maximation and it may seem staying away from capitalism. Yet, there are actual agents of FOSS in capitalism: FOSS firms and international companies (especially IBM and Novell). Additionally, hackers are real people who should sell their labor power in order to live. Even though hackers and firms have coinciding interests in promotion of FOSS, sometimes they pull FOSS in opposite directions. This reveals itself especially in the emergence of FOSS licenses.

In the next three parts, FOSS is discussed as a social relation that is included in:

1. Hackers' relation with their work
2. Firms interest in FOSS

3. Objectification of Conflicting Interests: FOSS Licenses

2.2.2 Hackers

In informational capitalism, ICTs made flexible work models possible and introduced new workers which Castells called as "flex-timers" (Castells, 2001b: 281). According to Castells this was

the reversal of the historical trend of salarization of work and socialization of production that was the dominant feature of the industrial area. The new social and economic organization based on information technologies aims at decentralizing management, individualizing work, and customizing markets, thereby segmenting work and fragmenting societies (Castells, 2001b: 282).

Flexible forms of employment has four dimensions. First, it is not constrained by traditional work patterns of 35 or 40 hours work per week. Second, it is task oriented. There is not any future guarantee for the worker. Third, there is an increasing number of workers who work at home. Some works in another company whom the worker's company subcontracts. Last, the social contract between the employee and employer is changing. Workers do not have well-defined rights anymore (Carnoy, 2000).

Sennett (1998) also compares the new economy with the industrial economy and presents the personal consequences of work in the new capitalism. In his book, *The Corrosion of Character*, Sennett (1998) explores the uncertain social world of those moving from job to job. He discusses their flextime work and questions the flexibility of the workers: "A flextime worker controls the location of labor, but does not gain greater control over the labor process itself" (Sennett, 1998: 59) which results in the powerlessness of flex-timers.

Under these social circumstances, hackers' passionate relation with their work becomes a vital question for most of the studies in the FOSS literature. If the work in the informational economy is so corrosive on the characters of the workers¹¹, why the hackers goes on working in leisure time? Moreover, how can the hackers use the adjectives "fun", "joyous", "passionate" and "entertaining" to describe (Chance, 2005) their work?

¹¹ Including high-skilled IT professionals (as IBM workers) (Sennett, 1998)

Indeed, Sennett and hackers do not speak of the same kind of work. Sennett's work conceptualization remind Marx's words about work that is avoided as a plague. However, Marx also criticizes Adam Smith who undervalues the essence of work in one's life :

It seems quite far from Smith's mind that the individual, 'in his normal state of health, strength, activity, skill, facility', also needs a normal portion of work, and of the suspension of tranquility. Certainly, labor obtains its measure from the outside, through the aim to be attained and the obstacles to be overcome in attaining it. But Smith has no inkling whatever that this overcoming of obstacles is in itself a liberating activity - and that, further, the external aims become stripped of the semblance of merely external natural urgencies, and become posited as aims which the individual himself posits - hence as self-realization, objectification of the subject, hence real freedom, whose action is, precisely, labor (Marx, 2003: 94).

This study points out that hackers' work is different from the contemporary meaning of work which is ascribed by Sennett. Indeed, the essential difference between them is that an external power controls the worker in Sennett's ascription in spite of the self-autonomy in hacking.

The most important fact about the hackers is that they enter voluntarily into the work and this work includes operational autonomy, a variety of productive and creative activities. Chance describes this work as meaningful, since

hackers can choose what code to work on; they can influence the agendas that direct their work, both by entering into the open discussions about the direction of the project(s) they work on and by simply opting out of any projects whose agendas conflict with the hackers' own priorities; and finally though it is not forced nor prevalent in every productive forum, hackers can and often do enter into discussions about work itself (Chance, 2005)

On the contrary to contemporary society, hackers reject any distinction between work and leisure, instead they emphasize active realization of themselves as a creative and autonomous person. Consequently, instead of investigating about anomaly in hackers behavior, this study asks for the social conditions of hackers work and its sustainability in the informational capitalism.

Furthermore, societies act and react to informational capitalism differently according to their cultural and historical specificity. Although all societies are affected by informational capitalism, it happened in "different settings with specific cultural

/ institutional expressions” (Castells, 2001b: 21). For this reason, as one can not speak of homogeneity of informational society in the all countries, one can also not speak of homogeneity of the hackers. It is sure that there are some common traits among hackers. However, emergence and propagating of hackerdom are not the same in the all countries. For example, in the US, “the culture of freedom, individual innovation, and entrepreneurialism that grew out from the 1960s culture of American campuses” (Ibid.) played essential role in the emergence of FOSS. European, Chinese or Indian hackers appropriation of FOSS happened in different historical and social contexts. So did the Turkish hackers.

Thus, despite the hackers’ common attitude towards work , specificity of the hackers will be the second premise of the study about hackers.

2.2.3 Firms

There have been two main questions since the rise of academic interest for FOSS. First question is about the hackers’ motivation: *If the hackers are not paid, why do they contribute to FOSS projects?* As it is discussed in the previous chapter, source of this question is a presumption about man’s inclination to resting rather than working. Second question is about the firms’ participation in FOSS projects which is also counter to the conventional wisdom: *Why do firms contribute to FOSS?* It is also counter to economic intuition, since:

private agents, without property rights, will not invest sufficient effort in the development of public goods because of free-rider externalities (Bessen, 2005).

As emphasized in the discussion of the gift economy, because of its features being nonexcludable and nonrival, FOSS has been named as the impossible good. However, this specificity of FOSS in the capitalist economy still does not give a sound economic reason why firms contribute to FOSS development even when they also make profit from proprietary software. Additionally, approaching FOSS business models by only looking at the differences between FOSS and proprietary software, fails to understand real nature of FOSS and convergence between FOSS and proprietary software business models. Instead, studying FOSS firms in the context of

informational capitalism presents a more explanatory framework for comprehension of FOSS in firms' business strategies.

As Castells (2001b: 149) states there were Internet-related firms at the heart of IT industries and he predicts that this situation will not change in the 21st century. As he explains, first, they have essential influence on the conduct of business processes. Second, they have played a major role in the development (e.g increase in revenue and employment) of informational economy. CREC's (Center for Research in Electronic Commerce) report¹² classifies those Internet-related firms into four layers. First layer is the Internet infrastructure layer. In this layer, there are firms who supply products and services for network infrastructure. This layer includes networking hardware companies (e.g CISCO, 3COM) as well as Internet service providers (e.g AOL). Second layer is the Internet applications layer. This layer makes network infrastructure technologically feasible for business activities. This layer includes web-enabled databases (e.g Oracle, MS SQL Server), multimedia applications (e.g Macromedia), Internet commerce applications (Microsoft, Netscape, Sun). Third layer is the Internet intermediary layer. Role of this layer is facilitating the interaction between sellers and buyers over the Internet. This layer is a catalyzer in transformation of investments into business transactions. Online travel agents (e.g. TravelWeb.com), content aggregators (e.g Cnet, ZDnet), portal/content providers and online advertising (e.g. Yahoo, Google) belong to third layer. Last layer is the Internet commerce layer which is related with selling products and services to customers or businesses over the Internet. In this layer, there are e-tailers (e.g. Amazon.com), airlines selling online tickets, online entertainment services, manufacturers selling online (e.g. IBM, CISCO).

Furthermore, the Internet brought about a new *ware*, which is called by O'Reilly (1999) as *infoware*. He argues that killer application of the IT is no longer desktop productivity tools such as office applications but it is web site applications that might be called as information application or infoware. In his another article (O'Reilly, 2005), he distinguishes between the web before and after the infoware as Web 1.0

¹² Retrieved April 18, 2006 from http://crec.mcombs.utexas.edu/works/articles/internet_economy.pdf
The report is sponsored by CISCO.

and Web 2.0. In this aspect, while Netscape¹³ is a kind of Web 1.0 software company as Microsoft and SAP, Google is a typical Web 2.0 company as eBay and Amazon. Google is different from Web 1.0 software companies, since

Google, by contrast, began its life as a native web application, never sold or packaged, but delivered as a service, with customers paying, directly or indirectly, for the use of that service. None of the trappings of the old software industry are present. No scheduled software releases, just continuous improvement. No licensing or sale, just usage. No porting to different platforms so that customers can run the software on their own equipment, just a massively scalable collection of commodity PCs running open source operating systems plus homegrown applications and utilities that no one outside the company ever gets to see (O'Reilly, 2005).

In fact, Google is also transcendence of the trinity (hardware, software, wetware) and including each of them that is called as *googleware* (Vise and Malseed, 2006).

As it is presented in the fourth chapter, FOSS has been a fact of IT since the emergence of software. However, only after the rise of infoware, FOSS signaled a breakthrough in the software industry. As a result of increasing demand for a "silver bullet"¹⁴ (Brooks, 1995), success of FOSS projects proposed feasible solutions for the firms.

In this context, understanding software's specificity, software industry's previous solutions for complexity and unique characteristics of FOSS is necessary for understanding FOSS business models.

Beside being nonrival, as any information good, software is a complex product. A software product has a number of features and has a significant number of interactions between these features. An additional feature increases the complexity of software. However, :

a scaling-up of a software entity is not merely a repetition of the same elements in larger size; it is necessarily an increase in the number of elements. In most cases, the elements interact with each other in some non-linear fashion, and the complexity of the whole increases much more than linearly (Brooks, 1995: 182).

¹³ a web browser as Internet Explorer.

¹⁴ It is a well-known metaphor in software engineering literature. Silver bullets are used to lay werewolves to rest. In *Mythical Man Month*, Brooks used this metaphor to say that there is not any solution for software projects which can make software costs drop as rapidly as computer hardware costs do.

This complexity does not rise merely from internal elements of software (e.g. packages, source code, libraries). Complexity includes "unvisualized states that constitute security trapdoors" and "difficulty of communication among team members, which leads to product flaws, cost overruns, schedule delays" (Ibid.). Additionally, a software product is inclined to be more complex. Since change is inherent in software products.

The software products is embedded in a cultural matrix of applications, users, laws, and machine vehicles. These all change continually, and their changes inexorably force change upon the software product (Ibid: 185).

Bessen (2005) discusses the software industry's three solutions against the complexity of software: pre-packaged software, API (Application Programming Interface) and FOSS. Pre-packaged software combines group of features in a bundle which multiple customers can purchase. If the customer's needs are met by bundled features, it is an efficient solution. However, as emphasized in the above paragraph, most of the customers have continuously changing needs. Thus, if the firms insist on producing software according to these changing needs, this solution will be inefficient. For this reason, some proprietary software firms also sell the tools (APIs) to call functions in the bundled software for solving the problem of increasing and changing needs. In fact, APIs are means of production that enable the customers to develop software for themselves. Consequently, last solution is FOSS which have the same solution principle as giving APIs approach, but in a wider context. This solution will be more efficient than giving APIs, because

these firms share their code in turn, the base of available code can grow far greater than the code available in commercial APIs, allowing far more complex applications to be developed efficiently (Bessen, 2005).

In this context, solutions, pre-packaging and giving APIs are preferred by proprietary software vendors. Furthermore, according to CREC's report, this firms mostly belong to second layer of the Internet. Their business model is based on selling right to use licenses.

On the other hand, third and fourth layers mostly prefer FOSS solutions in order to solve complexity problem of the software. In these sense, FOSS firms can be put into two broad categories. In the first category, there are FOSS integrators. FOSS

integrators either sell software projects to customers or they are infoware firms. IBM and many large companies fall into this category and their business model is based on selling integration services. Additionally, infoware firms, Google and Yahoo run on FOSS and they earn money by advertising.

Second category FOSS firms are more directly related with the FOSS products. As their business might be based on selling services for infoware, their product might be equivalent of second layer's proprietary software. They differ from proprietary software by charging no license fees for using the product or distributing its source code (Valimaki, 2005). Instead, they sell complementary services for FOSS products which are publicly available or they sell products which are complemented by FOSS (Hecker, 2000).

The earliest and the simplest FOSS business model is accesorizing by selling CDs and books associated with FOSS. Although this business model was founded by enthusiasts for enthusiasts, later it is extended and the other complementary services such as consultancy, support and training are included in this business model. Furthermore, their customers are extended from enthusiasts to less technical people (Golden, 2005).

Contrarily, some FOSS firms may sell another product and FOSS complements this product. This product may be hardware as well as a proprietary software product (Hecker, 2000). Firm's strategy while making its hardware's driver FOSS is "the more places you can enable your device to function, the greater your potential market." (Fink, 2003: 182). FOSS firms may sell proprietary software with the same strategy as selling hardware (Hecker, 2000) or proprietary software firms may make a part of their product FOSS with the same expectations. These strategies points out the convergence between FOSS and proprietary software business models and it is becoming widespread under the name of hybrid business models.

On the other hand, convergence may occur with dual-licensing. Dual licensing means FOSS is released with both FOSS license and proprietary software license. However, there might be differences between these releases. For example, proprietary release might be newer, more tested or with more enhancements than its FOSS licensed release.

2.2.4 FOSS Licenses

The common point of FOSS business models is, although FOSS firms sometimes contribute to the FOSS projects, they highly owe to the volunteer contribution of the hackers. In other words, while source code is conceptualized as crown-jewels which must be hidden from the others in the proprietary software business, in FOSS business source code is continuously accumulated social knowledge. Thus it needs sharing for its existence and giving freedom for attracting contribution of hackers. Thus, firm's success depends on ongoing involvement of the hackers (Golden, 2005).

There are many studies which point out the role of hackers in the firms FOSS business model. For example, Fink warns the firms about dual licensing:

Depending on the project, the community may object to copyright assignment. Make sure you seek council with potential members of the community before assuming that everyone will agree to assign you their copyright (Fink, 2003: 182).

As Hecker remarks,

much of your potential success will depend on the efforts of others willing to work for you "for free" you need free-software developers who will contribute their work to your company, and to the developer community at large, without demanding or receiving money or other tangible payment in return. However, many free-software developers will not (and should not) do such work unless you treat them fairly and provide them with the freedoms and other in-tangible "payments" that they do want (and demand). This stems in part from your company's attitudes and actions toward developers working with its products, but is also formalized in the company's choice of an open-source license, specifying the terms and conditions under which the company's open-source products can be used, modified, and redistributed (Hecker, 1999).

Additionally, Onetti and Capobianco (2005) also underlines the key role of hackers' work in FOSS business. There are also studies which emphasize about supporting the FOSS developers in order to keep away from the danger of freeloader image in hackers' eyes (DiBona et al., 1999).

However, despite these statements, possible conflicts of interests between individuals and firms are usually undervalued and not studied in detail. In one of the rare studies that discusses this conflict, Bonaccorsi and Rossi (2003, 2004, 2005) point out "altruistic individuals" and "selfish firms" from the perspective of relationship

between their different sets of motivations. In their large-scale survey, they conducted interviews with 146 firms and dealt with their business models and their attitudes towards FOSS hackers. Later, they compared their results with the previous studies about hackers' motivation.

Bonaccorsi and Rossi (2005) emphasize that while firms enter the FOSS field in order to profit from FOSS, monetary rewards are not so crucial for the hackers. In this respect, they make distinction between the extrinsic and intrinsic motivations. Extrinsic motivations are mostly related with monetary rewards and intrinsic motivations are related with the pleasure of activity itself. For this reason, while firms mostly have extrinsic motivations, intrinsic motivations are more essential than extrinsic motivations for the hackers. In other words, hackers are more interested in the activity itself than the product.

Bonaccorsi and Rossi (2005) approach the relationship between the hackers and the firms from the perspective of if the firms really comply with the rules of the hackers. Additionally, they compare the firms behaviors and attitudes towards FOSS, since what they claim may not be what they really do. For instance, while a FOSS businessman may claim that his firm supports FOSS activities, his firm may not have a budget for this support.

Another approach in investigating the relationship between the hackers and the firms may be studying the evolution of the FOSS licenses and contradiction between free and open source software. Since as discussed in the fourth chapter, the term open source software was coined as a marketing strategy for free software.

In this context, first of all, a misconception, FOSS as a public good must be corrected. Being in public domain means there is not any copyright holder who has the right to restrict copying or using the software.

On the contrary, McGowan argues that

Open-source software production is not about the absence or irrelevance of intellectual property rights. Open-source production instead rests on the elegant use of contractual terms to deploy those rights in a way that creates a social space devoted to producing freely available and modifiable code....Open-source production, therefore does not take place within a true commons... (McGowan, 2001)

FOSS licenses are based on copyright and some of them (especially GPL) restrict

copying or using according to the statements in their license agreements. This happens in two steps. First, they reverse the traditional copyright and give users copying, using and distributing rights. For example, according to GPL ¹⁵:

- You have the freedom to run the program, for any purpose.
- You have the freedom to modify the program to suit your needs. (To make this freedom effective in practice, you must have access to the source code, since making changes in a program without having the source code is exceedingly difficult.)
- You have the freedom to redistribute copies, either gratis or for a fee.
- You have the freedom to distribute modified versions of the program, so that the community can benefit from your improvements.

However, in the second step , GPL puts restrictions¹⁶:

- Any derivative work-that is, any work containing a nontrivial amount of GPLed code-must itself be distributed under the GPL.
- No additional restrictions may be placed on the redistribution of either the original work or a derivative work. (The exact language is: "You may not impose any further restrictions on the recipients' exercise of the rights granted herein.")

Furthermore, these restrictions made GPL the most popular FOSS license by preserving the source code for hackers. As DeLanda (2001) emphasizes, GPL has been like a virus. If one developer uses a GPL'd software, the derivative work should also be licensed with GPL that will be source for the latter works. In fact, FOSS with GPL is the most suitable license for conceptualization of software as continuously accumulated social knowledge. On the other hand, OSI (Open Source Initiative) does not insist on GPL. Moreover, according to some FOSS advocates, GPL's reciprocal statements make it less free and an important barrier for attracting business world to the FOSS, so they encourage using less restrictive licenses (Fogel, 2005). Among these licenses BSD is the most popular. Its mere statement is writing the author of the code. It gives any right to user including making source code private (Laurent, 2004).

According to open source advocates, giving freedom to make source code private increases firms incentive to participate in the FOSS projects. Indeed, Microsoft also

¹⁵ Retrieved December 5, 2005, from <http://www.gnu.org/philosophy/free-sw.html>

¹⁶ Retrieved December 5, 2005, from <http://www.fsf.org/licensing/licenses/gpl.html>

agrees with open source advocates. For Microsoft, open source has some positive elements, but GPL is a threat for IPR: ¹⁷:“The GPLs viral nature poses a threat to the intellectual property of any organization that derives its products from GPL source”.

In this context, although BSD and other open source licenses encourage firm’s participation in FOSS projects (by giving them right to proprietarize source code), they do not guarantee the continuity on social accumulation of knowledge. Additionally, they are less successful than GPL in building collaborative development around a project (Fink, 2003). On the other hand, if project’s goal is making software be accessible by the greatest possible number of users and implementing a standard in a part of software sector, BSD-style licenses gain importance.

2.3 FOSS In The World

2.3.1 Unconscious FOSS

Although, conceptually, software is as old as computers, the term software did not appear until the late 1950s¹⁸. The emergence of modern computers is highly inter-related with the World War II. Development of computers during and immediately after World War II was directed toward scientific and technical rather than business objectives (Steinmueller, 1996). For example, between 1943 and 1947, US Army developed the first all-electronic computer ENIAC (Electronic Numerical Integrator and Computer) for their Ballistics Research Laboratory for calculating ballistic firing tables. ENIAC was capable of being reprogrammed by rewiring as electromechanical computers. It was more than 1000 times faster than any contemporary electromechanical computer and it was used for simulating the ignition of the hydrogen bomb. However, ENIAC had no immediate commercial applications. Its programmability by rewiring made ENIAC highly specialized to particular information processing tasks. In other words, software did not have separate and permanent existence in ENIAC (Seidel, 2002).

¹⁷ Retrieved December 5, 2005, from <http://www.microsoft.com/presspass/features/2001/may01/05-03csm.mspx>

¹⁸ The term software was first used by John W. Tukey in 1957 (Retrieved December 5, 2005, from <http://en.wikipedia.org/wiki/Software>)

Despite its shortcomings, ideas generated from ENIAC had important impacts on Von Neumann's revolutionary conceptual architecture for computers. In Von Neumann architecture, computer uses a single storage structure to hold both the set of instructions on how to perform the computation and the data required or generated by the computation. Ability to hold the set of instructions as data made stored-program machines available. In this way, the instructions could be easily changed. As a result, software started to become a more independent part of the computer and gained separate and permanent existence. More importantly, re-programmability enabled usage of computers for business applications.

In 1950s, the project SAGE (Semi Automatic Ground Environment) funded by US Air Force, played an important role in the commercial development of the American computer industry. It has been evaluated as the most important learning experience in computer history which led to many inventions (Campbell-Kelly, 1995). It revolutionized computer industry from batch processing to the world of interactive systems. Furthermore, SAGE project showed that computers could be used for purposes other than scientific computation.

Increasing commercialization of computers stimulated a market for software services. Computer producers needed the production of software to embellish and sell their computers. For this reason, leading computer manufacture, IBM, started to provide programming services and software tools. Programming languages and their end, source code, appeared at that moment of history:

Providing these complements accelerated adoption of new general-purpose computers, reinforced links between computer producers and users, and laid a foundation for the reuse of software in future machines. If software instructions could be made less machine-specific, the costs of adopting new machines could be reduced. Computer manufacturers accordingly focused on producing the tools for creating applications programs rather than developing application program themselves. Corporations using computers thus needed to develop software for their own information processing applications (Steinmueller, 1996: 22).

After this breakthrough, wetware as a hacker, would write programs easier. In this context, comparing the relationship between wetware and software, before and after emergence of programming languages is essential in understanding FOSS. A software application (e.g web browsers, editors, calculators, translators) is sets of in-


```
11100001 11110000 00110011
11100000 11100001 10101010
01101011 00001100 11010010
11111100 00001111 00110000
```

Figure 2.1: Machine Code

```
int main()
{
    printf("Hello world");
    return 0;
}
```

Figure 2.2: Programming Language Code

structions and data. For example, let one wants to write a program, that prints "hello world" on the screen. Before programming languages, s/he would have to write series of 1s and 0s in machine language (Figure 2.1).

On the other hand, in a programming language, source code is human readable (Figure 2.2). The source code is translated into a lower-level language, assembled by compiler. In the final step, it is translated into 1s and 0s of machine-language. It is obvious that a program, first written in machine language can run faster than the same program written in a programming language because it does not have to make any work for compiling source code into assembly language and translating assembly language into machine language. However, comprehending and modifying any code written in a programming language is easier than series of 1s and 0s. Thus, emergence of programming languages was an important step to overcome scarcity problem in software industry because of its two outcomes: reusability and share of software.

In late 1950s, hardware manufacturers supported the formation of user groups for sharing software routines, in other words source code. In this sense, IBM sup-

ported user group, SHARE¹⁹, was the ancestor of FOSS. In 1950s and early 1960s, what makes a computer commodity was its hardware. Software was still being considered as part of hardware. Thus, the main incentive behind the SHARE was to stimulate the use of computers.

However, in the second half of 1960s, three events had important impacts on the disintegration of hardware and software industries. The first event was the introduction of IBM System 360 family of computers which gave chance to independent software vendors to develop and market the same product to a variety of users. The second event was IBM's announcement, on December 6, 1968, of its intention to unbundle hardware and software sales which means pricing them separately. Until then, software was part of hardware and IBM's customers had to pay for it whether they used it or not. According to IBM, this decision was because of rising software costs. On the other hand, according to OECD (1985), this was because of antitrust pressure on IBM. Whether commercial judgment or antitrust pressure was behind this decision, it provided an incentive for independent software vendors to produce software compatible with IBM products. Lastly, the third important event was the development of the minicomputer industry. Minicomputers made it possible for small organizations to begin to purchase and operate their own computers (Campbell-Kelly, 1995).

Early in the 1970s, as a result of improvements in semiconductor technology, faster, cheaper and better hardware made an impetus for software industry growth. The personal computer software industry effectively began in 1975-78. Financial and technical barriers to entry in this new industry were low. Since existing software firms could not recognize the potential for a personal computer software market and specifications of early personal computers were very low which did not need engineering knowledge (Ibid.).

In 1975, the first personal computer, Altair 8000, was advertised to hackers in the popular technical press. Bill Gates, the founder of Microsoft with Paul Allen was one of those hackers. Microsoft developed its first program, BASIC, a simple programming language, for Altair. One year after this, Gates would publish an Open Letter

¹⁹ Retrieved December 5, 2005, from <http://www.share.org/About/>

to Hobbyists for the users of the first personal computer, the MITS Altair. He warned the users about piracy, making illegal copies of the program, BASIC, developed by Microsoft for Altair. He was arguing against the long tradition of sharing software. According to Gates, hobbyists should be aware that software was not something to share but must be paid for as hardware. He condemned most of hobbyists stealing and neglecting if the people who worked on it get paid. Additionally, he told that piracy was the prevention of good software from being written. He asked ironically:

Who can afford to do professional work for nothing? What hobbyist can put three man-years into programming, finding all bugs, documenting his product, and distribute it for free? (Moody, 2002: 2)

Although sharing had been a widespread tradition in software world, the economy was following its own inclination, making software also a commodity produced for the market. After the early 1980s, the success of the personal computer had led to an explosive growth in the mass-market software products industry; PC revolution had begun. Software part of computer started to become commodity as its hardware. By the advent of PCs, the users of computers were not only scientists and engineers anymore. For this reason, it did not take long for companies to get aware that most people just wanted to run programs. Ordinary users did not need source code that could be hacked. As a result companies quickly learned to take the source code to themselves, in other words made source code proprietary, as a marketing strategy. While their customers were satisfied, they would lock out their competitors. PC users had lost their freedom in the beginning (Wayner, 2000).

Actually, users, who would never want to re-program their applications, did not mind about source code being proprietary. Since they were not programmers and freedom to produce, in other words, to look at source code was not necessary for them. Additionally, proprietary software in PCs did not also bother the hackers, who consider PCs as toys. At this moment, hackers were programming on more sophisticated systems using PDP-10s²⁰ and UNIX which had grown to be very popular in universities and laboratories. AT&T had designed and built UNIX at Bell Labs throughout 1970s. In the beginning, AT&T shared the source code with researchers and computer scientists in universities. At first, UNIX was just an experiment for

²⁰ Later it became PDP-11

the company to run the next generation of telephone switches which were already using another system. In the development of UNIX, contribution of universities was essential (Wayner, 2000).

But project which was an academic exercise at first turned into a nice operating system. When company started splitting up in 1984, AT&T wanted to turn a profit from their investment. In other words, they wanted to follow commodification process in PCs and asked universities to sign nondisclosure agreements for UNIX (Stallman, 2002).

At this moment, hackers had lost the source code, which was means of production to program. More importantly, this end was pointing out a new stage in the commodification of computers. After the commodification of hardware and software, it was source code's turn.

2.3.2 Hackers and Free Software Movement

On September 27, 1983, an unusual message with a subject, New UNIX Implementation, appeared on the Usenet newsgroup, net.unix-wizards:

Free Unix!
Starting this Thanksgiving I am going to write a complete Unix-compatible software system called GNU (for Gnu's Not Unix), and give it away free to everyone who can use it. Contributions of time, money, programs and equipment are greatly needed. (Stallman, 1984b)

This message was unusual in the sense that it did not release a new version of UNIX, but it suggested to rebuild UNIX from the ground up. The owner of this idealist and arrogant message introduced himself as:

I am Richard Stallman, inventor of the original much-imitated EMACS editor, now at the Artificial Intelligence Lab at MIT. I have worked extensively on compilers, editors, debuggers, command interpreters, the Incompatible Timesharing System and the Lisp Machine operating system. I pioneered terminal-independent display support in ITS. In addition I have implemented one crashproof file system and two window systems for Lisp machines (Stallman, 1984b).

More interestingly, his goal was not just a new operating system:

I consider that the golden rule requires that if I like a program I must share it with other people who like it. I cannot in good conscience sign a nondisclosure agreement or a software license agreement.

So that I can continue to use computers without violating my principles, I have decided to put together a sufficient body of free software so that I will be able to get along without any software that is not free (Ibid).

This was a move against current commodification of software and source code. Additionally, it was a turning point in the history of software, transformation of software from a form which is free in itself to a conscious form which has become the objectification of definite goals (Fogel, 2005: 12). For this reason, comprehension of the leading figure in this transformation, Richard M. Stallman, is essential. In fact, his writings were usually theory of hackers practice and in his writings, he conceptualized FOSS from the vantage point of the hackers²¹.

In his words, Stallman was the inventor of EMACS and programmer of some famous software (Stallman, 1984b). On the other hand, Levy described Stallman as "The Last of the Hackers, who vowed to defend the principles of hackerism to the bitter end." (Levy, 2001).

Hackerdom played an essential role in the history of FOSS. According to Levy, hackers were programmers and designers for whom computing was the most important thing in the world. He described them as heroes of the computer revolution (Ibid.). Hackers are different from ordinary users in the sense that they explore the details of systems instead of the users who learn only the necessary ²². This is the difference between producers/users and consumers/users (Castells, 2001a).

