

The Gift of the Code: A Culture of an Operating System¹

Maria Kastrinou-Theodoropoulou, Durham University

Abstract

"The Gift of the Code" explores the boundaries between technology and sociality, computers and cultures. Based on long-term ethnographic research among users and developers of GNU/Linux Operating System, this work analyses how Linux developers and users consume, create and exchange an as much technical as cultural discursive construction of sociality. Like a modern-day *kula* ring, the Linux code is analysed in terms of a gift: one cannot keep it for one's self, it contains obligations and a promise of future reciprocity. It is a collective gift of the self-ascribed Hackers that come from different geographic places and meet in lines of code, socializing by exchanging ideas about the code and about themselves. This work shows by what means the computer hackers of Linux, abiding to the original definition of the word, actively constitute their community using discourse: language, e-mail, internal meritocratic hierarchies based on technical ability and ethics of the group, boundaries of exclusion and inclusion. This project is about power relations, resistance networks and the hegemony of a techno-scientific self-indulgence of some post-residents of an imagined cyber-West. Equally, it is about the giving of gifts, hacker culture and the 'fun' of hacking, creating and maintaining a 'guerrilla' operating system. Studying the anthropology of GNU/Linux² operating system is a journey towards an investigation of what makes the social into technology and how technology is translated into sociality.

Keywords

reciprocity, gift, gnu-linux, internet, cyber-sociality

¹ This paper was originally submitted as a final year dissertation as part of the Human Sciences programme at Durham University. It was the winner of the David Brookes Memorial Prize for the best ethnographic student dissertation in the Anthropology Department at Durham University in 2007.

² Hereafter called 'Linux' for convenience.

Introduction³

"Why does every anthropology student seem to think that free and/or Open Source software is a fascinating topic worthy of study?"
(e-mail #19, received on 22/02/07)

Never before I received this e-mail had I thought that so many anthropology students would have asked my informant from the East Coast of USA about the significance of Open Source and Free Software! Surprised and somewhat disappointed by his response to my 'unoriginality', his question was valid: why would an operating system be "a fascinating topic worthy of study", and what an undergraduate dissertation on GNU/Linux has to offer to anthropology?

This is a project about power relations, resistance networks and the hegemony of a techno-scientific self-indulgence of some post-residents of an 'imagined' (Anderson 2003) cyber-West. Equally, it is about the giving of gifts, hacker culture and the 'fun' of hacking, creating and maintaining a 'guerrilla' operating system. Studying the anthropology of GNU/Linux⁴ operating system is a journey towards an investigation of what makes the social into technology and how technology is translated into sociality. Not that examples of the dynamics of such relations are rare: most of anthropology and most of human interaction, as a matter of fact, is comprised of people interacting with other people, creating and using tools, which in an ongoing feedback loop may change both their interactive relationships as well as themselves. Hence, the study of Linux is a study of a hybrid (Latour 1993) of actions, reactions and interactions between people, hardware and software, historically situated within a 'restructuring' capitalism (Castells 1996), and politically active and challenging in its definition of freedom and exchange.

Inevitably, this study stands simultaneously on several rather 'imagined' domains of academic and popular discourse: exploring the anthropology of Linux inevitably means talking about science, technology, society and culture, as well as their intricate and complex interrelations amongst themselves. On the other hand, through the anthropological endeavour of participant observation, 'traditional' issues of community, reputation systems, boundary formation and reciprocity arise. The aim of this work is to venture in the exploration of the active discursive construction of an emergent networked global community, to observe and analyse the mechanisms by which the geographically scattered individual developers and users are bound together. Yet, this 'traditional' ethnographic journey is somewhat unique: Linux-Hacker culture has only been described by members of the culture itself (see Raymond 1999), while the anthropological literature on the matter is spare: Ratto (2005) has conducted a study on the negotiation of trans-local space in Linux development through focusing on the psychological ties

³ The exploration in the world of GNU/Linux is almost now beginning, after the completion of this dissertation. It has been a challenge and an honour to venture in such 'deep waters'. For rescuing me from definite drown, my gratefulness goes to my supervisor, Dr. Stephen M. Lyon and to Dr. Elisabeth Kirtsoglou. For swimming next to me, even at times of difficult currents, many thanks to my friends John Lyle and Elena Poughia. Special thanks to Panagiotis Varelas for introducing me to GNU/Linux almost by accident, in a sunny afternoon, not too long ago, in 2005. Many Thanks to Mr. Manolis Darkadakis, the President of the Greek Radio-Amateur (Hams) Association, for explaining to me the relationship between Hams and Linux. Also, Many Thanks to my friends Fadi Shehadeh and Nikos Nikoleris for all the enthusiasm, patience and technical 'inside' support they have given me throughout this project. I'm grateful to my cousin Emilia Kanta and her friends in SomeThink Creative Group for the design of the cover page. Without the sea of the Linux community, this dissertation wouldn't have been ethnographic... Linux developers and participants of this study gave me the sea to swim in and the harbour to aim for. Words are not enough to thank them. I only hope to be able to reciprocate adequately, now and in the future. And to my mother, who taught me how to swim in the first place....

⁴ Hereafter called 'Linux' for convenience.

and symbolism embedded in the psychological relationship between developers and Tux, the Linux logo; while Kollock (2000) has analysed Linux in terms of the gift-exchange dynamics of online cooperation. The originality of this ethnographic study is that it aspires to further our understanding of hacker culture by exploring specifically how organisation and cooperation within Linux development form discursively a ‘culture’⁵.

The structure, organization and reciprocal development of this operating system offers anthropological insights and challenges of the forms and structures of cyber-sociality: the source code of the kernel (the most basic ‘unit’ for OS) resembles a ‘gift’: it is open and free for everyone to see, use and change. Also, GNU/Linux develops through the voluntary cooperation of millions of developers world-wide. This ‘transnational’ (Kearney 1995) community promotes its own distinctive form of globalization in ‘cyberspace’, challenging and crystallizing simultaneously many ‘given’ stereotypes about the effectiveness of hierarchical organization, dominant power structures, ways of resisting economic and ‘cultural’ hegemonic forces (i.e. Microsoft’s Windows OS), by its open, free and seemingly ‘anarchical’⁶ organization.

Finally, because GNU/Linux has equally to do with software and hardware, peoples, cultures and ‘imagined’, virtual but always real communities, with technological innovation combined with a distinctive ‘maussian’ sociality, it posses challenging questions about the interaction of humans and machines, ‘modernity’ and ‘scientific development’, the ‘ethos’ of capitalism and the future of profit-driven ‘laisser faire’, the importance of everyday practices in their constant negotiation of power and resistance of hegemony in the realm of ‘cyberspace’, and in an age of proclaimed ‘digital revolutions’.

What is Linux?

Like the brain or the nervous system that connect our internal biological processes to the external socially constructed world, an operating system (OS) “coordinates the interaction between a computer’s hardware and application software” (Rheingold 2002: 48). An OS is the most basic structural and functional unit of a computer, controlling and determining the activities and abilities of the machine, while making possible the connection between hardware (machine) with software (applications, Graphic User Interface (GUI), etc.). The most commonly used OS is that of Microsoft’s Windows (XP, 1998, Vista, etc.), while other OS include Mac OS and Linux distributions (such as Debian, Fedora, RedHat, SuSe, Ubuntu, Gentoo to mention a few popular ones). Now, because this is an anthropological study and not a computer science one, and because in anthropology computer-literacy is not assumed, I will try to simplify the technical terms to the best of my abilities, undertaking the risk of oversimplification⁷.

Due to the fundamental importance of an OS, its creation is considered one of the hardest and most noble things, while the creation of the OS’ kernel⁸ is considered a much harder and honourable job within the community of computer programming and developing (Raymond 1999). By building a kernel “you get to create your own world” (Torvalds & Diamond 2001: 74), says Linus Torvalds, the man who as a university student in Helsinki created the kernel of GNU/Linux OS in 1991. Although one man started the Linux kernel, the success of both the kernel

5 ‘Culture’ defined by Clifford (1983: 137) as “an open-ended, creative dialogue of subcultures...”

6 Although, part of this work is to challenge the stereotypes concerning Linux itself.

7 Therefore, I ask from the beginning for the pardon of the readers who might be computer-literate for my potentially reductive analysis and simplification of the technical domain of computer technology

8 The kernel is the central control unit of an operating system- like a nucleus in a cell or an administrative system.

and the OS in general lies on its wide social base: its users, developers and co-developers who engage voluntarily with the project and improve it. As Raymond notes:

“Linus’s cleverest and most consequential hack was not the construction of the Linux kernel itself, but rather his invention of the Linux development model.” (Raymond 1999: 37)

In the design of the Linux kernel, Linus relied heavily on previous developments and systems (such as UNIX and Minix OS as well as the GNU Project⁹), and yet, the true innovation was not in the technical domain: the greatest, maybe, innovation was Linus’ decision to release and distribute his efforts to other programmers, so that they can find problems ('bugs'), suggest ways to fix them, and offer their insights and help. In the literature of Linux and hacker culture, this is described by two rules:

1. Release early. Release Often. And listen to your customers.
2. “Linus’ Law” or “Given enough eyeballs, all bugs are swallow”

(Raymond 1999: 39- 44)¹⁰

These two ‘rules’ relate as much to the beginning of Linux as well as to its continuous development since today. In short, Linux is a free, open, collaborative operating system, in which dispersed people from all around the world offer voluntarily their time and efforts for high-quality development of software. It is ‘free’ in the sense of “freedom of speech”¹¹, and it is ‘open’ in that the source code is not hidden or closed (as in the case of most proprietary software, like Windows) but available for anyone (granted they know programming!) to see, use, and change.

Today, Linux has 138021 registered and twenty nine million estimated users, while its growth is continually increasing¹².

QuickTime™ and a
TIFF (Uncompressed) decompressor
are needed to see this picture.

Graph: Growth of Linux Users per Year

9 See ‘The GNU Project’, <http://www.gnu.org/>, March 2nd, 2007.

10 Also see Appendix I for a summary of Raymond’s Linux development rules.

11 See “Free Software Foundation”, <http://www.fsf.org/>, and “Philosophy of the GNU Project”, <http://www.gnu.org/philosophy/>, March 2nd, 2007

12 Estimates and Graph from The Linux Counter, <http://counter.li.org/>, accessed on April 26, 2007.

“Seemingly overnight, the Linux operating system caught the world’s attention. It had exploded from the small bedroom of its creator, Linus Torvalds, to attract a cultish following of near-militant geeks. Suddenly it was infiltrating the corporate powerhouses controlling the planet.” (Torvalds & Diamond 2003: ix)

Notes from the History of Internet, Computer Technology and Culture

The above section served as an introduction to Linux OS and the social basis of its development. However, as with most things, Linux did not happen overnight. So, how did Linux came about? What are the connections between Linux and the rise of the Internet and computer technology? What are these “cultish near-militant geeks” and where did they come from?

The history of Linux is inextricably related with the history and development of the Internet, the growth of computer technology as well as the rise to self-consciousness of hacker culture and ethic. In this section, I will briefly sketch the connections between these which are relevant to this anthropological enquiry¹³, namely connections that pertain to the sociality of innovation, change and technology, bearing in mind that:

“All innovation is social innovation. Innovation does not happen ‘out there’ in the world of objects, but in society and in minds. More particularly, it happens in the minds of the users, which are intrinsically integrated with the activities of the users. Those cultural and material recourses that are available to the users, therefore, become key resources in the innovation process.” (Tuomi 2002: 5)

Castells characterizes the history of the Internet as “a unique blending of military strategy, big science cooperation, and countercultural innovation” (1996: 351), and this statement can also characterize the rise of computer technology after WWII. Already by 1945 computing technology had begun emerging and along with that the first “real programmers” (Raymond 1999: 7): “a more or less continuous and self-conscious technical culture of enthusiast programmers, people who built and played with software for fun” (*Ibid.*). From the 1950’s onward, big collaborative projects were organized in the United States of America, by the Defense Department’s Advanced Research Projects Agency (DARPA), aim of which was to create technologically advanced ways of computer mediated communication (CMC). DARPA brought together political and military strategy with academic and scientific aims, creating the first decentralized network of information exchange and communication (Castells 1996: 352). DARPA gave rise to ARPANET in 1969- “the first transcontinental, high-speed computer network” (Raymond 1999: 9). With the ARPANET and with renewed political

13 This is not only because of word-limit constraints but also because there are excellent, much more complete and detailed accounts of such histories, written by people experienced on the field- see Stallman. “The GNU Project” (<http://www.gnu.org/>); Raymond 1999. “A Brief History of Hackerdom” In *The Cathedral and the Bazaar*, pp. 5- 26; Castells 1999, Chapter 5: “The Culture of Real Virtuality” in *The Rise of the Network Society*, pp.; Tuomi 2002 *Networks of Innovation: Change and Meaning in the Age of the Internet*.

interest in technological advancement¹⁴, computer technology was accelerated. In this period, major scientific- technological centers emerge throughout the USA, in universities such as MIT, Stanford, Berkley as well as businesses such as the Bell Labs. This was a time of free and open exchange of information, a time of sharing innovative ideas and dynamic development. It was also the time that UNIX OS and computer language C were developed in Bell Labs: flexible and portable software tools, that could be used in any computer and a pre-cursor to Linux.

Starting from political and military agendas, computer technology merged the worlds of academia, military and business, creating a unique social opportunity through the development of technology (such as time-sharing computer machines), for like-minded people to meet, create and exchange their ideas. Stallman explains the mode of technological innovation in MIT Artificial Intelligence Lab in the '70's as the first software-sharing community¹⁵, in which people were exchanging software ideas without constraints of price or copyright. They were constantly creating, using and changing a pool of knowledge. Such were the social and technological conditions that gave rise and nurtured the early culture of self-ascribed hackers.

In the 1980's a series of events started changing the ways the 'software-sharing community' had operated, the most important of which is the beginning of the 'proprietary- UNIX era' (Raymond 1999: 19). First of all, with the development of many different OS and computers, there existed three different programming 'cultures', "overlapping at the edges but clustered around very different technologies" (Raymond 1999: 16): the ARPANET/PDP-10, the UNIX, and an emerging one of "microcomputer enthusiasts" (Ibid.). Secondly, major University centers, such as the MIT AI Lab, started using proprietary software: contra the elder practices of free sharing software, now they "had to sign a nondisclosure agreement even to get an executable copy"¹⁶. This caused great dissatisfaction within the academic community, many people leaving the academic institutions in order to be employed by companies.

It was in this time that Richard M. Stallman (known as RMS- his login name) left MIT AI Lab to start the GNU¹⁷ Project (1984) and the Free Software Foundation (FSF) (1985). It is this phase that can be described as "countercultural innovation" (Castells 1996: 351). Stallman, described as the "God of Free Software" (Torvalds & Diamond 2003: 58), or as "the last true hacker" (Raymond 1999: 17), "would largely define the public ideology of the hacker culture, and Stallman himself would be the only credible claimant to leadership of the tribe" (Ibid.: 18). Stallman, despising the commercialization of software, longing for the days where the community of hackers were exchanging ideas freely, created the FSF in order not only to promote high-quality free software, but the essential ideals and ethos of the hacker culture¹⁸. 'Free' was to mean "freedom of speech, not free beer":

14 Let us not forget that in the 1960's and 1970's USA (the center of computer technology activity) goes through the most intense arms race in the Cold War Era, and thus, investment in science and technology is politically stimulated and funded.

15 The Hacker Community and Ethics: An Interview with Richard M. Stallman, 2002. <http://www.gnu.org/philosophy/rms-hack.html>, April 3, 2007

16 Ibid.

17 GNU= Gnu's Not Unix, one of the many recursive acronyms, following the hacker tradition, "a kind of computer science in-joke that nobody else gets. Geeks- we're just tons of fun to be around." (Torvalds & Diamond 2003: 58)

18 The Hacker Community and Ethics: An Interview with Richard M. Stallman, 2002. <http://www.gnu.org/philosophy/rms-hack.html>, April 3, 2007

“Free software is a matter of the users' freedom to run, copy, distribute, study, change and improve the software. More precisely, it refers to four kinds of freedom, for the users of the software:

The freedom to run the program, for any purpose (freedom 0).

The freedom to study how the program works, and adapt it to your needs (freedom 1). Access to the source code is a precondition for this.

The freedom to redistribute copies so you can help your neighbor (freedom 2).

The freedom to improve the program, and release your improvements to the public, so that the whole community benefits (freedom 3). Access to the source code is a precondition for this.”¹⁹

The underlying motives of the FSF and GNU were as much about software as about ideology (and maybe more about the later). As the brief description of the history of computer technology shows, the culture of hackers emerged from an environment which promoted both free sharing of ideas as well as the building of a cooperative community. In this light, Stallman's ideology grew from the history and culture that preceded it. As proprietary software and copyright in general were a later development in the history of computer technology and communication (Harvard Law Review 2001: 2439- 2448), they did pose a serious threat to the cohesion as well as the development of both the community and the software it produced. Judging the “proprietary –software social system” as “antisocial, unethical and simply wrong”²⁰, FSF based its ideology on the following points:

1. Copyright is not a natural right, but an artificial government-imposed monopoly that limits the users' natural right to copy;
2. Software developers care about software as much as about the society they are allowed to have;
3. Free software movement has demonstrated that plenty of useful software can be made without putting chains on it- that is without offering the power of restrictions over companies.²¹

Basing its arguments on the principle that helping each other is the basis of society, the FSF gained a lot of momentum during the 1980's, becoming the center of hacker community (Raymond 1999). GNU was able to develop a variety of software tools, protecting its interests and the sharing community under Gnu Public License (GPL)²², however, it's kernel project,

19 “The GNU Operating System- Free as in Freedom”, <http://www.gnu.org/>, March 5, 2007.

20 “The GNU Project”, <http://www.gnu.org/gnu/thegnuproject.html>, April 4, 2007.

21 Ibid.

22 GPL is based on the ideology of freedom of speech for the sharing, exchanging and changing ideas and software technology, this is widely known as copyleft which means that anyone can copy and change somebody else's ideas- see <http://www.gnu.org/copyleft/copyleft.html>, April 5, 2007.

called HURD, was not very successful. It was on HURD that Linus Torvalds built his own kernel, Linux²³.

The birth of Linux correlated not only with the existence of FSF and GNU, but also with the “Great Web Explosion” (Raymond 1999: 25) of the 1990’s, as Raymond notes: “the early growth of Linux synergized with another phenomenon: the public discovery of the Internet” (*Ibid.*). The Internet is another example of a decentralized non-hierarchical computer mediated communication network, whose origins also date back to the DARPA projects. The Linux development model, that is the architecture of structural cooperation, emerged simultaneously with Internet explosion (Tuomi 2002: 170), while also the existence of the Internet facilitated the communication and emergence of the developer community per se, as it allowed (1) new ways to distribute development work; (2) a new distribution channel; and (3) the development of a global community of sophisticated users²⁴, being “geographically broadly distributed since the very beginning” (Tuomi 2002: 170-171).

In sketching the historical links between the development of computer technology, the Internet and Linux it is easy to note that an underlying thread is that all three cases have in common a network of people exchanging and sharing ideas, a network of people that through computer-mediated-communication (CMC) came to represent a community: the “sprawling computer counterculture” (Castells 1996: 353) of hackers. As the Internet became a mass ‘pull’ medium (Castells 1996), the community was able to widen its geographic base, influencing and being influenced by and from the world outside the USA. Yet, although the geographic base widened, the “information inequality” (Schiller 1996) which characterizes the Internet can be sufficiently, and maybe necessarily, be extended to the community of Linux, as CMC “remains the domain of an educated segment of the population [representing] an elite on a global scale” (Castells 1996: 359).

