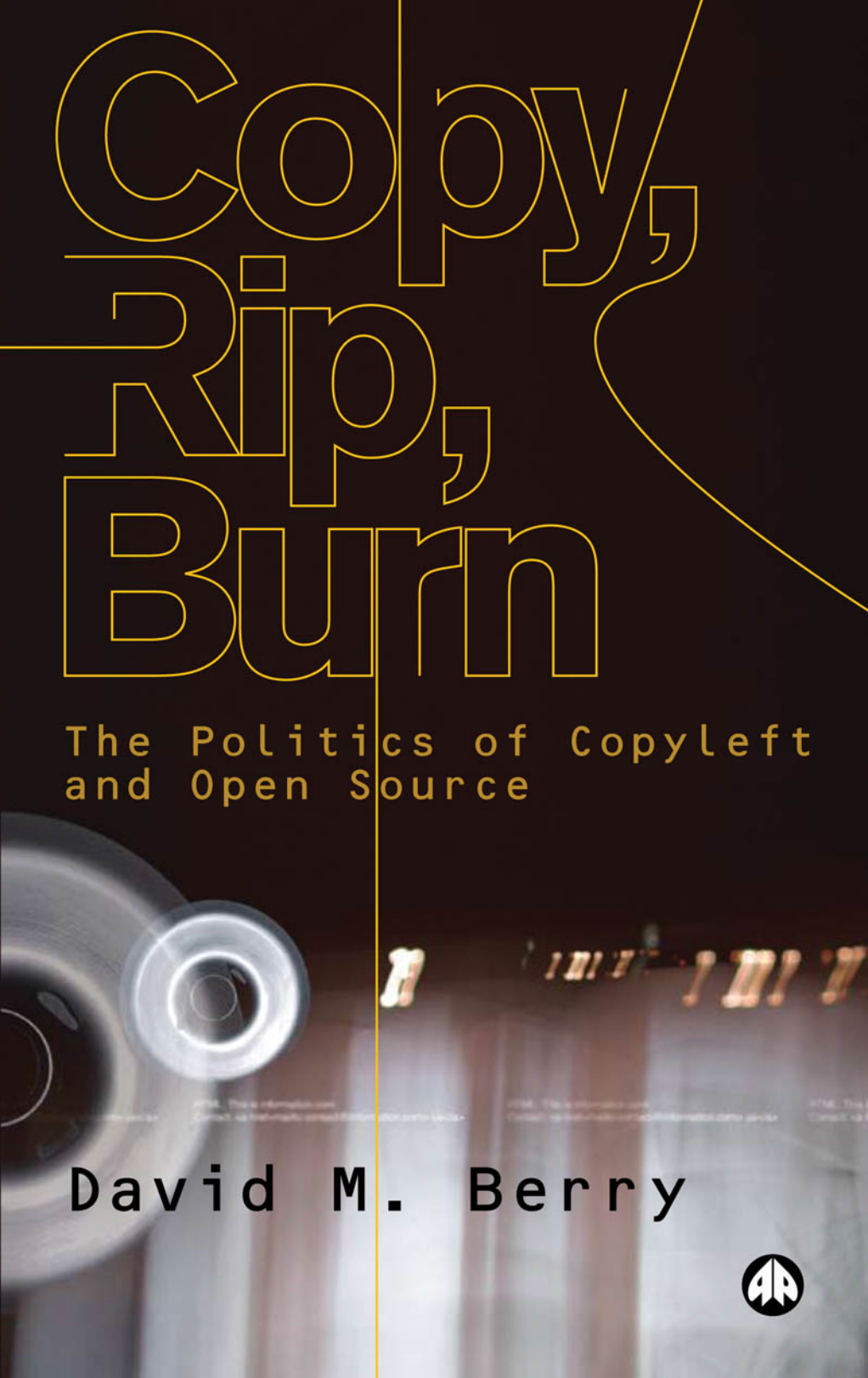
Copy, Rip, Burn

The Politics of Copyleft and Open Source



David M. Berry

Copy, Rip, Burn

Copy, Rip, Burn

The Politics of Copyleft and Open Source

DAVID M. BERRY

PLUTO PRESS
www.plutobooks.com

First published 2008 by Pluto Press
345 Archway Road, London N6 5AA

www.plutobooks.com

British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library

ISBN 978 0 7453 2415 9 Hardback
ISBN 978 0 7453 2414 2 Paperback

Library of Congress Cataloging in Publication Data applied for

This book is printed on paper suitable for recycling and made from fully managed and sustained forest sources. Logging, pulping and manufacturing processes are expected to conform to the environmental standards of the country of origin. The paper may contain up to 70% post consumer waste.

10 9 8 7 6 5 4 3 2 1

Designed and produced for Pluto Press by
Chase Publishing Services Ltd, Sidmouth, EX10 9QG, England
Typeset from disk by Stanford DTP Services, Northampton
Printed and bound in the European Union by
CPI Antony Rowe, Chippenham and Eastbourne

CONTENTS

For Trine, Helene and Henrik.

Prometheus: I caused mortals to cease foreseeing doom.
Chorus: What cure did you provide them with against that sickness?
Prometheus: I placed in them blind hopes.
Chorus: That was a great gift you gave to men.
Prometheus: Besides this I gave them fire.
Chorus: And do creatures of a day now possess bright-faced fire?
Prometheus: Yes, and from it they shall learn many crafts.
Chorus: These are the charges on which –
Prometheus: Zeus tortures me and gives me no respite.