Hackerdom existed in the relational world of hardware, wetware and software. In literature, Internet is usually considered as the indispensable part of hackerdom. There were also vast amount of MS-DOS and Mac hackers. Moreover, Bill Gates was also a hacker. However, only Unix hackers, which organized on Internet, became a self-aware culture (Castells, 2001a). Unix hackers did not care about commercialization contrary to Bill Gates . They just wanted better tools and cheaper Internet (Raymond, 1999a). Furthermore, MS-DOS and Mac hackers had PCs which an individual could afford. On the other hand, hackers were using more powerful but ex-

²¹ In the interviews of the study, it was clear that Stallman was the reference point. Hackers answered the questions about free software in terms of Stallman's writings.

²² The Jargon File, Version 4.4.6, Retrieved December 5, 2005, from <http://www.jargon.8hz.com/html/>

pensive hardware which could only be funded by universities and research centers. For this reason, hackerdom was inherited from techno-meritocratic culture of academic world. Values and habits of academic world diffused in the culture of hackers. According to Castells(2001a: 40), "shared pursuit of science, reputation by academic excellence and openness in all research findings" was the tradition inherited from academic world to hackers.

While hackers shaped the Internet, Internet played a prominent role in the culture of hackers²³. Growth of Internet has been determined by three principles which had their roots in hackers culture:

1-) networking architecture must be open ended, decentralized, distributed, and multi-directional in its activity, 2-) all communication protocols and their implementation must be open, distributed and susceptible of modification 3-) institutions of the governance of the network must be built in accordance with the principles of openness and cooperation that are embedded in the Internet (Ibid. :28).

In this context, free software was a natural consequence of interrelation between hackers and Internet. Most of the main Internet services are run by free software: BIND (Berkeley Internet Name Domain) for the DNS (Domain Name System), sendmail for e-mail, latter Apache and Perl for the Web. In other words, as emphasized in the first part of this chapter, saying of hacker culture, "Information wants to be free" (Boyle, 1996) was the theory of practice of hackers.

Hackerdom had twofold aspects in the history of FOSS. First of all, they added a social dimension to the technology sharing. Additionally, freedom of expression, security and individual's own activity became the ideals of hackers (Himanen, 2001). In these years, authorities shut their eyes to those activities of hackers, since they

understood that the extra overhead was a small price to pay for attracting an entire generation of bright young people into the computing field (Raymond, 1999a).

On the other hand, it had the roots of techno-business entrepreneurs' culture. Commodification of source code and Stallman's reaction against this was the early realization of contradiction between two cultures. Hackers had started to be hired. Stallman describes this situation "Everything was dying"(Moody, 2002: 18).

²³ The reciprocal relation between wetware and hardware

Message of Stallman, "New Unix Implementation", appeared in this social context. The GNU project formally began in January 1984. The GNU Manifesto, written by Stallman in the beginning of the project, was about the sharing of source code. When GNU Manifesto emerged as the most radical reaction against locking up the source code, people looked at it with confusion(Wayner, 2000).

Main goal of GNU project was to re-create the software sharing community after its collapse as a result of commodification. In other words, software, which was free in itself and commodified later was being defended by a conscious free software form by GNU Project. Practice of "Free" is theorized by Free Software Movement (FSM) as computer users' right to use, copy, modify, and redistribute a GPL'd software. FSM strongly emphasizes that "free" is about freedom not about price. In other words, free is as in "free speech", not as in "free beer". Despite the claim that software can be sold, above principles were making software also with zero price as a side effect of sharing (Stallman, 2002).

Stallman was aware of possible problems about free software. He pointed out that if a program was free software when it left the hands of its author, this did not necessarily mean it would be free software for everyone who had a copy of it. In this context, he gave X Window System as an example. It was developed at MIT and released as free software with a permissive license. Soon it was adopted by computer companies and sold in binary form with their proprietary Unix Systems. Consequently, users lost the freedom which the original authors of software had given them (Stallman, 2002).

For this reason, Stallman offered another licensing method *copyleft*. In fact, copyleft uses copyright, but flips it over to serve for opposite purpose. Instead of privatizing software, copyleft prevents software from privatizing. The central idea in this method is making user free, but preventing user from restricting other users. According to the GNU General Public License (GNU GPL) which is based on copyleft, you can not make your modifications private. For example, if X Windows were licensed, any commercial software vendor would also have to release it with its source code instead of only in binary form. An additional restriction is you are not allowed the

incorporation of a GNU GPL'ed program into a proprietary software ²⁴.

Stallman's aim was conserving the user's freedom. But "it was like a virus" (DeLanda, 2001). If you used a GNU GPL'ed software, you would have to release it with the same license. As a result, number of GNU GPL'ed programs increased very fast and forced a vast amount of software to be free.

Perens (1999) remarks that political rhetoric of GNU GPL puts some people off. It protects the users' freedom but restricts the software vendors when they want to make their software proprietary. Stallman's (2002) response to this critic is to remind the goal of GNU was to give users freedom, not just to be popular.

Since the free software emerged as a radical reaction to commodification of source code, GNU project grew by accepting contributions from many hackers. In the early 1990s, people began to realize that a completely free operating system, as announced by Stallman, was a serious possibility (Wayner, 2000). There had been a vast amount of software donated to the GNU Project and a free operating system was almost ready.

The gradual commodification of computers was trying to be reversed. For FSM, the problem was only commodification of source code. But their resistance to source code's commodification resulted in decommodification of software as a side effect. In spite of the fact that FSM insisted on 'free' in free software is like 'free speech', not as 'free beer', for people, who do not program, opposite of this fact was valid.

2.3.3 Internet and Linux

As mentioned in the previous part, the first item on the free software agenda was a free operating system. Writing a whole operating system was a very huge work which includes development of a kernel, compilers, editors, text formatters, mail software, etc. (Stallman, 2002). By the 1990, GNU project was almost complete, except the major component, kernel. The planned kernel, GNU HURD (Hird of Unix-Replacing Daemons), would run on top of a microkernel which had been developed at Carnegie Melton University and then at the University of Utah. FSF (Free Software Foundation) waited for the free release of Mach to start development of kernel. Yet,

²⁴ However this is allowed in a less-restrictive license: GNU Library General Public License

HURD has not been officially released and still being developed (Raymond, 1999a).

FSF and its leader Stallman, played a prominent role in the history of FOSS, especially in recovery of FOSS after commodification of source code as a conscious form. However, at that moment, they were not the only actors confronting against the proprietary software. FOSS also owes to hackers in the Department of Computer Science at the University of California at Berkeley whose participants were the direct victim of AT&T during commodification (Wayner, 2000).

As a response to AT&T's attempt to make UNIX proprietary, one of the core teams in Berkeley, leaded by Keith Bostic, started to replace the source code that belongs to AT&T with their own code. In June 1991, a complete version of Unix, Network Release 2, was ready but without six files. It is released under the BSD (Berkeley Software Distribution) License which was less restrictive than GNU GPL. New versions of BSD (386BSD, FreeBSD and NetBSD) stemmed from Network Release 2 by adding the missing six files. BSD source codes were being distributed through Internet. But those free software did not seem to catch AT&T's attention. Since Internet was not so popular at that time, almost only universities and hackers were online. BSD variations were caught when a company called BSDI (Berkeley Software Design Incorporated) started putting advertisements in press. They were offering source code for \$995 which was a huge discount for AT&T's UNIX. AT&T sued BSDI for actually stealing source code. Later Berkeley involved in the battle. In the end, Berkley won the battle by January 1994. However until that time, developers of BSD worried about the result of the court which had negative effects on BSD projects because of uncertainty (Wayner, 2000).

In addition to AT&T which did not hear about the announcement of BSD versions, a Finnish student Linus Torvalds had not also heard about it and had to reinvent the wheel. He needed an operating system as cheap as DOS but as functional as UNIX. But he could only afford a PC with a 386 processor²⁵. Linus looked at Tanenbaum's Minix which was another version of UNIX. It was developed by Tanenbaum for students education. But it was again too expensive, \$150 (Torvalds and Diamond,

²⁵ Indeed, he also could not pay monthly payments of his PC and Linux users collected money for him.

2001: 60-64).

So he started writing his own operating system for the 386. GNU had provided a wide variety of tools which he could use freely. Especially, GNU's biggest programming project GNU C Compiler (GCC) was an important tool for Linus. GCC converts the human readable code, which is produced by C programming language, to machine code which is understood by computers. GCC was a cornerstone for the GNU project. First of all, it was one of the best compilers. Second, it was portable, can be moved through different platforms. Third, the GCC was free and a poor student like Linus could use it in his project (Wayner, 2000).

On one hand, it was an unambitious project, such that it did not aim to compete against AT&T or Microsoft. On the other hand, it was an ambitious project whose creator wanted it to have many features of UNIX versions on the market. In fact, as Linus admits, he was writing just a toy and writing an operating system was a fun experiment for him (Raymond, 1999a).

In this sense, Linus's decision to create an operating system was not extraordinary. There were lots of brilliant students who did brilliant things. But Linus's real talent was not what he did but how he organized the project. In GNU Manifesto, it was written that the kernel would require closer communication and would be worked on by a small, tight group²⁶. On the contrary to GNU's kernel development, Linux kernel is developed with a large group. In the Manifesto, closer communication presupposes a small and tight group. However, increasing number of Internet users, 7 years after the GNU Manifesto, made project organization in a large group possible and surprisingly better. Linux project used the entire world as its talent pool as a consequence of cheap Internet. Linus play well by the new rules that pervasive Internet access made possible (Raymond, 1999b). According to Torvalds (1999), the power of Linux is as much about the community of cooperation behind it as the code itself. In this respect, cheap Internet was a necessary condition for Linux-style development. Yet, it was not sufficient. Success of Linux also highly owes to its modular design. Otherwise, parallel work on the kernel would be impossible (Torvalds, 1999).

Developers' organization and modular design are issues about programming.

²⁶ Retrieved December 5, 2005, from <http://www.gnu.org/gnu/manifesto.html>

They were necessary but one of the most important fact about Linux is "treating your users as co-developers" (Raymond, 1999b). In other words, instead of separation between producer and consumer, a new concept emerged on Internet, prosumer. Linux's development strategy was simple: release early and often, delegate everything you can, be open to the submissions from any prosumer(Raymond, 1999b).

This strategy worked well. In 1992, GNU/Linux was almost ready for prosumers as a free operating system. Moreover, in the second half of the 1990s, Linux became a real alternative for commercial UNIX systems and Microsoft's Windows family.

2.3.4 New Economy and Open Source Initiative

The meaning of the word, *improve*, does not mean just 'make better'. It also means "to do something for monetary profit"(Wood, 1999: 80), especially cultivate land for profit. As emphasized by Wood (1999), in the early period of capitalism, productivity and profit²⁷ were inextricably connected in the concept of improvement.

In late seventeenth century, these new conceptions of property were being conceptualized by John Locke. His arguments were based on the notion of improvement. Theme of his writings could be summarized as the earth was here to be made productive and profitable which could be only reached by private property (Wood, 1999).

According to Locke (1952), unimproved land was waste and any man who took it out of common ownership and made it his own private property had given something to humanity. In other words, exclusion of others from land did not mean taking something away from humanity, instead giving them something. In Locke's writings property was not for consumption of its proprietor but increasing labor productivity, output per hour.

Private property and competition (which is strongly related to private property) have been necessary conditions in commodity markets for the emergence and development of capitalism (Wood, 1999). However, in 1984, Stallman (1984a) refused the spread of private property and exclusive rights on software:

The paradigm of competition is a race: by rewarding the winner, we en-

²⁷ Based on the old French for 'into', en, and 'profit', 'pros'-or its oblique case, preu

courage everyone to run faster. When capitalism really works this way, it does a good job; but its defenders are wrong in assuming it always works this way. If the runners forget why the reward is offered and become intent on winning, no matter how, they may find other strategies—such as, attacking other runners. If the runners get into a fist fight, they will all finish late.

Proprietary and secret software is the moral equivalent of runners in a fist fight. Sad to say, the only referee we've got does not seem to object to fights; he just regulates them ("For every ten yards you run, you can fire one shot"). He really ought to break them up, and penalize runners for even trying to fight.

In addition, he claimed that proprietary software social system was highly anti-social and forbidding cooperation of community:

The rule made by the owners of proprietary software was, "If you share with your neighbor, you are a pirate. If you want any changes, beg us to make them." (Stallman, 1984a)

He was defining his goal as building a system where people were free to decide their own actions, particularly free to help their neighbors and "improve the tools they use in their daily life" (Stallman, 1984a).

Moreover, according to Stallman (2002), the term "Intellectual Property Rights" was misleading since it was presupposing what you talk about was property.

In fact, all of the participants of GNU project were not acting completely because of a moral choice. Some of them contribute to the project because of their daily job. Some call it fun. Some of them contribute for reputation. Some of them contribute for future careers (Wayner, 2000). It was a heterogeneous society working for the success of GNU project, at least side effect of their work was helping the success of project. But as mentioned above, despite the development of surprisingly good software, FSM was conflicting with the bases of capitalism: private property and competition. Despite FSM has not been the only social movement rejecting the norms of capitalist society, what made it different from the other social movements is, in the second half of the 1990s, this "pseudo-communist" (Wayner, 2000: 5) rejection became an alternative business strategy. Hackers created better free software than the corporations' proprietary software.

Wide-open cooperation also turned out to be wide-open competition because the best software won the greatest attention (Ibid: 13).

On the other hand, second half of the 1990s was also extraordinary for US economy which conditioned the rise of open source business strategy. US economy's performance astonished many economists. After a period of stagnation which began in 1970s, US economy has experienced an unexpected resurgence of growth. Moreover, inflation and unemployment rates dropped to historically low rates. Most of the economists suggested that investment in and use of information technologies (IT) played the most important role in bringing about that situation. US economy was evolving into a high technology-based economy, arising largely from new developments in the information and communication technologies (Bosworth and Triplet, 2000). The Internet diffused from the universities and research laboratories to society at large. Business firms were behind this diffusion of the Internet. As business transformed the Internet, the Internet also transformed the business. More importantly, hackers were transforming into entrepreneurs in this process (Castells, 2001a). And FOSS has:

reached the stage where an entire generation of students who learned computer science under the influence of GNU is now at work in industry, and have quietly been bringing free software in through the back doors of industry for years. They do so not from altruistic motives, but rather to bring better code to their work (DiBona et al., 1999).

As Castells (2001a) remarks entrepreneur culture was a culture of money. Even though hacker culture was still contained in it, free software movement was not still acceptable by the business world:

...radical message (the freedom part, not the beer part) led many software companies to reject free software outright. After all, they are in the business of making money, not adding to our body of knowledge. For Stallman, this rift between the computer industry and computer science was acceptable, maybe even desirable (DiBona et al., 1999).

Because of this, even publishing a technical book, GNU Emacs Manual, was difficult for Stallman. It was hard to find a publisher who would publish a book, which encouraged people to copy and share. Thus, Stallman published the book himself (Tiemann, 1999). It did not matter for a purist like Stallman. For him, ethical values were more important than technical matters. Yet, freedom and ethical values of FSM have been Achilles heel for the hackers of business world. Microsoft's CEO

Steve Ballmer claimed that GNU/Linux was cancer for Intellectual Property Rights²⁸ (Moody, 2002).

To overcome this problem, in 1997, leaders of FOSS world, including Eric Raymond, and Tim O'Reilly, met in California (DiBona et al., 1999). They thought that FSM's anti-business message leaded by Stallman was preventing free software's popularity. Consequently, it resulted in a new term for the free software: Open Source Software. They left the "freedom" of FOSS and their main arguments for supporting GNU/Linux were technical arguments and its profitability.

Success of FOSS companies, Red Hat and VA Linux, took attention to FOSS business model in 1999 coinciding with the success of the New Economy. But making money from FOSS was not new. For instance, Stallman himself was one of the first people who had earned money from FOSS by selling EMACS tapes or for providing consultancy for his programs. Furthermore, FSF is still sponsoring GNU project in this way.

On the other hand, Tieman, one of the founders of Cygnus Solutions, saw the potential of free software as a business strategy and put it into practice in 1989 (Moody, 2002). While Stallman's GNU Manifesto was seemed as a socialist polemic for some, for Tieman it was a hopeful business plan. According to Tieman basic idea of GNU manifesto was:

Open Source would unify the efforts of programmers around the world, and companies that provided commercial services (customizations, enhancements, bug fixes, support) based on that software could capitalize on the economies of scale and broad appeal of this new kind of software (Tiemann, 1999).

Cygnus Solutions' was based on a service-based business model. They were providing technical support for free software. In other words, they were changing the rule of hackers, "from engineers according to their ability, to engineers according to their need" (Moody, 2002: 224). New rule was, "from engineers to users if they pay". They pioneered the latter open source business models.

In 1990s, after the emergence of Linux, Slackware, Red Hat and SuSE²⁹, followed

²⁸ In 2000s, Microsoft strategy was on property rights rather than technical advantages of its products.

²⁹ These are well-known GNU/Linux distributions. Concept of distribution is explained in the

Cygnus Solutions. For example, founder of Red Hat, Eving, was aiming at providing solutions for users who were less technically able. Red Hat tried to achieve what Microsoft did in PC revolution: Making FOSS for everyone. So installation of Red Hat GNU/Linux distributions were easier than most of the other GNU/Linux distributions (Young, 1999).

Red Hat won the confidence of not only the users, but also big companies like Oracle, IBM and HP. However, the most critical point was their relationship with the hackers. Red Hat had understood that it had to support hackers' world (DiBona et al., 1999). The main asset for open source business model is the people who involved in the projects. Making money, without giving anything would cause problems and the people would consider it freeloader (DiBona et al., 1999). Moreover, new hackers were not much interested in the ethical issues as the hackers of the free software movement, but some of the norms, beliefs and values of hackers were still essential.

Red Hat, Cygnus, Slackware, Sendmail etc. founded from bottom. They made money from ideas rather and did not invest much money. If they lost in their business strategy, they would lose just their dreams (Castells, 2001a). They were pragmatist, wanted to earn money. Hacker's ethic toward work was different from Weber's Protestant Ethic. In the Protestant Ethic, which mediated capitalist society, work and money were ends in themselves. On the other hand, in this new work ethic, there is a passionate relationship between the hacker (worker) and the work. Hackers are not blind to the facts of capitalism. But while money is important, desire to create something is also important for hacker ethic. For this reason, while there is a contradiction between work and leisure in Protestant Ethic, in hacker ethic work is leisure in itself (Himanen, 2001). Otherwise, Torvalds would not add Virtual Memory support for his Linux kernel in a Christmas break as the other hackers who work on holidays or nights after work (Moody, 2002).

Indeed, increasing commercialization changed the dynamics of FOSS (Moody, 2002). New actors became parts of FOSS.

While the practice of open source business model by young companies took at-

glossary.

tention of business world, more surprising moment was when old companies decided to free their source. Netscape announced to give away the source code of their browser, Netscape Communicator in 1998 (Hamerly et al., 1999). At the same time, IBM replaced its own web server with one of the most popular FOSS, Apache Web Server³⁰ (Moody, 2002), in other words returned to FOSS world after SHARE. Lastly, Microsoft announced its Shared Source License in 2001. Microsoft's new license was complementary to the Halloween Documents, and FOSS was not completely an enemy anymore³¹:

Microsoft has been learning from the OSS community regarding the benefits of deeper collaboration and increased transparency leading to better communication with customers. We believe the most effective pathway for a commercial software company is to strike a balance between investing in research and development and the release of intellectual property assets in the form of source code for both reference and collaborative purposes.

The increased competition resulting from the proliferation of OSS has been constructive for the industry as a whole. The implications of OSS within multiple market segments are causing organizations to figure out what is most important to them. It has placed a higher premium on innovation and a drive to deliver greater value for lower costs. The big winner in this equation has been the software consumer, whose choices have increased dramatically.

Those companies started new FOSS projects. As a consequence, new property relationships emerged that revealed themselves on licenses. In this aspect, FOSS firms adopted commercial models and sometimes never hesitated to include proprietary software in their products. That was an unacceptable behavior for FSM. Commercial firms adopted open source business models in their strategy (Smith, 2002). New licenses emerged as a renewed interrelation between FOSS and capitalism. For this reason, the only FOSS license that Microsoft can not accept is GPL, which forces the freedom of user against proprietary software in any way (Lessig, 2002).

Contradiction between Free and Open Source software is the basic point for comprehension of FOSS's historical movement. This forking in the FOSS was the assimilation of it by capitalism. Whereas the free software emphasizes the freedom, open

³⁰ More than 50% of web servers in the world are Apache.

³¹ Retrieved December 5, 2005, from <http://www.microsoft.com/resources/sharedsource/Articles/MicrosoftandOpenSource.mspx>

source does not care about it. According to open source advocates, the word freedom is harmful for making FOSS compatible with the business people's thinking (DiBona et al., 1999). For open source advocates, "openness" means speed and more profit in other words, "Free Software for Business" (Richardson, 2001).

Stallman (2002) contrasts this forking in community with the splits in 1960s radical groups. At that time organizations splitted because of details of strategy and it was not rare that after the split, they saw each others as enemy. On the other hand, Stallman (2002) emphasizes that relation between FSM and open source advocates is opposite of this. They disagree on basic principles but agree on most of the practical recommendations. In this sense, they are not enemy camps and work together on many projects. Popularity is very important for open source advocates. Thus, open source licenses are less restrictive than GNU GPL. For example, GNU GPL'ed software forces the user to release the newly developed software again in GNU GPL . In addition, GNU GPL does not give permission to link with proprietary software. This policy had negative impacts on the popularity of FOSS. Stallman (2002) explains it as freedom for sake of popularity. On the other hand, open licenses do not have those restrictions.

However, some problems occur in FOSS world. For example, Linux kernel uses a non-free version control software. Torvalds prefers BitKeeper's software instead of a free version control software as CVS. Torvalds preferred it because it was technically better. Stallman criticizes Linus for devaluing freedom. Although Bitkeeper is proprietary software company, they give their software without money for FOSS developers. But if FOSS developers use it as a version control in their project, none of them can participate in development of a competing version control software (Williams, 2002).

Another dispute was about KDE, a popular GNU/Linux desktop environment. When Trolltech firm decided to use proprietary in KDE, FSM was the opponent. For the freedom of user, they started GNOME Project, another desktop environment. Red Hat also supported this project. After this, Trolltech had to step back.

Open source succeeded in persuading business world and made FOSS more popular. Nowadays, most of the developers and trade press are using the term open

source software more than free software³². Furthermore, the term, free software, is in long-term decline. But, FSM is still a radical group in FOSS world whose actions are based on freedom rather than the slogan, "technically better", of open source. FSM is playing decisive roles in critical moments despite open source advocates' hesitancy because of its increasing relationship with business world.

Indeed, new hacker ethic, which is more pragmatist than the purist ethic of FSM, is strongly related with the venture capitalists. New hacker ethic does not insist on all software should be free and in business, if it is required, proprietary software is acceptable. On the other hand, purists believe that all software should be free and proprietary software is only acceptable if it is a mean for making all software free.

2.3.5 Governments and Intellectual Property Rights

Germany, Singapore, China started to migrate their information systems from proprietary software to FOSS. In Germany, Bundestag mandated GNU/Linux on servers and Windows on desktops. According to German-IBM agreement there were discount on machines with pre-installed SuSe Linux. Singapore government offered tax reductions and funding GNU/Linux related projects. Legislation in Brazil mandated FOSS given preference in municipal governments of Recife, Campinas, Solonopole, Amparo, Sao Carlos and Porto Allegre. There are pending legislations in other countries. In French Government, there is a bill forbidding use of proprietary software in government related institutions. In Italy, there are bills mandating preference of FOSS in all governmental offices. Indeed, European Parliament is advising to use FOSS whenever possible (Hahn, 2002). Furthermore, China which has a different socio-economic structure is rejecting proprietary software solutions and supporting national software industry which is based on FOSS (Bessen, 2002).

On the other hand, a growing debate is taking place in Europe about software patents. Although emergence of debate is not new, it is evident that software industry is in a turning point. Possible result of this debate will not only effect on European Union's software industry, but also the rest of the world. For this reason,

³² Terminology wars: Retrieved December 5, 2005, from <http://www.catb.org/esr/writings/terminology/>

it is not surprising that United States (US) warned European Union (EU) Parliament about software patents. Since it is part of US Government's "Action Plan" which aims to strengthen the rights of American intellectual property holders³³. Microsoft threatened Danish Government to move 800 jobs from Denmark to US if they continued to oppose the software patent directive. Indeed, Microsoft CEO of Brazil, Emílio Umeoka, did similar in Brazil³⁴. While Microsoft's Chairman, Bill Gates, attended World Economic Forum at Davos on January 2005, FOSS was urged for poor nations at the World Social Forum³⁵. IBM, which is on the side of FOSS, is supporting software patents although hackers are against it. And more interestingly, IBM license their software patents loyalty-free to FOSS developers.

In addition to the patents, another risk for the future of FOSS is the trusted computing. Trusted computing is advocated by the Trusting Computing Group³⁶. The group defines its role as "developing, defining, and promoting open standards for trusted computing that will benefit users"³⁷. According to them, trusted computing will increase computer security. This will be done by a chip integrated into the hardware. On the other hand, opponents of the trusted computing claim that trusted computing power will just increase big companies power on users. Moreover, these companies will be able to control the software in user's computer. Thus, users may be forced to use proprietary software rather than FOSS:

The biggest profits in IT goods and services markets tend to go to companies that can establish platforms and control compatibility with them, so as to manage the markets in complementary products.

...

So a successful TC (Trusted Computing) application will be worth much more money to the software company that controls it, as they can rent out access to their interfaces for whatever the market will bear. So most software developers will enable their applications for TC; and if Windows is the first operating system to support TC, it in turn will get a further

³³ Retrieved December 5, 2005, from <http://swpat.ffii.org/papers/eubsa-swpat0202/usrep0309/index.en.html>

³⁴ Retrieved December 5, 2005, from <http://wiki.ffii.org/Navision050215En>

³⁵ Retrieved December 5, 2005, from http://seattletimes.nwsourc.com/html/nationworld/2002164876_wdig30.html

³⁶ Retrieved December 5, 2005, from <https://www.trustedcomputinggroup.org/>

³⁷ Retrieved December 5, 2005, from <https://www.trustedcomputinggroup.org/about/>

competitive advantage over GNU/Linux and MacOS with the developer community³⁸.

In this context, it is not surprising that Microsoft is a member of the Trusting Computing Group. Yet, companies which support FOSS are also members of the group: IBM, HP and Sun.

After the contribution of companies and transformation of hackers into entrepreneurs, governments contributed to the FOSS. In this new moment of FOSS, there are two interrelated processes which condition the FOSS. First is some governments' preference for FOSS over proprietary software. Second is the "trend toward harmonization or standardization of intellectual property laws, in the direction of greater protection" (Nayyer, 2001).

Yet, Evans (2002) argues that governments should not prefer one (open source software) model to another (proprietary software) model. According to him, software industry had performed well with little government interference and there is not any failure in the market which makes intervention of governments useless. But he misses two points.

First of all, there was a little sign of change in European countries between 1995 and 2000. There was not any new economy as in US. In its "New Economy" survey, *The Economist*³⁹ claimed that America's leading in IT sector would give it a big advantage and present formidable barriers for European firms. There were only four European firms⁴⁰ among the 50 biggest companies of the world. Additionally, in the same article it was emphasized that the more important point was use of IT not making of IT products. In the same survey, but in another article⁴¹, *The Economist's* advice for developing countries was the same. It would be very hard for them to compete with the American companies which had become dominant using the first mover advantage. Developing world should not reinvent the wheel and should open its economies to new ideas. In this context, even though there was not any failure in

³⁸ Retrieved December 5, 2005, from <http://www.againsttcpa.com/tcpa-faq-en.html>

³⁹ Catch up if you can, *The New Economy, A Survey*, September.

⁴⁰ There are 36 American and 9 Japanese firms.

⁴¹ Falling through the net, *The New Economy, A Survey*, September.

US software industry, there was failure in other countries' software industry or there was not any domestic software industry at all. For this reason, China rejected proprietary solutions for cultivating its domestic software industry. France also supports FOSS for encouraging the growth of small and medium-sized software companies (Bessen, 2002).

Secondly, Evans (2002) considers governments as private actors whose decisions are about maximizing efficiency or profit. However, Lessig (2002) criticizes the approach of Evans, since "the factors that define efficiency of governments are fundamentally different from the factors that define efficiency of private factors". Beyond cultivating domestic software industry or reducing costs, interoperability and security are important factors for governments.

Interoperability means "the capability of different programs to read and write the same file formats and utilize the same protocols"⁴². If a vendor's product is not interoperable, this may result in vendor-lock-in. For example, if you write documents in a proprietary format of a vendor, you will have to use the products of same vendor in the future. For this reason, if a government constitutes its eGovernment process, the transformation of public sector's internal and external relationships through Net-enabled operations, on a proprietary software, this may result in a vendor's monopoly over eGovernment.

FOSS is believed to be less vulnerable because of the availability of source code. This does not mean that it has less bugs than proprietary software. However, bugs are more easily discovered and corrected. Moreover, non-availability of source code may make backdoors invisible in computer systems. By the help of backdoors, an unauthenticated user may transfer your private data. Because of this, China says that dependence on Microsoft is like leaving the keys to the country's increasingly computerized economy in the hands of a potential enemy⁴³.