What's in a Name?

It is one of these terms which- followed with popular misleading connotations of security breaking, spread of computer viruses, and other negative associations when it comes to computer technology- rather serves in showing the boundaries of social inclusion and exclusion than generic, empirical applicability: Hacker.

While in the media as in the minds of those who are not intimately involved with computer technology, a hacker may mean one who breaks computer security, usually illegally²⁵, the history of the word and its meaning are somewhat different within the community of self-ascribed hackers. As mentioned in the above history section, the ascription ‘hacker’ was born in the 1960’s and 1970’s within MIT’s flourishing community of computer programmers (Raymond

23 Because of the connection of Linux with GNU there has been a debate on whether Linux should be called GNU/Linux- See “Linux, GNU, and Freedom” (<http://www.gnu.org/philosophy/linux-gnu-freedom.html>) and Torvalds & Diamond 2003: 163. Furthermore, there is a political/philosophical debate between FSF and “Open Source Initiative”, <http://www.opensource.org/>, April 10, 2007. Alternatively, a more ‘neutral’/inclusive terminology has been used, such as FOSS (Free and Open Source Software), <http://en.wikipedia.org/wiki/FOSS>, April 10 2007. Because this study is not directly dealing with intra-Linux development debates, FOSS will be employed to designate the larger community of hackers who work under Free and Open Source standards.

24 In the structure of Linux development, users are as important as developers, since they can be very useful in detecting and reporting problems with the software/programs (called ‘bugs’)- see Appendix: Raymond’s rules for development (# 8, 10, 11) (Raymond 1999: 41- 49).

25 The ‘proper’ name of people who break into computer security is ‘cracker’ and as Raymond (1999: 232) informs us, real hackers have something less than little respect for them: “Hackers build things, crackers break them.” (*Ibid.*)

1999: 8)²⁶. In this work, I have tried to employ²⁷ the meaning of the word as these people who ascribe to and identify with it understand it:

“It means someone how enjoys playful cleverness, especially in programming but other media are also possible. In the 14th century, Guillaume de Machaut wrote a palindromic three-part musical composition. [...] I think that was a good hack.” Interview with Stallman, 2002²⁸

From this view: “Hackers built the Internet. Hackers made the Unix operating system what it is today. Hackers run Usenet. Hackers make the World Wide Web.” (Raymond 1999: 232) The term hackers refers to technical ability in solving computer programming problems and is frequently associated with a certain delight (Raymond 1999) or aesthetic (Stallman 2002²⁹) in doing so. As an emergent CMC culture, it bears its own characteristics, stereotypes and boundaries. As hinted in the beginning of this section, even whether one knows the ‘real’ definition of hackerdom acts as a boundary marker. Similarly to the ‘rite of passage’ (Van Gennep 1960) of an initiation which serves to “rediscover [...] the threshold that separates the connoisseur from the ignorant and the ineffective” (Kirtsoglou 2004: 41), or to nation-building processes which entail the creation of hegemonic stereotypes and boundaries serves in the imagined homogenization of the members of the community (Alonso 1994: 393-394). The mechanism of stereotyping the community simultaneously with forming boundaries of exclusion/inclusion becomes a source of identity and pride: for example, hacker community’s association with ‘nerds’, ‘geeks’ and/or science-fiction movies (Raymond 1999: 245-246; www.slashdot.com). It is also indicative of the elitist boundary formation within the community that even the rules of becoming a hacker are not available to the wider public, hinting to a performative initiation ritual that includes a person’s effort to find out through hard work and alone what the meaning of hacker culture is. In fact, as Raymond notes: “oddly enough there don’t seem to be any FAQs or Web documents that address this vital question” (*Ibid.*, 231). The Jargon File (<http://www.catb.org/~esr/jargon/html/index.html>) is one of the few sources of the colourful history and folklore of this culture, as well as Raymond’s manual “How to Become a Hacker”³⁰.

As a community and culture which has emerged from but is not bounded to ‘West’, the hacker ethic can be described as countercultural in that it challenges as well as crystallizes the dominant, modern hegemonic discourses about individual freedom and collective action. Such themes, broadly fall within the theoretical debates on modernity and everyday practice (Bourdieu 1992, Lefebvre 2000), as well as power and resistance to both proprietary software as well as social exclusion of ‘nerds’ as the Other in a world of public relations and marketing (see Foucault 1982, Leung 2005, Raymond 1999). While there is a broad range of literature dealing with such topics, as well as identity construction in cyberspace, ethnographically there is

26 Also see Levy, S. 1984. Hackers. Anchor/Doubleday.

27 As well as to figure out the anthropological significance of the term through participant observation and empirical methods- see discussion.

28 The Hacker Community and Ethics: An Interview with Richard M. Stallman, 2002. <http://www.gnu.org/philosophy/rms-hack.html>, April 3, 2007

29 *Ibid.*

30 See Appendix II.

a gap of anthropological knowledge related to hacker culture in particular.³¹ Maybe, this is because as a culture borne out of the dominant value system, with highly sophisticated members, it is hard for an anthropologist to reach (maybe because by doing so many of her/ his own stereotypes would need to go under the microscope (Argyrou 2002)), or because of the disguises of power (Gledhill 2000) the very ‘modern’ almost religious dominance that technoscientific views exert). Therefore, having only just touched upon some of the issues around the terminology and the culture behind the ‘name’ hacker, we will return to this fascinating topic in the discussion section to juxtapose the findings with the literature reviewed.

The Architecture of Cooperation

Some, like Raymond (1999), argue that the architecture of Linux and FOSS resembles a ‘bazaar’ of random, self-organised, non-hierarchical structural model, while others, like FSF founder Richard M. Stallman, argue that in software development of Free Software there is always a bigger plan towards which developers concentrate their efforts. And yet others, usually people who have studied socio-technical phenomena such as Linux development, concentrate on the aspects of “dynamic meritocracy” (Tuomi 2002: 175) based on sophisticated systems of values and mechanisms such as reputation (Rheingold 2002) and reciprocity (Kollock 2000). In this section we will take a closer look in the actual architecture in order to aid the understanding of how Linux is socially organized.

The operating system of Linux is divided into three structural spheres:

1. hardware;
2. kernel space (including the operating system kernel and the system call interface); and
3. the user space (utility and application programs).

The focus of hacker activity has been on developing the kernel(Raymond 1999), a process that has been continuous, increasingly complex and active with one new version of the system released every week (Tuomi 2002: 166-167). The programmers’ community is organised around constantly evolving projects, resembling a “community of communities” (Tuomi 2002: 172) in specific, and a “network of communities of practice or fractal organization” (Ibid., 173) in general.³² The programmers in these projects are usually organised around a hub of “central gurus, old-timers and more peripheral novices.” (Ibid.) While everybody who has access to the system can make suggestions and minor additions, the important decisions filter up to a core group of people who are the key developers of the overall kernel project. Linus Torvalds, whether or not a ‘benevolent dictator’ acts as the ‘default’ in the decision making process, and also maintains some subsystems that have no defined maintainer. Torvalds, on his role in the system notes: “I control the Linux kernel, the foundation of it all because, so far, everybody connected with Linux trusts me more than they trust anyone else.” (Torvalds & Diamond 2001: 168)

Looking through the Glass: Prisms of Theoretical Analyses

“The more we forbid ourselves to conceive of hybrids, the more possible their interbreeding becomes.” (Latour 1993: 12)

Linux is equally about technology as is about sociality: “software is constructed as a set of social relationships by everyday discursive social practices” (Carter 2004). Understanding

31 Apart, perhaps, Raymond’s self- ascription as “the hacker culture’s tribal historian and resident ethnographer” (1999: 197).

32 This resembles a fractal organization of Romanesque networks, for an interesting connection see Strogatz, Steven H. 2005. “Complex systems: Romanesque networks”.

that social and technological merge and are in a dialectic relationship with each other is vital in understanding how Linux works and how the community of its developers and users are able to coordinate. In order to do this we have to firstly account for the limitations of the currently available prisms of analysis, modernist approaches that separate the two, namely, technological and social determinism (Escobar 1994). Because we will employ theoretical frameworks from both the social sciences (community, gift-theory, reputation systems) as well as from the ‘positive’ sciences (graph theory, networks, emergent properties), it is important to underlie some of the limitations of such analyses, simultaneously permitting “areas of hybridization” (Lyon 2004: 13) to emerge. This is the purpose of this commentary.

Modernity is an idea- an idea that became self-conscious during the period of the Enlightenment: the scientific and intellectual ‘revolution’ of the 17th century through the rise of a renewed social and political interest concerning knowledge (a rereading of ancient history, rise of science, and an attempt to formalize and organize that knowledge- i.e. Diderot’s Encyclopedie (1772)), the relations of state and church, state and citizens and an idea of equality (to mention just few) as expressed in the writings of the ‘French Trio’: Montesquieu, Voltaire and Rousseau, as well as in the emergence of the ‘philosophes’ in France. The Prussian philosopher Immanuel Kant was the first to coin ‘enlightenment’ (Kant 1784 in Sahlins 1999: ii), crystallizing, thus, the different currents of thought, denoting the start of the beginning of ‘Rationality’ and the start of the ‘Modern’ Era. And if Kant ‘captured’ the spirit of Enlightenment, those ideas gained momentum and substance during the European Expansion, colonization and rise of imperialism.

Enlightenment depicts a period of European struggle to ‘shake off’ the regime ancient of the Middle Ages- a period of middle class growth and strive for power. As such, it denotes a break from the ‘superstitious’ past with the attempt to acquire ‘scientific’ understanding of the world.³³ As such, it is based on the logic of rationality- i.e. human beings as capable of reason, and also that a certain moral ‘social order’ is logical, natural and somewhat harmonious. The idea of Modernity, thus, presupposes a certain (culturally specific, western) logic: the existence of causes and effects, or in its philosophic term, the existence of causality. I do not wish to show how causality is a ‘bad’ thing. Rather, I wish to show that causality is only just another “option” (employing, here, the ‘therapeutic’ model of Rorty 1980), another way to look at the world and explain what is seen, and not the only available prism.

Modernity encompasses the existence of opposite and often antagonistic categories (i.e. object/subject, mind/body, space/time, modern/traditional, technology/culture, developed/developing); which are not only empirically and analytically unreliable if not incorrect (in that they fail to account for the dynamic relationships of constant change), but politically dangerous because of their essentialisation and hierarchical/ hegemonic division of the world. The idealism of essentialisation rests in the presupposition that “relations are derived from the nature of things” (Deleuze 1991: 109), and as such not only are they natural, but they are also fixed and unchangeable in time. It is this invented modern idealism that fosters political manipulation of Kipling’s “white man’s burden”, in order to ‘civilize’ the ‘savages’, to export ‘democracy’ to the ‘primitive authoritarians’. Therefore, the idea of modernity is dangerous in its political disguise as ethical or moral imperative and implications: the exportation of power relations, domination, exploitation and hegemony (Argyrou 2002; Gledhill 2000). In the case of technology (Castells 1996; Wise 1997), different analytical prisms produce utopias and distopias accordingly,

33 Of course, maybe only too late, we now realize the fluidity of such ‘break’ and the cultural construction not only of ‘superstition’ but of ‘science’ as well.

determinisms that reduce accordingly either agency or impact of social/physical constraints (Latour 1993).

Yet another political, historical and philosophical problem with the concept of Modernity (and with post-modernity by extension) is that of the ‘here/now’ implication. Where does the authority to call ourselves ‘modern’ derives from? Are we ‘modern’ in opposition of what the people before us who were ‘primitives’, are we ‘modern’ because we are contemporary (but don’t all people are contemporary in relation to their historical period?), and what will happen to the people and to the generations after us- are they going to seize being modern, as they going to be post-modern, or are they going to continue being modern (and if yes, how are they going to account for their differences with ‘us-modern’)? A brief history of the word puts things in perspective³⁴. Transforming a word whose initial meaning was “being at this time, now existing” into an ascription of a historical time-period (which some argue that we have never been, others argue that we are still in it and yet others say we have gone beyond) is best analysed as a discursive “game of knowledge and power” (Argyrou 2002: 6). It pertains to the overall hegemony of certain power relations of the dominant part- showing how the dominant ‘West’ has power not only over other parts and people of the world, but how it controls both universal space and time by spatially transforming and reproducing its hegemonic relations into and endless ‘here’ and ‘now’. The concept of modernity, thus, clearly suggest the ‘eternity’ of those ‘modern’ power structures by definition.

Through introducing the history and the political implications of modernity, we have shown how its core ideal is based on illusionary ‘rational’ dichotomies, and on political and philosophical problems of imposing and perpetuating hierarchies and definitions upon people as well as time and space. Hopefully, this discussion has shown how modernity is related to discourses of science, technology and society. This discussion relates directly to Linux, as this ‘bundle’ of software, people and machines isn’t only a technological achievement, and it’s not only about sociality, but rather a hybrid of both(Latour 1993), and therefore we need to account for the limitations of the current analytical prisms available.

The Emergence of an Imagined Community: Networks and Social Analysis

“... the notion of community encapsulates both closeness and sameness, the distance and difference; and it is here that gradations of sociality, more and less close social associations, have their effects. For members of a community are related by their perception of commonalities (but not tied by them or ineluctably defined by them as are kin), and equally, differentiated from other communities and their members by these relations and the sociation they amount to. In short, ‘community’ describes the arena in which

34 “The word modern, first recorded in 1585 in the sense "of present or recent times," has travelled through the centuries designating things that inevitably must become old-fashioned as the word itself goes on to the next modern thing. We have now invented the word postmodern, as if we could finally fix modern in time, but even postmodern (first recorded in 1949) will seem fusty in the end, perhaps sooner than modern will. Going back to Late Latin modernus, "modern," which is derived from Latin modo in the sense "just now," the English word modern (first recorded at the beginning of the 16th century) was not originally concerned with anything that could later be considered old-fashioned. It simply meant "being at this time, now existing," an obsolete sense today. In the later 16th century, however, we begin to see the word contrasted with the word ancient and also used of technology in a way that is clearly related to our own modern way of using the word. Modern was being applied specifically to what pertained to present times and also to what was new and not old-fashioned. Thus in the 19th and 20th centuries the word could be used to designate a movement in art, modernism, which is now being followed by postmodernism.” (<http://dictionary.reference.com/browse/modern>, browsed on 03/03/07)

one learns and largely continues to practice being social. It serves as a symbolic resource, repository and referent for a variety of identities, and its ‘triumph’ (Cohen 1985:20) is to continue to encompass these by a common symbolic boundary.” (Rapport 2006: 116)

The correspondence of Linux development with other economic and technological, historical, political and socio-cultural phenomena has been already noted. The literature on Linux and the social movement of the FOSS, mostly written by principal actors in it such as Stallman and Raymond, mention the ‘community of hackers’, the ‘culture of programming’ and such other socio-technological hybridizing ascriptions. The previous section dealt with the cultural construction of the identity in ‘hackerdom’. In this section, the notion of ‘community’ will be explored

With a definitional debate as old as anthropology itself, the concept of ‘community’ is by no means universal neither in its terminological nor in its empirical application. It is rather a concept whose definitional origin usually suggests more about the analytical/philosophical prisms employed than the empirical reality it seeks to understand. Rapport (2006: 114- 116) separates the anthropological approaches in terms of ‘traditional’ and ‘symbolic’. Named ‘traditional’ because of their functionalist structuralist heritage, they share the view that communities “came to be marked by a fair degree of social coherence” (*Ibid.*, 114). This ‘bundle’ of theories have differing degrees of focusing on the ‘building blocks’ of communities, which can be on the grounds of (1) shared interests, (2) common locality, or (3) shared social structure (*Ibid.*). With its analytical focus on producing an ‘objective’ ‘universal’ definition of the characteristics of ‘community’ based on ‘check lists’, this approach inadequately describes the richness, diversity and flexibility of people and communities to reappropriate meaning and definition in their own actions and practices.

What Rapport (2006) categorizes as the ‘symbolic’ approach to ‘community’ is a more recent development in anthropological thought based on the critique of modernist-postmodernist analysis of social construction (Escobar 1994: 212). This theoretical position does not focus on the structural logic of communities, but on their relation with symbolic capital such as the construction, negotiation and context of meaning:

“... as Gregory Bateson put it succinctly: things are epiphenomena of the relations between them; or as Barth elaborated, social groups achieve an identity by defining themselves as different from other such groups and by erecting boundaries between them (1969).” (Rapport 2006: 115)

Underlying the fluidity, relativity and social construction of the concept, this approach doesn’t recognize an essentialist definition of ‘community’, since, as Deleuze puts it “relations are external to their terms” (1991)- or that meaning as well as social relationships are historically and socially situated, created manifestations of symbolic significance and prone to change (if not always in a state of flux and becoming). Making use of the inherent ambiguities and cultural appropriations of the concept to suggest that ‘community’ encompasses a combination of fluid and culturally specific notions, the concept as well as the theoretical debates have found prosperous ground in the anthropology of nationalism and ethnic-group formation (Barth 1969; Banks 1996). Thus, Anthony Cohen argues that community “must be seen as a symbolic construct and a contrastive one; it derives from the situational perception of a boundary which marks off one social group from another: awareness of community depends

on consciousness of boundary" (emphasis added, Cohen (1985) cited in Rapport 2006: 115). Hence, the symbolic construction of meaning as a process of forming boundaries and identities becomes the locus of defining 'community'. As such, 'community' becomes the "social milieu" (Rapport 2006: 116) on which meaning, identity and boundaries are symbolically constructed and enacted based on perceived homogeneity and difference (Alonso 1994).

Doesn't this approach sound much more appropriate for an understanding of the Linux and FOSS community?

Firstly and foremost, Linux is described as a 'community' by its members (Raymond 1999), an act of conscious active discursive construction of identity, sameness, otherness, and boundaries. Yet, this definition of community doesn't allude to a specific locality, as in traditional approaches, but to a geographically distributed collectivity (Ratto 2005) of culturally, socially and politically dissimilar individuals. In this continuous and conscious discursive formation, the members of the community choose to define themselves as members not only in terms of technical ability but also using several symbolic markers, such as the defining terms of hackerdom, legitimization of history(technology) and ideology (FSF and freedom of speech). An interesting approach to the community boundary formation is also the role of resistance to dominant/hegemonic and antagonistic views of software production. Microsoft corporation and Windows OS present themselves as the logical antithetical Other against which the members of Linux symbolic community will erect a consolidating boundary. In this effect, the section of findings/discussion will investigate whether and to what extent the anthropological theory of community construction in term of defining boundaries can adequately explain aspects of the formation of Linux community.

Networks

Simply put, a network is a graph of points and lines (Sanjek 2006: 396). The points represent units of analysis (actors, groups, organizations, etc), while the lines represent relationships or ties between them (Boissevain 1979; Nooy, Mrvar, Batagelj 2005: 7). The strength of network analysis or social network analysis (SNA) lies in the combination of:

- being a visual medium of representation of dynamic and interdependent relational ties between units of analysis (Boissevain 1979: 392); and
- providing 'objective' quantifiable results.³⁵

Network analysis derives from a branch of mathematics of Graph Theory (Watts 2004) and has a long history both as a theoretical as well as a methodological tool. Its origins in anthropology can date back to Leach, Barnes and Gluckman, whose works challenged both social structure and its relation to people as well as presuppositions of "equilibrium" (Gluckman 1968, Boissevain 1979). It has been used widely, more as a methodological tool, in areas such as kinship, and gift/commodity exchange (Cook & Whitmeyer 1992; Kollock 2000).