Aeschylus, *Prometheus Bound*

ACKNOWLEDGEMENTS

Writing a book is never a solo endeavour and I would like to thank all those who contributed in a number of different ways to make the finished book possible. In particular I would like to thank Caroline Bassett, Michael Bull, John Holmwood, Christopher May, William Merrin, Giles Moss, William Outhwaite, and Darrow Schecter, for their engaging arguments, critiques and generosity in sharing ideas, concepts and theories that have influenced my work. I would also like to thank Trine for being there.

PREFACE

The world is one and common to those who are awake, but that everybody who is asleep turns away to his own.

Heraclitus (2006, Fragment 89)

This book critically engages with the activities and theoretical exchanges between the free/libre and open source software groups who write and share computer code online. I place these groups in the context of the expansion of intellectual property rights and look at their discourses surrounding the enclosure of the 'intellectual' commons. In particular, I explore how free/libre software and open source software (FLOSS) articulates productive forms of self-knowledge and discipline (such as through discursive formations and code), which appear to establish a potential for uncoordinated and decentred models of creativity. In doing so, I investigate how code designates what is prescribed and what is not, what is articulated and what is silenced; how code structures our lives and our subjectivity. I hope to uncover the way in which the open source and free software groups are challenging our existing liberal categories (around cultural production, knowledge ownership and authorship) both in economic terms (that is, as a new form of commons-based peer production) and in terms of political liberties (for example, the question of free speech, democracy and its connection to code).

First, I am interested in the disciplinary nature of knowledge and power and this is an important element in my political-economy-influenced approach. Secondly, I am interested in political intervention as practice. Thus this book forms a normative project of both explanation and a contribution to further praxis in the field of creative research. Thirdly, I wish to offer readers a set of concepts that can be used both to think creatively about the questions I raise but also to offer political possibilities.

Heidegger (2000) called for a more profound interpretation of the epistemology of technology, arguing that technology is a special form of knowledge – a form of truth or disclosure. Here then I would like to explore the extent to which the social practices of the FLOSS groups are introducing a rupture or break with the immediately given and accepted 'nature' of technology. That is, rather than abandoning technology, these groups foreground technological approaches to the world and bring the given of technology into fuller consciousness. In a related manner, a *politics of code* asks fundamental questions about human relationships with complex technologies, technologies whose complexities sometimes exceed the human ability to manage their interconnected parts. Below I examine the activities and discourses of the FLOSS groups' approach to a 'politics of code' and whether they could contribute to such a Heideggerian project of disclosing technology.

Much of the literature on the open source movement is scanty theoretically – essentially popular journalism – or takes a particularly liberal approach to the understanding of the subject. Questions regarding the motivations of actors are addressed as individual preferences of groups operating within the sphere of artistic or cultural production and this individualistic outlook informs many rational-choice-oriented approaches to this issue.[1] To date a great portion of the literature is heavily concerned with questions of legal theory and intellectual property connected to the idea of the Romantic artist. The Romantic artist is the idea of an original author or *auteur,* as the French describe 'artistic' film directors, who has somehow wrought an original creation from nothing (*ex nihilo*), which is sometimes considered to be an act of genius. Clearly this liberal and one-dimensional explanation of creativity leaves a lot to be desired; in contrast I argue that creativity requires a social environment to flourish.[2]

By focusing on questions of collective creativity and desire I feel that we are better able to question the notion of the Romantic artist and also to offer the possibility of collective action as a creative moment. The common is a key aspect to thinking in terms of the ways in which a 'technology of the common' could

raise critical awareness of the collective moment in production. But it also contributes positively to new ways of approaching and applying methods of working, which legitimate and encourage the flourishing of social action and political practices.

This book also aims to question the assumptions of the 'information' or 'creative' society. One of the most common of these is the argument that 'incentivation' can encourage individual creativity and hence economic growth. Thus the motivation for the artist, musician, designer or writer is explained purely through their desire for profit; to stimulate their creativity and innovation more intellectual property rights (IPR) legislation is required. The argument for a 'creative' economy can therefore be used to cast everyone in the unlikely Thatcherite model of one-dimensional profit-motivated entrepreneurs rather than complex and multifaceted human beings. Additionally, there is built into much of the legislation a bias towards an understanding of creativity through the creative acts of lone genius, singularly creating works out of nothing. But as we must constantly remind ourselves, behind every musician, composer or author there is an army of teachers, friends, peers, producers, editors and managers who all contribute in different ways to the final artefact. No woman or man is an island and creativity is always a collective achievement.

There is a tension between the monopolistic granting of property rights in information and the democratic needs to expand the flow and access to this information. Copyright and other intellectual property laws seek to restrict access so that only those able and willing to pay might make use of the work. This restriction of access may therefore actually reduce the ability of certain members of society to get the information they require in order to make informed social, economic and political choices and widen the gap between an 'information rich' and 'information poor'. The actions of the free software and open source movement which are predicated on a sharing of both the structural code and the content that sits upon it (i.e. the algorithms and meaning of the code), places it squarely in conflict with the owners of copyright and other intellectual property rights. It does so particularly when understood in relation to the hugely profitable content industries,

which must be able digitally to restrict distribution and copying in an informational market relying on a notion of scarcity. This is because if the code is open, there is no way in which the protection methods, commonly known as digital rights management, which serve as the locks on creative works, can be kept secret. This is one of the critical issues that serves to explain the current actions of the content industries to undermine the free software and open source movements, and their focus on hardware driven technological protection that cannot be bypassed through software (e.g. trusted computing). If that fails (as increasingly seems to be the case) then it will not be surprising if the next approach is the co-option of FLOSS into new models of production (and Web 2.0 companies, such as Google, can be understood in that light).