However, while most of the governments in EU started to migrate their information systems to FOSS, there is also growing debate about software patents. As

⁴² Retrieved December 5, 2005, from <http://en.wikipedia.org/wiki/Interoperability>

⁴³ Retrieved December 5, 2005, from <http://list.wylug.org.uk/pipermail/wylug-advocacy/2000-July/000016.html>

offered by the writers of Halloween Documents (Raymond, 2004) , Microsoft put a strategy which is based on the software patents. Microsoft also had some success in this strategy. For example, Munich paused its Linux migration project because of possible risks of software patents. Since legal problems about FOSS would arise and there would be lawsuits about patent infringements. Uncertainty about the future of software patents was an important risk. Munich mayor Christian Ude told that project was not stopped, but they need risk assessment before the proceeding of project. Impact of this decision may not effect only Munich because it was a leading project for other cities of Europe⁴⁴.

The pause of migration projects may reverse the rise of FOSS in the world. Until recent time, European Patent Convention defines patents for inventions as a set of exclusive rights granted by a government to an inventor for a limited amount of time. For patentability, three requirements are necessary:

- to be new
- to involve an inventive step
- to be capable of industrial application

Yet, what is problematic in the context of the software patents is not requirements but the exclusions. In European Union countries, an invention is not patentable if it is⁴⁵:

- a discovery
- a scientific theory or mathematical method
- an aesthetic creation
- the presentation of information such as computer program

⁴⁴ Munich Reconsiders Linux Migration, Retrieved December 5, 2005, from <http://www.eweek.com/article2/0,1759,1631252,00.asp>

⁴⁵ European Patent Convention, Retrieved December 5, 2005, from <http://www.european-patent-office.org/legal/epc/e/ma1.html#CVN>

Patents are territorial. For this reason, EU and US have different styles of patentability. US-style patentability does not have the same restrictions of EU. Consequently, while algorithms and business methods are patentable in US, in EU they are not. For this reason, a growing debate is taking place in Europe about software patents. It is evident that not only FOSS but also the software industry is in a turning point. Possible result of this debate will effect international software industry. Moreover, US wants to preserve its first-mover advantage by blocking the governments possible strategies as migrating the systems to FOSS. In this context, it is not astonishing that, Germany and particularly France are the most important opponents of software patents. But one of the most interesting points about software patents is demonstrations of hackers in the streets. Indeed, software patents reveal the main contradictions between hackers and companies.

CHAPTER 3

General Outlook of FOSS In Turkey

3.1 Research Questions and Methods

3.1.1 Questions

In this study, goal of the research was to gain deeper understanding on the relationship between the hackers and the firms in Turkey from the perspective of their conflicting interests. In order to do this, the research focused on two questions:

- To what extent the conflicting interests between the hackers and the firms affect the development of FOSS business models in Turkey?
- What should (or should not) a firm do in order to benefit from voluntary contribution of the hackers?

However, as stated in the previous chapter, this study diverges from most of the studies in the FOSS literature since it does not consider hackers' voluntary work as an anomaly. Instead, it states that under certain social circumstances one may also prefer work. Then, the research should focus on the certain social circumstances that make hackers prefer work to leisure. In this sense, before answering the above questions, there are four more complementary questions to be answered:

1. What kind of FOSS works (or projects) do hackers engage in?
2. What are the differences between hacking and non-hacking activities of the hackers?
3. What are the differences between the hackers and non-hackers in the context of their relationship with their work?

4. What are the differences between the Turkish hackers and the other (European or American) hackers?

There is no doubt that hackers are not interested in all FOSS projects. Experience of the Netscape's web browser is a good example for this situation. At first, Netscape was a proprietary web browser which was losing its market share against Microsoft's Internet Explorer. Thus, Netscape company decided to release its web browser as a FOSS product. Contributions from outside developers started, but these contributions faded away and hackers drifted to another projects (Weber, 2004). Thus, a FOSS firm should know constraints and boundaries of working with the hackers either inside or outside the firm. First three questions investigate this subject.

On the other hand, Turkish hackers are different from the US and European hackers. There is a huge literature that studies hackers' common traits. Levy's book, *Hackers, Heroes of The Computer Revolution*, is one the earliest studies. According to Levy (2001), hackers have some principles and acknowledgments: access to computers should be unlimited, all information should be free, mistrust authority and promote decentralization, hackers should be judged by their hacking, not bogus criteria such as degrees, age, race and position, one can create art and beauty on the computers, computers can change people's life for the better.

However, development of the hackerdom happened in different social circumstances in each of the countries. In this context, there may be specific traits of the hackers in each of the countries as well as their common traits.

3.1.2 Methods

In the study, qualitative research methods, interviews and participant observation are used. The interviews are conducted from February, 2005 to March 2006 and observations include the time between the September, 2004 to April 2006.

Interviewees can be categorized into three groups each with its different sets of questions. In the first group, there are FOSS hackers and I conducted 66 interviews with them. In the second group, there are 12 managers who work for a FOSS firm. Most of these managers were either owner of the firm or have share in the firm. Furthermore, there was one freelance consultant in addition to those managers. Last

category consists of 14 software engineers who are not involved any FOSS activity as a hacker or a businessman.

Second research method, participant observation, includes many activities, from using/testing GNU/Linux distributions, participating discussions in the mailing lists to attending festivals and LKD's work groups. These activities gave me important insights about the hackers that I could never have by the interviews.

The research performed in six consecutive steps.

First Step: I subscribed to all LKD e-mail discussion lists¹. The discussions in those lists were best resources of information on the development of the FOSS in Turkey. There has been a huge e-mail archive since December 1993. These lists' archive helped me how the general tendencies of hackers had changed since 1990s. It was also good in figuring out that how today's businessmen were considering FOSS when they were amateur hackers. As I read the discussions of hackers from the mailing lists, I also followed forums and web sites which were popular between the hackers. In fact, I learned their popularity from the mailing lists. Among these websites, FM (fazlamesai.net) is the most popular and it is a portal of Turkish hackers, as slashdot.org.

Second Step: I recognized that for understanding them well, I should also use FOSS. Otherwise, it would be hard to understand hackers' jargon. I had given up using FOSS after graduation and because of my job I was using MS applications. So I installed GNU/Linux in addition to the Windows XP on my home computer. This helped me in many ways. First of all, it helped me to understand that technical discussions were more than it seemed. For example, when a user asked the list members, "Which Linux distribution should I use?", long discussions were happening. Everybody was proud of his distribution. I installed all well-known GNU/Linux distributions Mandrake², Red Hat³, Slackware, Gentoo, Debian and lastly beta version of Pardus. I used each of them for a while to understand their social back-

¹ see <http://liste.linux.org.tr/>

² One year after my experience it became Mandriva.

³ Six month after my experience it became Fedora.

ground. Secondly, as I used GNU/Linux, it was clear for me that why hackers prefer GNU/Linux rather than Windows. For example, one has ten files whose names are 1.mp3,2.mp3,3.mp3 etc. If extensions of these files are wanted to be changed from mp3 to MP3, a windows user has to rename each file since there is no alternative. However, in GNU/Linux, one can write a small script and change hundreds of file names in seconds and that is hacking. In this sense, GNU/Linux gives power to hackers which make them hackers as mentioned in Raymond's book, *The Art Of Unix Programming* (Raymond, 2003).

Third Step: After following discussions, I thought that if I had participated in a FOSS project, I would have learned more about hackers. Thus, I joined a project. But while I was looking for a suitable project, I saw that there were only small projects, consisting of one or two men, in Turkey. I made a small contribution to the project that I joined. Latter, project manager gave me another task. But I realized that they were using proprietary software for compressions which was wrong according to free software. Thus. I left the project. But this experience, became one of my questions in the interviews: if they experienced the same problem, what would they do. Whether it would be really a problem for them.

In this step, I also searched on the international magazines, forums, mailing lists and usenet groups. In this process, I followed the international debate about the commercialization of FOSS among hackers.

Fourth Step: The turning point in my study was when I realized that programmers were not the only people who added value to FOSS. There were also many people who added value to FOSS by giving seminars, finding bugs of software, internationalizing FOSS, helping others in discussion lists. Thus, I decided to consider hackers in a wider spectrum and included them in the research.

However, my research on e-mail discussion lists can not be named as lurking. Since lurking implies reading messages rather than participating discussions. On the contrary to lurking, I participated in discussions, posted questions/answers about technical topics, and sometimes triggered discussion of subjects about social sides of FOSS to learn what the members of the mailing lists thought.

Fifth Step: After following discussions lists in Turkey and in the world, I passed

another step of the research, interviews. At first, I planned interviewing with hackers and firms.

I almost asked the same questions for hackers and firms⁴ (Appendix B). Focus of the interviews were how they perceive FOSS, their practice and awareness about the licenses. Some questions were asked again in different words to determine if they really well-informed about the FOSS. For example, I asked if they prefer using either Free Software or Open Source Software. After this, I asked if they preferred GPL⁵ or BSD⁶ licenses. Additionally, as Bonaccorsi and Rossi (2003, 2004, 2005), I investigated the discrepancies between the attitudes and behaviors of the firms.

Participants of the interviews were not selected randomly. Instead, active hackers in the mailing lists and hackers who have FOSS projects in the web page⁷ of LKD are selected. Furthermore, I attended activities such as conferences, festivals, LKD members' meetings. In these activities, some hackers and firms did not want to talk to tape recorder, instead they said "I will e-mail answers to you.". As a result, some of the interviews happened face-to-face, some of them by e-mail and latter by ICQ.

Another source of information about the hackers was their weblogs⁸. From these weblogs, one can learn a lot of thing about everyday life of the hackers⁹.

Sixth Step: During the interviews, I recognized that only a small number of interviewees were software engineers. However, in the US and European researches (Ghosh et al., 2002), there were significant number of computer scientists, software engineers, computers science and software engineering students. So, I decided to interview with software engineers and searched on their discussion mailing lists. I made informal interviews with software engineers about hackers, programming and future of FOSS in Turkey and in the world. Yet, when I met messages about FOSS

⁴ Some questions were not applicable, so I changed them during interview.

⁵ It is Free Software license.

⁶ It is the most widespread Open Source License.

⁷ <http://www.linux.org.tr/index.php?Pg=LinuxYerliYazilimlar>

⁸ A website that displays in chronological order the postings by one or more individuals and usually has links to comments on specific postings. src: The Free Dictionary.

⁹ see <http://gezegen.linux.org.tr/>

in software engineering discussion lists, I decided to make formal interviews (Appendix B), mostly investigating if they consider FOSS as a (or a part of) business strategy.

Beyond this, I gave seminars about FOSS to people who had never heard about it. These seminars were useful since I had the opportunity to observe FOSS from different perspectives of different professionals, e.g software engineers, electric engineers and social scientists.

3.2 Main Actors and Organizations

First days of 2005 was a critical moment for FOSS in Turkey as the emergence of GNU Manifesto in 1984. It was critical in two aspects. First of all, it revealed the contradictions of government's IT policy. Secondly, FOSS advocates turned into political actors.

On 31 January 2005, Bill Gates visited Turkey after accepting an invitation by the Turkish Prime Minister, Recep Tayyip Erdoğan. Gates attended a conference about the developments of IT in Turkey¹⁰. He promised to give support for Turkish IT projects. There was nothing extraordinary in his speech, but Erdoğan's speech was interesting. He told about security of information and asked how one could trust an information system which was not developed domestically. Furthermore, he said that any chosen operating system should be inspected in detail because of security reasons. Another interesting point about Erdoğan's speech was his emphasize on open standards and file formats, in other words interoperability.

However, most of the governments prefer FOSS rather than proprietary software because of national security and interoperability problems of proprietary software. Indeed, Microsoft had to defend itself on courts (both in Europe and US) because of interoperability problems of its products.

But the most fascinating point was the release of the Turkish GNU/Linux distribution, Pardus, in the same days. Pardus was the product of Uludağ¹¹ Project. Although Pardus was not the first Turkish GNU/Linux distribution, it was the only one

¹⁰ Retrieved December 5, 2005, from <http://www.milliyet.com.tr/2005/01/31/ekonomi/axeko01.html>

¹¹ National Distribution, in Turkish, Ulusal Dağıtım

that was directly funded by Turkish Government. Moreover, Uludağ Project introduced an alternative for people who were dissatisfied with Bill Gates's visit. Project management team declared the possible ways of contribution (fixing bugs, testing, localization) to the project. Consequently, messages were sent to Uludağ Project's discussion lists by wide spectrum of people including IT professionals, artists, students, lawyers. Contents of the messages were simply, "I am happy with a national operating system. I want to contribute to the project."

On the other hand, the events before and after the visit of Bill Gates showed that FOSS in Turkey had reached a critical mass and qualitative transformation was occurring. After this visit, OSP (Open Source Platform)¹² was built and it published a declaration. Main theme of the declaration was showing that FOSS was a real alternative for national software policy. As a naming convention, "Open Source Software" was preferred rather than "Free Software" following the trend in the world.

OSP, leaded by LKD¹³, did not publish any additional declaration and did not play any role after its declaration. Yet, it was the first concrete and organized reaction of FOSS advocates in Turkey.

In this part, main actors and organizations of FOSS in Turkey are introduced. Among these organizations, LKD is the essential one in the development of FOSS in Turkey. In the last section, FOSS activities, including projects and festivals, in Turkey are presented.

3.2.1 LKD

LKD¹⁴'s history goes back to 1992. In 1993, Turkish Linux users started to organize on a mailing list, linux@bilkent.edu.tr. Their first face-to-face meeting happened at the conference, "Internet in Turkey" in 1995. From 1995 to 2000, number of Linux users grew on mailing lists and in 2000, Linux users started to discuss about having a juridical statue. Source of discussion was problems of being a mere virtual organi-

¹² Retrieved December 5, 2005, from <http://www.acik-kaynak.org.tr/>

¹³ <http://www.lkd.org.tr>

¹⁴ Information in this part is gathered from the web sites of LKD (<http://www.linux.org.tr> and <http://www.lkd.org.tr>), the mailing list archives (<http://liste.linux.org.tr/>) and interviews.

zation.

One of the LKD's activists explained the necessity in having a juridical statue:

When you want to organize an activity, such as a festival or a seminar , you should overcome some legal restrictions. For instance, you should take permission from concerned authorities. Yet, more importantly, we live in a real world. Go to a firm, introduce your self as a member of Linux users' mailing list and ask for sponsorship. None of the firms will take your demand seriously. OK, be optimistic; assume that your demand is accepted. But again, there is a problem. For their accounting records, they need a juridical organization.

LKD was established in November 2001 and had its first general committee. Even though LKD is organized around GNU/Linux, it tries to be an umbrella organization for all FOSS. It does not have a physical meeting place and any strict rules for membership. Indeed, there are not any significant differences between people who have membership in LKD and people who are merely subscribers of LKD's mailing lists. Number of e-mails that are distributed on LKD's e-mail discussion lists is about 1,000,000 and its servers have about 27,000 visitors in each month. In other words, LKD is still a virtual organization.

LKD's activities are organized through work groups which are the essential part of its organization. In each work group, there is a member of board of directors who is responsible for the communication between the work group members and the board of directors. In my research, I also became an active member of one of these groups and observed their activities. The most important points about these work groups are, first they entirely depend on voluntary contribution of the hackers as FOSS projects and members of the work groups have significant amount of initiative in the works. Consequently, the common point of members of successful work groups is their high self-confidence. Some of the work groups are:

Seminar Work Group: It is responsible for organization of LKD's permanent seminars (These are generally in Istanbul and Ankara) and LKD's seminars at festivals. Additionally, it organizes seminars if there is demand from anywhere (Sivas, Adana, Samsun e.g.) in Turkey .

Web Site Work Group: It is responsible for organization of the web site, <http://www.linux.org.tr>.

e-Magazine Work Group: It is responsible for publishing LKD's magazine, Penguence¹⁵, periodically.

Penguen Work Group: It is responsible for organization of new members and establishing relationship with them.

Festival Work Group: It is responsible for organization of LKD's *Linux and Free Software Festival* which is discussed in the next part in detailed.

Additionally, if there is any need, LKD's activists quickly organize sub work groups. However, LKD's most important success in the last years is its major role in the formation of *Guide for Interoperability in Public*¹⁶ circular which is against the use of proprietary formats in the public sector. In this sense, while LKD's activities increased awareness about FOSS in Turkey, its lobbying activities effected on the formation of Turkey's IT policy.

Furthermore, general trend in the world may not be valid for LKD in near future. Barr (2005) compares past and recent days of the LUGs (Linux User Groups). According to him, in the the early days,

a typical LUG brought together early adopters from every walk of life who had a missionary zeal for Linux (Ibid.).

But the recent LUGs are different:

Today, by and large, the early members are gone. The current membership's system administrators take a more objective view towards Linux and its strengths and weaknesses compared to other operating systems (Ibid.).

As result, as LUGs started to include more, it lost its homogeneity. Barr (2005) asks if LUGs still matter. His answer is ,

LUGs do still matter, but not as much as they did in the early days. They are not the primary drivers of Linux adoption that they once were. Improvements in the ease of installation of modern distributions, Linux's widespread adoption, and its acceptance as an enterprise tool have all combined to lessen the need for what LUGs offer. Today's LUG is less a vibrant beacon of a community of users and more of a professional/social club for admins (Ibid.).

¹⁵ see <http://penguence.linux.org.tr/>

¹⁶ <http://rega.basbakanlik.gov.tr/Eskiler/2005/08/20050805-11.htm>

However, as some of the LKD activists argue LKD is not as European and American LUGs. They say that they are not at LKD for just technology sharing. They want FOSS become Turkey's IT policy. Thus, their activities are not as narrow as LUGs, but one can find some similarities between LKD and FSF. Since both LKD and FSF are political organizations which are dedicated to promoting free software¹⁷. On the other hand, LKD does not have any international connection with any international organization.

3.2.2 Debian Turkey

In Turkey, none of the GNU/Linux distributions has an organization like Debian. In fact, this situation is result of Debian's distinctive characteristic features from other distributions since its emergence. For this, comprehension of Debian Turkey requires comprehension of Debian's emergence and its three foundational documents.

When it was first announced on August 16, 1993, goal of the Debian project stated as creating a non-commercial GNU/Linux distribution that could compete with commercial distributions. It started as a small project of a group of hackers and it grew gradually and became a well organized group of hackers¹⁸.

Debian Project is organized around three documents.

First document is *Debian Social Contract* which consists of Debian's commitments to the hackers. According to this contract, Debian declares:

1. Debian will remain 100% free
2. We will give back to the free software community
3. We will not hide problems
4. Our priorities are our users and free software

Second document is *Debian Free Software Guidelines*. In this document, Debian defines the criteria for free software. Definition of free software is wider in this docu-

¹⁷ LKD promotes open source software, too.

¹⁸ see <http://www.debian.org/doc/manuals/project-history/>

ment than the FSF's definition. Indeed, definition of the open source is based on this document¹⁹.

Last document is the most distinctive side of the Debian, *Debian Constitution for the Debian Project*²⁰. This document is related with the organizational structure for formal decision-making within the Project and relationship between the members (e.g. Project Leaders, Project Members, Project Developers) according to their powers and appointments. In this document, Debian also describes its relation with the SPI (Software Public Interest). SPI is a non-profit organization that "help organizations develop and distribute open hardware and software" ²¹. In this context, Debian and SPI share the same goal, "to create, form and establish an organization to formulate and provide software systems for use by the general public without charge"²².

In brief, Debian is a GNU/Linux distribution whose social sides are more evident than any FOSS distribution. Furthermore, these are the facts that make Turkish Debian users different from other distribution users.

Debian Turkey started as a result of translating English web site of Debian GNU/Linux into Turkish. On December 2000, Debian-tr web site became active. Despite short intervals before 2004, Debian-tr has been active since that date.

On the other hand, despite its popularity and large user base in Turkey, there are only a few contributors to the Debian project in Turkey. In this respect, members of the Debian Turkey can be categorized into two groups.

In the first group, there are hackers who contribute to the Debian Project. Their main work has been localization of Debian for Turkish users. Additionally, as discussed in the next chapter, the most important common point of these hackers is their inclination to the social aspects of FOSS more than any distributions' hackers.

In the second group, there are hackers who only use Debian and do not make any direct contribution to the Debian Project. They are also inclined to social aspects of FOSS. However, for some, non-commerciality of Debian is an essential factor for

¹⁹ see http://en.wikipedia.org/wiki/Open_Source_Definition

²⁰ see <http://www.debian.org/devel/constitution>

²¹ see <http://www.spi-inc.org/about>

²² <http://www.spi-inc.org/goals>

their choice. Additionally, feeling of being a member of a community among this group is one of the most significant points that one can observe from the discussions in the Debian mailing list ²³. Thus, one can also come across e-mails that search for a Debian T-Shirt or plan to go to an activity with the other Debian users.

3.2.3 EnderUNIX

The best description for EnderUNIX ²⁴ is "hacker academy". The introduction passage in their web site affirms this description. They call EnderUNIX as school of developers who learn, teach while learning and produce from the learned subjects. They summarize their activities as developing software, writing or translating documents, giving seminars and supporting open source. Naturally, they give software with FOSS licenses.

The word, *ender* in EnderUnix means unique in Turkish. Their projects are unique, since, they "are selected in such a way that they should be unique, which means that there exists a need for the project."²⁵

They diverge from most of the Turkish FOSS hackers since they prefer BSD operating system rather than GNU/Linux. However, their software also works properly on GNU/Linux, because either BSD and GNU/Linux is UNIX-type operating systems.

Members of the EnderUNIX are compatible with the Raymond's hacker definition that is based on "technical adeptness and a delight in solving problems and overcoming limits" (Raymond, 2001). The most important point in their mailing lists is the technically higher level of discussions than the other Turkish FOSS mailing lists. One can easily observe that the members inclination to technical discussions on the contrary to Debian Turkey's interest for social aspects of FOSS.

Furthermore, they have a social contract similar to the Debian Project. In this document, they declare the responsibilities of its members to the other members.

²³ see <http://lists.debian.org/debian-user-turkish/>

²⁴ Information in this part is gathered from the web site of EnderUNIX (<http://www.enderunix.org/>) and interviews with the EnderUNIX hackers.

²⁵ see <http://www.enderunix.org/>

However, EnderUNIX's social contract is only concerned with its internal relations. In other words, EnderUNIX does not give any promise of social issues as the Debian's Social Contract.

3.2.4 Other Organizations

In Turkey, there also a few hacker organizations which mainly focus on the localization projects. These organizations localize the FOSS projects which are very widespread in the world. For example, there are localization organizations for Linux desktop environments, KDE Turkey²⁶ and Gnome Turkey²⁷. Open Office Turkey²⁸ organization localizes the Open Office which is also a widespread alternative for Microsoft Office. Another popular localization organization, Mozilla Tr²⁹ works on Mozilla products Firefox (web browser), Thunderbird (mail client) and Mozilla Suite (all-in-one Internet application suite). GNU-tr³⁰ localizes GNU software for Turkish users and they make Turkish one of the most localized languages of GNU software.

These organizations are not merely virtual organizations. They sometimes make face-to-face meetings. For example, release of the Firefox was celebrated both in Istanbul and Ankara³¹ parallel to celebrations in the world.

3.3 Works, Festivals and Activities

3.3.1 Works

3.3.1.1 Localization

Each software application does not have to be in English. In fact, computers which compiles everything to ones and zeros do not care the language. However, most of

²⁶ see <http://www.kde.org.tr/>

²⁷ see <http://www.gnome.org.tr/>

²⁸ see <http://www.openoffice.org.tr>

²⁹ see <http://www.mozilla.org.tr>

³⁰ see <http://gnu-tr.sourceforge.net/>

³¹ Celebrations were organized at the bars. Indeed, informal organizations at the bars is very common among Turkish hackers.

the software has been developed by and for American and European people. Thus, it is not uncommon that most of the software applications have English menus, buttons and alerts.

The Localization Industry Standards Association defines localization as: "Localization involves taking a product and making it linguistically and culturally appropriate to the target locale (country/region and language) where it will be used and sold." (LISA, 2003)

In this sense, localization is not a new work. Firms have been localizing software for the other nations for a long time. However, firms' motivation is determined by the market share. In other words, if there is a small number of Turkish users, firm will not be inclined to localize software for them. Since, despite localization is not technically difficult, it requires professional management and financial resources.

In this context, FOSS gives important opportunities for the countries who lag behind the US and EU for use and adoption of IT. These countries do not depend on the firm's motivation about localizing the software. By the availability of source code in the FOSS, they can do it by themselves. As a result, people may adopt FOSS, which is developed for another nation, for their own nation. However, FOSS should be internationalized before being localized. For this reason, developers should abstract

the functionality of a product away from any particular language so that language support can be added back in simply, without worry that language-specific features will pose a problem when the product is localized (LISA, 2003).

In this sense, FOSS, which is internationalized by its developers, becomes suitable for a nation by localization works.

Localization brings some benefits:

- Significantly reduces the amount of training necessary to empower end-users to use a computer system.
- Facilitates the introduction of computer technology in Small and Medium Enterprises (SMEs).
- Opens the way for the development of computer systems for a country's national, provincial and district level administration that will allow civil servants to work entirely in the local language and manage databases of local language names and data.

- Opens the way for the development of computer systems for a country's national, provincial and district level administration that will allow civil servants to work entirely in the local language and manage databases of local language names and data.
- Helps universities train more software engineers.
- Reducing costs and giving better service to citizens (Ibid.).

As Akkuş (2006) emphasizes, localization is inadequately thought as translating software into native language. However, it is a wider work, including overcoming problems of some characters (in Turkish, ğ, ü, ö, ç, İ), keyboard configuration for Turkish, font types, viewing date/time or currency in Turkish formats or adopting of specific applications (e.g accountancy applications) for Turkish.

As discussed in the previous part, hackers localize various FOSS products in voluntary organizations. Furthermore, essential documents of the FOSS are translated into Turkish. In the belgeler.org, a huge archive of FOSS manuals and documents are translated and archived.

Localization works have been one of the essential points in the development of the FOSS in Turkey. Firstly, they make FOSS available for the users who do not know English. Secondly, compatibility and integration problems caused by Turkish characters have been overcome in a larger extent. Consequently, overcoming of these problems convinced firms about use of FOSS commercially.

In this context, one of the essential points about the localizing is hackers relation with their work. In the FOSS literature, people speak of fun in the software development or "scratching an itch" (Raymond, 1999b) as the motivation of developing software. On the other hand, despite its contribution to the FOSS, localizing is considered as a "non sexy work"³² (Bonaccorsi and Rossi, 2003) by some of the hackers because of lack of pleasure in the activity itself.

3.3.1.2 Local GNU/Linux Distributions

GNU/Linux is a highly modular operating system which is composed of interchangeable parts. GNU/Linux borrows this important feature from the design of Unix that is summarized as, "Write simple parts connected by clean interfaces" (Raymond,

³² It is used for denoting jobs for hackers such as preparing manuals and coding GUIs (Graphical User Interfaces)

2003: 42). In fact, this feature of GNU/Linux gave rise to different many GNU/Linux distributions having "a Linux kernel, GNU tools and libraries, additional software, documentation, a window system, window manager, and a desktop environment"³³.

When hackers localize a specific application (e.g Firefox, Open Office, Thunderbird etc.), they do not contribute to just one distribution, but to all distributions. Reciprocally, when people work on a specific distribution, parts of the distribution also benefit from this work. For this reason, localization works and emergence of local GNU/Linux distributions have been interrelated with each other in Turkey.

Turkuaz GNU/Linux is the oldest Turkish distribution. In the second half of the 1990s, there were university students who had gained advance knowledge about GNU/Linux. In 1996-1997 the students started to ask for a Turkish operating system. Consequently, the Turkuaz project initiated by the core team, Turgut Uyar, F. Kağan Gürkaynak, Giray Devlet and Nurhan Çetin. After a short period of time, number of members of the core team approached to 10 and people who directly contributed to the project approached to 100.

On the other hand, in these days some of the hackers also started to ask if there was really a need for this work and instead of this work, they proposed contributing to the international GNU/Linux distributions such as Red Hat and Slackware. However, Uyar's and his friends were also sharing a similar thought about the international distributions. According to them, if they had provided Turkish support for the parts of Turkuaz, other distributions would have benefited from their work. Additionally, as Uyar (Pengence, 2006) says their mission was enforcing international distributions for Turkish support and after this, there would be no need for Turkuaz and the project would annihilate itself.

In this sense, Turkuaz accomplished its mission. Before Turkuaz, Red Hat could be installed only in English. Turkish and Portuguese³⁴ became the additional languages for installation. In this context, one can also observe the interrelation between

³³ http://en.wikipedia.org/wiki/Linux_distribution

³⁴ In these days, Brazilian hackers were also working on a localized GNU/Linux. Turkish and Brazilian hackers worked together.

the internationalization and localization. Localization efforts of the Turkish hackers encouraged Red Hat for a more internationalized software.

According to the other project members, most of the work in the project were done by Turgut Uyar. When he stopped working on the project, the progress of Turkuaz also stopped. However, Turkuaz had important effects on the development of FOSS in Turkey. First, it played a significant role in introduction of GNU/Linux to the Turkish users and bringing out new hackers. Second, it proved that people who came together with a shared goal could develop FOSS projects.