Today, the social sciences undergo a revival in SNA/ network theory, partly because it can explain coherently decentralized/self-regulatory phenomena such as the Internet (Tuomi 2002; Escobar 1994; Castells 1996), emergence of globally distributed individuals into virtual communities (Carter 2004; Rheingold 2002; Dicks & Mason 1998), and other social phenomena such as the globalised tensions between local, global and "transnationalism" (Kearney 1995; Leung 2005; Ratto 2005). In an increasingly interconnected world³⁶ in which, as Castells

35 Of course, 'objectivity' has its own bias and problems, for an interesting approach see Latour 1993: 6.

36 With a bit of 'salt': our world has not only recently become interconnected due to capitalist hegemonic forces of the West, historically, our world has always been interconnected- see Wolf, E. Europe and the People without History

describes it, a restructuring capitalism is rebuilding around the emergence of a different mode of development, that of “Informationalism as the new material and technological basis of economic activity and social organization” (Castells 1996: 14), network theories of connectedness seem to be able to make order out of chaos:

“Chaos is in the air. Somewhere, a butterfly (is it always the same one or are there lots of them?) has flapped its wings, and we can already hear the wind whistling among the word processors. [...] Chaos is to be the order of the day.” (Abrahams 1990: 15)

Contrary to theories of processualism³⁷, ‘complexity’ theory “focuses on understanding the emergence of self-organizing structures that create complexity out of simplicity and superior order out of chaos” (Castells 1996: 64), arguing for the unpredictable contingency of some phenomena in showing how simple systems can have complex behaviour and vice versa (Escobar 1994: 221), while at the same time showing how phenomena that are unique in particular maybe similar in general (Bentley & Maschner 2003: 3). The history of this theory is in some respects very similar to that of Linux- it begun from Artificial Intelligence labs in the USA during the 1980. Below, we will describe the nature of Linux community network, however, it is important to keep in mind that although complexity theories have had a prominent influence in social sciences in their attempts to explain social phenomena, a certain scientific determinism derived from the authority of making ‘scientific’ predictions.

Tuomi (2001: 173) describes the structure of the Linux community as a “dynamic meritocracy”, in where authority and control is closely associated with the produced technological artefacts, thus forming a “network of communities of practice”. As a system of collaborating networks, Linux appears to work under self-organised principles and having more than one centre. Without a strict hierarchy and with such lack of centrality, Linux could appear very anarchical and inefficient (indeed, this was the debate between Torvalds and Tanenbaum, creator of Minix, in 1992). However, the case is the opposite; Linux is very successful exactly because of its openness to innovation and its flexible nature. Indeed, one of the characteristics of the network theory is the challenge it poses against the criticality of centrality: “one of the great mysteries of large distributed systems [...] is how globally coherent activity can emerge in the absence of centralized authority or control.” (Watts 2004: 51) For Watts this is possible if we take into account three parameters regarding communication between the members (nodes) of any network: the critical point of the system (transition), the correlation length between its members and the overall condition of criticality (Watts 2004: 63-65). Recent studies on software development and computer programs have shown that Linux and Open Source development suggest “signatures of small-world and scale-free properties” (de Moura, Lai, and Motter 2003; also Challet and Le Du 2005). The question that arises here is that maybe Linux is a too small and too connected a world to be functional. If only half of the programmers attempted to reach the only one central hub, the kernel (Linus), at once, how could ever the kernel be able to maintain stability?

Indeed, controlling the kernel is one of the greatest challenges in Linux. The constant flow of suggestions and improvements equates with constant risk of loosing maintainability, while the golden rule appears to be the equilibrium between control and innovation/technology design (Tuomi 2001: 176). According to network theory, there are two ways in which it would be

³⁷ Processualism is a form of evolutionism and holding that change in social or physical environments has a unified gradual process. (Castells 1996: 29; Bentley & Maschner 2003: 3)

interesting to look at Linux. Firstly, its lack of centrality in the sense of central organization and hierarchy (apart from Linus' role in maintaining the kernel) and its open accessibility to every potential user in changing it are very similar to the properties behind the theory for random networks, where nodes are supposed to have the same probability of links (in this case it could be the probability of having an equal share on the kernel). On the other hand, scale-free networks are not as 'democratic' as random ones. Indeed, one of the properties of scale-free networks is that "some nodes have a tremendous number of connections to other nodes, whereas most nodes have just a handful. The popular nodes, called hubs, can have hundreds, thousands or even millions of links. In this sense, the network appears to have no scale" (Barabasi & Bonabeau 2003: 62). Although no specific research has been undertaken to collect empirical data on Linux' network structure, the evidence of the existence of communities within communities maintained by some 'core' programmers, and even the existence and function of Linus Torvalds as a 'super-hub' does strongly suggest that a scale-free network would be more appropriate for the Linux community. Furthermore, Tuomi's compromise between democracy and oligarchy in a meritocracy (which in Greek meridion means the part that someone deserves), although it reminds Plato's idealism, it does seem like a fair and above all accurate description of the structural organization of the Linux community network.

Linux and The Anthropology of Gift: The Ties that Bind

The idea of the 'Gift' has been one of the most compelling theoretical forces within the discipline of social anthropology, enduring in the passage of time radical waves of criticism and change from within as well as outside the confines of the discipline. The history of the concept goes as far back as the history of the discipline itself: the first functionalist analysis of the then 'traditional', 'archaic', or 'savage' societies. Mauss was the first one to write a coherent history and analysis of gift exchange (Mauss 1967), based on a radicalization of Durkheimian functionalist social theory (Komter 2005). Malinowski and Boas were among the first ethnographers to describe instances of gift-giving like the kula ring and the Indian potlatch, making descriptive connections between the 'thing' given, the giver, the receiver and the socio-cultural relations formed and maintained through the transaction. Later, Levi-Strauss through his structuralist analysis demonstrated the 'alliance' theory of kinship through the exchange of women as "the supreme gift" (Levi-Strauss 1963; Komter 2005), rising important questions that would lead to feminist critique and focused analysis of the role of women in gift exchange, as well as the cultural construction of the meaning, value and gender of the gift (Strathern 1988).

The development of Linux OS, and specifically the way its developers sustain the growth and expansion of both the community as well as the software technology, makes an anthropological analysis based on 'gift' theory not only relevant but highly necessary. Eric Raymond, one of the gurus of Linux and FOSS, relegates the analysis of hacker culture in the realm of 'gift culture':

"Like most cultures without a money economy, hackerdom runs on reputation. [...] Specifically, hackerdom is what anthropologists call a gift culture. You gain status and reputation in it not by dominating other people, nor by being beautiful, nor by having things other people want, but rather by giving things away. Specifically, by giving away your time, your creativity, and the results of your skill." (Raymond 1999: 241-242)

And, indeed, studying and connecting the anthropological theory on gift with the apparent paradox of people giving their time, creativity and thought to a voluntary activity, may shed some

light into not only why so many people contribute to a project like Linux, but what the benefits might be from doing so. Thus, the anthropological concept of the gift, with its long history and debate within the discipline, is a robust analytical tool with which to examine the emerging structure of the techno-sociocultural reappropriations within the Linux community.

But what exactly is the concept of the ‘gift’ and, how does it relate to the development of an operating system?

Mauss (1954) in his classic work defines the ‘gift’ as a form of social deception which creates obligatory reciprocal ties, as:

“prestations which are in theory voluntary, disinterested and spontaneous, but are in fact obligatory and interested. The form usually taken is that of the gift generously offered; but the accompanying behaviour is formal pretence and social deception, while the transaction itself is based on obligation and economic self-interest.” (1967: 1)

Mauss’ definition underlines some important characteristics of gift exchange, such the social ties of ‘debt’ created to the receiver from the giver to reciprocate. The creation of ‘debt’ marks a beginning in the formation of social ties: the acceptance of the gift and the underlying promise to reciprocate constituting the beginning of interdependent social relations, and a promise for its continuation. When Mauss defines the ‘gift’ in terms of the social expectations and relations it creates and sustains, he does so in order to show how gift exchanges are “total social phenomena” (1967: 1) because in them “all kinds of institutions find simultaneous expression” (*Ibid.*). As the encompassing concept of a total social phenomenon, the interactions and transactions emerging from gift-giving become themselves the “technology of social relations” (Bell 1991, cited in Kollock 2000), in which and by which groups and individuals engage in ongoing interdependent relationships. These relationships are characterized by constant negotiation, demonstration of power and prestige (Foucault 1982, Rheingold 2002, Raymond 1999, Bourdieu 1992), transcribed by cultural and social norms. This particular form of sociality is what makes communities possible and allow the existence of networked links in the first place.

Mauss analyses gift exchange through the prism of contract (Mauss 1967: 1-6).³⁸ There are two points of importance in this contractual exchange: firstly, the transaction is not necessarily made between individuals but between “moral persons” (*Ibid.*: 3), and secondly, the value of the gift can be material, informational and/or symbolic (Cook & Whitmeyer 1992). In Maussian analysis, ‘moral persons’ meant clans, tribes and families. In the realm of cyberspace, the symbolic value of “moral persons” is translated into publics: a public sphere of commons, which produce and consume public goods through collective as well as individual action (Rheingold 2002; Smith & Kollock 2000). Could Linux be seen as a public in which a community of commons interacts based on the contractual sociality of Maussian gift-exchange between not only individuals but virtual ‘moral persons’?

As both the Internet and Linux exhibit a state of change in the content and in the people who compromise it at a given time, and as its character is openly accessible to its users, it becomes a public: yet another freely available field of social interaction (Bourdieu 1992; Escobar 1994). In this field, when its inhabitants make a contribution (from the creation of a ‘home page’ to the

³⁸ The concept of the contract is based on the Kantian assumption of rational individuals as well as dichotomies- an ‘enlightened’ idea, which should be handled with care in order not to pass cultural assumptions as theoretical universals.

development of the kernel of Linux), it is like they are offering a ‘gift’ for the whole cyberspace. Since there is not necessarily a peer-to-peer interaction, and taking into account the distributed and open nature of the Internet, the potential recipient of the gift can be anyone. When a resource is made available so that all may benefit without having helped in its creation and without the obligation to reciprocate, this resource becomes a public good (Rheingold 2002: 32; Kollock 2000: 223).

The word commons, Rheingold informs us, means a common resource and it originated from the designation of pastureland which was used by individual herders as a common feeding ground (Rheingold 2002: 34). In an article entitled “The Tragedy of the Commons”, Garrett Hardin describes how common grounds with unregulated access are deemed to depletion and ruin: “ruin is the destination toward which all men rush, each pursuing his own best interest in a society that believes in the freedom of the commons. Freedom in a commons brings ruin to all.” (Hardin cited in Rheingold 2002: 34)

The connections between public goods and commons are many: anyone can have free access, both can be viewed as ‘gifts’. Whatever the context might be, from herders’ pasturelands, to the Internet to Linux, one aspect remains constant: both a public goods as well as a commons allow unlimited access and use of the offered resource. And if we’re talking about pastureland the danger of depletion appears to bring ruin, if we talk about the Internet and Linux, the danger of ‘free-riders’ poses the threat, in both cases uncovering the limitations and potential inequalities of gift-giving and gift-receiving. Yet, if all public goods and commons are deemed to fail due to selfish individual exploitation, how, then, has Linux succeed in making it so far, and how close or far away is the theoretical failure of this collective commons project predicted? Isn’t there an alternative for the existence of a successful public good?³⁹

The answer partly lies in the relationship between the notion of ‘community’ (see above section) and the mechanism of its expression and manifestation through the production of social ties and increase of solidarity (Komter 2005) through gift-giving. The balancing of the emerging social dilemmas of costs and benefits are also important: similarly to the common pastureland that is deemed to be ruined by unlimited grazing, in the sphere of free software there emerges the problem of ‘free-riders’. The designation ‘free-riders’ applies to people who use the resources produced without contributing. For example, Kollock (2000: 225, 230) analyses the potential risks for the failure of Linux as the risk of free-riders temptation: if the number of people who use Linux without contributing something back to the community increases so that the people who do contribute are outnumbered (‘production function’ in negative), then the result will be a ‘tragedy of commons’ due to the imbalance between limited resources and unlimited demand. However, both mechanisms of gift-giving and community solidarity (such as the formation of boundaries and creation of ‘public’ debt) are inextricably related in that they provide an answer to the social dilemma of ‘free-riders’.

In order to understand the mechanisms of gift-giving with their relation to community solidarity, let’s return to Mauss’ three principles of gift exchange:

1. the spirit of the thing given;

³⁹ Before proceeding to an answer, it is wise to acknowledge the nature of the above questions, and subsequent debates. We have already explored how many modes of analyses take for granted a modernist, evolutionary perspective which prefers quantifiable generalizations and ‘applicable’ universals, resulting in totalizing deterministic outcomes. So, it is important to keep in mind that although pasturelands and the Internet have some important structural similarities, this does not mean that they are the same nor that we can expect *de facto* outcomes. More often than not, in the complicated or simple (depending on how one looks at it) world of everyday interaction and change, the parameters and contexts of action and interaction are so numerous, that we often need to look closer and tenderly, in order to be able not only to spot them, but to challenge deterministic approaches in the rich, fruitful and undeterminable mosaic of human interaction.

2. the obligation to give; and
3. the obligation to receive.

Mauss argues that these mechanisms of gift exchange create and maintain social ties, reproducing and challenging simultaneously the social structure (see Leach (1970) 2006: 128-135). The explanations and interrelations between the three fundamental principles of gift exchange will help us understand how the mechanisms of gift exchange not only promote solidarity (Komter 2005), but how they can potentially be the building blocks of social functionality.

It is not only in ‘traditional’ societies that people believe that when a gift is made the spirit of the giver is somehow intertwined in the gift (Mauss 1967). From Mauss’ analysis of Malaysian gift-giving (*Ibid.*) to modern day Japan (Henry 1999: 60- 62), there is an underlying assumption that in the materiality of the gift exchanges there is something of the giver given as well. And while the giver may appear as giving something of himself within the gift, the action of giving perceives returns prestige to the giver while demonstrating the power relationships between gift giver and receiver (Mauss 1967: 72; Cook & Whitmeyer 1992: 112-113; Komter 2005: 23- 24) The imbalance created by the act of giving in turn, creates the necessity or the obligation to reciprocate. Hence, gift- giving not only transmits the spirit of the given but creates reciprocal ties between giver and receiver. As already demonstrated by the history of Linux and FOSS, the ‘spirit of the thing given’ bears upon the ‘signature’ of the giver- translated in symbolic capital of reputation/prestige and recognition within the hacker community (Raymond 1999), but also viewed as products of collective/public property (Mauss 1967: 65) by the underlying GNU GPL ideology: the view that knowledge is a collective artefact belonging to all.

A Foucaultian approach to the power relations of gift-giving demonstrates that the subject of power (Foucault 1982) is to be found in the symmetrical or asymmetrical power relations formed. Hence, economies and/or cultures based on gift-giving “are driven by social relations while commodity economies by price” (Kollock 2000: 222), and the currency they run on is power accumulated through status, prestige and reputation (Kollock 2000, Raymond 1999, Rheingold 2002).

The bipartite structure of 'the obligation to give and to receive' (Mauss 1967: 10) establishes relations and alliances, In terms of Linux success this can be understood as the paradox of “the wisdom of crowds” (Suroweicki 2004: 72-74). The on-going interaction facilitated by technological advances (Internet), groups and the specifics of cyberspace are examples of how social life is a constant give and take (Mauss 1967: 27). Thus, both the Linux community as well as the code itself are open to suggestions and innovations- the source code can be described as a gift on its own, which, following a hacker ideological taboo, much like in the kula ring one may not keep for one’s self!

Having explored the broad theoretical reasons for the successful cooperation in the production and maintenance of public commons such as Linux, let us now turn to a specifically related study. Kollock’s work, the only relevant anthropological study of Linux cooperation, describes Linux as the “impossible public good” (2000: 230), focusing on precise reasons for its success and the motivations implicit in its community basis. The success of on-line cooperation rests on an intricate balancing between the costs and benefits of individual and collective engagement, while one of the important ‘rules’ is that the ‘production function’ (“relation between the proportion of the group contributing to a public good that is produced”, *Ibid.*, 225) should be large enough for continuous growth and for withstanding free-riders’ depletion of the common. Kollock’s findings are illuminating and precise and are summarized in the tables below:

Table: Motivations for Contribution in Linux (Kollock 2000: 227-229)

1 Anticipated Reciprocity	<ul style="list-style-type: none"> • network-wide accounting system • potential benefits from group reciprocity • rough balance over time • well defined and defended group boundary
2 Reputation	<ul style="list-style-type: none"> • prestige • on-going interaction • identity persistence (archives) • group boundaries
3 Efficacy	<ul style="list-style-type: none"> • an effect/ action on the environment (social/technological)
4 Altruism	<ul style="list-style-type: none"> • self-image • someone needs and I have it
5 Attachment or Commitment to the group	<ul style="list-style-type: none"> • costs of sharing minimized • no social dilemma

Table: Reasons for the Success of Linux Cooperation (Kollock 2000: 230-231)

1 Ability of Internet to facilitate communication	<ul style="list-style-type: none"> • reduction in communication and coordination costs • history • decentralization
2 Shape of production function	<ul style="list-style-type: none"> • volunteers
3 Intrinsic interest and challenge	<ul style="list-style-type: none"> • issues of coordination • personal usefulness • GNU GPL creates and incentive structure
4 Internal inspection and critique	<ul style="list-style-type: none"> • Informal monitoring system

Methodology

As a curious and cautious intruder, I sneaked into the ‘world’ of Linux development in the early hours of Monday 20th in November 2006. A few hours before, I was talking to a friend from back home about my dissertation plans, about my desire to study and try to understand how the phenomenon of Linux successfully exists, how people manage to coordinate and cooperate without apparent hierarchy, based on principles of freedom of speech and sharing of information. He told me that if I wanted to study developers, and not just Linux users, I should stop looking in forums and newsgroups: “developers prefer mailing lists” he said, prompting me to a relevant website (<http://vger.kernel.org/vger-lists.html>). And so, with ridiculously very little prior computer knowledge, I went to the website and randomly subscribed to eight of technically-oriented mailing-lists, shown below with current subscription numbers:⁴⁰

- Linux-lugnuts: 41
- Linux-net: 1021
- Linux-new-lists: 112
- Linux-newbie: 431

⁴⁰ Mojomo Lists at VGER.KERNEL.ORG, <http://vger.kernel.org/vger-lists.html>, April 20, 2006. At the time, the website did not include information about what each mailing list is about and how many subscribers it has. Therefore, my choice was quite random; luckily, some of the mailing lists turned out to be more active than others rather quit ones!