As governments around the world begin to consider their legislative agenda for the expansion of copyright and patents (mostly influenced by corporate financed lobby groups), it is time to ask whether the steady expansion and extension of the privatisation of shared knowledge are things we can afford to ignore. It is crucial that questions about the new political economic structure of knowledge are critically discussed in the public sphere.

Globally, manufacturing is being eclipsed in the rich world as the internationalisation of trade and globalisation encourage the movement of capital and labour around the globe to cheaper locations. It is sometimes argued that the industrial base upon which these world economies have depended for centuries is shifting to that of information, knowledge and communications. Thus we may be standing at the crossroads of a new form of economic system that is a creation of the North, the holders of the majority of the world's copyrights, patents and trademarks.

As we enter a knowledge age, increasingly structured by corporate desires for profit, democratic debate within civil society helps citizens and publics to contest the ownership, control and direction of the 'information' society and potentially shape it towards more democratic ends. I hope that this book will contribute to that debate.

1

THE CANARY IN THE MINE

There's something I don't understand about the open-source movement. Oh, I understand open-source intellectually. I understand that it means that source code is open to be read and reviewed and perhaps revised by anyone who wants to… What I don't understand is something more sociological. I don't understand who those folks are who want to do all that code reading and reviewing for no recompense. It goes against the grain of everything I know about the software field. (Glass 2000: 104)

Man produces himself through labour. (Marx and Engels 1999: 21)

In 1995, two scientists from the University of Mississippi were granted a patent on a method of increasing the effectiveness of treatments of wounds and cuts by the use of turmeric in a 'special preparation'. They calculated that the estimated market for this product could be worth billions of dollars a year. Turmeric as a treatment for minor skin cuts and wounds has been used in India as a traditional remedy for hundreds of years. However, intellectual property law in the US does not see anything that constitutes 'originality' or 'inventiveness' in traditional remedies and so is unwilling to grant any protection to traditional knowledges.[1]

In America in 1998, a man who had his spleen removed by doctors as treatment for leukaemia discovered that the doctors had proceeded to patent some of the genetic material they removed from his body. After the patient sued the doctors, the court found that the man did not have any claim to his own bodily material as it was a 'naturally' occurring substance and he was classified as a 'source' who had 'abandoned' his genetic material. Yet it was argued that the doctors, due to their 'expertise' and 'ingenuity', had contributed to an 'original' and creative act by 'discovering' this cell-line and were awarded the property rights

to this portion of the patient's genetic code.[2] The products that are being developed using his genetic material are estimated to be worth over $2 billion annually (Boyle 1996: 22–4).

In 2002, a nursery in the US innocently painted pictures of Disney characters on the walls for the amusement of the children, aged between one and five years old, who played there. Somehow, the Disney corporation (estimated market capitalisation – $20 billion) found out and their lawyers sent a cease and desist letter to the nursery explaining that this represented an infringement of copyright. They warned that the nursery should remove the offending paintings and images from their walls. Failure to comply would mean an expensive and drawn-out court action that would most likely bankrupt the nursery. Even though the children from the nursery went on national television to plead for their beloved nursery walls, the Disney representative claimed that they viewed the nursery as a for-profit organisation and didn't feel a need to distinguish between it and other organisations. They stated that any infringement by anybody else would be dealt with in the same harsh way (Cox, quoted in Coombe 1998: 53).

In 2005 in the UK, the government discussed developing a new campaign to teach children and young people that copying music, pictures or text without permission is 'theft' and that intellectual property should be respected in the same way as physical property.[3] The programme is largely funded by the content-industry (e.g. music, film and publishing multinationals) intent on educating children into a 'better' understanding of how intellectual property should be used. The aim is to teach children that whenever they produce any work they should mark it with a copyright symbol to prevent other people (presumably also children) 'stealing' from them. Nobody seems to have borne in mind that children learn by repetition and copying, and teaching 'property' rights in this corporate-approved way is likely to undermine learning and education. Combined with this 'education' programme, the Federation Against Copyright Theft (FACT) has been running a campaign attempting to draw a link between terrorism and copyright infringement (the poster images from which were hastily removed from the web following a critical outcry).[4] Even

language itself is being manipulated with our ongoing corporate re-education about copyright infringement through compulsory property-theft DVD trailers, cinema adverts and assertive control of trademarks and corporate slogans.[5]

Again in 2005, a reborn Napster (the company that was originally infamous for allowing the allegedly illegal copying of music until it was forced into bankruptcy by the music industry) introduced a service that for $15 a month allows customers to rent music online by downloading music to your portable music player from their catalogue. In contrast to the download purchase-type services (such as Apple's iTunes), this service is designed to be more like an online music library that you rent from month to month which effectively gives you a huge variety of music from which to choose. The service is aimed particularly at those between the ages of 15 and 25 whom the music industry has identified as most likely to pirate and download illegal music. However, in contrast to purchasing the music, should you fail to keep up payments then the technology will automatically cancel your rights to play on your computer and portable player and your Napster music collection will vanish (Rothman 2005).