After Turkuaz, Turkish hackers developed more local distributions, e.g. Gelecek, Boreas, Turkix, IstanbulX, Pratih Linux, Teknowall, Truva Linux and Pardus³⁵. Among these distributions, two of them are the most important.

First is Gelecek³⁶. Gelecek Linux is the product of Gelecek A.Ş. In their web site, it is claimed that it had reached the 90% market share of Linux in Turkey. As Turkuaz, it uses Red Hat as its base distribution. However, Gelecek not only uses Red Hat as a base, additionally has a business model which is similar to Red Hat. It sells support for its product. On the contrary to the Turkuaz, it has not been developed by a group of volunteers and on the contrary to the Red Hat firm, it has not got hackers' support behind its product. In other words, there are not any outside hackers who find and fix bugs or develop additional features for Gelecek.

Second is Uludağ Project's product, Pardus. It has many distinctive features. First, it is the only GNU/Linux distribution which is supported by the government. The project has a core team of developers paid by TUBITAK (The Scientific and Technological Research Council of Turkey - Türkiye Bilimsel ve Teknolojik Araştırma Merkezi) UEKAE (National Research Institute of Electronics and Cryptology - Ulusal Elektronik ve Kriptoloji Araştırma Enstitüsü) . Second, it is supported significantly by the Turkish hackers. More importantly, it is having major role in making of the hackerdom in Turkey. It is not uncommon to see messages in its mailing lists saying, "Hi! I like Pardus. What can I do for it?". Project management team of the Uludağ

³⁵ see <http://www.linux.org.tr/index.php?Pg=LinuxYerliYazilimlar>

³⁶ see <http://www.gelecek.com.tr>

Project is encouraging users to contribute to the project as prosumers³⁷. Thirdly, it is not only a localization project. Even though Pardus has a Gentoo Gnu/Linux base, new packaging and configuration software are developed for the project.

3.3.1.3 Local FOSS Projects

Local FOSS applications, which are developed in Turkey, can be found in the web page of LKD³⁸. EnderUNIX's project team has significant number of projects in this page.

I wanted to examine these projects. However, some of the links were not working and some of the projects were stopped by their maintainers. However, one of these projects, BasiliX³⁹ was a good example for the cycle of FOSS software projects. On September 11, 2003, Murat Aslan, owner of the projects announced that he would give up the project.

As most of you know, the BasiliX project was not updated for more than a year now for a number of reasons. As of September 11, 2003 BasiliX project has been discontinued. The download page, mailing list archive, etc. are still available. If you are interested in taking over BasiliX development please let me know. ... Time changes. Since then, I had several (private) problems, and then lost my energy and will to keep on developing BasiliX. I think this is the soul of GPL. You create a free software, and then switch to and focus to another one⁴⁰.

After this announcement, Mike Peters took over the project.

BasiliX is once again under active development. Over the last couple of weeks, with the help of Murat, I have been working to get the BasiliX project moving again. We now have the project hosted on Sourceforge and, hopefully, this will see the start of a new productive development cycle led by myself, Mike Peters. Over the next couple of weeks I hope to poll opinion from current users about what direction they would like to see the future development of BasiliX take. Also any code which anyone has written over this recent period of latency will be gratefully accepted for review and possible inclusion in future versions. Once I have an idea of what people want, I'll get a cvs branch started and we can start to move towards a new version (Ibid.).

³⁷ It is discussed in detail in the fifth chapter

³⁸ Retrieved April 5, 2006 <http://www.linux.org.tr/index.php?Pg=LinuxYerliYazilimlar>

³⁹ a WebMail application

⁴⁰ Retrieved April 5, 2006, from <http://basilix.sourceforge.net/news.php?lang=en>

In this context, one can not always speak of FOSS projects' nationality. In large projects, there may be people from different nations. In small projects, after a time, anyone can take over the project.

The projects at the web site of LKD have two important common points. First, they have been developed by small project teams and most of these project teams have one or two members. Second, Turkish hackers are more inclined to GPL for their products rather than other FOSS licenses.

3.3.2 Festivals and Activities

3.3.2.1 Free Software and Open Source Days

The activity, Free Software and Open Source Days⁴¹ has been organized by Bilgi University's OSEC⁴² (Open Source Enterprise Center) since 2003. OSEC also established inside the Computer Science department of Bilgi University in 2003.

OSEC defines its goals as working for the development of FOSS in Turkey and supporting firms and individuals by training and consultancy services. In this context, Free Software and Open Source Days is organized for accomplishing these goals.

The well-known figures of the FOSS (Richard Stallman, Ian Murdock⁴³, Miguel de Icaza⁴⁴ etc) attended the organization in each year. Another important fact about this activity was its changing vision in each year. For example, in the first year, they only came together to understand the unique dynamics of FOSS. In the second year, the slogan was "World is discussing FOSS. Turkey will also discuss FOSS". In the third year, the slogan became "Transformation started" and in 2006 it was "Future: Now".

Furthermore, OSEC has strong relationship with the international companies which are the supporter of FOSS in the world. In 2004, 2005 and 2006 IBM sponsored the Free Software and Open Source Days. In 2006, OSEC became the only Novell

⁴¹ see <http://open.bilgi.edu.tr/freedays/>

⁴² see <http://open.bilgi.edu.tr/>

⁴³ Founder of the Debian Project.

⁴⁴ Project Manager of the GNOME and Mono projects

Practicum Center in Turkey which was authorized to make exams for Novell Certified Engineer and Novel Certified Linux Professional.

3.3.2.2 Free Software and Linux Festival

Linux and Free Software Festival⁴⁵ has been organized by LKD since 2002. It is different from Free Software and Open Source Days in some ways. First of all, while social aspects of FOSS is more essential than its technical aspects in Free and Open Source Days, festival is more inclined to technical discussions. Second, one can easily observe the academic atmosphere in the Free Software and Open Source Days. However, the best word for describing Linux and Free Software Festival is festival and it is more informal. In addition to the technical seminars, it has a game tournament and a medal ceremony named as Penguins of the Year.

The most important point in the festival is its organization. Although there are many sponsors in the organization, everything in the festival is organized by hackers who participate in the organization voluntarily. A few months before the festival, members of LKD's festival work group come together and start working on it. In this sense, it is an argument to prove that voluntary work is not constrained to programming among hackers. Additionally, one can observe their programming habits in non-programming activities. For example, if they think that an activity is complex, they divide it into sub-activities. Programmers also divide complex functions into small, more understandable parts.

3.3.2.3 METU Computer Club Programming Contest

METU Computer Club Programming Contest⁴⁶ is not a mass organization as the other two organizations. But it is older than them and has been organized since 1997. It is a programming contest for university students. Contest has two steps. In the first step, students answer the questions which are asked on the web site of METU Computer Club. If they pass this step, in the second step, they will solve the problems on the computer by the programming languages either C or C++.

⁴⁵ <http://senlik.linux.org.tr/>

⁴⁶ <http://www.cclub.metu.edu.tr/yarisma.php>

However, students must use GNU tools on the GNU/Linux platform. In other words, proprietary compilers and editors are not acceptable in the contest. METU Computer Club explains their choice with both technical advantages of GNU/Linux and its philosophy.

CHAPTER 4

Hackers in Turkey

4.1 Hackers

“Hey! Did you see Penguen¹? There is a Selcuk Erdem² caricature about FOSS”. After this announcement on a mailing list, there were rumors about Selcuk Erdem that he was also a Debian user. Even more, he designed the poster of Fourth Linux and Free Software Festival, gave a seminar about GIMP³ and attended a panel discussion about Debian. It is surprising to see a non-technical man’s active participation in FOSS, unless you recognize the similarities between artists and hackers. The panel discussion about Debian showed it well.

Selcuk Erdem explained his interest about Debian. It was mostly about its philosophy which was based on freedom. He also explained the meaning of freedom from the vantage point of artists. Then, questions about either technical or license issues started. One of the audiences asked about the next stable⁴ release of Debian. It was very late and people were waiting for it impatiently. Technically oriented participants of panel discussion talked about the reasons of delay. Suddenly, Selcuk Erdem, who was silent during those discussions, started to talk. He said he was happy about this delay. Impatient audiences waiting for Debian for along time were astonished. Selcuk Erdem explained the reason for his happiness:

Download testing version of Debian. It is still more stable than any

¹ ‘Penguin Haftalık Mizah Dergisi’ is a Turkish humor magazine.

² He is a very famous caricaturist.

³ A free graphical image manipulation program. It can be used in the place of Adobe Photoshop.

⁴ Debian has three different versions: Unstable, testing and stable. Stable version is released only after all the tests. But each can be downloaded in any time.

Linux distribution. But during this long delay, I know that Debian developers are delaying next release for a more perfect Debian. I had this problem a few times. I want a perfect work but the firms I work for want it earlier. As a result, even though the product is on time for the firm, the result does not satisfy me...

In another discussion panel⁵, Chris Stephenson⁶ talked about a similar conflict between the employees and employers at software development. One who gives money has the right to decide not only what to do but also how to do. Furthermore, project deadlines and budgets also constrain the programmers as Selcuk Erdem complains. For Graham (2004), both programming and drawing have a lot in common. In his view, composers, architects, writers, painters and hackers are makers and try to make good things. Knuth⁷ (1974) also describes programming as an art in his famous book, *The Art of Computer Programming*. For him, computer programming is an art, because:

... it applies accumulated knowledge to the world, because it requires skill and ingenuity, and especially because it produces objects of beauty. A programmer who subconsciously views himself as an artist will enjoy what he does and will do it better (Knuth, 1974: 673).

McBreen (2001) questions software engineering and proposes craftsmanship in software development instead of engineering. For him, software craftsmanship is a blend of art, science and engineering and it allows developers to acknowledge their craft both aesthetically and mechanically. On the other hand, software engineering tries to apply mechanical metaphors to software development. Indeed, each metaphors have different and contradicting approaches to work. While software engineering tries to deskill software developers, software craftsmanship needs highly skilled software developers. Chance (2005) also points out the mechanical routines of software engineering that provides "little or no scope for the development of skills and a growth in knowledge".

⁵ "FOSS from the perspective of Software Engineering" at Second National Software Engineering Symposium (September 23, 2005-Middle East Technical University, Culture and Convention Center-ANKARA

⁶ He is program coordinator of Computer Science Department at Bilgi University

⁷ He is one of the well-known computer scientists. He is also a programmer, hacker, who has authored Latex software that I used for writing this study.

IT professionals have attempted to approach software development from the perspective of engineering since 1969 (NATO, 1969). However, as a consequence of failures in software projects led by software engineering's mechanical approach, new approaches in software development have been emerged: Extreme Programming (Jeffries et al., 2000) and Agile Software Development (Ambler, 2002). In each of these approaches, mechanical routines are removed and skills of programmers started to play a prominent role in the development process. For example, instead of the relationship between the engineer who designs and the worker who implements the design, new relationship in Extreme Programming is the programmer "who defines the architecture, designs the system, writes the test and the code that supports them" (Jeffries et al., 2000: 10). In other words, programmer is both an engineer designing the system and a worker implementing the designed system. Ambler (2002), in Agile Software Development, also emphasizes the same point and argues that individuals and their interactions are more important than any processes and tools.

Who do you think would develop a better system: five software developers with their own tools working together in a single room or five low-skilled "hamburger flippers" with a well defined process, the most sophisticated tools available and the best offices could buy? If the project was reasonably complex, my money would be on the software developers, wouldn't yours? The point is that the most important factors to consider are the people and how they work together, because if you don't get right, the best tools and processes won't be of any use (Ambler, 2002: 6).

On the other hand, the absence of engineer-worker relationship does not mean the absence of any manager. There exists a manager who is aware of essentiality of programmers' skills in a software project and programmers' motivation in exploitation of their mental labor. Humphrey (1981) points out the motivations of technical people in organizations. He claims that in the absence of any motivation, technical organizations can not prosper. He discusses the motivation of technical people from the perspective of employers and makes suggestions for them. For him, managers should consider the needs of their employees such as appreciation, recognition and feeling of accomplishment. In this sense, Extreme Programming and Agile Software Development adapts Humphrey's management suggestions to software development giving programmers some rights such as,

...right to produce quality work all times ... the right to an overall plan, to know what can be accomplished, when and what cost (Jeffries et al., 2000: 17).

In this context, one may argue that new approaches in software development take developers' will into consideration. But Chance (2005) is against this argument. According to him, programmer is still susceptible to deskilling and loss of operational autonomy.

Additionally, in Extreme Programming, project deadline is directly determined neither by the manager nor the customer. Programmer himself decides the project schedule and tries to get software ready on time to satisfy his manager and the customer. This may sound fair unless one knows the famous optimism of programmers. According to Brooks, all programmers, especially young ones, are optimists and usually assume "all will go well, i.e., that each task will take only as long as it ought to take" (Brooks, 1995: 14). In this respect, pressure of project deadlines still exists. Moreover, programmers' right on the development itself does not change that they work to satisfy customer, i.e., most programmers does not like to code GUIs⁸, but customers focus on GUIs at first.

On the other hand, as it is discussed in the following paragraphs, hackers relationship with FOSS have "operational autonomy and variety of productive and creative activities that actualize the worker's capabilities" (Chance, 2005). In order to understand hackers' practice and their relationship with FOSS, interviews⁹ were conducted with Turkish hackers. Interviewees were mainly chosen from the members of LKD, EnderUNIX and Debian Turkey. Most of these interviewees were leading figures in the mailing lists and the social activities of LKD. There were also interviews with the owners of the FOSS projects which were developed in Turkey.

Second¹⁰ question of the interview was the comparison between the hacking and other types of work which include necessity:

If your firm gave up its FOSS policy, what would be your reaction against this decision? (e.g my motivation would decrease, I would look

⁸ This the main reason why the early days of FOSS applications were hard to use.

⁹ Interview questions are at Appendix B

¹⁰ First question is about hackers past experience. See Appendix-B

for another job, nothing would change for me etc.)

If you do not work for a FOSS firm, can you compare the programming activity that you perform at work with the work you do in your leisure time at home?

For amateur hackers, there is a clear difference between their professional and amateur work. They find hacking in their leisure time more satisfactory because of the same reasons with Selcuk Erdem: No deadline, no budget constraints of contracts. For them, deadlines and budget constraints of contracts have negative effects on the quality of software. Some of them thinks that failure of Microsoft or other proprietary software products are not because of their software developers. Indeed, they are talented programmers, but competence for the market is diminishing the quality of their work. In this context, they argue against Bill Gates who said amateurs could not develop software of good quality.

Furthermore, autonomy plays a significant role in the amateur works of hackers. Small-size projects¹¹ are widespread in Turkey. In these projects, they manage the project according to their wishes. They decide what and how to do in the project. For example, most of the hackers hate preparing detailed documentation or coding GUIs. In this sense, they are not interested in all software projects. One of the hackers explained this situation:

If there is challenge, I contribute to a project. Now, I am working on a system about remote car control. In fact, this is not a software project. Its mechanics and electronics design is more important than its software. Thus, in this work I am not proud of the code that I developed. I am proud of the differential in it. Here is challenge.

Later, he told me other challenging projects he contributed to or worked on. One day later, he showed me his answer to a question in a newsgroup. This time challenge for him was solving problem of a user in one line that no one had ever thought.

Raymond (1999b) explains the challenge as "scratching an itch". While most of the hackers are concerned with the social aspects of FOSS, few are not interested in it. For them, FOSS is only the necessary condition for realization of their passion and learning new skills. As, one interviewee said:

¹¹ Number of developers are one or two in such projects.

I wanted to learn multithreaded programming so I contributed to that project. Later, I wanted to improve myself on network programming which started another project.

In this context, Netscape's failure after making its browser FOSS is understandable. Project was almost complete and contributing to the project was hard because of its design. Its source code was so complex to improve it and more importantly, there were not any challenging tasks in it which might attract hackers. Thus, hackers' interest to the project slowed down. But their contribution increased only after throwing old project away and restarting the project. At this time, the project was designed to be a FOSS project at the start, such that many could contribute to it at the same time. Additionally, at this time challenging tasks of the project were not performed merely by the Netscape's staff; these tasks were also for the hackers (Weber, 2004).

Turkish hackers are also interested in projects in which they can realize themselves. This realization is also accompanied with appreciation by the other hackers. One said:

I am the happiest person in the world if my work is appreciated by a famous hacker. It is not very important for me to be known by ordinary people. Sometimes people send me congratulation e-mails because of my projects. But those are unimportant. If they do not know programming, can they really understand creativity in my work? No!

For this reason, some are eager to contribute international projects where they can find more challenging works and more famous hackers to be appreciated. Furthermore, there are hackers who do not contribute to the Uludağ Project because they do not find it challenging.

I do not know why they are re-inventing the wheel. And I can not understand why people exaggerate it too much. It is not hard to make a Linux distribution. For instance, give me one day, I can build it for you.

Professional hackers' answers were also parallel to amateur hackers. Most of the hackers talked about a decrease in their motivation if their firms gave up its FOSS policy. Some young hackers were more passionate that they would resign in this case. One said:

After the work time, I am working at home. Sometimes until 3:00 AM. Later in the morning I am again going to work. I think I would not do this for anything else.

Few hackers said that it would not bother them and went on developing FOSS at home. Some of them were sure about their firms. They claimed that it would never happen but they said that if it happened, they would resign. Professional hackers were satisfied with their job. Since they were happy working with FOSS at work time. Most of the time, there are not any difference between work and leisure time for the hackers. In this context, hackers transcend the duality between work and leisure. Additionally, description of FM affirms this situation: Web sites of people who want to go home for being busy with the computer.

These hackers are also among Castells's flex-timers or IT workers who are living under the circumstances that corrode the character. Interestingly, hackers did not mention any problems about this situation and in some of their sentences, it was clear that they were happy to be flex-timers. However, it should also be mentioned that almost all of the interviewees were not older than 30 years.

Bonaccorsi and Rossi (2004) study the motivations of individuals in three categories as economic, technological and social. In economic motivations, there are motivations such as monetary rewards, gaining a reputation among peers and gaining future career benefits. On the contrary to European hackers, almost none of the hackers talked about economic motivations in the interviews. For them, economic benefits were not a motivation, rather these benefits were result of their participation in FOSS as discussed in the gift economy. Here it can also be said that earning money from FOSS is not obvious for Turkish hackers. They do not know FOSS business models very well and small number of FOSS firms is not enough to present future careers for the hackers.

On the other hand, technological motivations, i.e., learning new technology, FOSS is better, scratching an itch, were widespread among Turkish hackers. Especially hackers in EnderUNIX group are more inclined to technical motivations. However, challenge plays more important role than the other technical motivations. On the other hand, they are not insensitive to the social issues. For example, even though

their blogs were mostly about technical issues, in the interviews, some of them explained their relation with FOSS as "For my country" and talked about problems of software industry in Turkey.

Furthermore, there are also a few hackers who participate in FOSS only because of its technical features. These hackers usually do not participate in anywhere such as LKD, Debian Turkey or EnderUNIX. They prefer either international or individual projects and not interested in anything in Turkey. For instance, one of them was very surprised and happy when I told him his project was on the web page of LKD and it was a candidate in the contest. He jokingly said:

Indeed, I am not FOSS advocate. Further, I am against it. Why are you surprised? I develop software, then a programmer take my code and use it in its firm. Result: I am unemployed and they earn money. It is real world. I also use windows and Microsoft Office. Because it is technically better. But in my embedded system projects, I have to use FOSS.

However, among Turkish hackers social motivations are the most widespread. If few merely technically-motivated hackers were not counted, most of the hackers talked about social motivations. Main theme was freedom, helping others and sharing. There were important number of hackers who talked about sharing, freedom and social awareness although they never explained their motivation in terms of such as socialism, communism, imperialism etc. except few hackers. Even people who never talked about politics in everyday life, started talking about politics of FOSS in the context of freedom. In this sense, Turkish hackers attitudes are consistent with the hackers' principles and acknowledgments that Levy (2001) mentioned. First of all, they believe that computers can change the life for the better and they want to share this belief with others. Secondly, they want unlimited access to the computer and information should be free for all people.

For example, some hackers argued that for equality in the society, information should be free for all and one should not steal software by using pirate software. In their view, Turkey could escape from technological dependence on proprietary software vendors (e.g. Microsoft, Oracle, SAP etc.) only by adopting FOSS. Additionally, they do not merely suggest solutions for the government. They also do work in or-

der to achieve their goals. In this sense, "Computer for Children"¹² campaign is a good example for their social attitudes and practice. In this organization, they are organized for gathering old computers and gifting it to the children who need. Yet, before gifting the computer they install GNU/Linux on it. While this is a technically necessary decision¹³, they explain their decision by giving a link of an article: Why software should not have any owners¹⁴?

Motivation for programming can be explained by challenge or the activity itself. But there is also great social passion involved in the other activities of hackers, i.e, translation, giving seminars, helping each other in mailing lists. Without demanding any money, they go anywhere in Turkey to give seminars about FOSS. A new/potential hacker observed that:

Well, last month I attended a Microsoft seminar which I had paid. There is not a contact between the speaker and the audiences. Speaker was as if he wanted to go home as soon as possible. But, at the festival, Linux seminars were different. Speakers were giving seminar just as talking about the recipe of a delicious cake.

In this respect, one can not speak of the homogeneity of the hackers' motivations. For hackers, who program FOSS, activity itself is essential. Thus, it can be called as intrinsic motivation (Kuster et al., 2002). On the other hand, hackers who work in localization or organizing seminar/festival activities are more concerned with extrinsic motivations. In other words, their motivation is more related with the products of their activity rather than the process itself. They use the words *sharing, Turkey, nation, society, helping* were more than the hackers who only program.

However, the borderlines between different types of motivations of the hackers are not absolute. Thus, in addition to the interviews, I tried to focus on their practice with FOSS. In this sense, *Distribution Wars* in Linux e-mailing lists was a good point of start.

By makings anyone's free contribution possible, FOSS also developed a new relation with the product, prosuming. Any user can produce (pro-) while consuming

¹² <http://www.cocuklarabilgisayar.org/>

¹³ Second-hand computers are older and most of the time MS Windows needs newer hardware. On the contrary, GNU/Linux can be configured for older and lower quality hardware.

¹⁴ In Turkish, Yazılımın Neden Sahibi Olmamalıdır?

(-sume) FOSS. In other words, FOSS enables consumers to produce the FOSS during consumption. One can not understand the GNU/Linux distribution wars in LKD mailing lists without the aid of the concept, prosuming. Story repeated many times.

A new (potential) GNU/Linux user asks,

I am new to Linux. I could not decide which distribution is good for me. Which one do you suggest? Suse, Red Hat, Mandrake, Slackware, Gentoo, Debian etc.?

Then, the war starts, each hacker claims his distribution is better and the others are worse. These discussions may seem absurd for a person who uses Windows OS or MacOS and if one sees the GNU/Linux distributions' user interfaces, possibly s/he can not recognize their differences. However, distributions have different philosophies and one can not entirely understand it without using it. Thus, I installed some popular GNU/Linux distributions (Red Hat, Mandrake, Slackware, Suse, Gentoo and Debian) and tried to grasp their philosophy. These distributions can be categorized into three groups: In the first group, there are Red Hat, Mandrake and Suse¹⁵. They are easier to install and manage. In fact, they are GNU/Linux distributions which either less technical users or beginners can install and use.

In the second group, there are Slackware and Gentoo. It is hard to advice a beginner these distributions, especially Gentoo. Because it is hard to install for a beginner. Moreover, it may take one day to install Gentoo. However, these distributions give the user more control. Automated installation process and tools are less than the distributions in the first group. However, some says that if one really wants to be GNU/Linux hacker, one should use Slackware.

In the third group, there is Debian. There is not any commercial company behind the Debian Project. Indeed, there is no company behind Gentoo. But Debian's discourse based on freedom makes it different from any distribution¹⁶. For example, In their web site, it is titled as "The Universal Operating System"¹⁷. In Turkey, there are not any widespread Turkish Red Hat, Suse, Gentoo or Slackware user mailing lists.

¹⁵ Here the subject is desktop/home versions, not server versions.

¹⁶ For example, one can not expect a major Debian Developer be employed by Microsoft. However, same is not true for the Gentoo Project. Retrieved December 5, 2005, from <http://linux.slashdot.org/linux/05/06/13/2137239.shtml?tid=109&tid=106>

¹⁷ <http://www.debian.org/>

On the contrary, there is the Debian Turkey list whose average daily mail traffic is about 16 e-mails per day in 2005¹⁸.

During the interviews, one of the questions was about the distribution wars. I asked for the distribution they use. I got answers parallel to my categorization:

In the first group, mostly there were beginners and IT professionals. It is easy to understand the beginners' preferences. These distributions are easy to install and manage as mentioned above. However, behind IT professionals' preferences, there are business models of these distributions. For IT professionals, to have power in the market, GNU/Linux must be more user-oriented instead of hacker oriented. However, some of these hackers also have a second choice, i.e Gentoo or Debian in their computer.

As stated above, Slackware is not a good choice for a beginner. Since its installation is not automated as in the first category distributions. Usually, one has to configure the Slackware through text files. Indeed, hackers suggest it. They argue that if one wants to really learn GNU/Linux, Slackware is the ideal distribution. Although it seems complex for beginners, the hackers explained their choice as "Because it is simple". Then explained its simplicity in terms: "Because I can control it. I use the computer, computer does not use me". Gentoo is similar to Slackware. Each one gives hacker more chance to affirm themselves on the operating system by giving almost all configuration jobs to the user.

Among the GNU/Linux distributions I used, Debian is the most stable. However, interestingly, none of the hackers talked about its technical sides in the interviews. They talked about freedom. At the Debian panel¹⁹, while audiences were waiting for hearing about Debian, the speakers again talked about free software and the Free Software license, GPL. This was a critical departure from the international declarations of the Debian. In its international declarations, Free Software and Open Source Software terms refer to the same thing²⁰. According to the The Debian Free Software

¹⁸ 5579 e-mails were sent in 2005. see <http://lists.debian.org/debian-user-turkish/2005/>

¹⁹ at Fourth Linux and Free Software Festival

²⁰ *What does free mean?* (Retrieved December 5, 2005, from <http://www.debian.org/intro/free.en.html>)

Guidelines²¹, the BSD license is also a free software²² license. In an opposite manner, Turkish Debian users speak of free software in the sense of FSF and in the interview questions they talked about GPL as they did in the panel discussion.

In this context, GNU/Linux is neither an alien product for the hackers nor it is just a tool for them as Windows or MacOS users pretend. It is objectification of hackerdom which they produce again and again in their productive activity.

In addition, there are also BSD (its philosophy is almost similar to Slackware) hackers. Furthermore, in near future, Pardus hackers are expected to appear. Since there are many people who contribute or want to contribute to the project. As the possibility of contribution increases, the distribution is owned by the prosumers. Moreover, national feelings are expected to play a prominent role in the formation of Pardus hackers.

However, Pardus is a sign of contradiction between the general acknowledgments of the hackers in the world (Levy, 2001; Himanen, 2001) and some of the hackers' attitudes in Turkey. In the general hacker culture, there is mistrust for the authority and tendency for de-centralization. Amateur hackers in Turkey also do not want government intervention in FOSS. They see it as a threat for freedom and autonomy in FOSS. While professional hackers agree with amateur hackers, they also think that government should not be a side for FOSS, but also for proprietary software.

On the other hand, attitude of hackers towards the Uludağ Project was opposite of this. Uludağ Project gained great support from Turkish hackers. But the real success of Uludağ Project was attracting new people. For old hackers, as it is discussed in the following part, GPL of the project was important. On the other hand, new people were attracted by TÜBİTAK's support and army's relation with the project. As some hackers underlined, there was strong trust for state's authority among Turkish people and when they recognized TÜBİTAK's support for Uludağ project, they were influenced by this authority. Furthermore, for new hackers, national feeling was an important source of motivation.

²¹ Retrieved December 5, 2005, from http://www.debian.org/social_contract#guidelines

²² According to FSF BSD license is not free (Retrieved December 5, 2005, from <http://www.gnu.org/licenses/license-list.html>)

For a few hackers, nationalism was against the spirit of the FOSS and there was no need for a national GNU/Linux distribution. One said:

When we translate or make a program compatible for Turkish we do not call what we do as nationalization. FOSS can not be a national thing. Instead we call it localization and more truly as localization of an internationalized software.

But some of the hackers were more moderate:

Nationalism or government intervention is not very good. But there is a fact in Turkey: "Domestic goods are Turkish goods. Every Turkish must use them!²³". Whether it is good/right or not. Nationalism is a reality in Turkey and people regard governments' approval in Turkey. Is it healthy? I do not think so. But I am sure that Uludağ will accelerate the spread of FOSS in Turkey.

In the last days of 2005, Pardus was ready for Turkish users. There were news about Pardus in national and international²⁴ press. Message traffic in the Uludağ project's main mailing list increased. New people subscribed to the list. For these new subscribers, Pardus was a national operating system rather than a national distribution which was a dialect of GNU/Linux. They were old MS Windows users and were eager to use Pardus. However, they did not know much about the goals of the project. For example, in the stated goals²⁵ of the project, compatibility with the UTF-8 (8-bit Unicode Transformation Format) standard was essential²⁶. In this sense, members of Uludağ project's mailing list was very sensitive about using UTF-8 in their e-mails. However, as new members subscribed, the number of e-mails with different encodings increased.