- Linux-word: 46
- Linux-hams: 383
- Linux-gcc: 256
- Linux-doc: 135

The rationale was to conduct an ethnographic study based on participant observation of the behaviour of the above-mentioned lists. Although I have tried to produce a ‘traditional’ anthropological ethnography of the community of Linux developers in that I have been looking in the discursive construction of culture and community, methodologically, this project belongs to the growing category of ethnographies of ‘cyberspace’ (Carter 2004, Markham 1998, Leung 2005, Rheingold 2000). Characterized as “Interface Anthropology” (Launel cited in Escobar 1994: 218), ethnography conducted in computer-mediated environments constitutes the research ‘field’ as a much more fluid entity, with its own rules and regulations, and a problematic that in some respects is different to the ‘traditional’ anthropological field, and in some bear the same epistemological issues. Taking into account the specifics of the research question, i.e. the technical function of the mailing lists and my technical incompetence, more observation than participation has been conducted.⁴¹

I received more than 700 e-mails in the course of my 5-month observation of the mailing-lists, of which most are exclusively of a technical nature (codes, programming, relevant developing questions). The statistics of my interaction are simple: I read most e-mails, I understood some, I sent a questionnaire for this study⁴², I received 50 replies, and I replied to one question sent through the mailing lists. In the course of this interaction I painstakingly tried to separate the social from the technical, the cultural from the scientific, only to find that within what is called the “religious programming” (<http://www.tux.org/lkml/#ss15>) culture of hackers, social and technological space merge⁴³, rendering such Kantian dichotomies obsolete. During my time as a ‘visiting ethnographer’ or an intrigued outsider, I have seek to understand obvious or hidden cultural traits and norms in between lines of code, often with the generous help of friends engaged in the culture, and it is some of these findings that I wish to present here. In this section, I wish to share some of the revelations of this journey, gathered by means of participant observation, a questionnaire sent to the mailing lists, and by discussions, interviews, personal e-mails and simply following informant’s links to the wider world web.⁴⁴

Through participant observation and through the use of the survey I have come to realize that some of the anthropological ethical guidelines may at times conflict with ‘hacker ethics’. For example, I have tried my best to avoid ‘undue intrusion’ (ASA Ethical Guidelines, http://www.theasa.org/ethics/ethics_guidelines.htm), however, it is considered intruding and poor ‘netiquette’ to send an e-mail to the aforementioned mailing-lists which is not directly related to its technical ascription. Thus, from past experience and for future reference, I have learned that Linux User Groups (LUG) are better places to carry out surveys. This may account for the relatively low intake of replies, and the diversity in the attitudes about the questionnaire.

41 Specifically, there was one time which I was able to provide an answer, and thus reciprocate mimicking behaviour, when an e-mail was asking for bibliographic references about the success of Linux (March 3, 2007).

42 See Appendix III.

43 Also see Wise 1997.

44 Unfortunately, the interview material that was conducted in Greece during 2006-2007 with Linux specialists and ‘newbies’, has been omitted from the study as the focus is solely on those developers and users of the mailing list.

And a last note: Having ‘stayed’ there for more than 5 months, now, that this dissertation is almost complete, I find it hard to unsubscribe from the mailing lists. Very often, due to my lack of technical skill and the nature of my anthropological enquiry, I have felt like an intruder, a strange outsider, an unaware tourist, or, at better times, a visitor, a traveller, an ethnographer. With their wit, humour, creativity, their sharp criticisms and encouraging comments, the people at the receiving terminals of the mailing lists have challenged and changed my ways of perceiving software, knowledge, sharing, who they are, who I am... Taking into account the basic technical skills which I lack, this experience- these finding and conclusions- may show more about my relationship to the culture studied than the culture *per se* (Ruby: 2001), a relationship just starting, like an unfinished initiation rite...

I'm a visitor here. I'm a violent fact.

I observe the stars in the depths of a pool. We both stare at the prospect of an artificial world.

A possibility before sundown. If he lets me I'll become a tenant.

K.B.

Participant Observation in Linux Kernel mailing lists

Having spent a 5-month period reading e-mails sent through vger.kernel.org mailing lists, this section attempts to summarize instances of culture, reciprocity and cooperation distilled from lines of code and technical questions. Firstly, I provide some general notes, and then I proceed to analyse two case studies, exemplars of not only the nature of e-mails sent but also of reciprocity and cooperation.

- E-mails about programming questions, kernel problems, bugs, patches, new releases
- Religion and Politics are prohibited issues; members of mailing lists avoid getting involved in any kind of political/ideological debate
- Writing style usually very straightforward, direct use of singular pronouns to address recipients
- Many replies to questions often include “do your searches better” and “read archives”
- Many members at the end of their e-mail their signature apart from their name/nick name includes kernel version, distribution and amateur radio location code
- Microsoft and Windows OS are frequently written as ‘Micro\$oft’ and ‘Windo\$’
- Questions are usually answered in a matter of hours or a couple of days. However, some questions take longer to answer, some question receive many answers and some none. This depends on the interest a question arises to the rest of subscribers.⁴⁵

Case Study A: Reciprocity

This e-mail discussion thread is in some respects representative of what takes place in the mailing lists, such as the technical questions asked, the ways they are dealt and answered, the characteristic Linux-hacker jargon, the networks of information flow, what makes a problem ‘interesting’ and worthy of attention. Yet, in some other respects it is very different from most of the e-mails sent through the list:

1. It is one of the longest, if not the longest, e-mail thread that I have received during the fieldwork period: 17 e-mails sent through a period of a week and a day (Tuesday 28 November to Wednesday 6 December);

45 See How To Ask Questions The Smart Way, <http://www.catb.org/~esr/faqs/smarter-questions.html>, April 20 2007.

2. The initial question was posted by a person who was not subscribed to the list. This is not considered ‘good practice’ or ‘netiquette’ by the mailing list members⁴⁶. Also, it underlies the importance or reciprocity in the community. However, a lot of people, some of the from the most ‘famous’ developers took an interest in the e-mail. Why?

The question sent by Bob⁴⁷ was about some very technical aspect of the “networking code”⁴⁸, and in the end it asked to be copied in any response as he was not on the mailing list. An hour later, one of the most active writers in the mailing list⁴⁹, Alex, replied:

“This is a common confusion. Read the mailing list archives.”

Such reply is rather common as it is considered a waste of precious time to ask questions that have been answered⁵⁰. However, it can also be seen as a boundary marker as the person asking the question is easily identified as an ‘outsider’, who probably hasn’t ‘done his homework’ and asks other people to do it for him. Yet, Bob chooses a wise strategy in his replying, in addressing Alex personally, and humbly clarifying his problem.

The next reply comes from Paul- one sentence in the tone of a command, demonstrating with it the authority and knowledgeable prestige of an ‘old timer’:

“At no point did you *delete* the address, it’s still there.

ifconfig –a ”

Bob’s next reply demonstrates that he has done his homework and is ready to follow advice. In a long mail filled with code he has followed Paul’s command and shows the results. This gains him the attention of one of the most well-respected and high-ranking developers, Chris⁵¹. Chris offers an alternative advice. However, the fact that one of the high-status developers has replied to a question sent by someone outside the mailing list members does point out that both Bob’s question as well as his conduct have deemed him and his enquiry ‘worthwhile’.

An e-mail is sent to the mailing list by Nick, commenting on Bob’s problem (but without copying Bob in recipients, so Bob never gets it). This e-mail is different from previous one as it doesn’t only offer a (vague) technical answer, but in its commentary about Microsoft, and the connotations of Windows OS ‘stupidity’, reflecting a resistant attitude and community formation as in opposition to the dominant power structure:

“Maybe you should search your solution in the Windows common bugfix
solution tutorials where the common rule is defined this way:

If (strange_problem) {

46 See “The Linux-kernel mailing list FAQ”, <http://www.tux.org/lkml/>, and “How to Ask Questions the Smart Way”, <http://www.catb.org/~esr/faqs/smarter-questions.html>.

47 All the names used are nick-names and not real.

48 Don’t ask what that is, I have no idea!

49 During my participant observation period, he was very often replying to e-mails, hence how he seems ‘prominent’.

50 Also see Appendix II, “How to Become a Hacker”.

51 ‘Chris’ is one of the people listed under the “Who’s Who” section of “The Linux-kernel mailing list FAQ”, <http://www.tux.org/lkml/>, among Linus Torvalds, Alan Cox and other founding figures.

Do "reboot the system"

}

(c) by BillySoft

[...[

It's simple a question of "trust"

Interestingly, no-one replied to Nick, or copies him in future e-mails, nor is he copied in the further communication as all other participants are (in terms of personal e-mail address). It becomes apparent that not only the technical aspect of the question is engaging on its own, but that people of the mailing list don't want to get involved in opinionated 'flame' wars.

The next replies are between Bob and James: James takes a genuine interest in Bob's problem, and after 8 e-mails they manage to solve the problem successfully.

The table below summarises the interactions, people and e-mails involved in this case-study:

Participant	# of e-mails that participants sent	# of nodes sent by participant (direction of link)	# of nodes sent to participant (direction of link)	Total links
Bob	9	22	6	28
Alex	1	2	14	16
Paul	1	3	1	4
Chris	1	3	10	13
Nick	1	1	1	1
James	4	14	3	17
Mailing-list	0	0	17	17

Case Study B: Coordination

The following query was sent to the linux-newbie mailing list:

Hi list,

Whenever there is a change in the kernel API (or a new API is introduced), all of the drivers that use the older API need to be changed (or recommended to be changed). I believe it is the responsibility of the person changing the kernel API, to change all the drivers that have found their way into the kernel code?

How does this happen? Because the person who brought the change in the API might not know the internals of all the drivers?

Is there any way volunteers like me can help in this exercise?

Thanks

Michael is asking one of the founding questions of this project: in such a self-organised, non-hierarchical, such as the Linux kernel, how is 'help' organised?

It takes 6 e-mail and 2 hours to answer to Michael's initial concern and another participant's, Tom, further questions:

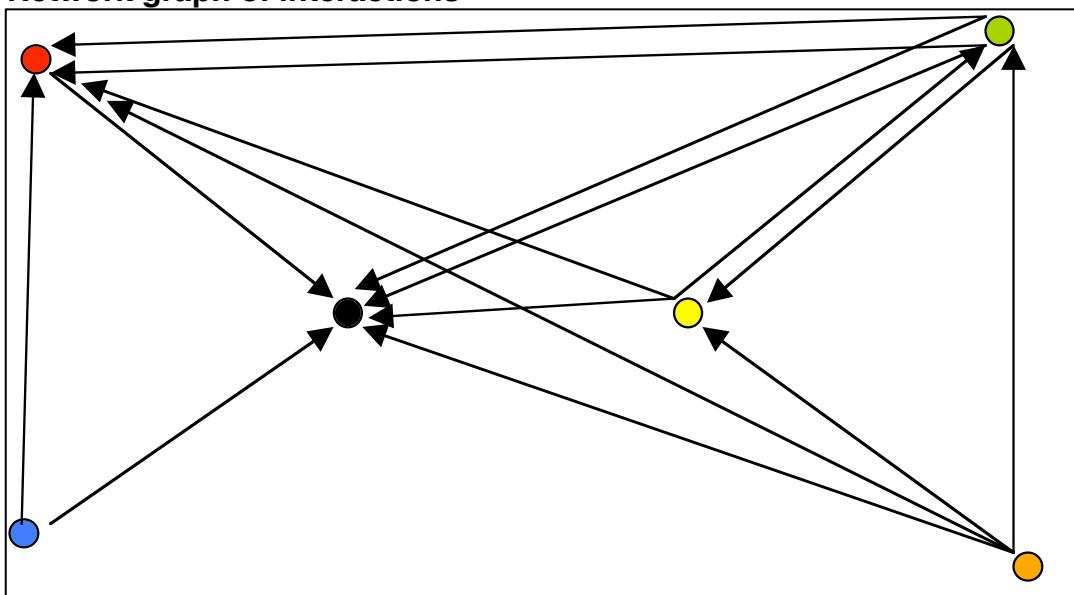
1. How do I make sure if some one is NOT working on any of the mentioned bullet points? Who coordinates? On what mailing list?
2. Do any patches for the above ToDo list have the chances of getting merged into the mainstream kernel? Who approves? I suppose the respective maintainer of the driver / subsystem getting affected?

While both questions refer to Linux kernel organization and coordination, the e-mail thread in itself resembles the answer: very quickly and very effective 3 people offered illuminating answers that show how coordination, reciprocity and efficiency work hand-in-hand with practicing the main function of mailing-lists and e-mail communication. The network graph below and the subsequent table, demonstrates the flow of information in the networked community.

Table: Exchange of E-mails

Participant	# of e-mails that participants sent	# of nodes sent by participant	# of nodes sent to participant	Total links
		(direction of link)	(direction of link)	
Michael (red)	1	1	5	6
George (blue)	1	2	0	2
Jeff (yellow)	2	5	2	7
Tom (green)	1	3	2	5
John (orange)	1	4	0	4
Mailing-list (black)	0	0	6	6

Network graph of interactions



Graph: Flow of Information during e-mail thread discussion

One ‘answer’ to the question of coordination is alluding to Raymond’s rules for ‘bazaar’ organisation⁵²:

“Release early and release often and there won’t be much duplicate work :”

The last reply by Jeff advices something that is referred to most Frequently Asked Questions (FAQ) documents for any forum/mail list⁵³

I advise lurking (following/reading) the list for at least 2 or 3 weeks and you’ll automatically understand how the “system” works.

Surowiecki, in his analysis of Linux notes that aggregation is “paradoxically important to the success of decentralization” (2004:75). Aggregation refers to both the availability of opinion diversity, as well as to the organisation of judgement in creating a coherent and functioning whole. This case- study both explains how this happens within the cooperative gift-culture of Linux, as well as practically demonstrating, through e-mails and the flow of information, how help, diversity, and aggregation are organised within the community.

Questioning Linux Hackers: the Survey

Looking through the history and the meanings associated with computer programming and developing culture, as well as their relationship to the development of GNU/Linux OS, we have noted in both the literature as well as through the participant observation of kernel development mailing lists that the ‘community of hackers’ is an exclusive group of dispersed people throughout the world, who share two important (although flexible in their definitions and somewhat ‘imagined’) characteristics:

1. The technical ability in relation to computer/software technology; and
2. The social dynamic defined by the ‘hacker ethic’.

As these people are highly educated and very conscious of both who they are and active in terms of the process of making their community and culture, and as their community exhibits a lot of similarities with other exclusive groups and communities (Alonso 1994; Kirtsoglou 2004)), they are also very critical of the people who want to ‘study’ them- i.e. make them the object of study, posing methodological and ethical problems of power and intersubjectivity (Foucault 1982; Argyrou 2002).

“How much do you understand about Linux? I’m asking because you’ve sent out this request to a specific genre of Linux users and developers. Do you intend to get a statistically significant amount of people to write you feedback? What papers and incentives do you base your work on? Since you’re working in the field of Social Anthropology, do you plan on using some of the established network theories? When is your thesis due and what’s the defending title?” (reply # 50)

52 See Appendix I.

53 See <http://www.tux.org/lkml/>.

The replies received from the questionnaire were more diverse, colourful and controversial than my highest expectations! In fact, the questionnaire itself turned out to be controversial, and my ‘innocent’-looking questions became a subject of ‘flaming’⁵⁴ debate, with some people really liking it, and some criticizing it hard:

- “Great to see someone studying the uses and options of Open & Free Software.” (reply # 28)
- “It is poor netiquette to send this to mailing lists.” (reply # 9)
- “I think this is a cool idea!” (reply # 27)
- “Have you never heard of SPAM?” (reply # 11)
- “I’m guessing you will get some flames for the multiple lists as well as serious answers, but all good fun :)” (reply # 12)

Bearing in mind the methodological problems generated by the research technique of surveys as well as the particular problems of the specific subject matter (sampling bias, the culture of developers, hacker ethic, exclusivity, see methodology), this section will present and discuss the results of the questionnaire.⁵⁵

1. When did you start using GNU/Linux OS?

The question starting the survey is a clear, straightforward question, aiming in both catching and engaging participants as well as in showing how familiar and how long the members of the lists have been engaged with Linux. Also, as the graph below shows, it is interesting to note the growth of Linux popularity over the years.

⁵⁴ Definition of ‘flames’: [at MIT, orig. from the phrase flaming asshole]: 1. vi. To post an email message intended to insult and provoke; 2. vi. To speak incessantly and/or rabidly on some relatively uninteresting subject or with a patently ridiculous attitude; 3. vt. Either of senses 1 or 2, directed with hostility at a particular person or people; 4. n. An instance of flaming. When a discussion degenerates into useless controversy, one might tell the participants “Now you’re just flaming” or “Stop all that flamage!” to try to get them to cool down (so to speak). (The Jargon File, version 4.4.7, accessed on 10/04/07, <http://catb.org/jargon/html/F/flame.html>)

⁵⁵ The key of the questionnaire can be found in the Appendix III; the cumulative results of questionnaire in Appendix IV.

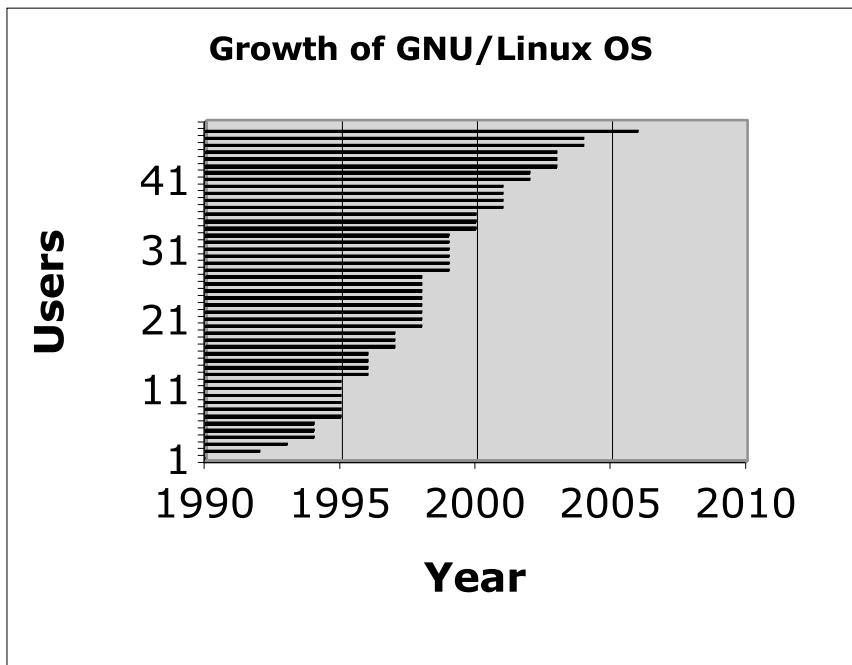


Chart: When did you start using GNU/Linux OS?

Average:	1998
Max:	2006
Min:	1990
Median:	1998

Table: Results for Question 1

As both graph and table show, the most of the members of the vger.kernel mailing lists have been using GNU/Linux OS for an average of 9 years- which, judged under ‘computer time’ where history seems to move faster, makes a lot of them ‘old-timers’.

2. What is your level of involvement? Newbie/ User/ Developer (delete as appropriate)

This question aims in revealing the degree of technical involvement of members with Linux OS as well as with the community since a developer is an active member of that community in the reciprocal ties formed in giving and receiving the ‘public gift’ of Linux collaboration (Kollock 2000). However, a useful criticism I received from this question was the following:

The results (below) serve to demonstrate that most people engaged with Linux are also active participants in its development (exhibiting a User/ Developer state), thus, avoiding a “tragedy of commons” (Rheingold 2002: 34) in that every newbie or user are potential developers or beta-testers of new software tools. This way, the problem of ‘free-riders’ is abolished, pertaining to a better understanding of the success of Linux collaborative network.