Lastly, in 2007, volunteers continued to develop a computer operating system collaboratively over the Internet called GNU/Linux (Stallman 1999). Started in 1991, GNU/Linux has challenged our understanding of the production of complex software projects and the best method of organising, controlling and managing them. In short, GNU/Linux eschews traditional methods of copyright protection and code secrecy in favour of a common-ownership model[6] (known as *copyleft*). It is then freely distributed with the source code for little or no cost and encourages contributions, comments, criticisms and bug-fixes from its users and developers. This has led to an exponential rate of growth both in terms of its code quality (which directly relates to the workable nature of the software in a production environment) and also in terms of its feature-list and capabilities. It is now a viable challenger to Microsoft Windows and is taken seriously as an important infrastructural software product (most noticeably supported by IBM).

These cases and wider arguments over intellectual property rights (IPR) may seem disconnected and distant from our everyday lives and worries. After all, it seems unlikely that multinational corporations would be bothered to look into all our collections of music, film and images. The worlds of IPR, legal copyright cases and patent infringement do not usually impinge on the lives of individuals going about their daily activities. However, corporations are becoming increasingly assertive and aggressive in their claims to rights in intellectual property, as well as increasing their holdings and portfolios. It appears that across the corporate world a new awareness is growing of the possibilities of profiting from the ownership of ideas, concepts, expressions and processes.

These examples serve to illustrate that the relationship is shifting between culture, creativity, and the ownership and control of intellectual property rights. The reconfiguration of IPRs is aimed at maximising profit through exclusion but may have repercussions across the whole of our social lives, transforming our ability to interact, contest meaning and to take part in culture and creativity.

IPR debates find their context in a broader shift, the move towards an 'information society', however this is conceived (and there are contradictory theories as to the extent to which there has been any change at all). However, it would be impossible to deny that governments, particularly in the North, are strengthening their intellectual property laws, and pressuring other countries to follow their lead. They are also investing heavily in the production of information, communication and affective services, either directly through subsidy and tax cuts, or more generally in terms of discursive shifts and exhortations for the population to engage in 'life-long learning' and 'creative' work and to become more entrepreneurial and alert to new technology. One only needs to look at the profound changes operating at the level of the university (and instituted through legislative and funding changes by central government), with the shift from a so-called Mode 1 form of knowledge generation (i.e. 'traditional knowledge' generated within a particular disciplinary and primarily academic

context) to that of Mode 2 knowledge (i.e. generated outside academic institutions in broader, cross-disciplinary social and economic contexts) (Gibbons et al. 1994). Increasingly, private funding is being sought to drive the research agenda (private-sector partnerships, research institutes, and research and technology parks on campus being the most prominent examples), research outputs are monitored and controlled, and non-performative individuals and disciplines, particularly in the humanities, are pressured through closure and funding difficulties. This new institutional justificatory discourse was demonstrated by Professor Philip Esler, chief executive of the Arts and Humanities Research Council (AHRC) in evidence presented to the Science and Technology Select Committee in Parliament, where the value of cultural research is no longer defended in terms of a public good, but rather is solely linked to economic growth and profit:[7]

> It may be that our leverage role will be sufficient here because as you go around the universities you discover that many of them are now introducing knowledge transfer into the heart of their research activity which is where I think it should be; it should be embedded in research activity from the beginning. Some of them are saying to their staff, 'Don't give us an application to a Research Council unless you have addressed the knowledge transfer possibilities' and 'Your promotion application will be helped if you have a knowledge transfer profile', so these sorts of things are already happening.
>
> (HC 310-I 2007: Q35)

Many theorists are now arguing that we are on the cusp of a profound change in the way in which our societies manage and organise the production of both material and immaterial goods. This has been variously termed biopolitical, immaterial or informational production and is said to require new laws, norms and institutions if it is to be financially viable or profitable.[8] This 'new' economics is being constructed through building on the existing institutional intellectual property system (through, for example, copyrights, patents, design rights and trademarks), new legal frameworks and new norms of criminality, but also

through the use of technical devices that can actively enforce or deter actions that infringe these legal rights and actions deemed unacceptable to informational property owners. These are known as *digital rights management* techniques.

Although these changes affect all aspects of agricultural and industrial production from plant and seed ownership to computer-controlled manufacturing, the focus of this book is particularly on the effects on computer code of this widespread drive to control and own information. By seeking to extend property rights to intellectual artefacts (immaterial products) and social relationships (business processes and methods etc.) these interests are strengthening and extending the concept of informational or immaterial proprietorship.

These issues are clearly global; however, due to space and analytical considerations this book concentrates mainly on the geographical areas of the United States and the European Union, and particularly on the free software and open source groups located within those areas. FLOSS, as a networked and increasingly global set of practices, clearly spreads beyond those boundaries; however, historically, the majority of the hacker debates have been located in the US/EU areas. Nonetheless, alternative sites of contestation should be expected in the future (particularly from Japan, Brazil and China) and this should open up interesting avenues for future research.

Throughout this book these issues will be explored with particular attention being paid to their relevance to understanding the political economy of FLOSS together with an analysis of the meanings and discourses of the groups studied. This methodological approach is broadly similar to that which Silverstone (2003) refers to as a 'double articulation', whereby he highlights the importance of paying attention in research to both the material and symbolic dimensions.