More interestingly, some users started to discuss national vs. universal. They said:

²³ In Turkish, Yerli malı Türk'ün malı, her Türk onu kullanmalı!

²⁴ Retrieved December 5, 2005, from <http://news.zdnet.co.uk/software/linuxunix/0,39020390,39245444,00.htm>

²⁵ Retrieved December 30, 2005, from <http://www.uludag.org.tr/hakkimizda.html>

²⁶ It is able to represent any universal character in the Unicode standard, yet is backwards compatible with ASCII. For this reason, it is steadily becoming the preferred encoding for email, web pages, and other places characters are stored or streamed. (Retrieved December 5, 2005, from <http://en.wikipedia.org/wiki/Utf-8>)

Why are you still using other GNU/Linux operating systems²⁷? We have Pardus!

Consequently, similar messages appeared in Debian Turkey mailing lists and LKD's mailing lists. Mailing list administrators put a stop for this type of messages in a strong tone giving the reason that discussing this topic was out of the scope of the list.

In the interviews, hackers' attitudes towards Pardus were mostly considering it as a mean for making FOSS widespread in Turkey. They were hacking and supporting Pardus, but they did not say that they would give up their present GNU/Linux distribution for using Pardus.

4.2 Hackers, Licenses and Commerce in Turkey

In the second chapter, history of FOSS is presented as a commodification process of software which had been free in itself. While conflicting interests of hackers and firms transformed FOSS into new forms, coinciding interests of hackers and firms made FOSS more widespread and as a network affect new actors with different interests engaged in FOSS.

In this respect, reconsidering the separation between Free Software Foundation and Open Source Initiative is necessary to analyze the tendencies of FOSS in Turkey. As discussed in the second chapter, open source software appeared as a need for "a marketing program to pitch it to the corporate world" (Fogel, 2005: 15).

As a consequence of the separation between the free software and open source software, each side started to favor different types of licenses²⁸. For free software advocates, hackers' interests have been more essential than anything, so GPL which is based on reciprocity has been the most appropriate license for them. On the contrary, open source software advocates are more concerned with firms participation in FOSS. So they prefer BSD-style licenses²⁹ which give rights for firms to make the

²⁷ As stated above, there was a strong misconception between the terms operating system and the distribution

²⁸ see section 2.2.4 FOSS Licenses for detail

²⁹ Licenses which do not enforce reciprocity.

software proprietary. In the recent times, Raymond, leader of the OSI, declared that for pragmatic reasons, hackers should prefer BSD license rather than GPL ³⁰.

The discussions are still going on in the world, but there is a different situation in Turkey. As I observed from the discussions in mailing lists and forums, although some Turkish hackers are aware of discussions around the world, FOSS as a relationship between hackers and firms is still in the formation process and being conditioned by the discussions in the world.

In this context, hackers were asked for the relation between Free Software and Open Source Software to understand how they perceived the discussions in the world:

Which term do you prefer? "Free Software" and "Open Source Software"? Why?

In the world, it is observed that there is a tendency for using "Open Source Software" rather than "Free Software". What do you think about this tendency?

Soon, this relation is investigated from the perspective of licenses:

Do you have any license preferences for the products that you develop? GPL, LGPL, BSD, Mozilla etc?

There were significant inconsistencies between the answers of the hackers. While some preferred to use Open Source Software instead of Free Software, they preferred to use Free Software License, GPL. On the other hand, while some preferred Free Software, they talked about using BSD licenses. Indeed, most of the hackers were not aware of the debate between Free Software and Open Source. They said that they preferred GPL license, because they did not know anything about the other licenses.

More interestingly, after a few interviews I had to change the question and added the term, unrestricted beside free and open source³¹. In English, the main problem about free was the confusion between different meanings of free. Thus, free software hackers explain it, "free as in free speech, not as in free beer". In Europe, 'libre' is also used for free software to escape from the confusions of free. However, in Turkey, even though the word, 'free' has not any problem as 'free' in English, some hackers started use the word 'unrestricted'. First of all, the word, 'unrestricted', is not

³⁰ Retrieved December 5, 2005, from <http://www.myfreebsd.com.br/static/raymond-20050604.html>

³¹ in Turkish, unrestricted:Serbest, free:Özgür, open source:Açık Kaynak Kod

capable enough to explain the philosophy of free software. Second, it is inclined to be understood as 'zero-cost' more than 'free'. Moreover, some of the hackers insistence on translating free software as unpaid software³² shows that they consciously avoid from using the Turkish word, özgür.

A hacker explained the problems of 'özgür' in Turkish:

In Turkey, unfortunately, 'free' means rebelism, anarchism, terror. Additionally, firms may not consider the word 'free' as a serious discourse.

And some hackers, especially Debian hackers, were strongly opposed to the term 'Unrestricted'. They said they did not talk about public good, so using the term 'unrestricted' was nonsense.

In spite of inconsistencies of some hackers' answers and their positioning in opposite poles, two groups' answers were clearer and more consistent than the others. They were also aware of the discussions concerning Free Software and Open Source Software.

First group is Debian Turkey hackers. They are more conscious than any group in Turkey in the sense that they are aware of hackers' interests and importance of licenses in objectification of their interests. Interviewees from the Debian Turkey list were aware of the differences between free and open source. In this respect, their answers were usually free software and GPL.

Second is EnderUNIX. They prefer BSD licenses rather than GPL. But they also have GPL'd software. They are also aware of the controversies between free and open source software. For example, instead of stating any preference about these terms, they talked about the differences between them and said that one should use them according to the context. However, they consider freedom different from Debian users. One argued that real altruism was gifting the software with BSD license, not restricting the firms with GPL.

There are also hackers beyond these groups who are aware of either controversial between Free Software and Open Source Software or licenses. It is an important fact that these hackers are also activists of LKD. They admitted that sometimes they

³² Retrieved December 5, 2005, from <http://listweb.bilkent.edu.tr/linux/06/0116.html>

called FOSS as open source just unconsciously, but the real problem was freedom and "open source" was a vague term.

On the other hand, hackers, who are not members of Turkish Debian Users, EnderUNIX or LKD, are usually not interested in social aspects of FOSS and do not know the details or importance of licenses. They said that they preferred GPL or LGPL. This is rather an ignorance than a conscious choice. For example, when I asked them reason of their choice for GPL, they admitted that they did not know any license beyond GPL.

Despite their different tendencies, Turkish Debian Users, EnderUnix and LKD build a platform of self-education. Their activities such as seminars and mailing lists discussions increase awareness of their members about licenses in the process.

For example, LKD's first governing statute was revealing its unawareness about the international debate between the free software and the open source software. Neither of these terms were used in the first governing statute. Instead, the term, unrestricted was preferred.

However, after three years, in its second general committee, the term unrestricted was replaced with the term, free. Additionally, after Gates's visit in 2005, in the platform, which was led by LKD, they preferred the term open source and started talking about business models.

In fact, the change in LKD's discourse and internalization of the international debate did not happen by coincidence. On the contrary, the change is strongly related with two processes.

First process is similar to the trend in the world: transformation of some hackers to entrepreneurs. When they graduated from their schools, they chose to make money from what they knew well, FOSS. In this sense, while nature of FOSS was changing from hacking object to business object, a change in LKD was inescapable. Since, these transforming hackers were also LKD's activists. This is discussed in detail in the next chapter.

Second process is LKD's and other Turkish FOSS organizations' practices. As discussed in the previous chapter, the terms free software and open source software, consequently the debate between them, happened and evolved under some social

circumstances. For this reason, without any social base, appearance of this debate would sound artificial. If there were not any hackers who desired continuous availability of source code and if there were not any firms who wanted to benefit from hackers' work, there would not be any conflict between hackers and firms. However, LKD's activities and internationalization works of small hacker groups, helped rise and making of the hackerdom in Turkey. Furthermore, success of the FOSS products and firms affected the IT firms' business strategies.

Additionally, Uludağ Project is accelerating the process by making licenses a current issue for hackers. In October 2005, a meeting announced:

GPL fans and Commonists are wanted!

Now, we have to discuss licenses that we had delayed for a long time.

Question: Firstly, GNU GPL, GNU LGPL, GNU FDL and CC... What is the validity of these licenses in Turkish law? Is it enough to translate them into Turkish?

We must do something unless we want to have accidental problems after the release of Pardus³³.

Until Uludağ Project, there were small GPL'd projects. However, Uludağ Project increased the importance of GPL in Turkey. Since project was also presuming emergence of new business models around its product, Pardus. Meeting happened on October 8, 2005. There were significant participation of hackers, including Uludağ project team members. Conclusion of the meeting was, "we must study more..." Since there were contradictions in the Intellectual Property Rights of Turkey, another meeting was planned and they emphasized the need for more lawyers' participation in the next meeting.

There is no doubt that with the advent of Uludağ Project's product, Pardus and new business models accompanying it, license issues in Turkey will be clarified and awareness about it will increase.

On the other hand, despite potential conflicting issues between hackers' motivations and firms' profit-oriented interests towards FOSS, it should be pointed out that hackers general attitude towards commercialization of FOSS is affirmative. Moreover, they see hackers who earn money from FOSS as fortunate people who earn money in a funny way and most of them consider FOSS in the context of free trade.

³³ Retrieved December 5, 2005, from <http://www.fazlamesai.net/index.php?a=article&sid=3348>

On the contrary to the common intuitive, Microsoft is not more than a proprietary software vendor and they are not radically against Microsoft. It is true that they do not enjoy it but for them, it is just an actor in the market. Hackers talked about Microsoft as the below:

Neither Microsoft nor any proprietary software vendor is important for me. It has not any place in my life. OK, they also have good products. But they develop software for the market. If GNU/Linux started to compete against Microsoft, it would lose its purity, freedom.

Furthermore, hackers were bothered by being known as enemies of Microsoft. For this reason, there is a general attitude among old hackers that rather than pronouncing the name of Microsoft, they call it as "the other operating system".

Few hackers explained their motivation by either socialism or anarchism:

I am happy in FOSS world. This is really another world which showed me another world is possible. Indeed, it is a communal life, lack of any authority. No one force another but help each other. There is not any property. Everybody is equal.

According to a NewsForge³⁴ survey, "the worst Linux/free software myth" is "free software is communism". Indeed, this myth is one of the bases of Microsoft's propaganda against FOSS all over the world. One, who is against copyright/property rights, is communist for Microsoft. For example, at TBD (Informatics Association Of Turkey³⁵)'s meeting of IT Departments' Directors, Microsoft Turkey's propaganda was again about the relation between FOSS and communism³⁶. There were two interesting scenes in Microsoft's presentation. In one of the scenes, there was a message on application's user interface, "If you are using pirate software, do you accept the God's punishment after death?". Other scene was from TKP's (Turkish Communist Party) cortege at May 1, 2003 saying: "Microsoft, get out of the way! Long live Free Software".

Turkish FOSS hackers' reaction against those attacks was generally rejecting the notion of either communism or any ideology in FOSS. One said that, "if you need an

³⁴ Retrieved December 5, 2005, from <http://www.newsforge.com/pollBooth.pl?qid=78&aid=-1>

³⁵ Türkiye Bilişim Derneği

³⁶ Retrieved December 5, 2005, from <http://merkez.tbd.org.tr/listeler/tbd-forum/2004/Jun/0042.html>

-ism, let's call it tuxism"³⁷. More interestingly, when a newbie posts a message to the mailing lists with the content of "Support FOSS (or Pardus) against imperialism" in a leftist tone, hackers usually behave as the message does not exist.

However, as discussed in the forums of NewsForge, although Microsoft's propaganda "Free Software is Communism" is widespread in US, it is not in Europe. According to this discussions, main reason behind this situation is anti-communism is stronger in US despite legitimacy of the communist parties in Europe. In the same sense, if one accounts for the strength of conservativeness and Islam in Turkey, Microsoft's propaganda will be more understandable.

4.3 Software Engineers

According to a survey conducted in EU, software engineers play an important role for FOSS development (Ghosh et al., 2002). Additionally, there are many computer science and software engineering students who are contributing to FOSS in EU and US. On the other hand, I could meet only a few people who were both software engineers and hackers in the interviews. This was contradicting with the studies in EU and US, so I decided to investigate it.

First, I discussed this contradiction with software engineers who were also hackers. I just asked "Why are the software engineers in Turkey not interested in FOSS? Why do not they contribute to the FOSS projects as in EU?". Second, I interviewed with software engineers who were not hackers and asked the questions in the Appendix-B. In this second step, I mainly focused on their practices.

In the first step, the answers mostly focused on the socio-economic conditions of Turkey:

First of all, you should not forget that most of the students do not choose their future professionals as a consequence of their capabilities and interests. They go to the universities after ÖSS (Student Selection Exam - Öğrenci Seçme Sınavı). And do not forget that only students who have the highest scores in the exam can be a student at Computer Engineering Departments. You should ask why do they prefer computer engineering? Yes, the reason is money. Furthermore, in Turkey, computer engineers

³⁷ Tux the Penguin is the official emblem of Linux. Retrieved December 5, 2005, from <http://listweb.bilkent.edu.tr/linux/2000/Jun/0133.html>

earn money by developing and selling proprietary software. In this sense, FOSS seems as a fact against their profession.

Additionally, Turkey is not a rich country. A student needs computer and a good Internet connection. Well, it is not possible for most of the students. When those students are graduated, as natural, they choose the easiest way for earning money, developing and selling proprietary software.

Another hacker, who was also an instructor at a computer engineering department answered similarly.

Our students are more interested in earning money. OK, one call free software as free in freedom, not as in free beer. But in Turkey, developing free software requires being altruist.

Other answers were not different. In brief, they said Turkey was a country that FOSS offered so little to the software engineers and proprietary software was best to earn money.

Additionally, software engineers answered similarly in the interviews, . They talked about ethical values, importance of sharing and national software industry. However, they concluded that they should earn money to live and developing software for free in Turkey was irrational. However, they also admitted that they did not know anything about FOSS business models.

Furthermore, when I asked them about programming,

Do you enjoy programming? Do you develop software in your leisure time? Can you compare programming at work and at home?

their frequent answers were,

Once upon a time I was enjoying programming. Now, I do not know. My work starts at 9:00 and ends after 19:00. Of course, additional work after 19:00 is not rare for us. We are legionaries of the IT revolution. We program, sleep, wake and program. Despite there were times that I was enjoy programming, I really do not know now!

As I investigated work practice in detail, I realized that the main difference between the hackers and the software engineers was their practice. Most of the software engineers in the research, work for projects³⁸ which are managed by the principles of the software engineering discipline. In these projects, they have very limited autonomy and standards are essential. But more importantly, pressure of project deadlines

³⁸ Most of these projects belong to the defense industry and financial projects.

and unpredicted extra work hours to develop software on time are very frequent in their work.

4.4 Main Findings of The Research About Hackers

As stated in the section 2.2.2, this study diverges from most of the studies in the FOSS literature, since it rejects any anomaly about the hackers' voluntary contribution to the FOSS projects. Instead, this study focuses on the social circumstances that make hackers work voluntarily. In this part, key findings of the search about Turkish hackers are presented by discussing these social circumstances and their attitudes according to the questions in part 3.1.

What kind of FOSS works (or projects) do hackers engage in?

Hackers' contribution to the FOSS in Turkey is not only by developing software. They localize FOSS, give seminars and organize social activities (festivals and meetings) in order to familiarize more Turkish people with FOSS. In this context, as one can not speak of the homogeneity of hackers' work, one can not also speak of the homogeneity of the social circumstances that make their voluntary contribution possible.

On the other hand, FOSS users are named as prosumers in the previous parts because of their specific relation with FOSS that is based on reproducing and improving it. In this sense, they are not passive computer users as most of the Microsoft or Macintosh users. Thus, on the contrary to ordinary computer users, most of the hackers in the research were at least capable enough to write small scripts³⁹. Although writing scripts is not a necessity, it gives important opportunities to automate or simplify long works that can not be done by Windows operating systems. For example, the new hacker⁴⁰ who was a Windows user two months ago told that:

At 17:00, system administrators said that there would be power cut at 22:00 because of maintenance at system room and added that it would be better shutdown my computer before I left, at 18:00. However, I was downloading a large file and it had at least two hours to finish. And after this, I should run the downloaded file to prepare data for tomorrow. Well,

³⁹ Small programs used only for simple, repeated actions.

⁴⁰ This new hacker also told about problems with his wife because of GNU/Linux. He said that his wife was angry with him because of his long hours in front of the computer.

I wrote a small script and go home. I am not a programmer, but it was really simple.

In other words, most of the hackers at least program for themselves in their daily use.

Weinberg discusses the psychology of the programmers from the perspective of different personality traits whether a person is compliant, aggressive or detached:

The compliant type is characterized by the attitude of "liking to work with people and be helpful." The aggressive type wants to "earn money and prestige," and detached type wants to be "left to myself to be creative." (Weinberg, 1998: 53)

According to Weinberg, programmers are more inclined to having a detached type. Moreover, this detachment is overdeveloped:

Although they are detached from people, they are attached to their programs. Indeed, their programs often become extension of themselves (Ibid.).

In this context, it can be argued that hackers make FOSS extension of themselves as they prosume it. For this reason, availability of source code means for the hackers is its *prosumeability*. This is the most common point between the all hackers whether they contribute to FOSS projects by programming or in the other ways. Indeed, it is the main reason behind the distribution wars, since each hacker personalize their GNU/Linux distribution. However, for the hackers who program for the others ⁴¹, challenge of work, autonomy and increasing realization of one's powers are essential in their relation with FOSS. Most of the hackers implied that they did not like all programming tasks. For example, a few said that they did not contribute to the Uludağ project, because developing a distribution was an easy work. As I observed from the mailing lists, when a member of the list asked a question, hackers' response to this question is determined by its challenging degree. When it is a so easy question, hackers do not reply it or reply to mail as "It is simple question. Search on Google". When it is harder question, hackers themselves search on Google and compete for finding the best solution. Furthermore, if a well-known and technically superior hacker posts his problem (it is very rare), number of hackers who are eager to answer the question is higher, too.

⁴¹ Hackers who are releasing their product for public use.

For most of the hackers, hacking means autonomous work. They do not like to be told what to do and how to do. As some of the hackers said hacking was something for them as playing chess, solving puzzles or reading a book and difference was product might also be a valuable good in the market. For them, the most fortunate people were who could also hack at daily job. Furthermore, autonomous work also means work which is away from market pressure. Thus, in their view, Microsoft programmers are not enemies, but who suffer from competence and market pressure.

Increasing realization of powers is also related with challenge and autonomy. In each challenging autonomous work, they try to overcome their boundaries. Additionally, hackers preference for UNIX Operating Systems (e.g GNU/Linux and BSD) is not by chance. Raymond explains difference between the philosophy of UNIX and the other operating systems:

Many operating systems touted as more 'modern' or 'user friendly' than Unix achieve their surface glossiness by locking users and developers into one interface policy, and offer an application programming interface that for all its elaborateness is rather narrow and rigid. On such systems, tasks the designers have anticipated are very easy - but tasks they have not anticipated are often impossible or at best extremely painful.

Unix, on the other hand, has flexibility in depth. The many ways Unix provides to glue together programs mean that components of its basic toolkit can be combined to produce useful effects that the designers of the individual toolkit parts never anticipated (Raymond, 2003).

As UNIX-style Operating Systems, hackers prefer the software which can be extension of themselves by their work. Since, it does not give the tools to do something, but gives tools to make new tools.

On the other hand, for hackers who participate in mostly non-programming activities (localizing software, giving seminars, organizing meetings/festivals), although there may be pleasure in the activity itself, products of the activity are important. In this respect, these hackers were more inclined to use the words, sharing, freedom, helping others etc. for describing their relation with FOSS.

However, Uludağ project and release of its first product, Pardus, added another dimension to the development of FOSS: Nationalism. Project management team made the ways of contribution possible through mailing lists with a clear statement of goals. After this it was ordinary to see messages like:

Hello, I am a student. I know C, php and Java programming languages. I can spend 6 hours a week for the project. I can also translate English documents. What can I do for Uludağ?

However, nationalism is not enough to understand hackers' interest for it. Since there were also local GNU/Linux distributions before it. The most critical point about the project is project management team's strategy: Releasing the project with GPL and having good relationship with the hackers since the beginning.

What are the differences between hacking and non-hacking activities?

To understand hacking, I wanted hackers compare it with their non-hacking activities. In this respect, they compared programming at work with hacking at leisure. There were hackers who were also hacking at work time professionally, so they said that there was not any difference.

On the other hand, for the others, there was an important discrepancy between hacking and non-hacking. For Weinberg, the deepest difference between the professional and amateur work is the user of the program.

Almost invariably, the sole intended user of an amateur's program is the amateur himself, whereas professional is writing programs which other people use (Weinberg, 1998: 122).

It is also true to some extent for hacking and non-hacking activities of the hackers. Since, even though hackers mostly write programs for themselves as "scratching an itch" (Raymond, 1999b), the product may be released for public use. For this reason, explaining the difference between these activities should depend on the voluntariness of the work. All hacking works (programming, localization etc.) can be considered from the perspective of voluntariness.

However, if the differences are only explained from the vantage point of hackers as the difference between voluntary and involuntary work, one can undervalue the specificity of the FOSS hackers in the informational capitalism. FOSS hackers relation with FOSS is different from the ordinary leisure time activities. Since they relate with their product through FOSS licenses. It is true that interviewees were not well-informed about the FOSS licenses. Nevertheless, they were inclined to use GPL. According to Fink,

While the development process is not about the license, it is the license that provides the boundary conditions that make the process work. These

conditions establish a foundation that ensures communal and collaborative development. Since the GPL, LGPL, and other licenses that contain a reciprocity requirement provide that changes and modifications be returned to the community, this is the foundation that ensures collaborative development (Fink, 2003: 53).

In this context, by establishing their relationship with FOSS through FOSS licenses, first, they ensure continuous accumulation of social knowledge. On the contrary to the common sense, hackers are not people who dedicate their whole life to the FOSS. As I observed hackers from the e-mailing lists, most of the hackers' of the 1990s are not hackers now. However, as Ghosh's (1998) cooking-pot market, during their hacking process, they prosume FOSS without exhausting but with increasing its value by their small to large contributions. Second, hacking product with the FOSS license socializes the software.

... the license guarantees that the hacker and the user receive exactly the same rights with respect to the product, and that both are endowed with the product's full creative and productive potential. ... whereby the more hackers produce and relate to one another through sharing, the more productive and communicative powers they have (Chance, 2005).

In summary, the main difference between hacking and non-hacking activities is the voluntary work that is defined through FOSS licenses that both ensures continuity of the knowledge and socializes hackers work.

What are the differences between the hackers and non-hackers in the context of their relationship with their work?

In order to answer this question, I tried to compare hackers' practice with the software engineers. As the answers of previous questions, key finding of the research was the major role of voluntary work in hackers' relationship with the software and the major role of obligation in software engineers' relation with the software.

First of all, as some of the interviewees emphasized, there was an essential difference between the formation of these relationships and role of the voluntarism appearing at the first step. Students prefer computer engineering departments because of its popularity and higher income. For most of the students and software engineers, earning money from FOSS is not clear and they choose proprietary software business. Furthermore, they consider hackers' contribution as altruism. On the other hand, hackers' relation with FOSS starts voluntarily.

Secondly, software engineers' lives are similar to the characters in Sennett's (1998) book. They are flex-timers and usually work under the pressure of project deadlines and competition in the market. When I asked if they programmed at home in their leisure and enjoyed programming, they started to complain about extra hours, project-based working, difficulties of catching up with the technology etc.

However, as their firms work on FOSS, some of the hackers did not declare any difference between hacking for the firm and hacking for themselves. They answered, humorously, "I worked 30 hours a day". But they never complained about anything.

Thirdly, there are significant differences between the work processes of the hackers and the software engineers. For the software engineers, the product is a whole entity. They can not say, "I want to develop this part rather than that part". On the other hand, hackers can do this in their personal hacking and some firms overlook⁴² the hackers' attitudes. Indeed, there is again a difference between the kind of work: while software engineers seek for external rewards (money), hackers are mostly interested in the activity itself.

What are the differences between the Turkish hackers and the other hackers?

Despite the huge literature about the motivations of the hackers and the FOSS business models, there is an undervalued issue: Lower participation of the females in the FOSS activities. According to Levy, hackers

formed an exclusively male culture. The sad fact was that there never was a star-quality female hacker (Levy, 2001: 84).

Although I used the term, hacker, in a wider meaning, I could not interview with any female hacker⁴³. Thus, I discussed lack of female hackers both with male hackers and female software engineers/programmers. They attributed the phenomenon to genetic and cultural differences. A few women explained the genetic differences as,

females prefer permanent relationship. Their first operating system was Windows. They are not like males. They can easily move from Windows to Linux.

Nevertheless, there is a slight difference between Europe and Turkey. According to a research in EU, "women do not play a role in the development of Open Source

⁴² This behavior of firms is discussed in the next chapter.

⁴³ I tried, but some of them refused to answer because of their business

Table 4.1: Motivations of FOSS Developers in Europe

Motivation	%
Learn and Develop new skills	78.9
Share Knowledge and skills	49.8
Participate in a new form of cooperation	34.5
Improve FOSS products of other developers	33.7
Participate in the FOSS scene	30.6
Think that software should not be proprietary good	30.1
Solve a problem that could not be solved bu proprietary software	29.7
Improve my job opportunities	23.9
Get help in realizing a good idea for a software product	23.8
Get a reputation in FOSS community	9.1
Distribute not marketable software products	8.9
Make money	4.4
I do not know	1.9

Table 4.2: Motivations of FOSS Hackers in Turkey

Motivation	%
Learn and Develop new skills	10.9
Share Knowledge, collaboration, contribute to a better world	67.2
Nationalism	4.7
Think that software should not be proprietary good	7.8
Solve a problem that could not be solved bu proprietary software	4.7
Improve my job opportunities	1.6
Enjoy programming, fun	21.8

and Free Software; only 1.1% of the FLOSS sample is female.” (Ghosh et al., 2002). There were less female hackers in Turkey. For example, there were not any female among the hackers who gave seminars until the Free Software and Open Source Days in 2006.

On the other hand, there are significant motivational differences between European and Turkish hackers. Results of research (Ghosh et al., 2002) in EU are presented in the Table 4.1.

In the interviews, I asked open ended questions. For this reason, it is not possible to find one-to-one correspondence. However, Table 4.2 at least gives the tendencies of the Turkish hackers.

The first difference between the results of these studies is Turkish hackers’ are

more inclined to social motivations. Indeed, the EU report only included FOSS developers in its research. On the other hand, in this research, interviewees belong to a wider spectrum. Furthermore, 100% of the hackers, who were not developers, declared their interest in social aspects of FOSS. Yet, if the interviewees were only developers, social motivations of the Turkish hackers would be also higher than the Europeans.

Second difference is lack of monetary motivations in Turkey. Some hackers earn money in the process of contribution-reputation-money. Nevertheless, they do not contribute to the projects for money. Additionally, there are only few FOSS firms which may employ these hackers and only a small fraction of them monitor hackers' activities for employment.

Last difference is nationalism. Despite some government's national software policies on FOSS, supporting FOSS in the name of nationalism makes Turkish hackers distinct. Percentage of the hackers who are motivated by nationalism may seem inconsiderable. However, hackers who talked about Uludağ project with nationalist terms was more than this percentage. Furthermore, after announcement of the project, many new people started to pay attention to FOSS because of its national aspects.

In this context, although most of the hackers are not under the influence of nationalism, it can be argued that most of the hackers who contribute to the Uludağ project are motivated by nationalism. Continuity of their contribution may alter the formation of FOSS in Turkey. However, second possibility is more probable: Internationalization of FOSS may affect their national feelings in the opposite direction.

Although it is not observed from the comparison of tables, another big difference between Turkey and Europe is the organizational ability of people. For example, in most of the LKD panels and meetings, if the LKD's board of directors was there, one of the frequent question was,

Hello,
We are students from X university. We want to organize at the university as GNU/Linux users. Does LKD have future plans for organizations in the universities?

As one of LKD activists told people organized by themselves in Europe and Turk-

ish people had less organizational ability than the European. On the other hand, as I observed in my participation at the LKD work groups, they overcome the problems of organizational ability.

In a work group, they plan their work in detailed and document it for future activists. In case of any problem, they quickly reorganize in order to overcome this. For example, they create sub work groups immediately for the problem or they review the work process for finding problems and clearing unnecessary steps. In this sense, they organize as they program: Divide the long functions into sub functions and remove unnecessary statements in it ⁴⁴.

⁴⁴ Indeed, this process is called as refactoring in software development.

CHAPTER 5

FOSS Business in Turkey

5.1 FOSS Firms

At the seminar *Free Software Economy*¹ seminar:

computer engineering student (A): Linus had told that one should not aim to earn money at first. In other words, money should not play prominent role for a programmer in FOSS development process.

Speaker (B): No, no... Free Software is not against commerce. You must take Linus's words as 'FOSS developers have different motivations from proprietary software developers. In this sense, people do not program for money, but they enjoy programming; money comes as a natural consequence...'

A: Please do not understand me wrong. In fact, I am anti-capitalist. But I must earn money to live. You are telling something different from capitalism. It does not seem rational. For instance, sometimes programmers develop software and give it away without receiving any money. Yet, some of them has donation links in their web pages. Is there really any donations?