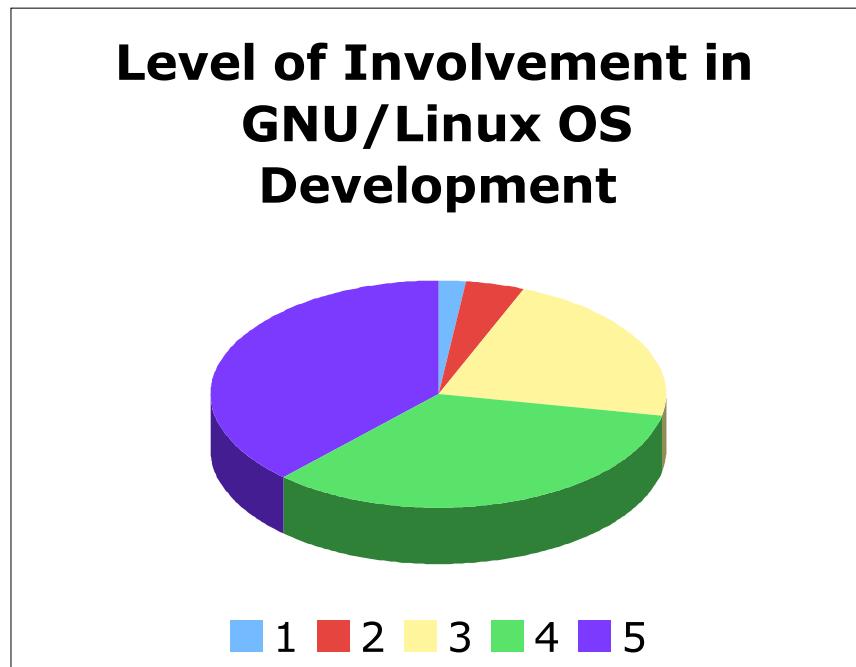


Chart: What is your level of involvement?

Graph Legend	Level of Involvement	# of Replies	%
1	Newbie/User/Developer	1	2
2	N/A	2	4
3	Developer	11	22
4	User	17	34
5	User/Developer	19	38

Table: Results for Question 2

Most people describe their level of involvement with Linux as both User and Developer, occupying the 38% of all replies. If we were to merge the three different categories which include development (Newbie/User/Developer, Developer and User/Developer) it would result in an overall of 62% of participants having done some developing work in the course of their engagement with the OS. This, hence, pertains to suggest the power of reciprocity in that in the specific community most members are inclined to “give something back” (reply # 28) and to “spread the wealth” (reply # 31) “because you get what you give” (reply # 33).

On a last note, I would like to add that the same way technology and culture merge in the techno-social realm of Linux development exists in the ‘research-imposed’ categories above. Most members of the lists, as shown by the results of question 2, are both users and developers and that is simply explained by their professional qualifications; for example most of them work in the software industry, so they are developers by job description. As one reply to the question notes:

“What does the personal self-qualification have to do with the involvement?” (reply # 50)

3. Why are you using Linux?⁵⁶

Although this question may appear very simplistic to computer programmers, it encloses an intriguing aspect to social scientists and anthropologists: what is that which makes people prefer an operating system over another? Is it purely technical, or does it also have to do with social and cultural associations? To what extent can technology be separated from the wider context of its social genesis and development, and why or why not should we separate them?

Table: An approximate categorization of the results to question 3.

A first glance at the above table shows that mentioning technical reasons has a higher frequency than the social reasons for using Linux. However, it also shows how it is somewhat artificial to divide the reasons into two categories, as both technical and social seem to overlap. For example, the most frequent reply in terms for technical preference is that of customization/flexibility. Yet, this aspect of Linux is connected with the social organization, i.e. the particular philosophical and organizational view of freedom both of speech as well as of choice. In fact, in the highly interconnected networked community of Linux most technical reasons of its success pertain to its meritocratic (Tuomi 2002) social organization and the social reasons of the existence of the community are underpinned by the technical sphere of development. Therefore, based on the results for this question, one can conclude that the social and the technical, at least within the realm of Linux development, are positioned in an ongoing feedback loop, supporting and been supported by each other. For example, most replies mentioned both technical and social reasons behind the choice of using Linux:

“It is very solid and predictable, I am a network researcher, and I have access to the source code of the operating system, so that makes my job much easier. I intrinsically like the free aspect of the software. I enjoy programming Linux because it is logically designed and cleanly implemented. I like the many choices of interfaces and programs to use for different tasks. (e-mail # 15)

“It's open source. At work I use it extensively because it is so configurable. I have a handful of keyboard shortcuts I use to navigate my windowing environment (fvwm2) and can't live without it. I practically don't need a mouse any more. Linux is very stable, it is a perfect platform for the curious Engineer on which to quickly explore new programming languages

⁵⁶ In this question, as one of the replies correctly points out, ‘Linux’ means the GNU/Linux OS and not just the Linux kernel (although the nature of the answer wouldn’t be significantly altered if supposed otherwise).

or quickly hack together a script that automates some mundane task." (e-mail # 18)

"I hate monopolies (read: Microsoft) and Linux is the obvious alternative. Also, Linux is well documented, most configuration files are in text (i.e. not binary), so it's an excellent system for fiddling about with. I don't like owning things (whether cars or computers) that have 'black boxes' I'm not allowed to open. And, if I crash the system, or just get bored, reinstalls (and upgrades) are

virtually free, I can try a different 'flavour' (distro) at almost no cost. I can keep my data in separate partitions from the operating system so a crash or a reinstall doesn't lose any data files. And Linux installs are now *easier* than installing, say, Windows 98, quite aside from being possible in any partition (...) Also a huge variety of software, I like being able to choose from 5-6 browsers, 3-4 email apps, any number of editors, etc." (e-mail # 24)

"Many reasons. A few are:

Linux is very stable (have had uptimes of more than 300 days),

It comes with perl/C/C++ - tools I need and use

It is free

It has great user support..." (e-mail # 33)

A) In order to know what goes on in my computer at any time.

B) In order to learn the mechanisms embodied in a modern operating system.

C) In order to be able to materialise my ideas without ridiculous constraints, that try to support the 'stolen'! (i.e. copyrights) (e-mail # 40)

Specifically, the results lead themselves to the following conclusions:

1. There's no specific or deterministic break between the dichotomies of technical and social reasons;
2. As a single most mentioned reason, the concept of freedom emerges as essential for both software development as well as community development.
3. Reasons of the success and maintenance of the community of Linux developers include:
 - 3.(a) technical quality
 - 3.(b) freedom in sharing knowledge
 - 3.(c) community and boundary formation through resistance to dominant ideologies/values (i.e. disliking of monopolies such as Microsoft, alternative view of 'copyright').

4. Is Linux fun? How?

"It's similar to the fun felt by an amateur car mechanic. To drive around all day in a car that you might not have built entirely from scratch but you have fixed problems on. Better still, when you fix a problem (or even contribute to a fix) in free software, everyone else (which is millions of people) often benefit from your fix too." (e-mail # 35)

"Linux enthusiasts like it because they can peek inside to know what they are using and also fix what nobody is. But think about if you have to fix every product or service that is available in the market place. How nice will life be if you have to redesign the engine of car, sew the zip of the skirt that you buy, get shoes without soles attached and so on..?" (e-mail # 21)

"If you have to ask, you wouldn't understand. Linux is fun/challenging and frustrating." (e-mail # 19)

These three answers are exemplar of the high degree of diversity and range in the answers I received to the above question. From reading about technology, computers and hackers, as well as from friends who are part of this culture, I had often come across what Linus Torvalds says about developing: "it_should_be fun" (e-mail # 16). 'Fun' is a rather strange concept within hacker/computer cultures, combining 'interesting problems' with challenge, innovation and a particular philosophical aesthetic.⁵⁷ Also, as the results reveal, Linux development 'fun' has to do a lot with the social basis of the community: GNU basis, free and open aspect of the community that allows both a high degree of individual freedom and choice as well as networked interaction and reciprocal on-going intersubjective ties. The question also aims in shedding some light in the varieties of ways of thinking of the participants, as well as being a bit 'fun' in its own.

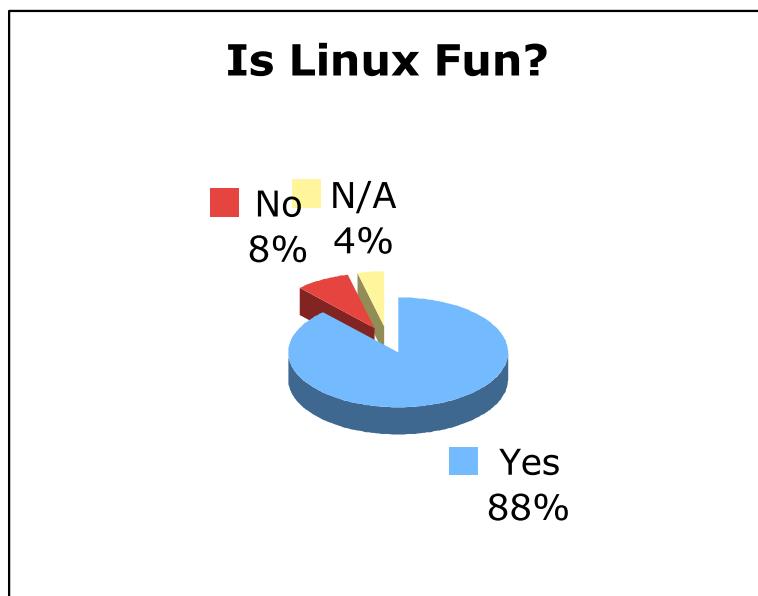


Chart: Is Linux Fun?

⁵⁷ Also see- Appendix II "How to become a hacker", and "The Jargon File, 4.4.7", accessed on 02/04/07.

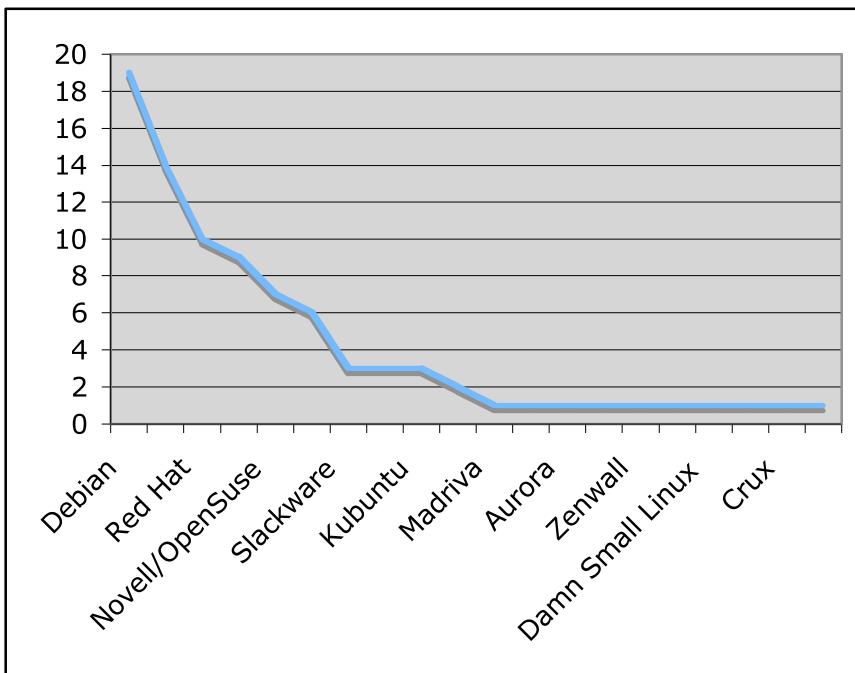
5. Which distribution of Linux do you use?

"Far too much importance is attached to distributions. Mostly I use whatever works. Sometimes we (my company) start with one distribution and add pieces from another. A distribution is like a label on a shopping bag of groceries. It's much more meaningful to ask what kind of tomato sauce you bought, not what the label on the grocery bag is." (e-mail # 23)

And this is quite right. Already Linux distributions have been described as different sets of packages of software tools. However, the main aim behind the question was to make my informants more interested, even maybe on somewhat technical grounds, in the questionnaire. The results below can be analysed in the light of intra-cultural boundary formation and differentiation, such as competitions of popularity between different Linux distributions. However, such an analysis includes some vital points:

- Most developers, as most replies to the questionnaire suggest, use more than one Linux distribution at different times;
- Most developers are competent enough as to be able to pick-and-choose between distributions and often even make their own. This can, in turn, increase the overall diversity and aggregation of the community as a whole, as well as the space of knowledge sharing (which is essential for the maintenance of the community).

The results at the time of the survey were the following:



Graph: Which Linux distribution do you use?

The above graph hints to the possible existence of a power law distribution in the preferences of Linux users and developers, based on the property of scale-free networks for preferential attachment (Barabasi 2003). Yet, an explanation for this attachment may be the notion of 'community' itself, as developers and users tend to prefer to have a distribution with wide community base and support.

6. What in your opinion constitutes a ‘good hack’?

“The word hack doesn’t really have 69 different meanings”, according to MIT hacker Phil Agre. “In fact, hack has only one meaning, an extremely subtle and profound one which defies articulation. Which connotation is implied by a given use of the word depends in similarly profound ways on the context. [...]”

Hacking might be characterized as ‘an appropriate application of ingenuity’. Whether the result is a quick-and-dirty patchwork job or a carefully crafted work of art, you have to admire the cleverness that went into it.” (The Jargon File, 4.4.7, Appendix A: The Meaning of Hack, <http://www.catb.org/jargon/html/meaning-of-hack.html>, accessed on 03/04/07)

Meaning or What’s in a Name?

The words ‘hack’ and ‘hacker’ have earned their popularity from the misusing in mass media as something connoting illegal activity and computer trespass. However, the history of the word and hackers themselves (as already shown in the Literature Review and as the results suggest) tell a very different story:

“This is related to what constitutes a "hacker". This term has been corrupted by the media (which has never been known to get anything right). A good "hack" is a computer-related success, or achievement. For example, writing a piece of code which is so elegant that it gives pleasure, or solving a problem in a “neat” fashion.” (e-mail # 25)

The original definition, which is also the definition employed within the hacker community and ethic, has nothing to do with “cracking” (illegal computer trespass), but with the skill, originality and creative attitude towards how a problem (usually but not exclusively) in computer programming is solved. This question aims in gathering and distilling the current meanings and attitudes towards the word from the participants in the survey. By asking what a ‘good hack’ is, I was hoping to obtain the definition they employ as well as their attitude about ‘hackers’ and ‘hackerdom’ (Raymond 1999).

Both questions 6 and 7 go hand-in-hand, in that they both encompass a very controversial aspect of what is called ‘hacker culture’: as Raymond reveals in his manual “How to Become a Hacker”⁵⁸, the intricate cultural complexities of this ‘subculture’ (Clifford cited in Gledhill 2000: 238) constitute and are constituted by power relations and hierarchies (such as reputation, status, and skill: Kollock 2000; Rheingold 2002), values of humbleness, and community boundary formation (exclusion/inclusion: Alonso 1994; Banks 1996).

Yet, before continuing the analysis of this rather fascinating aspect of the developer community, let’s take a look at the results.

58 See Appendix II

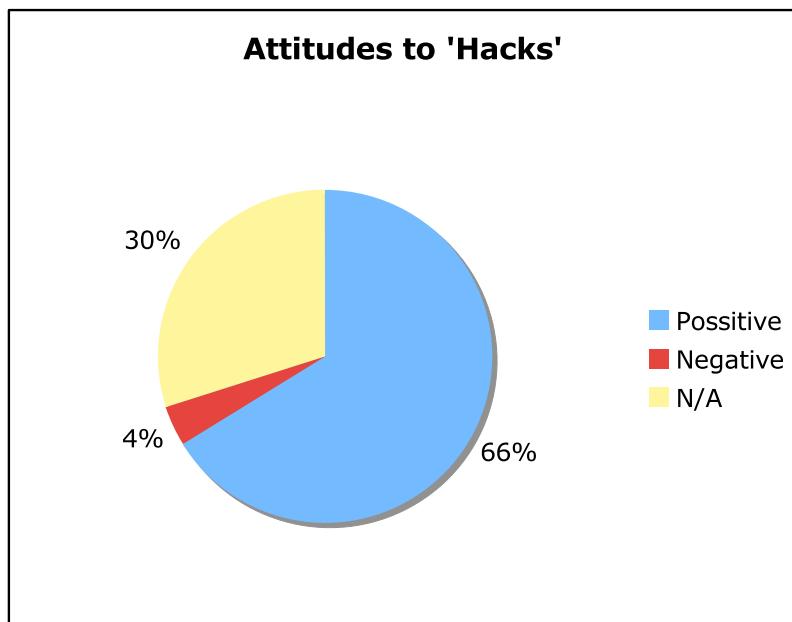


Chart: What in your opinion constitutes a good hack?

In the above graph, the results of question 6 have been categorized in three categories (Positive, Negative, N/A), based on the attitude of the reply given. As shown in the graph, the majority of answers (66%) view 'hacks' as something positive, endorsing the original definition of the word as :

"an elegant solution to an interesting or difficult problem" (e-mail # 49)

KISS or What does a Hack Entail?

"something clear something fast something smart KISS" (e-mail # 10)

KISS is one of the funny hacker acronyms standing for "Keep It Simple, Stupid"⁵⁹. Simplicity, both in design implementation as well as in communication (i.e. straight forwardness) are characteristics highly valued in the culture of networked subcultures of hackerdom (Raymond 1999; see Appendix II). This is due because of the necessity to keep technical complexity manageable (Tuomi 2002) in the arena of free/open development (a space where millions of people worldwide are working on millions of projects at the same time), but also it is safe to argue that it relates to the specifically 'cultural' trait of developers and hackers for an aversion to unnecessary complexity (either in technical or communicational styles) which may result in wasting of other people's time. And, wasting hacker's precious thinking time (Raymond 1999) is one of the worse cyber-crimes within the hacker community. Keeping that in mind, the table below summarizes some of the factors that make hacks good, and serves as an illustration of what is valued and esteemed in the culture in general.

A 'good hack' is/has...	Mentioning Frequency
• Clever/ingenious/novel/original	10
• Useful/improvement/utility/functionality	9

59 "KISS Principle: A maxim often invoked when discussing design to fend off creeping featurism and control development complexity. Possibly related to the marketroid maxim on sales presentations, "Keep It Short and Simple". (The Jargon File 4.4.7, <http://www.catb.org/jargon/html/K/KISS-Principle.html>, accessed on 03/04/07)

• Elegant	9
• Simple	8
• Unforeseen/Unobvious solution	5
• Quick/ Fast solution	5
• “It works”/“gets the job done”	4
• Clear/clean	3
• Pleasure/nicely developed	3
• Successful/ achievement	3
• Programming skills/ well-constructed	3
• Efficient	2
• “neat”	1
• “cool”	1

Table: Some reasons often mentioned in answers to question 6.