The methodological approach taken in this book intends to draw upon the empirical, theoretical and policy work concerned with FLOSS through a theoretically informed understanding of the social order in which FLOSS is being studied. That is, in similarity to cultural studies, this research is concerned with the

construction and exercise of power. In doing so, this approach attempts to keep in mind the importance of meaning and 'how it is produced and through particular expressive forms it is continually negotiated and deconstructed through the practices of everyday life' (Golding and Murdock 2000: 71).

This book will use a combination of Laclau and Mouffe's discourse theory (Laclau and Mouffe 2001; Phillips and Jørgensen 2002) together with elements of Fairclough's Critical Discourse Analysis (Fairclough 1992) to analyse both the contents of texts and how wider sedimented hegemonic discourses within society may intervene to suppress discursive conflict within this order of discourse.[9] This, it will be argued, may lead to a naturalisation (Fairclough 1992: 10) of the open source movement's (OSM) order of discourse.

Discourses can also interpellate individuals by creating subject positions for people to occupy. They imply certain expectations about how to act, what to say and what not to say (Phillips and Jørgensen 2002: 41). Examinations of the discourses of the Free Software Foundation (FSF) and the OSM will demonstrate the subject positions within their discourses and how they are constructed. The rights and obligations of these positions are different in the two traditions and the hierarchical relationships and interaction will be outlined. These have social and political implications (Phillips and Jørgensen 2002: 40). For example, the FSF utilise a discourse of ethics and a discourse of freedom (see Stallman 2003b), whereas the OSM draws on discourses of neoliberalism and technical efficiency (see Raymond 2001).

The critical political economy tradition that informs this research differs from economics in the four respects outlined by Golding and Murdock:

> First, it is holistic; second, it is historical; third, it is centrally concerned with the balance between capitalist enterprise and public intervention; and, finally… it goes beyond technical issues of efficiency to engage with basic moral questions of justice, equality and the public good.
>
> (Golding and Murdock 2000: 72–3)

Here, I discuss FLOSS as a 'limited totality' through a broadly political economy approach to the ownership and control of FLOSS by looking at the ideas, material capabilities and institutions that structure the social practices of the participants (Cox 1996: 98). This is combined with a critical examination of the meanings embedded within the discourses and social practices of FLOSS practitioners through close analysis of the discourses produced by the FLOSS actors themselves (see Fairclough 1992; Laclau and Mouffe 2001). FLOSS is a result of the interaction of the varied actors involved in FLOSS production, including the programmers and hackers themselves (highlighting the role of ideas), the particular technologies that they utilise and build (the material capabilities) and the networked arrangement of their associations and programming groups, which also include corporations and non-profit organisations (the institutions). In this book I am particularly interested in how questions of power are manifested in FLOSS (such as over the direction of FLOSS development – what is built and why not something else?) and how, even in networked groupings seemingly somehow beyond capitalism or in contradiction to it, certain actors can maximise their influence by their ability to control key resources. These actors use both material ownership (in terms of the copyrights on particular pieces of the FLOSS code or technology, for example) but also discursive argumentation and justification (that is, through ideas), and control of the institutions that are formed within the networks of practice that exist within FLOSS development (one revealing example is Linus Torvalds who is described, rather alarmingly, as the 'benevolent dictator' of Linux).

Two theoretical strands, namely free software and open source, dominate FLOSS and their followers express their ideological positions in terms of a universal or general interest, rather than of their own particular interest (Cox 1996: 99). That is, both attempt a hegemonic conception of FLOSS which involves the utilisation of all their key resources, arguments and ability to motivate and influence different actors (this is explored in particular in Chapter 4). Additionally it is important to note that I refuse to identify either structure or agency as the final determinant of social relations.

Rather, I argue that changes within FLOSS are overdetermined, in other words, 'no single set of forces or structures can provide change alone, nor any finally resist change alone' (May and Sell 2006: 33). This is not to argue that the actors involved have complete freedom: their options are constrained by structural forces, organisations and norms. Their actions are embedded in larger structures which can constrain and empower certain actors in disproportionate ways, nonetheless, they 'structure conditions but [do] not determine agency' (May and Sell 2006: 34).

Many of the early researchers into FLOSS used the concept of the gift economy as a means of explaining the behaviour of the FLOSS participants (Ghosh 1998; Barbrook 1998; Lancaster 2001; Raymond 2003), drawing particularly on the work of Mauss (2002) and Hyde (2006) and their explanation of the gift in a number of different societies through which they sought to link the social structuring of a gift economy to the organisation and structure of the group.[10] A gift economy is one in which goods or services are rendered without any market exchange taking place, for example the payment of monies or the bartering of goods. Usually, though, the exchange is mediated through cultural forms, such as a party or feast, and the participants are guided in their gift-giving by particular cultural norms (see Hyde 2006). Barbrook (1998) argues that the gift or potlatch[11] economy is part of the wiring of the Internet: gift giving is technologically determined by the structure of the code that makes up the communications networks and as such we should not be surprised to see certain gift-based cultures on the Internet. This form, though, he argues, is a compromised form that remains in symbiosis with commoditised capitalism in online spaces. Ghosh (1998), on the other hand, argues that it is a 'cooking pot' economy, that 'works on a … different model, of barter and division of labour (I provide the chicken, you the goat, she the berries, together we share the spiced stew)' (Ghosh 1998). Both of these models attempt to explain the gift-like nature of the contribution of software code to a larger project; however, a gift economy is usually made up of reciprocal gifts, which is not the case for FLOSS software. Indeed, one of the puzzles of FLOSS is

that although contributions are made, there is nothing given in return – there are no transactions between parties in most FLOSS development (although some have questioned whether reputation 'earned' is the return on investment – see below). The question of the circulation generated by FLOSS is one that I will explore later in the book, and certainly understanding FLOSS as connected to the circulation of capital is a critical part of understanding the motivations and social practices of FLOSS actors.