A well-known Turkish hacker (C): Yes, I got. In recent days, I got money as you wondered. I had just developed a software package, not for money, just for fun. Someone had used it and earned money. He wanted to give me money...

B: But...

[It was hard to convince the computer engineering student. Speaker and some of the audiences tried to convince him about the rationality of FOSS economy. But neither of them succeeded. At last, another hacker, who is overweight, stood up.]

C: Do I seem starveling? I earn money from FOSS. As you see, (shows his huge body) I live on.

This event is not an exception. In another seminar, named Introduction to GNU/Linux and Free Software philosophy,²

¹ Fourth Linux and Free Software Festival (May 19-22, 2005)

² Free Software&Open Source Days/2005 Bilgi University

Audience: I am an IT professional and unemployed now. If I want to earn money from FOSS, how can I earn?.

Speaker: First of all, you must give, then you will receive. Participate in the FOSS community. You will get what you give by the time.

[This time speaker himself was a little overweight, previous scene repeated]

Do I seem starveling? I earn money from FOSS. As you see [shows his body] I live on. [Additionally as I heard from the students of Hacettepe University, another speaker proved one could earn money from FOSS in the same way at a seminar in their university³.] ...

I knew the speaker from the LKD mailing lists and I had no doubt that he had studied FOSS literature. He was talking about the gift economy. Soon after I realized that gift economy was the direct result of hackers' practice. They participate in the FOSS development and localization projects, give seminars and answer the technical questions of the list members voluntarily. As a result of their contributions, they gain reputation among FOSS advocates. Consequently, this reputation becomes job reference for their future careers. For example, when a firm needs a GNU/Linux system administrator, archives of the LKD lists help it who is the most suitable candidate for the job position.

Nevertheless, the question, "how do you make money from FOSS?" is an unavoidable question of "Introduction to GNU/Linux and its philosophy" seminars. In these seminars, I observed the audiences' first impression about FOSS and talking about it after the seminars helped me comprehending FOSS from different perspectives. For example, when I talked with the computer engineering student and his friends after the seminar, they were still not convinced about feasibility of FOSS economy. It was also an essential point that computer engineering students were not much interested in FOSS and only a few computer engineering students attended the seminars.

Since, the gift economy, a contingent exchange process, does not sound rational for a Turkish computer engineering student who generally prefers computer engineering department because of its higher income than other professions. Moreover, some computer engineers and IT professionals are afraid of FOSS. For them, FOSS

³ Introduction to GNU/Linux and Free Software philosophy seminar, January 5, 2005

represents collapse of software industry and they suspect that increase in tendency towards FOSS will make them leave the field because FOSS is often understood as unpaid software. Raymond (2000) emphasizes the contradiction between hacker culture and people who learned software development outside FOSS. Their practice and way of earning money is contradicting with practice of hackers. Impact of practice also revealed itself in the seminars that I organized and gave. In the seminars that audiences were mostly from IT field, audiences were eager to learn new business models, but hard to be convinced about validity of these models despite of examples they easily understood. Because, they were used to earn money selling and developing proprietary software. On the other hand, when the audiences were not from IT field ⁴, they were more convincible about FOSS business models. They listened the speaker and accepted what he said because of lack of practice in proprietary software development. In other words, their inexperience made them more convincible about the viability of FOSS business models.

Furthermore, another fact that makes FOSS business models disputable is the limited number of FOSS firms in Turkey. Moreover, it is hard to give definite number of the firms in Turkey because of their temporariness. For example, on June 16, 2004 Turk.Internet.Com ⁵ announced⁶ the new FOSS firm, *Penguen Yazılım*. The news story was mostly about its FOSS business model and summarizing its revolutionary approach to the software business as selling support rather than license. Unfortunately, the firm could not become successful and resigned from FOSS business. According to a FOSS businessman, Penguen Yazılım's story was not an exception and there were more failures. However, he added "There is risk in any IT business. I also know many proprietary software firms which burst. The real problem is the immaturity of Turkish Software Industry".

Alican (2004) also points out the problem, underdevelopment of Turkish software industry. He mentions the major role of international companies in ERP (En-

⁴ They were not from IT field does not mean that they do not understand IT. They were good at terminology, but just working in other fields.

⁵ A popular IT News Portal in Turkey (<http://turk.internet.com/>)

⁶ Retrieved January 15, 2006, from <http://turk.internet.com/haber/yazigoster.php3?yaziid=10245>

terprise Resource Planning) software. Alican (2004) writes that according to some specialists, international companies' increasing interest are expected to impact Turkish firms negatively. However, Alican (2004) also emphasizes the increasing trend in the world towards GNU/Linux against Microsoft. Indeed, in 2005, except one firm, IT firms proposed GNU/Linux based solutions to the e-Government project⁷ which was an important moment in the development of the FOSS in Turkey.

In this context, in the near future, one can also expect convergence between the contemporary software business and FOSS business. In this process, external factors (as penetration of international companies) and internal factors (Turkish software industry's solution against complexity) will be important. For this reason, as FOSS businessmen, software engineers are also included in the research. In the following paragraphs, software engineers' view on FOSS is discussed, as well as emergence and development of FOSS business in Turkey.

There is a huge mail archive⁸ of Turkish hackers since December, 1993. In this archive, one can observe the evolution of hackers relationship with the market. In order to observe this process, I searched for some keywords (e.g. commerce, money, business, market, capitalism, socialism) from the list.

Until the late 1990s, there are not any signs of FOSS as business models in the archive. In one of the earliest mails, one wondered whether people would pay for Linux or not in the future⁹. In another mail, one questioned the idealism that one shared his/her knowledge with the other members of the list, but some used it to earn money without sharing any knowledge¹⁰. Furthermore, most of the young hackers were students. Thus, they did not write that firms or home users should use GNU/Linux. Rather, they wrote about their school e.g. if our university gave up windows and used GNU/Linux, it would save money.

⁷ Retrieved January 15, 2006, from http://www.btdunyasi.net/index.php?module=news&news_id=1638&cat_id=4

⁸ Retrieved December 5, 2005, from <http://liste.linux.org.tr/>

⁹ Retrieved December 5, 2005, from <http://listweb.bilkent.edu.tr/linux/01/0177.html>

¹⁰ Retrieved December 5, 2005, from <http://listweb.bilkent.edu.tr/linux/03/0989.html>

In this context, announcement of Gelecek Company in 1999 was a turning point¹¹. Content of the message was assertive that saying it was the first Linux firm selling support for Linux. Some hackers warned the owner of the message that they were not the first firm. Latter, he responded that it was the first firm with its full and high quality support. Whether it was the first or not, this announcement is a sign of transformation in Turkey. Hackers have started to consider FOSS as a way of earning money¹². However, despite some firms in the world, 1999 was a very early date in Turkey. In the interviews, I asked one of these entrepreneurs: "How did you decide for a FOSS business? Did you imitate a firm, e.g. Red Hat or another FOSS firm?". He said:

No, we did not imitate any firm. We are talking about 1999. In these days, FOSS business was not very well-known as today. We just grope for a business and failed [with a smile in his face]. Can it be different? I do not think so. Not only FOSS business, but IT itself was very very new in Turkey. Ours was just an early attempt.

In 2000s, earning money by selling support for FOSS was a natural business model and some of the members of the Linux discussion lists established their own firms or worked for the firms established by the other hackers.

Additionally, in order to conceive software engineers' attitude towards FOSS, I also subscribed to a mailing list of software engineers in Turkey¹³. There were also different mail discussion lists about software engineering. However, this list was both quantitatively¹⁴ and qualitatively¹⁵ superior than other lists. On January 15, 2006, there were 2753 messages in the list and interestingly 5th message of the list was titled "Open Source and free Software Development Platforms". Yet, the word 'free' in the mail had meaning "you do not have to pay any money", not in the sense of FSF uses of the term free. In the latter messages, FOSS was discussed in the same

¹¹ Retrieved December 5, 2005, from <http://listweb.bilkent.edu.tr/linux/11/1541.html>

¹² Date of the transformation coincides with the graduation dates of the hackers in the mailing list archive.

¹³ see <http://groups.yahoo.com/group/YazilimMuhendisligiTurkiye/>

¹⁴ The list has been established in 2004. In January 2006, the list had 1048 members.

¹⁵ There were well-known academics and businessmen in the list.

Table 5.1: Categorization of the FOSS firms in the study

Category	Description	# of firms
A	established or managed by the hackers	6
B	sell support for FOSS products but not managed by hackers	2
C	Sell both proprietary software and FOSS support	3
D	Sell support for specific GNU/Linux distributions	3
E	Develop FOSS	3

sense, "You do not have to pay any money to use FOSS. Then, why do you insist on using Microsoft products?". On the contrary, on June 29, 2005, 1466th message of the list was different, it has a title "Open Source Meets Venture Capital ". After this message, software engineers started to discuss if one could earn money from FOSS. Some of the firm owners tried to mention that a firm was established to make profit and questioned the viability of FOSS business. An argument against the nature of the firms was interesting: "Profit? Is it everything?". Later, owner of the argument talked about its philosophy, sharing with and helping. However, nobody gave arguments about FOSS business models and ways of earning money from FOSS.

Discussions in both of the lists gave insights about development of FOSS business models in Turkey. The interviewees were mostly selected from these lists. Questions were answered by managers or owners of these firms. Before the conduction of the interviews, no categorization was made. However, answers of the interviewees enforced a categorization for a more clear presentation of the research (Table 5.1 ¹⁶).

As discussed in the section 2.2.3, for firms, FOSS is a business strategy. For this reason, one can not speak of altruism of the firms to describe their relation with FOSS (Bonaccorsi and Rossi, 2004). However, one might expect to see similar traits of hackers' relation with their work on firms' relation with FOSS because of two reasons. First, businessmen who are in Category-A have a hacker background. Secondly, ecosystem of FOSS may have some effects on firms and may enforce them to comply with the hacker culture for making profit.

Nevertheless, it is hard to speak of homogeneity of the FOSS firms. Since, atti-

¹⁶ In the table, Categories A, B and C are discrete. However, Category-D's and Category-E's members are found either in Category-A or Category-B.

tudes of the firms and the traits of hackerdom on them varies in each firms according to the type of their relation with FOSS. Furthermore, their responses to interview questions and behaviors in mailing lists/seminars/meetings had significant conflicting. For example in some meetings I attended, FOSS businessmen were more inclined to talk about monetary sides of FOSS despite hackers' (especially young ones) passionate telling. For example, in a meeting, a young hacker started to talk passionately about beauty of "his" GNU/Linux versus his old experience with Windows. The FOSS businessman interrupted him:

Please... We know! Let's not tell each other about FOSS. One just satisfies himself by doing that. It is also an economic phenomenon. If we want to make FOSS widespread, we must tell its economic benefits. Indeed, we must not tell it each other about sustainability of FOSS business. Instead, our target has to be people, who have never heard about it.

Then, people started to talk about FOSS and business. Attendants of meeting talked about relation with firms and complained about its inadequacy. In the discussion, the FOSS businessman was passionate about FOSS, indeed, more passionate than the young hacker he criticized. He was the most participating person in the meeting. He told that he had chosen FOSS not because of prettiness of the penguin. According to him, people did not take up FOSS as a profitable business, in fact, it was. He told his business experiences on FOSS versus proprietary software. After the meeting, I got an appointment from him for an interview.

Before the interview, I expected to get answers of a businessman rather than a FOSS advocate. Indeed, the interview began as expected:

I do not only know Linux. Before Linux I was using Windows. But I chose Linux rather than Windows. Because it is better. And for a firm, like ours, it is profitable. There is zero-cost about licenses. For example, think about installing windows. Everybody can install it, e.g. your neighbor's son may know clicking next, next buttons. When you sold a computer with windows, you get small amount of money. Most of the money goes to Microsoft. But when you sell Linux installed computer, you get more than this. Sometimes great amounts. Additionally, Linux needs less hardware resources. Thus, when our firm makes old computers usable, we get money just using our labor. No money to either hardware vendors or software suppliers. OK, penguin is pretty and social relations among FOSS community are attractive. But people are always utilitarian. Only, relation of parents with their children is an exception. No, no. Sometimes there is also utilitarianism in this relation. Parents look after their

children, since they want to be looked after when they are old. I am interested in Linux. Because there is an utility for me. Because there is money that I can not make from proprietary software.

At first, it seemed as a genuine confession. As a rational businessman, making profit was his primary goal. In other words, he was acting as any businessman, in which money was an end in itself. A common point among professional FOSS hackers and FOSS businessmen in Turkey was that they were eager to talk about profitability of FOSS. On the contrary to Hardin's (1968) Tragedy of Commons, they consider an additional firm as a desirable phenomenon. For them, additional firms are not rivals if the new firms do not follow a business strategy excluding others. Since an additional firm increases possible support for FOSS in Turkey, and increasing support means persuasion of new users for migration from Windows to GNU/Linux. Later, the businessman returned to the beauty of FOSS, freedom, and interrelations among FOSS community:

Yes, money is important. But in our relations money is not the primary motive. I came to this meeting from another city. Instead of working at my firm, I am here to help the organization. I did not earn any money here, instead I spent money, e.g meal, travel, hotel expenditures.

After this, he talked about Turkey, FOSS and necessity of a national IT policy and FOSS's prominent role in it.

In addition to talking about profitability of FOSS, second common point among professional FOSS businessmen in Turkey was that they were also eager to talk about the importance of giving back or more truly, giving before getting something from FOSS (as in gift economy). In this sense, one may argue that attending national meetings, festivals and congress is important for firms for gaining reputation among potential customers, in other words, it is a type of advertisement. This is partly true and this is the best way for firms to be known. Especially for support sellers. Nevertheless, it is an obvious fact that FOSS businessmen are not excessively fond of money. They are under the influence of both hackerdom and motivation of making profit that each pulling in different directions (Himanen, 2001).

In fact, Turkish FOSS businessmen's behavior is not different from the international FOSS firms. In particular, Google's slogan summarizes FOSS firms strategy in

the market: Don't be Evil.

You can make money without doing evil.

Google is a business. The revenue the company generates is derived from offering its search technology to companies and from the sale of advertising displayed on Google and on other sites across the web. However, you may have never seen an ad on Google. That's because Google does not allow ads to be displayed on our results pages unless they're relevant to the results page on which they're shown. So, only certain searches produce sponsored links above or to the right of the results. Google firmly believes that ads can provide useful information if, and only if, they are relevant to what you wish to find ¹⁷.

As a business strategy, by not doing evil, firms consider long-term relationship with the users rather than destroying trust relationship with them for short-term profit maximizations. However, besides being a business strategy, businessmen's social relationships, their past and present experiences are also important facts. Hence, besides being businessmen, Google's owners Brin and Page were also hackers and this played essential role in the development of Google (Vise and Malseed, 2006).

Similarly, businessmen's past experiences and the business models they chose were also strongly related with their answers in the interviews and their postings for discussions on Internet. For example, FOSS businessmen (Category-A), who were amateur hackers in the past and activists at the moment in LKD, were more inclined to think as hackers. Their answers were almost parallel to hackers that are discussed in the previous chapter. On the contrary, businessmen (Category-C), who did not have any hacking experience and participated in FOSS just because of increasing demand, were more interested in earning money. Moreover, when I asked them about their hacking experience, they understood it as cracking passwords. They answered "No!" in a strong tone and reject any notion of hacking. When I explained them the meaning of hacking, they were all right. Yet, some insisted that I should not use hacker and advised me using programmer or software developer instead of hacker because of hackers' bad reputation.

On the other hand, firms' FOSS business model is an important factor in their relations with FOSS. What they do is also conditioning their tendency to either hacking or profit maximization. For example, firms whose business also includes de-

¹⁷ Retrieved April 5, 2006, from <http://www.google.com/corporate/tenthings.html>

veloping software or businessman who himself develops software answered more consciously and also hackerly, e.g. considering trust, collaboration in business as much as making profit. On the contrary, when the business model did not contain any type of developing software, businessmen's (Category-B) answers were more inclined to considering their business as a way of making money, even though, this inclination was less than Category-C businessman.

Furthermore, types of FOSS business models that are either widespread or rare in Turkey are also important in understanding development of FOSS business models in Turkey. For example, despite proprietary software firms' increasing interest, these firms are more inclined to use FOSS in their projects rather than developing FOSS. In other words, firms still do not prefer or know to benefit from the outside hackers. Consequently, these firms are more inclined to earn money as in proprietary software business: Selling right to use licenses. This inclination has two reasons.

First, FOSS business models are very controversial for most of the IT professionals. When computer engineers, out of the FOSS world, were asked about business and FOSS, they talked about irrationality of giving away the source code and they were really considering the source code as crown jewels:

Well, FOSS is good. It may play a prominent role for Turkish software industry. But not yet for earning money. Tell me how can I? OK, it is ethically good. But business is business. What would you do if another competitor take away your source code, and rename it? Is it rational? Secondly, development of project as FOSS will also make your future strategies and plans transparent to all. What about competition? Lastly, do you really think that customers in Turkey are capable enough to understand the source code? Are our programmers capable enough to understand the source code of Apache or Linux Kernel? NO!

Second reason is external to the software engineers. One of the interviewees was both a FOSS businessman and a software engineer. He explained the problem from the perspective of public sector:

As we know, people can not make money by selling FOSS Licenses. FOSS business is a service-based sector. You sell complementary services for FOSS products. Is it possible in Turkey? No. When you want to make business with public sector, you confront with lots of problems. Despite significant progress in public bids, default laws are still problem when you want to make business with public.

Moreover, it is not very easy to persuade managers of the IT departments in the public sector. They are escaping from any risk. Indeed, I can not see any risk in FOSS. But what is new is also risky for managers of the public sector.

As I learned from the interviewees, there are proprietary software firms which modify source code of FOSS and sell it to the public sector as proprietary software. For example, there are many continuing software projects (including ERP, Content Management, accountancy software etc.) at the web sites, sourceforge.net and freshmeat.net. However, there are also many software engineers who do not benefit from these projects. Some of them did not hear about this site and some of them were not sure about maturity of these projects.

In this context, the most widespread FOSS business model is selling support in Turkey. It includes selling FOSS media (e.g CD-Roms, DVD-Roms), education, offering technical support and customizing services to users (Weber, 2004). In this business model, firms do not directly contribute to the FOSS. Instead, they supply complementary services for the existing FOSS.

Selling CD-Roms, Linux hats or T-Shirts is the easiest business model. For example, although FOSS is downloadable from Internet, most of the users in Turkey do not have good connection capacity and they prefer buying CD-Roms. In this context, this kind of support sellers do not directly give back anything that they get. However, they give very importance to FOSS hackers and carefully avoid from having "greedy salesman" image as the strategy "Don't do evil" . For this reason, even though they do not contribute to FOSS in terms of source code, by different types of campaigns, they try to establish good relationships with the FOSS hackers. Additionally, they supply web hosting and developing web sites for customers. When they were asked about the reason why they chose FOSS, they were not capable enough explaining it in business terms. They talked about social and national aspects of FOSS and prettiness of FOSS.

Increasing popularity of FOSS in the world, increased interest for FOSS in Turkey. IT professionals wanted to learn more about FOSS for their future career. Furthermore, corporations wanted to migrate from proprietary software to FOSS and their staff who were used to Microsoft products needed education for using FOSS. FOSS

education as a business became widespread at this moment. This kind of FOSS firms are usually established by highly skilled hackers and give very importance for good reputation in Linux users' mailing lists. When they are asked about "why FOSS?", they talked about technical sophistication of FOSS. Additionally, with a smile, some said, "I started using computers with FOSS. I had no chance for another software business". One of the FOSS education firms declared that it would publish some of its education material with free documentation license. On the other hand, there are also firms which give proprietary software's education as well as FOSS education. They avoid contrasting FOSS and proprietary software and explained their business as "there was increasing demand for FOSS, so we added FOSS courses". They are not insiders of FOSS hackers in Turkey. They hire hackers but their answers in the interviews showed that they were more inclined to talking profitability of FOSS. They preferred to talk about increasing demand and strong avoid from contrasting FOSS and proprietary software. They said both have specific advantages and criticized any fanaticism among hackers.

As offering technical support, customizing services to users are also widespread in Turkey. These firms are often established by hackers and usually managed (at least their technical departments) by hackers. These firms' managers' answers were usually parallel to hackers who are discussed in the previous chapter. They explained their business, briefly, as:

I was hacker. I had to earn money and wanted to earn money from something that I enjoyed. So I preferred FOSS. Fun and money is together.

In addition to this subjective fact, they explained the technical advantages of FOSS over proprietary software. They are more conscious about hackerdom than the other businessmen. For example, when I asked them about hackers, there was a significant gap between category-A and Category-C firms. Category-C (and sometimes Category-B) businessmen's reaction to the word, hackers:

We are on the opposite sides. We want security and they want to break our systems. No, no... I am not a hacker.

But when I explained them about what I mean by the word hacking, they answered defensively:

OK, OK... Of course there are hackers in our firm. I also like programming.

Yet, Category-A businessmen firstly asked:

Hacker? What do you mean by it? There are different uses of it. Is your research about security? Or art of programming?

After explanation, they talked proudly:

Yes, I have a hacker background.

And their answers were highly parallel to hackers who were not businessmen. Another interesting point was that businessmen who also had programming activity (Category E) in business were more inclined to hackerdom. Additionally, they were the only FOSS businessmen whose answers were consistent about what they do as a business and their description about it. For example, when they are asked about:

Is there any international firm that you imitate its FOSS business model?

They give examples from famous FOSS firms, e.g Red Hat, Cygnus Cooperation or from less known ones. They were conscious about the business model they chose.

There also FOSS business models which are rare in Turkey. In this context, another business model is "Loss Leaders". Firms make their product FOSS and earn money from linked products. Firms may choose this model because of different reasons. They may want to gain reputation with their FOSS product and sell linked commercial products as proprietary software (Weber, 2004). However, this business model is also determined by consumer side. The product that is made FOSS must provide additional value for the customer. In other words, customers should be aware of FOSS. It must be expected that increasing awareness about FOSS will also increase loss leaders business model. There are also examples of this business model in Turkey. Indeed, one of them sent me manuals about their product. It was neither related with GNU/Linux nor any well-known FOSS tool. It was developed for and by Microsoft platform which revealed that FOSS business models were not constrained to GNU/Linux.

Another rare FOSS business model is "sell it, free it" which is similar to loss leaders. Firms may choose to make their proprietary software free or open source. But Hecker warns about this:

If your software business is doing fine, and you have neither problems you need to solve nor potential opportunities you want to take advantage of, then I really can't recommend you consider an open-source strategy; adopting such a strategy incurs definite costs in money, time, and effort, and I see no point in urging it on you if there's no benefit to be gained (Hecker, 2000).

Moreover, firms as Netscape chose to make its browser FOSS when it had almost lost against Microsoft's Internet Explorer. If Netscape had not almost lost competition against Microsoft, they would not have made their source code free. This type of business model may be observed in case of competition. In this sense, this strategy may also be put into practice in Turkey's market against the foreign ERP firms.

"Widget Frosting" FOSS business model is also similar to loss leader. However, at this time, firms sell hardware as a commercially linked product instead of software. Developing drivers, compilers or customizing operating system for hardware can be categorized in this model (Weber, 2004). But this business model is applicable to firms which produce computer hardware e.g. IBM, HP. Thus, it is inapplicable in Turkey. On the other hand, there are firms which supply complete IT solutions and use FOSS because of its licenses. Additionally, some computer sellers started to sell GNU/Linux installed computers because of the high price proprietary software licenses. In fact, BSA (Business Software Alliance) have major role in firms' practice about selling GNU/Linux installed computers. In this context, while the government is insisting on the IPR, firms are becoming more inclined to use FOSS.

Each of the above business models is a complementary service that adds value to FOSS which customers may pay for. Accesorizing is also this type of business model. Manuals and books are sold for easing use of FOSS. Partial information on English manuals are gathered, translated into Turkish and presented as a whole book for Turkish readers. This business model is performed by hackers who have good reputation among FOSS hackers.

On the other hand, business models as selling integration services are rare because of the reasons as discussed above: Software engineers ignorance and limitations of public bid law.

Besides Turkish firms, there are also two international companies in Turkey. Novel

and IBM. As their global policy, they are also promoting FOSS in Turkey. Although Novell Turkey's presence is new in Turkey, IBM Turk has been in Turkey for a long time and it is trying to establish strong relationship with the hackers. Furthermore, IBM Turk is encouraging public sector for FOSS.

5.2 Firms and Licenses in Turkey

In FOSS business, firms may take advantage of outside hackers. Every hacker is a potential developer of a firm's project. While hackers' direct code contribution is important, testing is also an important part of software development process. As hackers consume software, they also re-produce it by finding (and sometimes fixing) its bugs and deficiencies. Raymond (1999b) summarizes this process as "Given enough eyeballs, all bugs are shallow." Furthermore, localization works play major role for FOSS business in two ways. First, if a firm aims at the global market, it should internationalize its product and localize it for different countries. Thus, a firm has two chance: localizing by itself or encouraging FOSS hackers to localize it. Second, if a firm sells integration services, it will need localized version of the components that it integrates. Additionally, selling support business needs localized software. In Turkey, firms generally aim at the local market. Hence, second case is more essential for them.

In this part, relationship between the firms and hackers is studied from two vantage points. First, it is studied from the vantage point of "discrepancy between attitudes and behaviors in open source firms" (Bonaccorsi and Rossi, 2005). In this respect, firms' "social links with the community, involvement in OS advertising, and participation in OS projects" are investigated. Second vantage point is the FOSS licenses that is particularly about firms' perception and use of licenses.

In this respect, first of all, firms' managers were asked for if they studied on the tendencies of computer users in Turkey. Their answers were usually "No", but they added, "I can predict what they want as soon as I see them. It is not very hard." After this question, I also asked about the same question in the context of hackers. At this time not only for Turkey, but generally:

Do you have any research about the hackers? What do you pay atten-

tion in the relation with hackers?

Beyond a few "Yes", answers were again "No". Owner of "Yes" answers were hackers. Even though some hackers did not read about hackers, they were good about predicting their attitudes. They could easily predict what kind of work they liked. One said:

They like to work. But not every time. Work is fun for them. If there is neither fun nor challenge, they do not work properly. I was also a programmer once upon a time. Now, I saw that managing people is much harder than programming. I do not and can not make them perform programming tasks they do not enjoy. Furthermore, sometimes I want them to do something that is in fact useless for me. Just for fulfilling their passion. . . I have to do that for good creative work that I need later.

Others talked about flexible work time and one of the businessmen went on one step further and claimed that they also had flexible work place.

We have two offices. Hackers can choose either to work. They may come to work at any time. They can also work at home. They must do his work on time. It does not matter where and at what time of the day they did. In fact, we also had flexible payment system. Our programmers get money according to how much they produce.

They said their staff could do anything at work time. For example, they could study on another subject¹⁸ or check mails. But they also emphasized that "Of course, he must do his work on time." On the contrary, Category-C firms reject those kind of flexibilities. For them, no firm gives permission to this and this is contradicting with the business rules and professionalism.

In this context, it is interesting to see hackers' and firms' agreement on flexible time and work place. In fact, it is a compromise between them which is based on fun at work for the hackers and creative products for the firms. Hence, especially Category-E firms, which need more creative products at work, endure with hackers' relation with their work. For example, managers of these firms know "Hackers do not like routine. They are reluctant to document their work. Some enjoy working alone at home. Some likes working at night till the morning and sleeping at normal work hours." These are the rules of working with hackers for Category-E firms.

¹⁸ e.g. If one is a network administrator, one can study learning web programming.

Thus, when they need to work with the hackers, "They employ additional programmers who program routine development works, document the product and work at normal work hours."

On the other hand, Category-A firms also accept work-style of the hackers. Yet, they are less pragmatic than the Category-E firms and their acceptance is mostly concerned with their background. In addition to their background, they spoke of gifting back to the FOSS.

After this question, I asked about their relations with outside hackers:

How is your relationship with the FOSS community in Turkey? Do they support your firm? Do you support them? Do you have any budget for support?

Except two firms, none of the firms had a budget for FOSS, but they said they supported FOSS by being sponsor in their activities or helping users on mailing lists. Especially Category-A firms were sure of giving back what they got from FOSS. In this context, while Category-A firms considered this question from the perspective of supporting the community in any way, managers of Category-B and Category-C firms comprehended the question as, "Do you sponsor FOSS activities?" and explained their support in monetary awards. In this sense, Category-A firms often answered this question in terms of helping users on forums/ mailing lists, giving seminars and localization of manuals.

In this context, I should also add that interviews with the managers of Category-A firms occurred in a more friendly tone than the other interviewees. For example, they told that they were happy to attend in the interview and answered eagerly. Additionally, we talked in a friendly tone¹⁹. On the other hand, interviewees with the businessmen in Category-B and Category-C were more formal and they answered in short sentences.

In this sense, Category-A businessmen's answers were parallel to the hackers which were discussed in the previous chapter. It was usually impossible to understand if I talked to an amateur hacker or a businessman. Thus, I asked them:

Did you have any conflict being either a hacker or a businessman?

¹⁹ For example, except two of them, we use the word 'sen' rather than 'siz'

One answered this question with a quotation from Linus Torvalds:

I do not have anything to speak with Bill Gates. We have no common point. I am a hacker and he is a salesman.