The above table serves to illustrate not only just a summary of what the replies to the questionnaire think of a ‘good hack’, but also how these certain values- such as ‘cleverness’, ‘elegance’, and ‘usefulness’- dictate, depict and distil part of the general value system within the culture of development, since by definition, hacker culture and individual hackers must endorse and embody what a ‘good hack’ is supposed to be. Hence, it is not controversial that the individualistic value of ‘cleverness’ is highly valued in a community of open meritocracy (Tuomi 2002), borne out of a technical academic sphere, in the geographic space where the winds of the ‘American Dream’ have not only shaped ‘salesmen deaths’ but also culture and history. Also, it is not controversial that with just one less frequency “usefulness” follows. “Usefulness” relates to the community aspect of development and the structural maintenance of the community which is based on exchange of ‘gifted’ ideas:

“A good hack is making a thing(program) useful for a group of people, a good hack means to find and patch a bug, etc. That sort of thing.” (e-mail # 14)

“Fantastic scripting that will do something substantial for thousands of users in a pinch.” (e-mail # 31)

“If it helps people and can be built upon.” (e-mail # 50)

C. The Sound of Silence

Up to here, we have looked at two important outcomes of the results:

- how most responses correspond to a positive attitude of hack meaning; and
- how some particular responses to the factors that constitute a ‘good hack’ are embodiments of the value-judgments and ethic of the culture as a whole (by both definition as well as empirical data ;-).

However, these findings apply only to the 66% of the total answers received. And, while the majority of the answers offered positive remarks about what a good hack is, a disproportional high number of replies offered no comment at all. In this section, we will analyse the category of N/A, which correspond to the 30% of responses.

The ‘N/A’ category includes several types of replies, such as:

- “N/A” (e-mail # 11)
- “Don’t Know” (e-mail # 12)
- “Not sure how to answer this” (e-mail # 30)
- “No comment” (e-mail # 36)

as well as:

“I refuse to answer as the popular social connotations promulgated by the mass media have caused the term "hacker" and "hack" to indicate possible illegal activity.” (e-mail # 19)

Categorized by their ‘sound of silence’ in their denial to answer, this category deserves special attention as it introduces two intricately related cultural ‘conventions’ and mechanisms employed within the widely networked and free/open-based community culture of developing:

- the active dislike/apathy of ‘popular’ notions of what a ‘hack’ and a ‘hacker’ are, as well as the misuse of the term to mean ‘cracker’; and
- a marker of group/community boundaries.

As a lot has been written about the difference between a hacker and a cracker, mostly by members of the culture themselves⁶⁰, it does not come as a surprise that people don’t want to answer this question on the basis of fearing misuse and misunderstanding, or maybe even considered it a waste of time: “Unsurprisingly, hackers also tend towards self-absorption, intellectual arrogance, and impatience with people and tasks perceived to be wasting their time.”⁶¹

However, such reasoning resembles the mechanisms of inclusion/exclusion in the ‘traditional’ anthropological territory of studying ethnic group formation (Alonso 1994; Banks 1996).

Specifically, by choosing not to answer or by refusing to answer, 30% of the responses reaffirmed their intra-group solidarity by viewing the social anthropologist (to-be) as an outsider to their cultural and linguistic conventions. The outsider is marked on the basis of her/his knowledge of the culture’s characteristics- one, very significant, being the ‘right’ or inclusive definition of the word. Therefore, as a group whose solidarity and existence depends on a sharing of both technical and cultural forms of knowledge, it had developed semi-conscious mechanisms for identifying possible ‘intruders’ or ‘free riders’, and tends to exclude them.

Furthermore, a named outsider (such as a Social Anthropology student) which claims to be able to ‘study’ them, posses challenges in the form of objectifying power relations. And so, the sound of silence becomes a loud signal of exclusion and boundary formation.

7. Would you describe yourself as a ‘hacker’?

In personal conversations with technical people, I call myself a hacker. But when I'm talking to journalists I just say "programmer" or something like that.

Linus Torvalds

Having explained some of the seeming complexities and controversies of the meaning of the term ‘hack’ and how it may serve as a mechanism of boundary formation, neither Torvalds’

60 See Raymond 1999, The Jargon File 4.4.7

61 The Jargon File 4.4.7, Appendix X A Portrait of J. Random Hacker, <http://www.catb.org/jargon/html/weaknesses.html>, accessed on 03/04/07

quote⁶² nor the results of this question should come as a surprise. Question 7 asks informants whether they would describe themselves as hackers. The results are shown in the graph below.

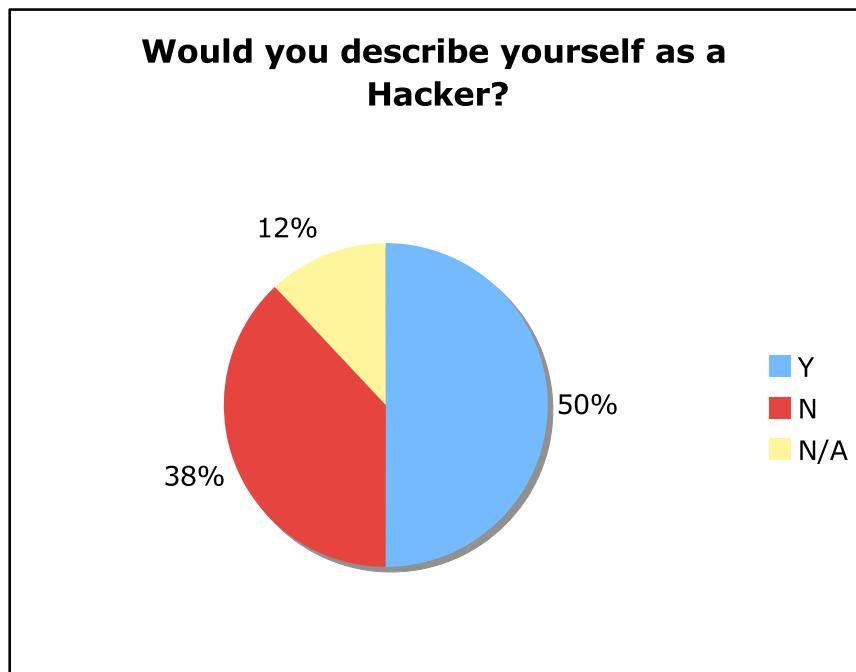


Chart: Would you describe yourself as a 'Hacker'?

Half of the respondents⁶³ replied positively, showing, as expected, the high degree of technical specialization within the chosen mailing lists, as well as challenging popular stereotypes by ascribing to the 'original' meaning of the word:

- "In the original sense, I design at component level and program in assembly (or directly in machine code in some cases) - my first computer was home-made with 256 BYTES of ram, & I managed to write games in that..." (e-mail # 12)
- "In one word? With it's original means? Yes." (e-mail # 14)
- "Yes, in the classical sense of the term which does not involve computer trespass." (e-mail # 22)
- "Maybe in the "old" sense, I like to play with things. I do though dislike greatly people who trespass onto other people's property or computer without their consent. In the "modern" usage of the word I'd never describe myself as a hacker." (e-mail # 39)

62 Used as signature in an e-mail sent to linux-net@vger.kernel.org and linux-kernel@vger.kernel.org mailing lists, as well as to Linus Torvalds, on December 9th 2006.

63 Bearing in mind that as the questions were open-ended I have, to the best of my ability, tried to estimate to which category a reply should belong.

It is worthy of note that 36% of the positive answers (9/25 replies) mention specifically that they are hackers not crackers:

Hmm, I 'hack' away at stuff so I guess so. however I do not attempt to access machines I haven't got responsibly for managing (e-mail # 9)
Yes, but not a cracker.. (e-mail # 16)

The findings point directly to the concept of a dynamic and performative (Butler 1999; Kirtsoglou 2004) identity construction process, by which the self-ascribed hackers as well as the community of Linux development as a whole, actively and consciously mobilize resources such as the history and original meaning of the term, as legitimizing forces towards their identity (Alonso 1994). Also, identity construction is linked with a process of defiance- a resistance movement against the “popular social connotations promulgated by the mass media” (e-mail # 19). Taking into account the role of resistance in identity and boundary construction, as well as the connection of hacker culture with ‘nerds’, ‘geeks’ and ‘social outcasts’ in general⁶⁴, as well as its counterrevolutionary history in the development of FSF and free sharing model of development, it would be useful to estimate that an identity of defiance and resistance to popular or hegemonic discourses and value systems is both actively constructed as well as emergent of the community (i.e. resistance against authoritative monopolies like Microsoft).

As the identity construction is always in the making- especially in this anthropologically/ historically recent culture (which is of course an ancient one on the time-scale of computer/ technology development), there have partially emerged and partially been constructed several different identification markers: signs used to define and separate the members who belong to the community from the members who don't (Bourdieu 1992; Kollock 2000). As Kollock notes, Linux being an economy of on-line cooperation, resembles the “impossible” public good because of its constant need of balancing the social dilemmas of personal gain versus collective (group/community) good (Kollock 2000: 230). The risk of free-ride temptation (*Ibid.*, Rheingold 2002), for example, is the most obvious one in undermining the purpose and development of the group. One way to reduce this cost is by maintaining reciprocal ties (see section below), while another mechanism is that of identity markers and boundary formation (Alonso 1994; Rapport 2006), in order not only to increase group cohesion and solidarity (Komter 2005), but also in order to be able to identify the ‘intruder’ and the ‘potential free-rider’. To this effect, I think, the hacker community has developed strong symbolic inclusion/exclusion markers and social conventions, such as the reputation/status system (Rheingold 2002), taboos regarding hacker identity ascription (Raymond 1999: 242), and a strong ‘insider’ knowledge of their history and ‘original’ meanings of words (such as ‘hacker’).

Such a marker is the ‘taboo’ of the title ‘hacker’. The Linux development community, which corresponds to the hacker community (see literature review- history), both because of its history as well as the meritocratic organization model of development, value attitudes which aid in the development of the ‘public gift’ of Linux and the free sharing networked community. Due to these reasons, the ascription ‘hacker’ acts as a status signifier, given by other people to those who have actively contributed to the ‘common good’:

64 See Raymond 1999. “The Hacker/Nerd Connection” in *The Cathedral and the Bazaar*. Cambridge: O'Reilly, pp: 245 : “Being a social outcast helps you stay concentrated on the really important things, like thinking and hacking”. Also, see *The Jargon File 4.4.7, Appendix X A Portrait of J. Random Hacker*, <http://www.catb.org/jargon/html/appendixb.html>, accessed on 05/04/07.

“...when you play the hacker game, you learn to keep score primarily by what other hackers think of your skill (this is why you aren’t really a hacker until other hackers consistently call you one). This fact is obscured by the image of hacking as solitary work; also by a hacker-cultural taboo (...) against admitting that ego or external validation are involved in one’s motivation at all.” (Raymond 1999: 242)

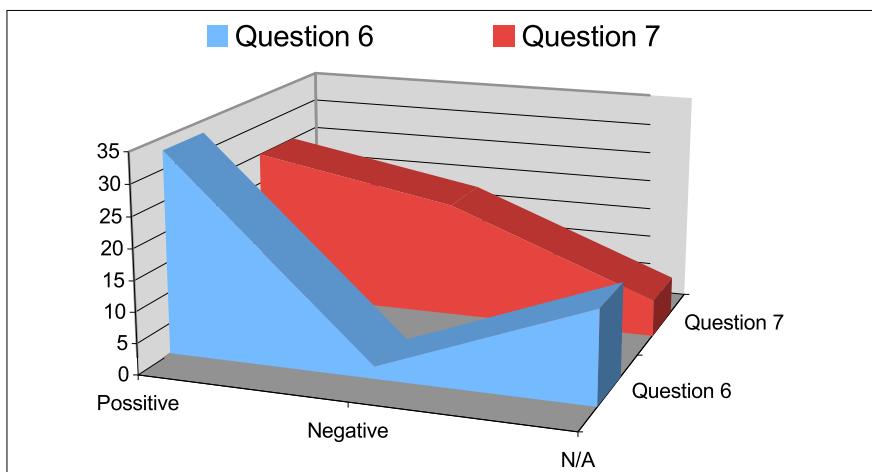
The following replies are examples of this, almost a rite of passage in the classical sense (Van Gennep 1960) and community boundary characteristic of the culture:

“...'hacker' is a title that other people may bestow on one but one shouldn't claim for oneself.” (e-mail # 24)

“The ethics of computer-hacking dictate that only other hackers can describe you as such. They are saying in effect that your ability suggests that "you are one of us".” (e-mail # 25)

“No comment, hacker has more senses...”(e-mail # 3)

Therefore, taking into account the ‘ethics of computer hacking’ and the mechanisms with which the community is built, it becomes clear that the controversial replies received for question 7 aren’t as controversial as they appear to be, but become understandable when explained under the light of the cultural norms and conventions, when explained by the power of reputation, prestige, resistance, and community boundary formation. Using such prism, it is easy to understand the apparent contradiction and polarization between question 6 and 7: namely the 16% drop in positive replies, the 32% increase in negative, and the 18% decrease of N/A:



Graph: Comparison between replies to Questions Six and Seven

As it becomes obvious from the analysis and the graphical representation above, although most of the participants agree that ‘hacking’ is a very positive activity, characteristic of ingenuity and sociality, due to cultural norms such as diversion of self-ascription, boundary marking, and skill involved, a smaller number of people agreed to tell the outside ethnographer their self or community-given identity.

8. Which super-hero (apart from Tux) do you think would represent Linux best?

In recent article, Ratto (2005) argues that the discrepancies of the geographically and spatially dispersed model of Linux development are bridged by the psychological relationship and the construction of symbolism with the Linux mascot, Tux:

"inhabitants of trans-local spaces such as Linux developers may manage some of the contradictions of local and global, compression and expansion, through a particular relationship to Linux itself, a relationship that allows them to maintain a productive tension between global affiliation and local situatedness and to construct an emotional and empathetic sensitivity to their shared creation" (Ratto 2005: 828)

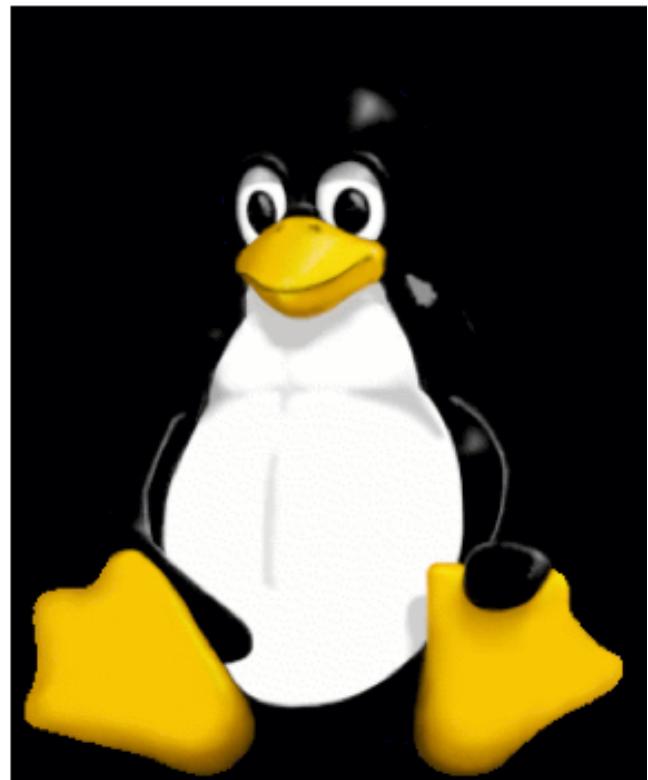
Although this question is not directly testing the psychological bond and the symbolic significance of the representational mascot⁶⁵, it does ask participants to visualize and metaphorically search for similarities between Linux and a super-hero. As the use of 'imagework' (Edgar 2004) can be particularly helpful in social science, by using a creative, imaginative and hopefully enjoyable question, I was hoping to discover something more about the individual relation and/or motivations of a developer to Linux, as well as how s/he visualizes the relationship between Linux and the wider social/technological context. Also, I was hoping to test the popular stereotype that wants computer enthusiasts, developers and hackers to also be science-fiction and comic-book fans (Raymond 1999: 245-246). The results challenge such stereotypes, as the majority of replies (50%) not only doubted the validity of such representation, but many did not know any superheroes:

"I don't believe superheroes exist.

Linux does..."(e-mail # 41)

"Linux is not a super-hero, in my opinion Linux is like an honest and good person in real life."(e-mail # 42)

"Hey, super-heroes are for Manichean fantasies of doing the good and being the strongest, does that really look like Linux?"(e-mail # 47)



Picture: The original Tux picture by Larry Ewing

65 Image by Larry Ewing (lewing@isc.tamu.edu), <http://www.isc.tamu.edu/~lewing/linux/>, accessed on 17/04/07.

"Linux doesn't need to be represented by a super-hero."(e-mail # 48)

"?? Tux is a super-hero? Cannot answer this question since I honestly do not know the world of super-heroes. It would be the submissive underdog, playing catch-up in the real world, but being a God-figure for its believers and followers." (e-mail # 50)

From the 50 replies I received, only 13 (26%) answered with a superhero, and 12 (24%) of them named a Linux-related mascot or developer, the results of which are shown in the table below:

Super-Heroes	Mentioning Frequency	Linux Developers and Mascots	Mentioning Frequency
Batman	2	Linus Torvalds	4
Superman	2	Tux	3
Mystique (X-Men)	1	Richard M. Stallman (RMS)	2
Charles Xavier (Professor X, X-Men)	1	Greg Kroah-Hartman (KH)	2
Arsene Lupin	1	Alan Cox	2
Mandrake	1	GNU	2
MacGyver	1	Alan Stern	1
Wonder Woman	1		
Neo (Matrix)	1		
Captain Freedom	1		
Captain Planet	1		

Table: Results of question 8 showing answers mentioning super-heroes and people/mascots related to Linux.

A qualitative analysis of the results shows that most replies (including those mentioning super-heroes) reflect a symbolic language of describing Linux as "underdog", "resistant" and "guerrilla" OS. This connects with the anthropological literature on boundary creation and group formation in terms of power and resistance to power (Alonso 1994; Scott 1985; Lukes 1974). It also links nicely with the next question...

9. Describe Microsoft OS in one word

In the anthropology of group and boundary formation, such as the creation of ethnic groups, nations and minorities (references), one of the mechanisms of such differentiation, and indeed one of the main causes can be described in terms of resistance to the dominant possibly exploitative ideology and system. In the realm of Linux development, resistance to copyright material which is thought it should be shared, helps in the creation of boundaries, and rhetoric against a common enemy- hence as a factor of unification, shared caused and mobilization. Such a relationship is antagonistic, at least in its rhetorical expression, and in this case is exemplified by the views and subsequent relationships of Linux developers and users with Microsoft Corporation and Windows OS. Microsoft's OS and software products are directly opposite to the ones produced under GNU GPL, both in their developmental standards (hierarchical vs. meritocratic) as well as in the treatment of the code (closed vs. open). One is a

business, the other a voluntary project. Yet, as Linux is a growing underdog, not only possess a serious threat to Microsoft's products due to the higher quality software it produces, but also because of the way its community is using the very real power of "weapons of the weak" (Scott 1985).

This question reveals the 'collective consciousness' of Linux users and developments in their shared negative perceptions of the Other, with 92% of replies describing Microsoft's Windows OS in a very negatively light, and 4% abstaining and more friendly views of it...