The World Wide Web

Our lives are increasingly mediated through digital technology (Castells 2001). Through computers, technical devices and countless databases, servers and storage systems, information has grown in importance and value. But, as information itself has become more crucial to modern society, so too has the desire to profit from it (Litman 2001: 89–99). Indeed, information, when viewed as a potential form of profit, justifies new ways of legitimating its ownership as a property right. And, of course, information when viewed as property seems to require fences; virtual fences that can both identify it as being owned and prevent others from taking it without paying (Bettig 1996; May 2000; Drahos and Braithwaite 2002: 15). This has begun to affect the way in which certain technological developments on the web are evolving. Online business models are constantly shifting to try to take account of the open-publishing model that currently dominates the Internet. This has taken place in conjunction with a growth in interactivity and user practices that have, to a large extent, populated the web with content. In fact, many of the recent moves in technology companies have been aimed at harnessing the creative power of their users in order to valourise their production, a process that has been described as that of 'free labour' (Terranova 2004: 73–94), in the sense of unpaid, user-generated content that is linked to profit-producing technologies such as subscriptions, services, advertising, social networking and so on.

Web-pages themselves are collected into groups of decentralised websites that lie within open unrestricted areas of access connected

by hyperlinks written mainly using HTML, itself an open standard that presents its 'code' as a freely viewable source code. This 'overt intertextuality' has the result that any user can access, view and download pages, as in principle every text is linked to every other text (Mitra and Cohen 1999: 182–3). This, combined with the persistent nature of the Internet's structure, can give rise to the mistaken assumption that all texts are created in a public domain or public sphere (Jones 1999: 5; Mitra and Cohen 1999: 183). Additionally, texts, such as web-pages, can remain in existence long after the author has forgotten about them and can be easily replicated in multiple forms across the Internet – a feature of Usenet groups, for example, that early contributors could not have foreseen (Sharf 1999: 246). These are where some of the current issues of copyright pervading the Internet are unfolding through numerous copyright-infringement cases and legal challenges (Lessig 1999).

The Internet has also provided the environment for new forms of social practices that are remarkable in their diversity, accessibility and persistence and which have excited researchers in many different disciplines. This social activity is predominantly manifested within code and through discourse, articulated online within a textual substrate which the Internet facilitates in low-cost reproduction, instantaneous dissemination and radical decentralisation. Further, due to the Internet's digital substructure the texts are stored in online repositories, web-pages, caches and so on, enabling easy accessibility and retrievability of texts which can be later viewed and easily manipulated without data loss or corruption.

The technologically flexible, dialogical and fluid nature of the Internet, which allows users to post and read texts with little restriction, lends itself to being conceptualised as a vast open-access public sphere, a position that is highly contentious (Bakardjieva and Feenberg 2001). Indeed, assumptions of the innate *public*-ness of the Internet contribute to some of the problems of understanding the Internet, due to the loaded nature of terms such as public and private and the difficulty of applying them to the online world (Herring 1996; Waskul 1996; Ess 2001). The

concepts of public and private are highly contested in the offline world too (Benhabib 1992; Habermas 1992), and I therefore use a provisional and minimal conception of 'private' and 'public' throughout this book.

Additionally, when using the Internet with a browser, the digital processes taking place behind the scenes operate on the basis of making copies. The browser is continually downloading web-pages and displaying them for the user to read and view, held locally as a 'copy' of the web files located on the website. In fact this is a critical issue when understanding how control over copyrights indicates who will have power over the digital environment in the future. To place a file on the web server or computer hard disk is to make a copy, to send via email is to make multiple copies, even to play a file as an MP3 or edit it as a document is to work on copies downloaded and opened temporarily into memory. These copies can then lurk in caches that are hidden in often-unexplored areas of the computer hard disk. But they are all copies, and copying is expressly a right that is controlled by a copyright holder. In the digital world, the owner of the copyright will no doubt assert the right to control the shape and direction of technologies, and to determine their use and the consumption models in this post-modern economy (as we have seen with Napster's enforced bankruptcy, for example). Digital technology functions by copying and manipulating digital files, an issue that conflicts directly with the copying right held by IPR owners. It is no surprise, then, that copyrights should become a key source of conflict in the information society as the common-sense dichotomy over 'legal' private copying (that is, as fair-use/dealing) is challenged when digital technology and networking are combined. When every node on a network can share an identical copy of any file, the difference between a public and private use becomes extremely blurred. It is no surprise, then, that the debate has moved to remove or outlaw the private rights for using digital files which in other media such as VHS and vinyl records were considered completely legitimate, such as making a tape backup, photocopying sections or selling on the goods (that is, the 'first sale' doctrine[12]).