After this quotation he told:

But of course, as every one, we have to earn money and people do not do what they enjoy in every time. So I am not so bothered about this situation. Yes, it is true our customers are not sophisticated computer users as hackers. Indeed, our business is that. FOSS firms play a complementary role. Firms do what hackers do not like for money.

Sometimes we also have deadline problems with our customers. Yet, our customers trust us, they know that they will get better.

Firms usually consider their business as a complementary service for FOSS. In this sense, they change the landscape of FOSS, from "from hackers to hackers" to "from hackers to users by the help of firms". They think that synthesis of FOSS with business made it a real alternative for proprietary software. In this respect, they think that the contradiction between hackerdom and business results a positive situation. According to them, FOSS is technically better than proprietary software and FOSS business models introduce this superiority to the customers.

Firms' attitude towards Microsoft was also moderate. There were only few answers that see Microsoft as an enemy ²⁰. Yet, they also added that Microsoft was only one of the enemies and one should not miss other proprietary software companies. On the other hand, Category-A firms general attitude towards Microsoft can be summarized by one of the professional hackers' words:

We earn money from software. So does Microsoft... Then what is the difference? Nobody can deny that Microsoft is one of the best companies in the market. What they do is developing what the customers want. Be careful! I say what the customers want. This means attractive GUIs. Of course, this market oriented development results in bugs for some time.

On the other hand, for FOSS hackers, the problem is not the customers' needs or expectations. They focus on the technical issues, such as security and modularity. The result is technically better software. But who knows this? Just we know. Since we are able to analyze software in a technical perspective. In fact, rule of FOSS business is keeping this problem in mind and making FOSS for everyone.

²⁰ Yet, it is an entertaining enemy. For example, one of these businessmen had put a Microsoft logo in his web page before he became a businessman. However, if you had clicked on the Microsoft logo, you would have reached web site of Playboy instead of Microsoft.

However, firms which sometimes sell Microsoft products told that they²¹ are businessmen and even though they did not like it, they had to sell it when customers wanted. But they were sorry while saying this. On the contrary, category-C firms were neutral between FOSS and Microsoft. They said that they were strongly against any technology fanaticism.

In the later questions, I investigated FOSS businessmen opinion about the terms 'free software', 'open source' and 'unrestricted software'. After this, I compared their answers with their FOSS license preferences. The term, 'open source' has been started to be used for persuading business world about benefits of 'free software'. In this sense, I was almost sure that when I asked the firms "Which do you prefer: Free Software, Open Source Software or Unrestricted Software?", the answers would be Open Source Software or Unrestricted Software. Yet, I was wrong.

Category-A firms were more inclined to use free software. When I reminded them they were sometimes using Open Source Software on mailing lists, they told that it was unconscious use and started to talk about freedom in the sense of hacking.

However, using "Open Source Software" unconsciously is related with the circumstances in Turkey. Some businessmen gave similar reasons as the hackers in the previous chapter about why they should use open source software or unrestricted software rather than free software: Freedom is a political term in Turkey which might not match with business.

For this reason, while the hackers in the world search for clarification for the double meaning of free and use instead of open, both businessmen and hackers in Turkey did the opposite of this. Despite clearness of the translation of free in Turkish²², they invented the word, unrestricted. I traced back this term and at last I reached the first users of the term. The common point of the inventors is they are businessmen. Even though they prefer to use free software at the moment, they admitted that they used unrestricted instead of free for political reasons. When I asked one of them about the reasons for changing his mind, he talked about the democratization of Turkey and EU's affect. He concluded that talking about freedom in Turkey was

²¹ Category-B firms

²² In Turkish, Özgür

not a problem anymore.

On the other hand, despite widespread use of the term free software in Turkey, people prefer using closed-source to describe opposite of either free software or open source software instead of using the term, proprietary software. There are not any widespread translation of proprietary into Turkish. Two different attempts were made by Uludağ Project members at Bilgi University's fourth Free and Open Source Days . First attempt was made by a member of the project as 'possession software'²³. On the other hand, project manager, Erkan Tekman proposed using 'owned software'²⁴.

In the interview, I used Erkan Tekman's translation and asked about 'hybrid models' (mix of FOSS and proprietary software) possible impacts on business and hackers. Although it is an international debate²⁵, I could not get satisfactory answers. Again, Category-A firms easily comprehended the question and when some of them looked through the interview questions before I asked, they thought that I was aware of internal discussions of FOSS and became more respectful. But this was an early question for them. Because most of them were support sellers and software developer firms' business was not large enough that could make it a fierce topic for them. Thus, I changed the question and asked if they earned money from proprietary in addition to FOSS.

There were Category-A firms which were working with only FOSS. They were also well-informed about FOSS licenses and their answers about licenses were consistent. Category-B firms' answers were also consistent, but they were not against selling proprietary software as Category-A firms, but they were still sorry about selling proprietary software. On the other hand, Category-C firms' answers did not have any consistency. In other words, they could not explain their opinion about FOSS licenses clearly. They were relatively newer to FOSS than other categories of businessman and very confused about the licenses.

Furthermore, when firms were asked about government's intervention in FOSS:

²³ In Turkish, mülk yazılım.

²⁴ In Turkish, sahipli yazılım.

²⁵ see http://www.freesoftwaremagazine.com/issues/issue_003 for a detailed discussion.

Do you have any license preferences for the products that you develop? GPL, LGPL, BSD, Mozilla etc?

Assume that government puts a directive that public sector should prefer FOSS. Is GPL a right decision for it?

Category-A firms' answers were heterogeneous. They answered either GPL or BSD. Other firms' answers were inconsistent which revealed ignorance about FOSS licenses among FOSS firms. Nevertheless, almost all of the firms began their answer with rejecting any law about governmental support for FOSS. They said that government should not be side between FOSS and proprietary software. This rejection was very obvious for Category-D firms. These support sellers also criticized Uludağ Project. But other critics emphasized that even though they did not want any law about FOSS, Uludağ Project would increase number of users in Turkey and all FOSS firms would benefit from this. FOSS businessmen were sure of FOSS's superiority. In this context, they want only equality in public bids and think that government is favoring Microsoft²⁶.

However, firms which are not support sellers for specific GNU/Linux distribution, are more optimistic to government's intervention. According to them, Uludağ Project will increase awareness about FOSS.

In this context, FOSS business in Turkey can be described as a developing market. Uludağ Project is expected to play an accelerating role in FOSS business. This expectation is particularly based on support seller business model. This might seem unfair competition between other distributions' support sellers and Uludağ Project's support sellers. Yet, it is a good example, which shows different interests of hackers and firms. There is a significant difference between hackers' interest for Uludağ Project and firms' reaction against government's support for any GNU/Linux distribution.

On the other hand, Uludağ Project which is discussed in detail in the next part, was a good lesson about how to treat hackers and benefit from their hacking. However, firms' answers show that there is significant ignorance about hackers and their tendencies, except the firms which are managed by hackers (Category-A firms). At first, it does not seem so important for support sellers. However, real power of FOSS

²⁶ Retrieved April 5, 2006, from <http://www.turk.internet.com/haber/yazigoster.php3?yaziid=13965>

is free contribution of outside hackers that a firm does not pay for and Turkish firms are not capable enough about exploiting this power. Furthermore, the firms which are outside FOSS and develop proprietary software have not noticed its potential yet. Additionally, in the interview with software engineers and lurking on software engineering discussion lists, I observed that software engineers did not know much about hackers.

5.3 Uludağ Project

Uludağ Project is a good example of how to make people contribute to a FOSS project. Despite success of many FOSS projects, "most free software projects fail" (Fogel, 2005: 9)

...anecdotal evidence from over a decade in open source, some casting around on SourceForge.net, and a little Googling all point to the same conclusion: the rate is extremely high, probably on the order of 90-95%. The number climbs higher if you include surviving but dysfunctional projects: those which are producing running code, but which are not pleasant places to be, or are not making progress as quickly or as dependably as they could (Ibid).

In the book, *Producing Open Source Software*, Fogel (2005) discusses about avoiding failure. On that point, he examines right and wrong things in the FOSS projects. Fogel's (2005) examination gives hints about understanding success of the Uludağ Project. In this context, the word, success, may sound exaggeration, since there are not any widespread use of Pardus. However, it's a successfully managed FOSS project that succeeded in gathering hackers participation around the project and encouraging new people for prosuming Pardus. In this part, important milestones of the project are presented by the help of Fogel's (2005) arguments about successful FOSS projects.

Before launching a FOSS project, existing projects must be examined for not reinventing the wheel.

Always look around to see if there's an existing project that does what you want. The chances are pretty good that whatever problem you want solved now, someone else wanted solved before you (Ibid: 19)

Uludağ Project started as a result of necessity for a national operating system. Before the decision for developing a national distribution, TÜBİTAK UEKAE examined similar applications and tendencies of the software industry in the year 2003. After these examinations, they decided to create a GNU/Linux distribution. Otherwise, if there were not any need for Pardus, it would be very hard to attract attention of the developers.

According to Fogel (2005), the hardest part of a FOSS project is transformation of a private goal into a public goal. For the Uludağ project, it was the transformation of TÜBİTAK UEKAE's goal into the public goal. The project management team had examined "open source software methodology and philosophy in detailed"²⁷. They started with choosing a good name for the project, Uludağ, which was the abbreviation for national distribution. This name was good at summarizing the project's objective. Secondly, the project team declared its goal with a clear statement. They said what they would do and not do.

Uludağ is GNU/Linux distribution developed by UEKAE according to computer literates basic desktop needs; uses existing distributions' dominant parts as concept, architecture or code; provides easy use, configuration, installation with configuration environment and tools that can be converted to an autonomous system.²⁸

After this statement, in the seminars and presentations, it was also declared that Pardus, product of the project would not be a distribution for the hackers. In other words, it was clearly declared that it would not be a Slackware or a Gentoo as hackers like.

However, the most important step of the project was declaration of its license and methodology.

Project will be GPL'd free software project.
Project will be public, compatible with free software philosophy, use contributors' work²⁹.

In fact, this declaration is the main difference of the Uludağ from other local distributions. GPL was a clear message to the public: Pardus is yours! As I emphasized

²⁷ Retrieved April 5, 2006, from <http://www.uludag.org.tr/eng/hakkimizda.html>

²⁸ Retrieved April 5, 2006, from <http://www.uludag.org.tr/eng/belgeler/sozlesme/index.html>

²⁹ Retrieved April 5, 2006, from <http://www.uludag.org.tr/eng/belgeler/sozlesme/index.html>

in the previous chapter, FOSS licenses are not well-known in Turkey. However, in the panels and seminars, project management team explained GPL by telling its importance for a FOSS project again and again. It was an important step for good relationship with the hackers. Indeed, it has been one of the key goals which the TÜBİTAK UEKAE achieved.

On the other hand, the project management team was also conscious about when to build a co-developer community around the project. They conformed to the Raymond's statement about the time of building a co-developer community around the project and waited until a first release that "one can test, debug and improve." (Raymond, 1999b). Only after the release of the first product, "Pardus LiveCD", they focused on public relations. In order to do this, they established the necessary technical infrastructure for the project.

Website of the project was also active before the first release. However, they enriched the content of the site with two essential documents: *How can I help?*³⁰ and *New Developer's Guide*³¹.

First document is for any user, who wants to contribute to the project. It introduces the ways of contributions. In that respect, it defines the ways of prosuming as testing, translating documents (either from English to Turkish or from Turkish to English) and reporting bugs. The document was very helpful. Since there are always people who want to contribute to the FOSS projects, but either they do not know ways of contribution or think that they are not good enough to do anything for the project. The document overcame this problem and helped the emergence of new hackers by encouraging and guiding them.

Second document is complementary to the first one. It defines the ways of contribution in more concrete terms. It also aims at more advanced hackers by defining technical infrastructure of the project.

Mailing lists are the most important parts for technical infrastructure of the FOSS projects. According to Fogel, "It is butter and bread of project communication" (Fogel, 2005: 37). These lists of the project, at least make people informed about the

³⁰ Retrieved April 5, 2006, from <http://www.uludag.org.tr/eng/belgeler/kalite/index.html>

³¹ Retrieved April 5, 2006, from <http://www.uludag.org.tr/eng/belgeler/gelistirici/index.html>

developments. Moreover, while people help for each others' problems, they learn the hacker culture, in which collaboration is essential.

In addition to the mailing lists, another part of the technical infrastructure is the bug tracker. By this application, users can report bugs or request new features in Pardus. Indeed, it is also an important fracture for Windows users. The project management team has encouraged the Pardus users about contribution since the release of Pardus LiveCD.

Lastly, the project management team announced another way of collaboration around the project: Pardus Voluntaries Program. Aim of this program is introducing Pardus to users by the help of users. Indeed, this strategy is similar to Google's strategy³² that can be summarized as making the users of the product, its advertisers. Besides introducing Pardus to the new users, voluntaries should also introduce the philosophy of the free software to the new users.

5.4 Main Findings of the Research About FOSS Firms

As emphasized in the previous part, if a firm wants to benefit from the work of outside developers in its FOSS project, the hardest point is transforming the private interest of firm into public interest of hackers. Success of the Uludağ project is concerned with this transformation. Although it gives important lessons about how a firm can conduct this transformation, its practice can not be imitated by firms exactly.

First, TÜBİTAK UEKAE is not a firm. For this reason, it was easier for TÜBİTAK UEKAE to constitute a public vision for the Uludağ Project. On the contrary, firms have more difficulties in order to accomplish this transformation. Although hackers are not against commercialism, if there were a firm instead of TÜBİTAK UEKAE, hackers would consider deliberate before participating in the project. Both government's support and army's interest in the project persuaded hackers. At least, they were sure that it would not be a project managed by a firm whose future was unclear.

Second, nationalism eased the most important step in the project: gathering the hackers around the project. One of the key findings of the research about the hack-

³² For example in gmail, users told their friends about it by sending them invitations.

ers was the role of social motivations in their work. In this sense, if there were a FOSS project about blind people, there would also be significant contributions from the hackers. However, one should not expect great contributions from high-skilled hackers. Most of the time, there would be modest contributions from hackers who localize some of the components of the project, test software's features, report bugs of the software and prepare documentation for it. Yet, as Ghosh (1998) emphasizes, these modest contributions accumulate continuously.

In this context, it is more difficult for firms to constitute a public vision around social motivations. Firms' profit-oriented character is a crucial obstacle for motivating hackers. On the other hand, as in the case of Apache, Mysql or Firefox projects, this transformation is not impossible for the firms. In the previous chapter, some of the hackers³³ explained their disinterestedness for the project, with the lack of challenge for them in it. For them, building a GNU/Linux distribution from scratch³⁴ was an easy work that they had done before. Thus, there was nothing challenging for them. More challengingly, some hackers were also developing their own operating systems by themselves.

Furthermore, this was the problem of Netscape in its Mozilla project. Lack of challenge in the project made hackers leave the project for the sake of more challenging works in other projects. In that respect, FOSS firms may also benefit from this tendency of the hackers. For example, a well-designed software project of a firm, with its clear innovative and challenging tasks, can also attract hackers' involvement in the project. In addition to the challenge, there are also hackers who are merely interested in learning and developing new skills in the FOSS projects. In this way, a compromise is made between the firm that seeks profit and the hackers who are interested in the activity itself.

Indeed, both strategies, using social motivations and presenting challenging works, have a common point with the some of Turkish FOSS firms' relationship with their hacker employees as discussed in the second part of this chapter: always considering

³³ These hackers were also brilliant programmers.

³⁴ Linux from scratch is a technical term which means "building your own customized Linux system entirely from source" in a step by step approach. See <http://www.linuxfromscratch.org/>

the boundaries of working with hackers. In other words, managing FOSS projects requires knowing the types of tasks in the project that hackers are willing to perform. Additionally, a firm should also consider the different motivations of different hackers.

In this sense, Uludağ Project is not as simple as some of the hackers assert. Since, it is not merely a Linux from scratch project. On the contrary, it has distinctive applications³⁵. However, those applications were not developed by the outside hackers, rather members of the core team, who were employed by TÜBİTAK UEKAE, developed them. In that respect, socially motivated hackers performed less technical and unchallenging tasks of the project and they complemented the work that was done by the core members of the team.

On the other hand, in the presented FOSS business model for a firm, firm plans and designs the project. Although firm's employees also contribute to the source code of the project, the essential point in this model is being able to benefit from outside hackers' creative work. In that point, firm itself complements the creative work of hackers by developing GUIs, customizing the product or selling support for the customers.

However, one of the important lessons of the Uludağ Project was the role of GPL in persuasion of the hackers. Role of nationalism was essential for the emergence of new hackers. Whereas for the old hackers, using GPL and compatibility with the free software philosophy were more important. In fact, GPL is the best license that sustains the collaboration and socialization of the hackers. GPL is an essential factor in FOSS projects, especially for motivating the hackers socially. Furthermore, although the theory of the gift economy has many controversial points, most of the hackers have explicit expectation of firms' gifting back to the FOSS what they get from it.

In the interviews, there were also firms who were more inclined to BSD licenses. However, this license can not attract the Turkish hackers, unless they participate in the project for only learning and developing new skills.

On the other hand, it is also possible for a firm getting help of socially motivated

³⁵ see <http://www.uludag.org.tr/eng/projeler/index.html>

hackers as the Uludağ Project. In order to do this, firm should comply with the rules of the hackers and support them for escaping from a free-rider image. Indeed, it is not coincidence that the most successful firms, which have FOSS business models, Red Hat, Google and IBM, are also the firms which have good relationships with the hackers. For example, both Red Hat and Google employ hackers and support some FOSS projects. IBM sponsors FOSS activities and gifts some of its proprietary products as FOSS to the hackers.

In Turkey, most of the firms are aware of the importance of supporting the hackers. Therefore, they are very careful about not having a greedy salesman image in the hackers' eyes. Additionally, firms and hackers have coinciding interests about making FOSS widespread in Turkey. Hackers work passionately for increasing the FOSS users. They believe that computers can change one's life and hence, information should be free for people (Levy, 2001). "Computer for Children" campaign is a good example of this.

From firms' standpoint, an additional user is more than a potential customer. Since as a network effect, an additional user also increases the value of FOSS in Turkey. Thus, firms' main attitude is supporting the hackers and their activities in Turkey. However, there are also discrepancies between their attitudes and behaviors. Moreover, these discrepancies are very significant in Category-B and Category-C firms' attitudes.

Firstly, Category-A firms' social links with the FOSS hackers are stronger than any firms'. Their managers' past experiences and ongoing activities in LKD's work groups play major role in their relationship with the hackers. On the other hand, Category-B and Category-C firms have weaker social links with the hackers. For example, when I asked managers of the firms if they were supporting the hackers, Category-A firms were eager to talk about their social support (seminars, localizations, helping others in the mailing lists) despite their monetary support. However, the other firms, understood my question only in terms of money. They talked about being sponsor in the LKD activities. In this sense, these firms prefer a relationship with the hackers only in monetary terms.

Secondly, although every firms' manager talked about monetary support for the

activities, only two of them had separate budget for supporting these activities. One also said that his firms employed a hacker just for his participation in the FOSS projects. Category-A and some of Category-B firms' managers needed to explain lack of separate budget for FOSS and gave reasons such as "I want. But our firm is very new.", "Being a IT firm is hard in Turkey. Yet, being a FOSS firm is harder.". On the contrary, Category-C firms' managers did not need to explain. Their response to the question about budget sounded as if they were implying "As you see, our firm is one of the sponsors of the FOSS activities. What else do you want? We are not a foundation for aiding people. We make business!".

Lastly, when I investigated FOSS firms' contribution to the FOSS projects and localization works in Turkey, there were some contributions only from the Category-A firms. Additionally, there were significant contributions from the Category-E firms to the international projects. As stated in the previous paragraph, one firm has also dedicated one of its employees for the FOSS projects. Although some firms are aware of the localization works and their benefits, none of the firms contribute to these projects. However, as I read from the web pages of the localization works, some of their employees contribute to these works.

CHAPTER 6

Conclusion

In recent years, there has been growing interest of governments and firms for FOSS. Social scientists were astonished when they realized that FOSS might be a perfect alternative to proprietary software. The source of their astonishment based on a presumption that one preferred leisure to work. They seek for the answers of voluntary contribution of hackers to FOSS.

Furthermore, IT professionals, as in Turkey, are also astonished and FOSS business models are not still obvious for them. They have used to earn money from proprietary software by hiding their crown jewels (source code) and selling right to use licenses. FOSS business models contradict with their previous practices.

In this study, instead of merely answering the questions of the FOSS literature, questions themselves were questioned because of their ahistoric presumptions. Since, one may also prefer work as well as leisure under certain social circumstances. Furthermore, as in the case of hackers' relation with their work, duality between work and leisure is not a historic necessity. Therefore, work, as a productive activity of the hackers, is also leisure for them.

On the other hand, in FOSS business models, software is considered as continuously accumulated social knowledge which needs sharing rather than hiding. In this context, FOSS business models depend on voluntary work of the hackers. However, as discussed in the study, hackers do not contribute to and support all FOSS projects. For example, while Turkish hackers supported Uludağ and Turkuaz GNU/Linux distributions, they did not support Gelecek GNU/Linux. For this reason, this study focused on the social circumstances that made hackers work voluntarily and ways of

conditioning these circumstances by the firms.

In that respect, types of works/projects that hackers prefer were investigated. In this investigation, it was concluded that hackers could be grouped into two broad categories. In the first category, there were hackers who were interested in challenging tasks in FOSS projects and realization of their passion in programming activity. In the second group, there were hackers who were interested in the product rather than the activity itself. Members of this category were, particularly, localizing software and giving seminars. Indeed, there was overlap between each category. Nevertheless, it can be said that while members of the first category were mostly attracted by the pleasure in the activity itself, hackers in the second category were interested in the social aspects of FOSS and in the interviews they talked about, freedom, sharing, helping others etc. In this sense, widespread use of Debian, which is well-known about its social sides, is not a coincidence.

On the other hand, all hacking activities have three common points. First, they are voluntary activities. Second, relationship between the hackers and their products is established through FOSS licenses to ensure continuity in the accumulation of the knowledge. Contrary to common intuitive, hackers are not always very high-skilled programmer. There are many ordinary hackers who make modest contributions to the FOSS projects. Most of them do not dedicate their entire life to FOSS. They prosume FOSS and their small contributions to the software continuously accumulate (Ghosh, 1998). Third, by making the software public, software is socialized that anybody can prosume and reproduce it (Chance, 2005).

Additionally, voluntary work is the essential difference between hackers and non-hackers (especially software engineers). In Turkey, software engineering is one of the most popular professions because of its higher income. In other words, software engineers' relationship with their work is conditioned by monetary rewards at the beginning. On the contrary, hackers' relationship with FOSS begins voluntarily.

However, there are some differences between European and Turkish hackers. First of all, Turkish hackers are more interested in the social aspects of FOSS than European hackers. For some hackers, FOSS is a national issue. Although general hacker culture is more inclined to internationalism than nationalism, Uludağ Project

attracted new hackers because of its national aspects.

Furthermore, it was also true for Turkish hackers that money was not an essential motive for them. This fact can not be explained only by the altruism of Turkish hackers. When the Turkish hackers are compared with the European hackers, it is true that Turkish hackers were more interested in the social aspects of FOSS and less interested in monetary rewards. However, it was also true that earning money from FOSS is not still obvious in Turkey. Hence, as discussed in the context of software engineers, participating in FOSS projects is considered as a kind of altruism. Additionally, there are not many FOSS firms in Turkey which might employ hackers.

On the other hand, most of the hackers did not show any tendency against commerce or market. For example, they did not deny that they disliked Microsoft, but they did not consider it as an enemy. Furthermore, some considered Microsoft and proprietary software as a complementary for FOSS. Young student hackers were more inclined to social sides of FOSS and sometimes they gave anti-commercialist answers. For the other hackers, commerce and market were the fact of life that they could not deny.

Additionally, despite hackers' inclination to GPL, they were not well-informed about the FOSS licenses. However, Uludağ Project triggered the FOSS licenses issue among hackers. The project management team's insistence on GPL and free software philosophy increased awareness about GPL. The project management's team message to the Turkish hackers was very clear: "Uludağ is a GPL'd project. Pardus is yours. Let's hack!". In this sense, hackers learned GPL in their practice.

However, there are not many FOSS firms in Turkey and existing firms generally sell support. In that point, they benefit from the Turkish hackers' voluntary contribution indirectly. For example, they sell localized product of the hackers' work. There are not any commercial attempts for directly benefiting from hackers' voluntary contribution by conducting hackers according to firm's interest. In this sense, Uludağ Project gives important hints for making hackers participate in a FOSS project.

TÜBİTAK UEKAE's success in the Uludağ Project can be summarized as its success in transforming its private vision into public vision of the Turkish hackers. In fact, two factors had major role in TÜBİTAK UEKAE's success that a firm would not

have. First of all, involvement of the government and army's interest in the project cleared doubts about the future of project. At least, TÜBİTAK UEKAE was not a software firm which had unclear future. Secondly, nationalism eased the process of constituting the public vision.

In this context, although a FOSS project is not impossible for firms, firms can not imitate TÜBİTAK UEKAE's strategy exactly. However, a closer look shows that there are similarities between the Uludağ Project and firms' which employ hackers: considering the boundaries of working with the hackers and fostering their motivation.

Uludağ Project's core team developed the most challenging parts of the Pardus. Other tasks of the project, testing, documenting and packaging software were performed by volunteers. In other words, hackers complemented the core team's work. In this process, hackers, who were mostly in the second category, played major role. In the project, basic element in fostering the hackers' social motivation was nationalism. In fact, the project management team almost never talked about nationalism and never introduced Pardus in nationalist terms. However, nationalism was already included in the project whose name was National Distribution¹.

On the other hand, there were also hackers² not attracted by the Uludağ Project. In this context, as the business strategy of the FOSS firms which employ hackers inside the firm, firms may foster the outside hackers' motivation by giving them challenging tasks. In that respect, firms themselves complement challenging and innovative works of the hackers (Hecker, 2000).

However, these FOSS business strategies are on the extreme points. There may be synthesis of these strategies, since hackers do not have only one type of motivation. For example, a hacker may have motivations which can make him included either first or second category.

Nevertheless, in Turkey, firms should consider social motivations of the hackers in most of the projects. In that point, greedy salesman image is an essential problem in FOSS business. In other words, firms should escape from free-riding problem, such that they should not have an image of getting from FOSS, but giving back

¹ In Turkish, Ulusal Dağıtım

² Hackers who were in the first category.

nothing. Since if hackers have an impression that their contribution is just for firm's profit, it is highly probable that they give up contribution to the project. Hackers are not very tolerant for the businessmen who considered FOSS merely as a profitable business. This type of businessmen are not supported by the hackers. Thus, in the interviews, most of the businessmen knew that and they were eager to show that they respected to the values of hackers. In this context, businessmen, who had hacker background, were more successful in their relations with the hackers. They were more natural in their relationships with the hackers and there were not any significant discrepancies between their attitudes and behaviors. In other words, these businessmen had more intense social links with the hackers, their firms sponsored FOSS activities and employed FOSS hackers in their firms. On the other hand, the other firms, especially the ones which were new in FOSS business, had links with the hackers only in monetary terms, sponsoring the FOSS activities.

Bonaccorsi and Rossi (2004) emphasize that pragmatist behavior of firms are acceptable by hackers if they comply with the rules of hackers, especially sharing knowledge. For this reason, firms should choose FOSS license carefully that draws the boundaries of sharing knowledge in their product (Fink, 2003). In this context, if a firm wants to attract socially motivated hackers to its project, GPL is essential. Since it is based on reciprocity and most of the time it is the necessary condition for collaboration around the project. It is a promise from firm for not making software proprietary.

On the other hand, firm may not choose between benefiting from outside hackers and making its contribution proprietary for competition. In this context, a FOSS license (e.g. BSD) without reciprocity is needed. Under this condition, firms would not attract socially motivated hackers easily. In that case, firms should focus on the technically motivated hackers who desire challenge in the work itself or want to learn and develop new skills. For example, in the fourth chapter, disinterestedness of the computer engineering students was explained by some of the software engineers as lack of altruism. In other words, computer engineering students do not have any social motivations as FOSS hackers. In this context, firms may attract computer engineering students with well planned and architected FOSS projects. In this sense,

computer engineering students will participate in the project to develop new skills for their future career and will not be against making software proprietary.

In summary, it can be stated that contradicting interests of hackers and firms do not seem to have significant negative effects on the future development of FOSS in Turkey. Instead, hackers and firms reinforce each other. FOSS, as continuously accumulated social knowledge, may present important opportunities for Turkish software firms and firms can conduct hackers' work for their own benefit. However, in order to be successful in FOSS business, they should always consider type of voluntary work they need (e.g. challenging works vs. ordinary works) , basic motivations of volunteers (e.g. pleasure in the activity itself vs. social motivation) and ways of fostering volunteers' motivations.