- CR@P
- Business
- Monopoly
- Closed
- Bloated
- Junk
- Expensive
- Clunky
- Jail/ Lock-in
- Shrewd
- Obfuscated
- Annoying
- Mediocre
- Flakey
- Imperialist
- FOOBAR
- Conservationism
- Kludge
- Erratic
- Gangling
- Inflexible
- Trash
- Buggy

10. How do you view the recent patent agreement between Microsoft and Novell?

On a similar vain with question 9, this question also asks the views regarding the patent agreement between Microsoft and Novell, raising issues of resistance, but also focusing more specifically of the impact of such agreement. The majority of the replies (48%) viewed the agreement as negative and possibly threatening to the community:

An attempt to damage the GPL and the FSF.(e-mail # 9)

Badly, should never have occurred in that form. A Novell/Microsoft agreement might have been a good thing actually, but the agreements that were made are bad for both parties, and the FLOSSMark community.(e-mail # 20)

On the Microsoft side, it looks to me like an attempt to convince the investors that they have OSS under control, without really doing anything drastic about it. On the Novell side, it looks they decided

that the up-front investment would help them enough to justify the long-term costs, and made the decision on solely financial grounds without giving much consideration to the community reaction.(e-mail # 22)

With disdain. I used to use SuSE Linux, but would never ever consider them again following this betrayal.(e-mail # 25)

Divide and conquer by M\$, \$\$\$\$ signs in the eyes of Novell. (e-mail # 39)

What is enlightening and particular to this question is the underlying assumed concept of community (Rapport 2006) that unites the threads of the replies, against a carefully constructed threat. Thus, by the erection of boundaries (Banks 1996) and through the construction and rhetoric against the Other OS, collective identity is formed and community's solidarity is reaffirmed (Komter 2005).

The summary of the results are provided in the chart below:

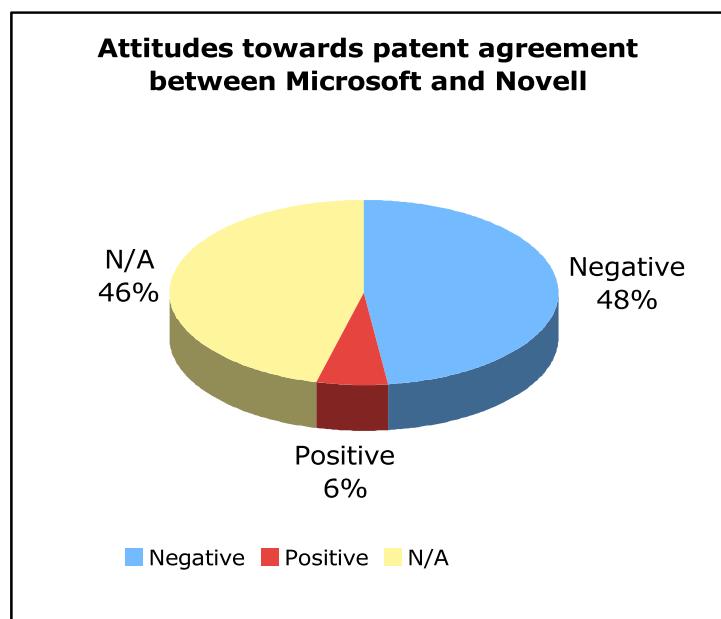


Chart: How do you view the patent agreement between Microsoft and Novell?

Characteristic of the hacker ethic of detesting getting involved in 'flame' and 'holy' wars, the greater minority of 46%, preferred not taking positions:

I suppose fed up with everyone making a big deal of it. I got bored of following the news stories.(e-mail # 32)

Two businesses looking after their own interests and nobody else's. Nothing new or surprising there.(e-mail # 35)

I don't care enough to know all the details. As long as I have a domain name and a public IP, I can do whatever I want, and I utterly laugh at people pretending to enforce any patent on me. If Novell think it is good for them to enter this game, well, it is their problem.(e-mail # 47)

11. GNU GPL, copyleft and freedom of speech: good, bad or irrelevant?

“Oxygen- good, bad or irrelevant?” (e-mail # 4)

With 90% of replies responding not only positively to the question, but also underlying the fundamental importance of the principle of freedom of speech and exchange of information, this question reveals and confirms the strong ideological prerogatives for the existence of both the technical as well as the social community of software development. The results of the replies are indicative:

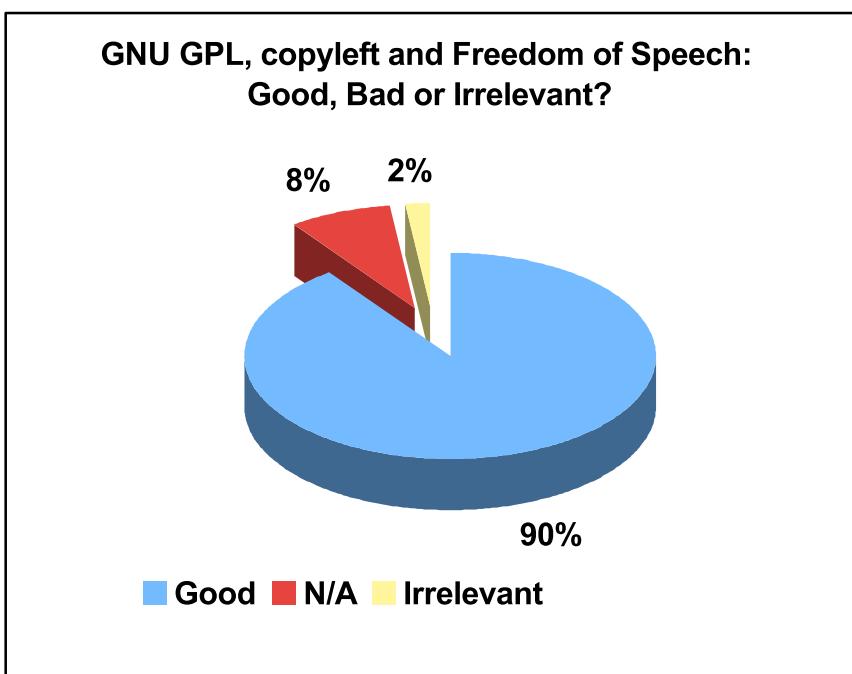
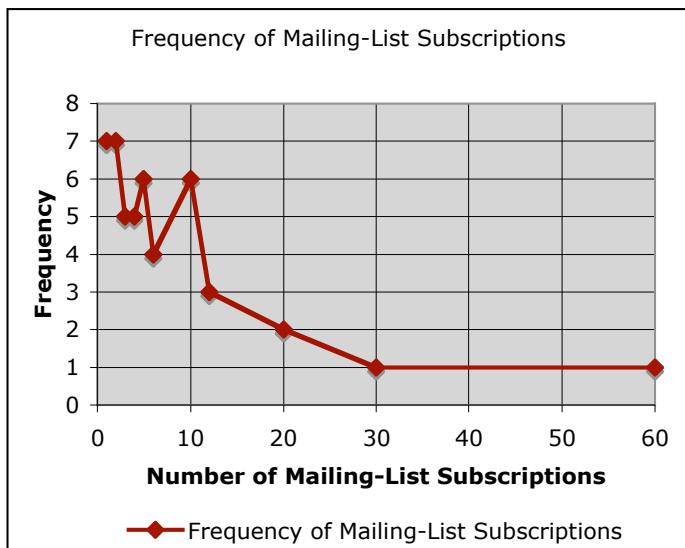


Chart: *GNU GPL, copyleft and freedom of speech: good, bad or irrelevant*

12. How many Linux mailing lists are you a member for?

This question acts as a quantitative precursor to the analysis of gift-giving within the free-sharing community of development. The results are displayed below, suggesting a strongly networked community, representing a small world phenomenon not in the actual number/degrees of separation between individuals (Watts 2003: 88) but by the potential of doing so. However, as more information would be needed in order to investigate the network properties of the community, this question deals strictly with the results derived. As shown, the average number of mailing lists a member is subscribed to is 7, while the most frequent mentioned numbers of mailing-lists subscriptions range between 1 and 2. Furthermore, a few individuals belong to a disproportional high number of mailing lists, like 30 or 60, which, maybe suggest the existence of a scale-free network with properties of preferential attachment and rich-get-richer phenomena (Barabasi & Bonabeau 2003).



Graph: Frequency of Mailing-List Subscriptions

Also, the graph hints that there might be a power law distribution between individual subscriptions and frequency of these, yet this would only be possible if the survey had a higher number of participants (to 'smooth' the graph curve). However, without having any evidence of who is connected to whom and what kind of relationships/links exist between them, this remains a hypothesis. Also, the tables below represent 47 (94%) of e-mails, as 2 (4%) where N/A, while I could not offer an estimation for one of the e-mails:

"I have lost track of my thunderbird filters :-P" (e-mail # 31)⁶⁶

Table: Statistics of Number of mailing list subscriptions

13. Would you reply to a question sent through Linux mailing lists and why?

"Evidently yes :-)." (e-mail # 32)

This question aims to understand the motivations behind the 'gift-giving' of time, thought and creativity resources- something that practically occurs when replying to a question sent through Linux kernel mailing lists. As shown in the chart below, 88% of the participants answered positively, 8% negatively and 4% did not provide an answer.

⁶⁶ Some of the answers received were estimations or I had to estimate between different figures a person gave me (this is because some of the participants don't keep track of how many mailing-lists subscriptions they have). When I had to estimate given on figures provided I chose the average. Therefore, the results reflect indicative estimations from both the participants' part as well as the researcher's.

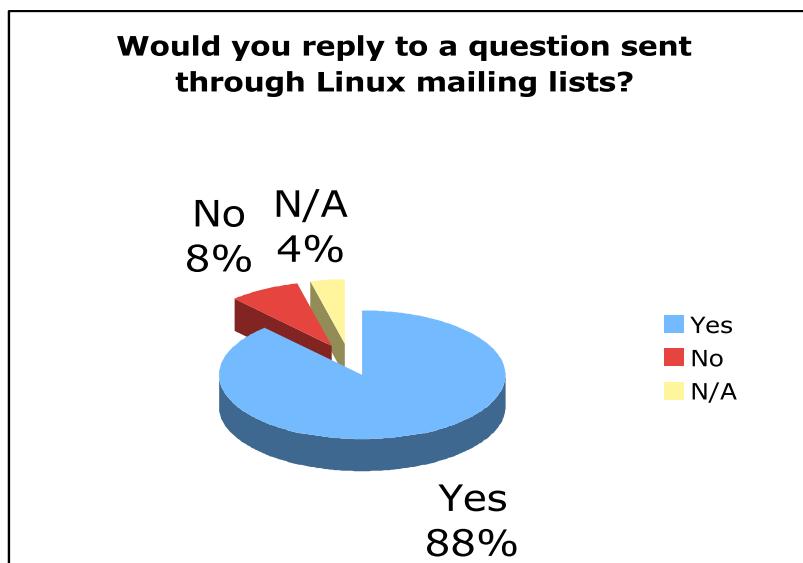


Chart: Replies to question 13

The chart shows that the majority of the survey participants would reciprocate by replying to an e-mail sent through the mailing lists. Of course, this only to be expected as they are already subscribed to the mailing lists (whose function is just that!), and as they replied to my questionnaire sent through it! Yet, the question reveals two fascinating findings about how this networked community works:

- Motivations and importance of gifts and reciprocity; and
- The politics of who gives and who receives.

The ties that bind: Gifts and Reciprocity

The capabilities I have in this area are largely down to others having given me the fruits of their labours. I see it as only proper to do likewise.(e-mail # 4)

The exploration of the anthropological theory on gift-giving suggests that gift-exchange is a basic function of sociality as it establishes links and reciprocal relations between individuals and groups (Mauss 1974). The creation of the obligation to reciprocate ‘locks’ “moral persons” (*Ibid.*) in ongoing interdependent relations, constituting interactive, interdependent and cooperative entities: such an entity is the network of Linux community. The existence of networks (Watts 2004; Barabasi 2003) is based on the existence of these ties that bind.

Furthermore, gifts are not only fundamental to the formation of networks and communities but also show that the exchange of gifts have social benefits such as gaining membership in the community, gaining reputation, prestige and power in the meritocratic community of Linux. Kollock’s (2000) suggested five principal motivations for contributing in Linux Community (see Literature Review).

Based on the replies, the motivations for contributing have been separated into four categories:

1	Ideological Reasons	<ul style="list-style-type: none"> • Greater good • Sharing knowledge • GNU GPL • Freedom/Open • Commons
2	Practical Reasons	<ul style="list-style-type: none"> • Efficiency in problem solving • Faster results • Aggregation of opinions
3	Reciprocity	<ul style="list-style-type: none"> • Obligation to give something back • Giving something and expecting reciprocity in the future • Collective responsibility
4	Personal Benefits	<ul style="list-style-type: none"> • Personal satisfaction/ fulfillment • Usefulness/ being helpful • Improvement of one's understanding/ experience by solving others' problems • Pleasure/ fun

Table: Motivations for answering a question sent through Linux mailing Lists

The above reasons correspond with Kollock's list of motivations for contribution to the "impossible public good" (Kollock 2000: 230). The only difference is that none of my informants listed 'reputation' as one of their motivations. On one hand, this is only to be expected, as it is a "hacker-cultural taboo [...] against admitting that ego or external validation are involved in one's motivation at all." (Raymond 1999: 242) On the other, reciprocity has been shown to have a direct relationship with reputation systems (Rheingold 2002; Kollock 2000; Scott 1985). Nevertheless, the replies suggest that all of the above motivations are interconnected and equally important. Also, the high proportion of answers listing ideological causes was surprising as computer-programs prefer to acquire a low-profile, actively a-politicized profile, sceptical of 'ideologies'⁶⁷. Moreover, this suggests an attachment to the community ideology- something that as Kollock informs us minimizes the social dilemmas of reciprocal interaction (Kollock 2000: 229).

Below are some indicative replies, demonstrating that the 'ties that bind' have social reasons, and when Linux is viewed as a public gift, its community members are collectively responsible for its maintenance.

Yes, and the same goes for most mailing lists, as I've had good help from them myself in the past, am very grateful, and would like to spread the love.(e-mail # 7)

I have and I do occasionally when I think I have an insight to offer. I do so because it makes me feel like I am contributing to a greater good. (e-mail # 17)

67 The Jargon File 4.4.7, <http://www.catb.org/jargon/html/politics.html>, April 11, 2007.

Yes, as I have in turn been helped the same way. Just giving something back. (e-mail # 28)

Yes, because linux is collaborative. You get better help from others if you are helpful. It's fun to solve problems and I usually benefit from taking part as it increases my own experience. (e-mail # 35)

With regards to the nature of progress the open source community made in the years passed, it should be obvious that it is based mainly on cooperation efforts (which are mostly not directly related to personal gain). Replies to questions one is capable of answering lowers the load on those on the list who could probably answer all questions but would get to nothing productive if they did. (e-mail # 36)

Well yes, if I think I know a relevant answer I would reply. Why? After all those others helped me, I could help someone else.

(e-mail # 38)

Obviously I just did. Because it is important to be open, free and transparent and to the good of the commons. That's what the GPL is about. (e-mail # 39)

Yes. If we share our expertise, many people can gain knowledge and help support other users in the community. We're all out here with each other in "GNU land" so I feel we should lend a hand where we can. Lists are one venue to do just that. (e-mail # 44)

The Politics of Gift Exchange: Entitled to Give?

While the statistical results of question 13 demonstrate that the majority of people would reply to a question sent through the mailing lists, a closer reading shows that although most people would like to reply, most of them, most often, do not.

Linux gurus are humans too, and are not always kind and patient with newbies: last time I posted something clever I just got ignored. As a simple user having limited knowledge, I am not sure my answers would be welcome.(e-mail # 47)

The above quote confirms something that was noted in the literature review section: that the act of gift-giving creates social obligations to recipients while simultaneously attaining a certain degree of status and prestige to the giver, hence underlying the 'reciprocal appeal of power relations' (Foucault 1982: 794; Rheingold 2002; Kollock 2000; Scott 1985) inherent in the exchange. Also, taking into account that the Linux community is based both on the sociality of cooperation and shared culture between members as well as in their technical abilities and competence, prestige, authority and thus, power of an individual member is counted against

both social and technical spheres. Due to the fact that “attitude is no substitute for competence” (Raymond 1999: 236), it is not a surprise that firm boundaries exist in showing and accounting the social hierarchy of Linux-hacker meritocracy (Tuomi 2002). This organizational hierarchy, however, is constantly negotiated and changed in order to reflect adequately the social values of the culture, such as the ‘ethical hacker attitude’ as well as the constant struggle and challenge of proving one’s intellectual skills of technical problem solving that grant a membership in the community. Thus, the politics of gift exchange and the hacker culture show how although reciprocity is an internal part of the core of Linux existence, the power of the gift is that it symbolized the authority of the person to provide a correct expert answer, not only in the technical problem solving aspect of the question, but with the prestige gained through a complex reputational system of social values and power (Kollock 2000; Rheingold 2002).

Hence, in a ‘second’ reading of the results, the ‘if’s’ were counted: ‘if’s’ that reflect the desire of users to answer on one hand, but also culturally constructed preconditions for doing so. Thus, out of the 88% of the people who replied positively, half of them, 44%, included ‘if’s’ in their replies pertaining to nature of question asked and their personal abilities of replying with an expert answer:

If it's something I'm an absolute expert on, yes. Otherwise, there's probably someone else who is an absolute expert, and I'll let them respond.
(e-mail # 22)

If I felt able to offer expert help. I have rarely done so. I have however participated in discussions, and asked questions myself.(e-mail # 25)
Yes, according to my ability.(e-mail # 46)

Conclusion

Like a modern-day *kula* ring, the Linux code resembles a gift: one cannot keep it for one’s self, it contains obligations and a promise of future reciprocity. It is a collective gift of self-ascribed Hackers that come from different geographic places and meet in lines of code, socializing by exchanging ideas about the code and about themselves. This work shows by what means the computer hackers of Linux, abiding to the original definition of the word, actively constitute their community using discourse: language (Raymond 1999), e-mail, internal meritocratic hierarchies (Tuomi 2002) based on technical ability and ethics of the group (Raymond 1999), boundaries of exclusion and inclusion (Banks 1996; Barth 1969; Rapport 2006).

In this study I reviewed the background history and development of Linux, developing the theoretical threads upon which the study is based. Issues of analytical and philosophical approaches to technology and science serve to detect and eliminate different kinds of determinisms in studies of techno-social cultures. A discussion on debates of ‘community’ (Rapport 2006) and ‘reciprocity’ (Kollock 2000) helped reinforce each other in the formation of social ties as well as in boundaries (Banks 1996, Barth 1969), and hence in the successful solidarity of a community (Komter 2005). Such a discussion not only demonstrated aspects of what kind a community Linux is, but offered valuable insights in understanding the motivations behind voluntary cooperation. The section of Findings and Results searched for the empirical application of the theories previously explored, uncovering colourful instances of the active ‘making’ of Linux-hacker culture and community, while at the same time analyzing the mechanisms that make the community successful in terms of the embedded sociality of ‘gift’ exchange, a system of reputation and statues, and the organization of cooperation.