Open Sources

The Free Software Foundation was established in 1985 and dedicated itself to promoting computer users' rights to use, study, copy, modify and redistribute computer programs. Over the past 30 years the popularity of free software (and its ideological competitor, open source) has increased dramatically, and has had a profound effect on computer developers' practices, challenging the activities of many major software firms (such as Microsoft and Apple Corp) and influencing popular culture through films such as the *The Matrix* (1999), especially in its concern with the 'source'. Particularly with the rise of the Internet, which was itself a product of the same liberal and libertarian values which drive the free software movement, free software and now also open source have gone on to provide a stimulus to the creativity and productivity of the entire software industry and have contributed to the emergence of new companies (such as Amazon, eBay and PayPal) and new forms of distributed creativity (manifested in projects such as Linux, Apache and Creative Commons and elements within the latest Web 2.0[13] craze).

Free/libre and open source software (FLOSS) has gradually infiltrated the deployment of software in the corporate sector (typically understood as infrastructure projects) and is now influencing the commercial off-the-shelf market (such as Microsoft Office). Some companies now energetically expound the new 'open source' mentality as a business opportunity that gives them a competitive edge over their rivals. Amazon, for example, has been a keen advocate of FLOSS and claimed a $17 million saving in licensing and labour costs for the first quarter of 2005 (Wheeler 2005: 18). As FLOSS has demonstrated its ability to provide extremely efficient and reliable solutions to complex computer needs, the term 'open source' has become popularised as a cultural term associated with freedom, progress, effectiveness and productivity. Indeed, corporations have been busy appropriating and promoting their 'open source' credentials, most memorably with Steve Jobs proclaiming that Apple Computer represented open source 'for the rest of us' (Cringely 1999) even as they were

investigating claims of 'theft' of their operating system source code.[14] As FLOSS increasingly takes on a cultural meaning in addition to its status as a software development practice, it has begun to spread from its home in the technical world of computer software programming and influence other spheres. We now have 'open source democracy', courtesy of Demos (Rushkoff 2003), 'open source politics' (Osborne 2007) and even 'open source beer' (Boyd 2005). Surprisingly, though, despite its wider usage as a marker of progressive and future-oriented policy,[15] the underlying logic of open source and free software, called the 'Wealth of Networks' by Benkler (2006), has been subject to limited empirical and critical attention. Indeed, it is only recently that these values and ideas have been questioned and their emergence and development become contested.

The considerable impact of FLOSS on the discursive and social practices of developers working within the information and communication technology (ICT) sector makes this new social form of production a pertinent object of inquiry. However, the increasing cultural resonance and influence that this technical practice is having within social and cultural spheres adds to the importance and relevance of research into the free software and open source movements. Although this book aims to examine the historical development of FLOSS from its origins as an infrastructural research project to a commodity, and the established structure for its production, distribution and consumption (with a special emphasis on the GNU/Linux operating system widely considered to be the most successful and influential open source project), it also seeks to engage critically with the underlying political and economic logics that FLOSS represents.

An important focus of this research is the open source technologies that make up the technological infrastructure of the Internet and the World Wide Web. The Internet is built upon a constellation of technologies that were written under free software licences that expressly allowed the copying and reproduction of their code (for example, BIND, TCP/IP, DNS, HTTP etc).[16] These technologies were released under open source and free software licences that loosened the bonds of copyright in this field and allowed for the

phenomenal expansion of the web, particularly as code could be freely copied, installed and improved. This greatly aided both the expansion in use of many experimental and untested software ideas (such as HTML) but also meant that getting the different parts of the Internet to interact and work together (known as interoperability) was made immeasurably easier through the use of standardised connections called protocols.[17] It also massively expanded the knowledge base of users who could work together to share ideas and programming solutions in a very efficient networked manner.

Essentially, software that is supplied without source code, usually under restricted licence[18] terms, is known as *proprietary* software. The majority of available commercial software falls into this category (Stallman 2003a). In contrast, software that is sold or distributed with the source code, which provides no restrictions on further copying and modification, is known as *free software* or *open source*. Richard Stallman[19] argues that unless software is supplied with the source code then it should not be considered software at all – if it's not source, it's not software (Stallman, 2004).

For source code to be usable, it must be transformed using a compiler into the machine code that can be read and executed by the computer, and which is known as the 'object file' or 'executable'. This process removes the original source code files after the executable has been produced and thereafter the source code files will not be required for subsequent processing of the executable. This process facilitates the commodification of computer software, as the easily read intellectual property held within the source code is stripped at the compilation stage and the program logic is disguised by the sheer complexity of the resultant object code. The executable may then be sold as a licensed 'product', such as Microsoft Word, and users pay both to be able to use the software and in order to upgrade to new versions. Although it is possible to reverse-engineer object code back into source code, it is complex, time-consuming and usually illegal.

One of the immediate problems facing research into FLOSS is a definition of who exactly is involved in 'free software' or

'open source' software development. To date, this has been investigated through a number of web-based surveys, ethnographical studies and textual-based surveys of code that have given some empirical information about the gender division (mostly male), the nationalities (mostly US) and the kind of interests (mainly technical) that FLOSS developers share (see Chapter 2). But they fail to provide much in the way of a definitive feeling for the kinds of people involved in FLOSS and how and why they do it. It is interesting to note that developers involved in these projects often do not themselves feel they are members of a shared 'community' at all; indeed many reject wholesale the arguments and values presented, particularly by the free software foundation and its evangelical leader Richard Stallman, for example. Many more see themselves as engineers or scientists doing what scientists and engineers have always done, sharing information, methods and ideas to save time and improve their own work. For them there is nothing new here and nothing to see.