REFERENCES

- Akkuş, D. (2006). Yerelleştirme Çalışmaları. *Pengence Dergisi*, issue 4, p. 27, Retrieved April 3, 2006 from <http://pengence.linux.org.tr/>.
- Alican, D. F. (2004). *Türk Yazılım Sektörü: Sorunlar ve Çözüm ÖnÖrileri*. Istanbul.
- Ambler, S. (2002). *Agile Modelling: Effective Practices for eXtreme Programming and the Unified Process* (pp. 1–25). New York: Wiley Computer Publishing.
- Asay, M. (2005). Does 'community' still exist in open source? NewsForge, Retrieved November 21, 2005 from, <http://trends.newsforge.com/article.pl?sid=05/06/08/2017218&tid=29>.
- Baldwin, C. Y. and Clark, K. B. (2003). *The Architecture of Cooperation: How Code Architecture Mitigates Free Riding in the Open Source Development Model*. Harvard Business School.
- Barr, J. (2005). *Do LUGs still matter?* NewsForge, Retrieved December 24, 2005 from, <http://software.newsforge.com/article.pl?sid=05/12/23/193238&tid=150>.
- Bell, D. (1976). *The Coming of Post-Industrial Society: a Venture in Social Forecasting*. New York: Basic Books, 2nd edn.
- Bessen, J. (2002). What good is free software? *Government Policy toward Open Source Software*, AEI-Brookings Joint Center For Regulatory Studies (pp. 12–33).
- Bessen, J. (2005). Open source software: Free provision of complex public goods. Retrieved March 21, 2006, <http://www.researchoninnovation.org/opensrc.pdf>.
- Bitzer, J., Schrettl, W. and Schroder, P. J. (2004). Intrinsic motivation in open source software development. Retrieved October 3, 2005 from <http://opensource.mit.edu/papers/bitzerschrettlshroder.pdf>.
- Blumer, H. (1969). *Collective Behaviour in A. McClung Lee(ed), Principles of Sociology*. New York: NY:Barnes and Noble.
- Bonaccorsi, A. and Rossi, C. (2003). Why open source software can succeed. Retrieved October 3, 2005 from <http://opensource.mit.edu/papers/rp-bonaccorsirossi.pdf>.
- Bonaccorsi, A. and Rossi, C. (2004). Comparing motivations of individual programmers and firms to take part in the open source movement. from community to business. Retrieved October 3, 2005 from <http://opensource.mit.edu/papers/bnaccorsirossimotivationlong.pdf>.
- Bonaccorsi, A. and Rossi, C. (2005). Intrinsic motivations and profit-oriented firms in open source software. do firms practise what they preach? *Proceedings of the OSS 1st International Conference on Open Source Systems July 11 - 15, 2005 Genova, Italy*.

- Bosworth, B. P. and Triplet, J. E. (2000). What is new about the new economy? *Brookings Institution*.
- Boyle, J. (1996). *Shamans, Software, and Spleens: Law and The Construction of Information Society*. Cambridge, Mass. : Harvard University Press.
- Brooks, F. P. (1995). *The Mythical Man-Month, Essays on Software Engineering Anniversary Edition* (pp. 13–26, 180–202). Addison Wesley.
- Burnett, R. and Marshall, P. D. (2003). *Web Theory An Introduction*. Routledge.
- Campbell-Kelly, M. (1995). Development and structure of the international software industry, 1950-1990. *BUSINESS AND ECONOMIC HISTORY, Volume 24, no. 2, Winter*.
- Carnoy, M. (2000). *Sustaining flexibility: Work, family and community in the information age*. Cambridge, MA: Harvard University Press.
- Castells, M. (2001a). *The Internet Galaxy: Reflections on the Internet, Business, and Society*. New York: Oxford University Press Inc.
- Castells, M. (2001b). *The Rise of The Network Society*. Blackwell Publishers Ltd.
- Chance, T. (2005). The hacker ethic and meaningful work. <http://opensource.mit.edu/papers/chance.pdf>.
- Cohen, P. A. (2000). *The symbolic construction of community*. London ; New York : Routledge.
- Crossley, N. (2002). *Making sense of social movements*. Buckingham ; Philadelphia, PA : Open University Press.
- Dalle, J. M., David, P. A., Ghosh, R. A. and Steinmueller, W. E. (2004). Advancing economic research on the free and open source software mode of production, version 3 december 2004. Retrieved October 3, 2005 from <http://opensource.mit.edu/papers/davidetal.pdf>.
- DeLanda, M. (2001). Open-source: A movement in search of a philosophy. presented at the Institute for Advanced Study, Princeton, New Jersey, Retrieved November 21, 2005, from <http://www.cddc.vt.edu/host/delanda/pages/opensource.htm>.
- DiBona, C., Ockman, S. and Stone, M. (1999). Introduction. *Open Sources: Voices from the Open Source Revolution, 1st edition*, Retrieved January 31, 2005, from <http://www.oreilly.com/catalog/opensources/book/toc.html>.
- Evans, D. S. (2002). Politics and programming: Government preferences for promoting open source software. *Government Policy toward Open Source Software, AEI-Brookings Joint Center For Regulatory Studies* (pp. 34–49).
- Eyerman, R. and Jamison, A. (1991). *Social Movement: A cognitive approach*. Cambridge Politics.
- Fink, M. (2003). *The Business and Economics of Open Source and Linux* (pp. 175–208). New Jersey: Prentice Hall.

- Fogel, K. (2005). *Producing Open Source Software*. O'Reilly.
- Gates, B. (2002). State of new york vs microsoft corporation, direct testimony. <http://www.microsoft.com/presspass/legal/billgates/billgates.asp>.
- Ghosh, R. A. (1998). Cooking pot markets: an economic model for the trade in free goods and services on the internet. *First Monday*, Retrieved October 3, 2005 from http://www.firstmonday.org/issues/issue3_3/ghosh/index.html.
- Ghosh, R. A., Krieger, B., Glott, R. and Robles, G. (2002). Free/libre and open source software: Survey and study. *International Institute of Informatics, University of Maastricht, The Netherlands*, Retrieved November 21, 2005, from <http://www.informatics.nl/FLOSS/>.
- Golden, B. (2005). *Succeeding with Open Source* (pp. 1–39). Addison Wesley Professional.
- Graham, P. (2004). *Hackers & Painters, Big Ideas from The Computer Age* (pp. 18–34). O'Reilly Media, Inc.
- Hahn, R. W. (2002). Government policy toward open source software: An overview. *Government Policy toward Open Source Software*, AEI-Brookings Joint Center For Regulatory Studies (pp. 1–11).
- Halverson, M. (1998). On the implementation of knowledge: Toward a theory of hardware. Retrieved August 15, 2004 from <http://www.medialounge.net/lounge/workspace/nettime/DOCS/zkp5/pdf/software.pdf>.
- Hamerly, J., Paquin, T. and Walton, S. (1999). Freeing the source, the story of mozilla. *Open Sources: Voices from the Open Source Revolution*, 1st edition, Retrieved January 31, 2005, from <http://www.oreilly.com/catalog/opensources/book/toc.html>.
- Hardin, G. (1968). The tragedy of the commons. *Science New Series*, Vol. 162, No. 3859, 1243-1248.
- Hecker, F. (2000). Setting up shop: The business of open-source software. Retrieved November 21, 2005, from <http://www.hecker.org/writings/setting-up-shop>, version 0.8.
- Himanen, P. (2001). *The Hacker Ethic, A Radical Approach to The Philosophy Of Business*. Random House Paperbacks.
- Humphrey, W. S. (1981). *Managing Technical People*. 1997: Addison-Wesley.
- Jeffries, R., Anderson, A. and Hendrickson, C. (2000). *Extreme Programming Installed* (pp. 9–27). Addison-Wesley Professional.
- Kelty, C. M. (2004). Free software/free science. *First Monday*, Retrieved October 3, 2005 from <http://www.firstmonday.org/issues/issue6-12/kelty/index.html>.
- Knuth, D. E. (1974). Computer programming as an art. *Communications of the ACM*, December, Volume 17, Number 12 (pp. 667–673).
- von Krogh, G., Haefliger, G. S. and Spaeth, S. (2003). Collective action and communal resources in open source software development: the case of freenet. <http://opensource.mit.edu/papers/vonkroghhaefligerspaeth.pdf>.

- Kuster, B., Osterloh, M. and Rota, S. (2002). Trust and commerce in open source ??? a contradiction? MIT Working Paper, Retrieved October 21, 2003 from http://www.wiwiwiss.fu-berlin.de/w3/w3sydow/EURAM/pdf_2002/EURAM%20Paper%20Rota%20et%20al.pdf.
- Laurent, A. M. S. (2004). *Understanding Open Source and Free Software Licensing*. O'REILLY.
- Lehman, F. (2004). Floss developers as a social formation. Retrieved November 21, 2004, from <http://www.firstmonday.org/issues/issue9-11/lehmann/index.html>.
- Lerner, J. and Tirole, J. (2004). The economics of technology sharing: Open source and beyond. Retrieved October 3, 2005 from <http://www.nber.org/papers/w10956>.
- Lessig, L. (2002). Open source baselines: Compared to what? *Government Policy toward Open Source Software*, AEI-Brookings Joint Center For Regulatory Studies (pp. 50–68).
- Levy, S. (2001). *HACKERS: Heroes of the Computer Revolution* (pp. 1–90). Penguin Books.
- Lindman, J. (2004). *Effects Of Open Source Software On The Business Patterns Of Software Industry*. Master's Thesis, Helsinki School Of Economics (HSE) Department of Management, Retrieved November 21, 2005 from <http://opensource.mit.edu/papers/lindman.pdf>.
- LISA, T. L. I. P. (2003). The localization industry standards association, 2nd edition. Retrieved October 3, 2005 from <http://www.lisa.org/interact/LISAprimer.pdf>.
- Locke, J. (1952). *The second treatise of government edited, with an introduction by Thomas Peardon* (pp. 1–30). New York : Liberal Arts Press.
- Margret, E. and Scacchi, W. (2003). Free software development: A case study of software development in a virtual organizational culture. <http://opensource.mit.edu/papers/eliottscacchi.pdf>.
- Marx, K. (1993). *1844 Elyazmaları* (pp. 138–204). (K. Somer, Trans.). Sol Yayinlari.
- Marx, K. (2003). *Grundrisse Ekonomi Politigi'nin Eleştirisinin Temelleri 2. Cilt* (pp. 94–99). (A. Gelen, Trans.). Sol Yayinlari.
- McBreen, P. (2001). *Software Craftsmanship* (pp. 1–30). Addison Wesley.
- McGowan, D. (2001). Legal implications of open-source software. *University of Illinois Law Review*, Vol. 2001, p103.
- McMillan, D. W. and George, D. M. C. (1986). Sense of community: A definition and theory. *Journal of Community Psychology*, 14, 6–23.
- Moody, G. (2002). *Rebel Code: The Inside Story of Linux and the Open Source Revolution*. New York: Basic Books.
- NATO (1969). *Report on a conference sponsored by the NATO Science Committee, Garmisch, Germany, 7th to 11th October 1968*. Retrieved November 21, 2005, from <http://homepages.cs.ncl.ac.uk/brian.randell/NATO/nato1968.PDF>.

- Nayyer, K. (2001). Globalization of information: Intellectual property law implications. *First Monday*, Retrieved October 3, 2005 from http://www.firstmonday.org/issues/issue7_1/nayyer/index.html.
- OECD (1985). *Software: An Emerging Industry*. Information Computer Communications Policy, Paris.
- O'Reilly, T. (1999). Hardware, software, and infoware. *Open Sources: Voices from the Open Source Revolution*, 1st edition, Retrieved January 31, 2005, from <http://www.oreilly.com/catalog/opensources/book/tim.html>.
- O'Reilly, T. (2004). The open source paradigm shift. Retrieved August 15, 2004, from <http://tim.oreilly.com/opensource/paradigmshift0504.html>.
- O'Reilly, T. (2005). What is web 2.0. Retrieved January 31, 2006, from <http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html>.
- Pengence (2006). Turkuaz. *Pengence Dergisi*, issue 3, p. 18, Retrieved April 3, 2006 from <http://pengence.linux.org.tr/>.
- Perens, B. (1999). The open source definition. *Open Sources: Voices from the Open Source Revolution*, 1st edition, Retrieved January 31, 2005, from <http://www.oreilly.com/catalog/opensources/book/toc.html>.
- Pestimalcioglu, G. (2003). A study of free software movement: Towards a new society? *Unpublished master thesis, Department of Sociology, Graduate School of Social Sciences, METU*.
- Pinchot, G. (1999). The gift economy. *Business On A Small Planet*, IC 41, p.49, Retrieved October 2, 2005, <http://www.context.org/ICLIB/IC41/PinchotG.htm>.
- Raymond, E. (1999a). A brief history of hackerdom. *Open Sources: Voices from the Open Source Revolution*, 1st edition, Retrieved January 31, 2005, from <http://www.oreilly.com/catalog/opensources/book/toc.html>.
- Raymond, E. (1999b). *The Cathedral and the Bazaar*. Retrieved August 15, 2004 from <http://www.catb.org/esr/writings/cathedral-bazaar/>.
- Raymond, E. (2000). *Homesteading the Noosphere*. Retrieved October 3, 2005 from <http://www.catb.org/esr/writings/cathedral-bazaar/homesteading/>.
- Raymond, E. (2001). *How To Become A Hacker*. Retrieved December 4, 2005 from <http://www.catb.org/esr/faqs/hacker-howto.html>.
- Raymond, E. (2003). *The Art of Unix Programming*. Retrieved October 3, 2005 from <http://www.faqs.org/docs/artu/index.html>.
- Raymond, E. (2004). The halloween documents. Retrieved December 5, 2004, from <http://www.opensource.org/halloween/>.
- Richardson, J. (2001). Free software & gpl society —interview with stefan merten, oekonux. Retrieved January 31, 2005, <http://www.free-soft.org/FSM/english/issue01/smerten.html>.

- Seidel, R. W. (2002). *Government and the Emerging Computer Industry*. In *From 0 to 1: An Authoritative History of Modern Computing*. A. Aker and F. Nebeker (pp. 189–201). New York, NY: Oxford University.
- Sennett, R. (1998). *The Personal Consequences of Work in the New Capitalism*. London: W. W. Norton & Company.
- Smith, B. L. (2002). The future of software: Enabling the marketplace to decide. *Government Policy toward Open Source Software, AEI-Brookings Joint Center For Regulatory Studies* (pp. 69–86).
- Smith, M. A. and Kollock, P. (1999). *Communities in Cyberspace* (p. 230). London: Routledge.
- Stallman, R. (1984a). The gnu manifesto. Retrieved December 5, 2004 from <http://www.gnu.org/gnu/manifesto.html>.
- Stallman, R. (1984b). Initial announcement. <http://www.gnu.org/gnu/gnu-history.html>.
- Stallman, R. M. (2002). *Free Software, Free Society: Selected Essays of Richard M. Stallman*. GNU Press www.gnupress.org Free Software Foundation Boston, MA USA.
- Steinmueller, W. E. (1996). *The US Software Industry: An Analysis and Interpretive History* (pp. 15–52). Oxford University Press.
- Tarrow, S. (1998). *Power in Movement*. Cambridge University Press.
- Techonology, B. (2003). Brazil: Microsoft, go home. *International Institute of Informatics, University of Maastricht, The Netherlands*, Retrieved November 21, 2005, from <http://www.brazzil.com/2003/html/news/articles/aug03/p139aug03.htm>.
- Tiemann, M. (1999). Future of cygnus solutions : An entrepreneur's account. *Open Sources: Voices from the Open Source Revolution, 1st edition*, Retrieved January 31, 2005, from <http://www.oreilly.com/catalog/opensources/book/toc.html>.
- Tonnies, F. (1958). *Community and Society*. The Michigan State University Press.
- Torvalds, L. (1999). The linux edge. *Open Sources: Voices from the Open Source Revolution, 1st edition*, Retrieved January 31, 2005, from <http://www.oreilly.com/catalog/opensources/book/toc.html>.
- Torvalds, L. and Diamond, D. (2001). *Just for Fun: The Story of an Accidental Revolutionary*. HarperBusiness.
- Valimaki, M. (2005). *The Rise of Open Source Licensing: A Challenge to the Use of Intellectual Property Licensing in the Software Industry* (pp. 13–49). Turre Publishing.
- Veale, K. J. (2003). Internet gift economies: Voluntary payment schemes as tangible reciprocity. *First Monday*, Retrieved October 3, <http://www.firstmonday.org/issues/issue8-12/veale/index.html>.
- Vise, D. A. and Malseed, M. (2006). *Çağımızın en yeni iş, medya ve teknoloji başarısı: Google* (pp. 9–20). (G. Köse, Trans.) Koridor Yayıncılık.

Wayner, P. (2000). *Free For All : How Linux and the Free Software Movement Undercut the High-Tech Titans*. Retrieved August 15, 2004, from <http://www.wayner.org/books/ffa/>.

Weber, S. (2004). *The Success of Open Source*. Harvard: Harvard University Press.

Weinberg, G. M. (1998). *The Psychology of Computer Programming, Silver Annivesary Edition*. New York: Dorset House Publishing.

Williams, S. (2002). *Free as In Freedom, Richard Stallman's Crusade for Free Software*. O'Reilly, Retrieved August 15, 2004, <http://www.oreilly.com/openbook/freedom/index.html>.

Wong, K. (2003). Free/open source software and governments. *International Open Source Network, Asia/Pacific Information development program, United nations development program, Kuala Lumpur, Malaysia*.

Wood, E. M. (1999). *The Origin of Capitalism*. Monthly Review Press.

Xiaonan, Z. (2003). China blocks foreign software use in gov't. *News & Technology*, Retrieved November 21, 2005 from <http://asia.cnet.com/newstech/applications/0,39001094,39146335,00.htm>.

Young, R. (1999). Giving it away how red hat software stumbled across a new economic model and helped improve an industry. *Open Sources: Voices from the Open Source Revolution, 1st edition*, Retrieved January 31, 2005, from <http://www.oreilly.com/catalog/opensources/book/toc.html>.

Zeitlyn, D. (2003). *Gift economies in the development of open source software:Anthropological reflections*. Retrieved November 21, 2005 from <http://opensource.mit.edu/papers/rp-zeitlyn.pdf>.

APPENDICES

APPENDIX A

Glossary

Definition of terms are taken from The Free Encyclopedia, Wikipedia ¹ and slightly altered for clarity.

Compiler: a computer program that translates a series of statements written in one computer language (called the source code) into a resulting output in another computer language (often called the object or target language). Compilers usually translate source code written in a programming language to a machine language which is consist of 1s and 0s.

Desktop Environment: In graphical computing, a desktop environment offers a complete graphical user interface solution to operate a computer. e.g KDE, XFCE and GNOME.

Domain: a name that is entered into a computer (e.g. as part of a website or an email address) and then looked up in the global (Domain Name System) which informs the computer of the Internet address(es) with that name.

Domain Name System (DNS): a system that stores information associated with domain names on networks, such as the Internet. The domain name system associates many types of information with domain names.

EMACS: A class of text editors, possessing an extensive set of features, that is popular with computer programmers and other technically-proficient computer users. The original EMACS, a set of Editor MACroS for the TECO editor, was written in 1975 by Richard Stallman, initially together with Guy Steele.

¹ see <http://www.wikipedia.org>

GNU/Linux Distribution: A typical distribution comprises a Linux kernel, GNU, additional software, documentation, and some multimedia (such as fonts, desktop wallpapers, system sounds, etc.). Although GNU/Linux distributions are comprised of same programs, they may have different system installation procedures, graphical user interfaces or philosophies. For example, Slackware is the oldest distribution. It's for advanced users or users who want to be advanced user. Red Hat and Mandriva are very easy to install and use. One who does not anything about GNU/Linux, but knows Windows can easily use these distributions. Debian is the most stable distribution, it passes from many testings and its philosophy gives more importance to freedom than other distributions. Indeed, Slackware, Red Hat and Mandriva are supported by firms and there is no firm behind Debian.

On the other hand, there are localized GNU/Linux Distributions e.g (Turkuaz, Pardus, Gelecek (Turkish) and Red Flag (Chinese)).

High Level Programming Language: a programming language that is more user-friendly, to some extent platform-independent, and abstract from low-level computer processor operations such as memory accesses (e.g Java, Pascal, C#, Visual Basic, Python)

Interoperability: The capability of different programs to read and write the same file formats and utilize the same protocols. e.g. an interoperable document is vendor independent. It should be opened by any vendor's software.

Kernel: It is the core of an operating system. It is a piece of software responsible for providing secure access to the machine's hardware and to various computer processes (a process is a computer program in a state of execution). Since there are many programs, and hardware access is limited, the kernel also decides when and how long a program should be able to make use of a piece of hardware, which is called scheduling. Kernel is similiar to the mind of body. Furthermore, Linux is the name of kernel, not the operating system. As Stallman emphasizes, the operating system's name is GNU/Linux

Localization: Means of adapting products such as publications or software for non-native environments, especially other nations and cultures.

Low-level programming language : a programming language that provides little or no abstraction from a computer's microprocessor. However, this definition is relative. For example, C is low level for Java programmers while assembly is low level for C programmers.

Microkernel: it is a kernel design approach that consists in defining a very simple abstraction over the hardware, with a set of primitives.

Monolithic Kernel: it is a kernel design approach which tends to be easier to design correctly, and therefore may grow more quickly than a microkernel-based system. Microkernels are often used in embedded robotic or medical computers because most of the OS components reside in their own private, protected memory space. It is used in design of Linux kernel.

Operating System: the system software that is responsible for the direct control and management of hardware and basic system operations. Additionally, it provides a foundation upon which to run application software such as office programs, web browsers, games and others. DOS, Microsoft Windows 98/2000/XP, Unix, GNU/Linux, Mac OS are the most well-known operating systems.

Programming Language: a standardized communication technique for expressing instructions to a computer. It is a set of syntactic and semantic rules used to define computer programs. A language enables a programmer to precisely specify what data a computer will act upon, how these data will be stored/transmitted, and what actions will be taken under various circumstances. C, Java, Perl, Python, Lisp, Prolog, Scheme, Pascal, Visual Basic, C# are the most well-known programming languages.

Protocol: set of rules governing communication within and between computer networks e.g file transfer protocol (FTP), Internet Protocol (IP), Simple Mail Transfer Protocol (SMTP)

Source code: Any series of statements written in some human-readable computer programming language

Version Control Program: it is used for the management of multiple revisions of the same unit of information. It is most commonly used in engineering and software development to manage ongoing evolution of digital documents like appli-

cation source code, art resources such as blueprints or electronic models and other critical information that may be worked on by a team of people. It is must for organization of FOSS projects on Internet. e.g CVS, BitKeeper, Microsoft Visual Source Safe,

Vendor lock-in: A situation which arises in the lack of interoperability. A good example is proprietary file formats. e.g Microsoft's Word file format is proprietary. If you write your documents with it, you should use Microsoft's products forever. Although there are software products which can open Microsoft documents, there may exist problems. In this context, user is in a situation of vendor lock-in such that user's friends must also use Microsoft's products in order to open the prepared documents.

Web Browser: A software application, that enables a user to display and interact with HTML documents hosted by web servers or held in a file system. Popular browsers available for personal computers include Microsoft Internet Explorer, Mozilla Firefox, Opera, Netscape, Apple Safari and Konqueror.

Web Server: A computer that is responsible for requests from clients, which are known as web browsers, and serving them web pages, which are usually HTML documents (Wikipedia). The most popular web server is Apache which is FOSS. Microsoft's Internet Information Server (IIS) follows Apache.

X Window System: It provides the standard toolkit and protocol to build graphical user interfaces on Unix and Unix-like operating systems. In this sense, Unix is different from Microsoft Windows. Unix is comprised of many programs. X Window System is one of these programs. On the other, in Windows, programs are integrated. Because of this Unix is more modular than Windows, its parts are interchangeable which makes Unix philosophy different from Windows.

APPENDIX B

Research

B.1 Hackers

1. How long have you been participating in the FOSS activities? Which projects did you contribute to? What were the types of contributions? (e.g. programming, internationalization, giving seminars etc)
2. If your firm gave up its FOSS policy, what would be your reaction against this decision? (e.g my motivation would decrease, I would look for another job, nothing would change for me etc.)

If you do not work for a FOSS firm, can you compare the programming activity that you perform at work with the work you do in your leisure time at home?
3. Why do you participate in the FOSS activities?
4. In some e-mail discussion lists, Microsoft is named as Micro\$oft, Mikisoft or Mikrosoft. What do you think about it?
5. When a new (potential) user asks the list "I am new to GNU/Linux. Which GNU/Linux distribution should I use?", distribution war among users start. People praise the distribution they use. Is there any distribution or product that you prefer rather than the others? Why do you prefer it?
6. Which term do you prefer? "Free Software" and "Open Source Software"? Why?

In the world, it is observed that there is a tendency for using "Open Source Software" rather than "Free Software". What do you think about this tendency?

7. Do you have any license preferences for the products that you develop? GPL, LGPL, BSD, Mozilla etc?
8. The debate in European Union about the patentability of software is directly related with the future of the FOSS. What do you think about it?
9. What must be done to make FOSS widespread in Turkey? What do you do for achieving this goal?
10. What do you think about the role of Uludag Project in the future development of the FOSS in Turkey?

B.2 Firms

1. How long have you been participating in the FOSS activities? Do you have any hacker background?

if yes: which projects did you contribute to? Do you still have hacking activities beyond business? if no: What were you doing before you met FOSS? (e.g developing proprietary software, selling support for proprietary software, not in IT sector etc)

2. What is your firm's FOSS business model?

- Support Seller (Selling CDs, consultancy, technical support, education etc)
- Loss leader (Earning money from a complementary proprietary software for FOSS)
- Selling hardware and developing FOSS for complementing hardware.
- Selling books for FOSS

Do you have any proprietary software business model in addition to the FOSS business model?

3. Did you study on the FOSS business models of the international firms? Is there any firm that you take as your model?
4. Did you have any research about the tendencies of computer users in Turkey?

5. How many employees do you have in your firm? Are there hackers among them? Can your staff study on FOSS at work hours even though the issue he studies is not concerned with his work? If yes, do you also motivate their participation in the FOSS?
6. How is your relationship with the FOSS community in Turkey? Do they support your firm? Do you support them? Do you have any budget for support?
7. Do you have any research about the hackers? What do you pay attention in the relation with hackers?
8. Why did you choose a FOSS business model? What are the advantages of FOSS business over proprietary software? Or disadvantages?
9. Did you have any conflict being either a hacker or a businessman?
10. In some e-mail discussion lists, Microsoft is named as Micro\$oft, Mikisoft or Mikrosoft. What do you think about it?
11. When a new (potential) user asks the list "I am new to GNU/Linux. Which GNU/Linux distribution should I use?", distribution war among users start. People praise the distribution they use. Is there any distribution or product that you prefer rather than the others? Why do you prefer it?
12. Which term do you prefer? "Free Software" and "Open Source Software"? Why?

In the world, it is observed that there is a tendency for using "Open Source Software" rather than "Free Software". What do you think about this tendency? Furthermore, hybrid models (mixing FOSS with proprietary software) are becoming more widespread. What do you think about its possible effects on hackers' motivation?
13. Do you have any license preferences for the products that you develop? GPL, LGPL, BSD, Mozilla etc?

Assume that government puts a directive that public sector should prefer FOSS. Is GPL a right decision for it? Hackers answered question about the FOSS licenses from two vantage points.

Firstly, there were hackers who emphasized that public's source must always be public's. Thus, GPL is a good decision. Secondly, for motivating firms less restricted licenses should be used, e.g. BSD-style licenses.

What is your opinion about licenses?

14. The debate in European Union about the patentability of software is directly related with the future of the FOSS. What do you think about it? Does it effect your firm's business model?
15. What do you think about the role of Uludag Project in the future development of the FOSS in Turkey?
16. In Turkey, selling support as a FOSS business is the most widespread. On the contrary, firms do not choose FOSS as software developing strategy and they do not have any benefit from the contribution of outside developers. What are the reasons behind this?

B.3 Software Engineers

1. Which university did you graduate from? When? Have you ever used GNU/Linux? For which purpose? Can you compare it with Microsoft Technologies?
2. Summarize your thought about FOSS?
3. What is the difference between Free Software and Open Source Software? Which term do you use?
4. There are many computer scientists and software engineers among FOSS developers in Europe. On the contrary, computer scientists and engineers are not interested in FOSS? Why?
5. Can people make money off FOSS in Turkey? Do you think it is a viable business strategy?

6. Do you enjoy programming? Do you develop software in your leisure time?
Can you compare programming at work and at home?

B.4 Activities

Third Linux and Free Software Festival, May 13-16, 2004 - Milli Kütüphane (Ankara)

Seminar: Open Source from the Economics perspective (given by Abdullah GOK),
October 9, 2004 - EMO Ankara Şubesi

Seminar: History of Free / Open Source Software (given by İzlem Gözükeleş), Octo-
ber 16, 2004 - EMO Ankara Şubesi

Seminar: Free Software Movement: Towards a New Society? (given by Güzin Peştimalcıoğlu),
October 23, 2004 - EMO Ankara Şubesi

Free Software and Open Source Days, March 4-5, 2005 - Bilgi Üniversitesi (İstanbul)

Meeting: LKD Members, April 16, 2005 - Fisek Enstitüsü (Ankara)

Fourth Linux and Free Software Festival, May 19-22, 2005 - Milli Kütüphane (Ankara)

Panel: Free/Open Source Software from the perspective of Software Engineering -
METU Culture Convention Center (Ankara)

Free Software and Open Source Days, February 24-25, 2006 - Bilgi Üniversitesi (İstanbul)