The mechanisms for the construction of the Linux community are parts of a social exchange, whether of e-mails, patches, or new kernel distributions, obey the rules of ‘gift exchange’ (Mauss 1967) and which pertain to the vitality of sociality for the production of technological advancements:

1. Literature as well as the results of the ethnographic study show that Linux resembles an “area of hybridisation” (Lyon 2004: 13-15), in which social and technical merge, sociality is translated into technical gifts, which establish on-going interdependent reciprocity ties, which means that changes are always negotiated, always happening, as part of the gift exchange.
2. Linux is a software product- but its functions as well as history make it something more than a product in that its consumers are its potential developers (which means that gift-receivers are not only empowered but promoted to become future gift-givers). This both establishes and maintains complex and interdependent social ties between the users, developers and technological product. Also, this makes possible the successful exploitation of the network qualities of the system, such as the aggregation of expert knowledge to solve problems quickly and efficiently (Surowiecki 2004).
3. In terms of culture, Linux promotes and shapes ideas, ideals and identities of the programmer through the conscious discursive construction of the hacker culture (Raymond 1999, Torvalds & Diamond 2001). The results aid in showing how culture and hierarchies are both constructed out as well as emergent properties of everyday practices (Bourdieu 1992, Lafeyvre 2000, Deleuze 1991). The Linux community of hackers is an exemplar of the active and conscious constructive practice of culture. Also, The social ties of reciprocity establish and promote an intra-group competition in terms of reputation and prestige (Rheingold 2002), while simultaneously building boundary markers of exclusion and inclusion (Alonso 1994; Banks 1996), establishing a unique sense of solidarity (Komter 2005).

With an emergent as well as constructed cultural tradition, the ‘gift of the code’ seizes to be “impossible” (Kollock 2000), not only does it become possible, but it also becomes practical, efficient, and through membership to the culture, it becomes fulfilling. In this sense, Linux developers and users consume, create and exchange an as much technical as cultural discursive construction: codes of sociality...

References

- Abrahams, R. 1990. “Chaos and Kachin”. *Anthropology Today*. Vol 6, No. 3, pp. 15-17.
- Alonso, A. M. 1994. “The Politics of Space, Time and Substance: State Formation, Nationalism and Ethnicity”. *Annual Review of Anthropology*, Vol. 23. Pp. 379-405.
- Amit V & Rapport N. 2002. *The Trouble with Community: Anthropological Reflections on Movement, Identity and Collectivity*. London: Pluto.
- Anderson, B. 2003. *Imagined Communities: Reflections on the Origin and Spread of Nationalism*. London: Verso.
- Argyrou, V. 2002. *Anthropology and the Will to Meaning: A Postcolonial Critique*. London: Pluto Press.
- Banks, M. 1996. *Ethnicity: Anthropological Constructions*. London: Routledge
- Barabasi, AL & Bonabeau E. 2003. “Scale-Free Networks”. *Scientific American*. May 2003: 60-69.
- Barabasi, Albert- Laszlo. 2003. *Linked: How everything is connected to everything else and what it means for business, science and everyday life*. New York: PLUME

- Barth, F. 1969. *Ethnic Groups and Boundaries: the social organisation of cultural difference*. London: Allen & Unwin.
- Bentley AR & Maschner HDG. 2003. *Complex Systems and Archaeological Research: Empirical and theoretical applications*. University of Utah Press.
- Boissevain, J. 1979. "Network Analysis: A Reappraisal". *Current Anthropology*. Vol. 20, No. 2. (June), pp. 392-394.
- Bonabeau E., Dorigo M., & Theraulaz G. 1999. *Swarm Intelligence: from natural to artificial systems*. Oxford: Oxford University Press.
- Bourdieu, P. 1992. *Language and Symbolic Power*. Cornwall: Polity Press.
- Brown, John Seely & Duguid Paul. 2002. *The Social Life of Information*. Boston: Harvard Business School Press.
- Butler, J. 1999. *Gender Trouble: Feminism and the Subversion of Identity*. London: Routledge.
- Carter, DM. 2004. "New Locations: the Virtual City". *Anthropology Matters Journal*. Vol 6 (1), pp. 1-8, URL: <http://www.anthropologymatters.com> (retrieved on November 2006)
- Castells, M. 1996. *The Rise of the Network Society*. Oxford: Blackwell Publishers.
- Challet, D. and Le Du, Y. 2005. "Microscopic Model of Software Bug Dynamics: Closed source versus Open source". *International Journal of Reliability, Quality and Safety Engineering*. Vol. 12, No. 6 (2005) 521-534.
- Cook, K.S. & Whitmeyer, J.M. 1992. "Two Approaches to Social Structure: Exchange Theory and Network Analysis". *Annual Review of Sociology*. Vol. 18, pp. 109-127.
- Coomber, R. 1997. "Using the Internet for Survey Research". *Sociological Research Online*, Vol. 2, No. 2. URL: <http://www.socresonline.org.uk/home.html>
- De Moura, A.P.S.; Lai, Y-C; Motter, A.E. 2003. "Signatures of Small-World and Scale-Free Properties in Large Computer Programs". *The American Physical Society: Physical Review E*. Vol. 68, pp. 017102-1-4. (<http://prola.aps.org/abstract/PRE/v68/i1/e017102>)
- Deleuze, G. 1991. *Empiricism and Subjectivity: an essay on Hume's theory of human nature*.
- Dicks B & Mason B. 1998. "Hypermedia and Ethnography: Reflections on the construction of a Research approach." *Sociological Research Online*, Vol. 3, No. 3
- Edgar, I. R. 2004. *Guide to Imagework: Imagination-Based Research Methods*. London: Routledge.
- Escobar A; Hess D; Licha I; Sibley W; Strathern M; Sutz J. 1994. "Welcome to Cyberia: Notes on the Anthropology of Cyberspace [and Comments and Reply]". *Current Anthropology*, Vol 35, No. 3 (Jun 1994), pp. 211-231.
- Foucault, Michel. 1982. "The Subject and Power". *Critical Inquiry*. Vol. 8, No. 4. (Summer), pp. 777- 795.
- Gledhill, J. 2000. *Power and its Disguises: Anthropological Perspectives in Politics*. London: Pluto Press.
- Gluckman, M. 1968. "The Utility of the Equilibrium Model in the Study of Social Change". *American Anthropologist*, New Series, Vol. 70, No. 2 (Apr), pp. 219-237.
- GNU. *The GNU Operating System*. <http://www.gnu.org/home.html>. 30/03/2006.
- Harvard Law Review Association. 2001. "Exploitative Publishers, Untrustworthy Systems, and the Dream of a Digital Revolution for Artists". *Harvard Law Review*, Vol. 114, No. 8. (Jun.), pp. 2438-2461.
- Hendry J. 1999. *An Introduction to Social Anthropology: Other People's Worlds*. Hampshire: Palgrave.
- Kearney, M. 1995. "The Local and the Global: The Anthropology of Globalization and Transnationalism". *Annual Review of Anthropology*, Vol. 24, pp. 547-565.
- Kirtsoglou, E. (2004) *For the Love of Women: Gender, Identity and Same-Sex Relations in a Greek Provincial Town*. London: Routledge.

- Kollock, P. 2000. "The Economies of Online Cooperation: Gifts and Public Goods in Cyberspace." in *Communities in Cyberspace*. Smith MA & Kollock P. (eds), pp. 220-239. London: Routledge.
- Komter, A. E. 2005. *Social Solidarity and the Gift*. Cambridge: Cambridge University Press.
- Kreimer, SF. 2001. "Technologies of Protest: Insurgent Social Movements and the First Amendment in the Era of the Internet". *University of Pennsylvania Law Review*, Vol. 150, No. 1, (Nov), pp. 119-171.
- Latour, B. 1993. We Have Never Been Modern. Translated by Porter, C. London: Harvester Wheatsheaf.
- Leach, ER. (1970) 2006. "Introduction to Political Systems of Highland Burma". *Anthropology in Theory: Issues in Epistemology*. Moore HL and Todd S (eds). Oxford: Blackwell Publishing.
- Lefebvre, H. 2000. Everyday Life in the Modern World. Rabinovitch S (translation). London : Athlone, 2000.
- Leung, L. 2005. *Virtual Ethnicity: Race, Resistance and the World Wide Web*. Haunts: Ashgate.
- Lévi-Strauss, C. 1963. *Structural Anthropology*. Jacobson C. and Grundfest Scheopf B. (translation). New York : Basic Books.
- Levy, S. 1984. *Hackers*. Anchor/Doubleday
- Lukes, S. 1974. *Power: A Radical View*. London: MacMillan Press LTD
- Lyon, S. M. 2004. *An Anthropological Analysis of Local Politics and Patronage in a Pakistani Village*. Lampater: The Edwin Mellen Press.
- Markham, A. N. 1998. *Life Online: researching real experience in virtual space*. Oxford: AltaMira Press.
- Mauss, M. 1967. *The Gift: Forms and functions of exchange in archaic societies*. Translated by Cunnison, I. London: W.W. Norton
- Negreponte, N. 1996 *Being Digital*. London: Coronet Books Hodder and Stoughton.
- Nooy W. de, Mrvar A., Batagelj V. 2005. *Exploratory Social Network Analysis with Pajek*, Cambridge: Cambridge University Press.
- Rapport, N. 2006. "Community". *Encyclopedia of Social and Cultural Anthropology*. Barnard A & Spencer J (editors). London: Routledge.
- Ratto, M. 2005. "'Don't Fear the Penguins': Negotiating the Trans-local Space of Linux Development". *Current Anthropology*. Vol. 46, No. 5 (December), pp. 827- 834.
- Raymond, E. S. 1999. *The cathedral and the bazaar : musing on Linux and Open Source by an accidental revolutionary*. Cambridge : O'Reilly.
- Rheingold, H. 1991. *Virtual Reality*. London: Secker & Warburg.
- Rheingold, H. 2002. *Smart Mobs: the Next Social Revolution*. Cambridge: Basic Books.
- Rheingold, Howard. 2000. *The Virtual Community: Homesteading on the Electronic Frontier*. London: The MIT Press.
- Rorty, R. 1980. *Philosophy and the Mirror of Nature*. Oxford : Blackwell, 1980.
- Ruby, Jay. 2000. *Picturing Culture: explorations of film and anthropology*. London : University of Chicago Press.
- Saco, Diana. 2002. *Cybering Democracy: Public Space and the Internet*. London: University of Minnesota Press.
- Sahlins, M. 1999. "What is Anthropological Enlightenment? Some Lessons of the Twentieth Century". *Annual Review of Anthropology*, Vol. 28, pp. I-xxiii.
- Sanjek, R. 2006. "Network Analysis". *Encyclopedia of Social and Cultural Anthropology*. Barnard A & Spencer J (editors). London: Routledge.
- Schiller, Herbert I. 1996. *Information Inequality: the Deepening Social Crisis in America*. London: Routledge.

- Scott, James C. 1985. *Weapons of the Weak: Everyday Forms of Peasant Resistance*. London: Yale University Press.
- Smith MA & Kollock P. (eds). 2000. *Communities in Cyberspace*. London: Routledge.
- Strathern, M. 1988. *The Gender of the Gift: Problems with Women and Problems with Society in Melanesia*. London: University of California Press.
- Strogatz, Steven H. 2005. "Complex systems: Romanesque networks", *Nature* 433, pp. 365-366.
- Surowiecki, James. 2004. *The Wisdom of Crowds: Why the Many are Smarter than the Few*. London: Abacus.
- Torvalds, L. & Diamond, D. 2001. *Just for Fun: The story of an accidental revolutionary*. London: TEXERE Publishing Ltd.
- Tuomi, I. 2002. *Networks of Innovation: Change and Meaning in the Age of the Internet*. Oxford: Oxford University Press.
- Van Gennep, A. 1960. *The Rites of Passage*. London : Routledge & Kegan Paul
- Watts, DJ. 2004. *Six Degrees: the Science of a Connected Age*. London: Vintage.
- Wise, MJ. 1997. *Exploring Technology and Social Space*. London: SAGE Publications.
- Wolf, E. R. 1997. *Europe and the People without History*. London: University of California Press.

Websites

Dictionary.Com. "Modern". <http://dictionary.reference.com/browse/modern>, browsed on 03/03/07.

"Philosophy of the GNU Project", <http://www.gnu.org/philosophy/>, March 2nd, 2007

"How to Ask Questions the Smart Way", <http://www.catb.org/~esr/faqs/smart-questions.html>.

"Linux, GNU, and Freedom" (<http://www.gnu.org/philosophy/linux-gnu-freedom.html>)

"The GNU Operating System- Free as in Freedom", <http://www.gnu.org/>, March 5, 2007.

"The GNU Project", <http://www.gnu.org/gnu/thegnuproject.html>, April 4, 2007.

FOSS (Free and Open Source Software), <http://en.wikipedia.org/wiki/FOSS>, April 10 2007

"The Linux-kernel mailing list FAQ", <http://www.tux.org/lkml/>,

ASA Ethical Guidelines, http://www.theasa.org/ethics/ethics_guidelines.htm

Larry Ewing (lewing@isc.tamu.edu), Original Tux Image, <http://www.isc.tamu.edu/~lewing/linux/>, accessed on 17/04/07.

Mojormo Lists at VGER.KERNEL.ORG, <http://vger.kernel.org/vger-lists.html>, April 20, 2006.

The Hacker Community and Ethics: An Interview with Richard M. Stallman, 2002. <http://www.gnu.org/philosophy/rms-hack.html>, April 3, 2007

The Jargon File (<http://www.catb.org/~esr/jargon/html/index.html>), April 3 2007.

“Open Source Initiative”, <http://www.opensource.org/>, April 10, 2007.
‘The GNU Project’, <http://www.gnu.org/>, March 2nd, 2007.

“Free Software Foundation”, <http://www.fsf.org/>, March 2nd 2007

The Jargon File 4.4.7, Appendix X A Portrait of J. Random Hacker, <http://www.catb.org/jargon/html/weaknesses.html>, accessed on 03/04/07)

The Jargon File 4.4.7, <http://www.catb.org/jargon/html/K/KISS-Principle.html>, accessed on 03/04/07

The Jargon File 4.4.7, <http://www.catb.org/jargon/html/politics.html>, April 11, 2007.

The Jargon File, version 4.4.7, accessed on 10/04/07, <http://catb.org/jargon/html/F/flame.html>
The Linux Counter, <http://counter.li.org/>, accessed on April 26, 2007.

“Copyleft” <http://www.gnu.org/copyleft/copyleft.html>, April 5, 2007.

APPENDIX I

Raymond’s Rules of Bazaar Organisation as Applied to Linux OS: A Summary

1. Every good work of software starts by scratching a developer’s personal itch.
2. Good programmers know what to write. Great programmers know what to rewrite (and reuse).
3. “Plan to throw one away; you will, anyhow.” (Fred Brooks, the Mythical Man-Month, Chapter 11).
4. If you have the right attitude, interesting problems will find you.
5. When you lose interest in a program, your last duty to it is to hand it off to a competent successor.
6. Treating your users as co-developers is your least-hassle route to rapid code improvement and effective debugging.
7. Release early. Release often. And listen to your customers.
8. Linus’ Law: Given a large enough beta-tester and co-developer base, almost every problem will be characterized quickly and the fix obvious to someone. Or: “given enough eyeballs, all bugs are shallow.”
9. Smart data structures and dump code works a lot better than the other way around.
10. If you treat your beta-testers as if they’re your most valuable resource, they will respond by becoming your most valuable resource.
11. The next best thing to having good ideas is recognizing good ideas from your users. Sometimes the latter is better.
12. Often, the most striking and innovative solutions come from realizing that your concept of the problem was wrong.
13. “Perfection (in design) is achieved not when there is nothing more to add, but rather when there is nothing more to take away.”
14. Any tool should be useful in the expected way, but a truly great tool lends itself to users you never expected.
15. When writing gateway software of any kind, take pains to distribute the data stream as little as possible- and *never* throw away information unless the recipient forces you to!
16. When your language is nowhere near turing- complete, syntactic sugar can be your friend.
17. A security system is only as secure as its secret. Beware of pseudo-secrets.

18. To solve an interesting problem, start by finding a problem that is interesting to you.
19. Provided the development coordinator has a medium at least as good as the Internet, and knows how to lead without coercion, many heads are inevitably better than one.

Raymond, ES. 1999. *The Cathedral and the Bazaar*. Cambridge: O'Rilly, pp: 27- 78.

APPENDIX II

Raymond's Guidelines on "How to Become a Hacker": A Summary

I. The Hacker Attitude:

1. The world is full of fascinating problems waiting to be solved.
2. Nobody should ever have to solve a problem twice.
3. Boredom and drudgery are evil.
4. Freedom is good.
5. Attitude is no substitute for competence.

II. Basic Hacking Skills:

6. Learn how to program.
7. Get one of the open-source Unixes and learn to use and run it.
8. Learn how to use the World Wide Web ad write HTML.

III. Status in the Hacker Culture:

"Like most cultures without a money economy, hackerdom runs on reputation."

9. Write open-source software.
10. Help test and debug open-source software.
11. Publish useful information.
12. Help keep the infrastructure working.
13. Serve the hacker culture itself.

IV. The Hacker/Nerd Connection:

"Contrary to popular myth, you don't have to be a nerd to be a hacker. It does help, however, and many hackers are in fact nerds. Being a social outcast helps you stay concentrated on the really important things, like thinking and hacking."

V. Points for Style:

-DO's:

- Read science fiction. Go to science fiction conventions (a good way to meet hackers and proto-hackers).
- Study Zen, and/or take up martial arts. (The mental discipline seems similar in important ways.)
- Develop an analytical ear for music. Learn to appreciate peculiar kinds of music. Learn to play some musical instrument well, or how to sing.
- Develop your appreciation of puns and wordplay.
- Learn to write your native language well. (A surprising number of hackers- the best ones I know- are able writers.)

- DONTs:

- Don't use a silly, grandiose user ID or screen name.
- Don't get in flame wars on Usenet (or anywhere else).
- Don't call yourself a 'cyberpunk', and don't waste your time on anybody who does.
- Don't post or email writing that's full of spelling errors and bad grammar.

"The only reputation you'll make doing any of these things is as a twit. Hackers have long memories- it could take you years to live it down enough to be accepted."

Raymond, ES. 1999. *The Cathedral and the Bazaar*. Cambridge: O'Railly, pp: 231- 250.

APPENDIX III

GNU/Linux Mailing List Questionnaire

Hello,

This is a rather strange e-mail for these mailing lists, I know. I am a third year Social Anthropology student in the University of Durham doing my dissertation (thesis) on the Anthropology of GNU/Linux. I would really appreciate if you could help me out and offer some of your time to fill in the questionnaire below- it will only take 2 minutes. Replies will be confidential and everything in the dissertation will be anonymous. Results will be e-mailed to participants upon request.

Thanks in advance, and... enjoy!

Maria Kastrinou

QUESTIONS:

1. When did you start using GNU/Linux OS?
2. What is your level of involvement?
newbie/ user/ developer (delete as appropriate)
3. Why are you using Linux?
4. Is Linux fun? How?
5. Which distribution of Linux do you use?
6. What in your opinion constitutes a 'good hack'?
7. Would you describe yourself as a 'hacker'?
8. Which super-hero (apart from Tux) do you think would represent Linux best?
9. Describe Microsoft OS in one word.
10. How do you view the recent patent agreement between Microsoft and Novell?
11. GNU GPL, copyleft and freedom of speech: good, bad or irrelevant?
12. How many Linux mailing lists are you a member for?
13. Would you reply to a question sent through Linux mailing lists and why?

Personal Info:

- a) Age:
- b) Gender:
- c) Occupation:
- d) Your current geographic location:

Any other comments?

Would you like me to e-mail you the results? YES/ NO

Hope you enjoyed it!

THANK YOU!

Maria