Yet clearly the FLOSS movement does represent something different and interesting to the researcher in terms of production and also as a contrast to discourses of production and privatised ownership. Here are individuals seemingly committing large amounts of time and effort to a common project that has produced some of the most startling technical developments in recent memory. The Internet itself can be conceived as a vast FLOSS project linking together a diverse range of software elements that have been freely shared and are available for others to download, use and adapt.

One example is the GNU/Linux operating system that has grown from a small unstable student project into one of the world's cheapest, modern, stable and reliable operating systems, putting some commercial products to shame. That is, a software project running into millions of lines of computer code that has no formal organisational structure, no corporate hierarchy and few of the problems that seem to plague commercial software projects. But even more startling is that the software is available free of charge, incorporating within it the rights to modify, improve and share the software, with the only proviso that future versions are equally

free from constraint. This is a software project that has caused the largest and most profitable software development company, Microsoft, to describe open source as a 'cancer', as 'communist' and above all, and perhaps most revealingly as 'a strategic threat' to the Windows monopoly on the desktop (Greene 2001).

But there is also the question raised by the form of interaction that FLOSS development seems to engender. This is a form of co-operation seemingly divorced from direct monetary remuneration – high-quality and complex work that is contributed freely to a common repository. Interactions are conducted across vast geographical distance and amongst people who rarely, if ever, meet. Clearly here there are real questions about how these multiple and shifting encounters can be stabilised in order to ensure that a working and viable project remains coherent, but also about how this networked model of production remains reproducible over time and space without falling apart because of the weak bonds between the participants. This model of 'commons-based peer production', as Yochai Benkler (2006) calls it, seems inexplicable, and yet it has produced some notable successes.

But that has not stopped many people from trying to explain it, and in doing so they have run the gamut from economic explanations of individual benefit by contributing to a common project to Marxist descriptions of the coming revolution ushered in by an informational mode of production built on the free exchange of products and services. There has been growing academic interest in free/libre and open source software movements particularly in regard to questions as to its technical and economic impact (DiBona et al. 1999; Weber 2004; Ghosh 2005; DiBona et al. 2006). Most of this literature has concentrated either on issues of individual motivation for these projects (which are usually presumed to be voluntary and provided without remuneration or payment for the labour expended) or else with the viability of FLOSS as symbolic/informational production in distributed and de-centred networks online. These debates have tended to be centred either around economic constructions of a 'rational' actor and the problem of collective provision of public goods or on highly theoretical or abstract conceptions with little or no

connection to empirical reality (such as the technical gift economy). Within the boundaries of these types of research programmes, the FLOSS phenomenon has looked either increasingly irrational and inexplicable or curiously reactionary and unable to meet its purported revolutionary potential as a new mode of production.

Perhaps one of the most important mechanisms for free software and open source software production is that of licences. These documents are essentially copyright licences that permit uses of the source code that would normally be restricted or expressly prevented by copyright (such as copying the files). The most famous of these is the GNU General Public License (GPL) which grants generous freedoms to users of the software and source code, providing that any further software built from this source code is also licensed under the same agreement. This is known as 'copyleft' licensing. These licences can be understood as a mechanism for social ordering, in that they help constrain and stabilise the interactions of the developers involved in FLOSS development. The licences operate as 'machines' for producing further FLOSS software. Further the licences embed particular values and norms that are transmitted across the networked, dispersed and often spatially and temporally separated members of FLOSS development groups. The licences therefore function as an organising device in managing people involved in FLOSS projects and a means of social control. They also have an important educative and reinforcement role ensuring that the agency of the developers is 'framed' within a particular field of action. Those who expressly 'break' or ignore the terms of licences such as the GPL will be variously flamed, hacked or subjected to email and discussion board altercation.

Although there is a variety of different open source licences, such as the Berkeley Software Distribution License (BSD), Apple Public License and the Mozilla/Netscape Public License, a large percentage of software written under open source is released under the GPL and the BSD licences. This book will focus on the GPL and the BSD as examples of an ideal-type of public licence (generally the others either water down provisions from the GPL/BSD or else strengthen the corporate ownership and control of the

software)[20]. Additionally there are proprietary software licences (known as End User Licence Agreements or EULAs) which, generally speaking, are extremely restrictive licences which are deemed to be accepted by the customer merely by opening the shrink-wrapping or clicking the agreement licence splash-page on the software. EULAs restrict the rights of the customer to use the software in any way that might jeopardise the proprietary selling of the software, such as preventing disassembly and reselling. They are also notable in making the customer accept that the software publisher accepts no liability at all for anything the software might inadvertently do, such as crashing, wiping data, or causing computer failure or breakdown.[21]

Hackers and Users

One of the key questions related to FLOSS software is how the computer software hacker[22] identity is constructed and contested. Today, the computer hacker has gained a stature in popular culture that is completely at odds to the nerdy social misfit of the 1970s and 1980s. We only need look at films such as *The Matrix* (1999) to see how hacking has become absorbed into fiction and the political unconscious as a cool and trendy but illicit and criminal activity. This, I argue, is a result of the threefold expansion of intellectual property protection, pervasive digital devices and the ubiquity of the Internet. Interestingly too, there has been a rise in government discourse regarding the emergence of a 'creative' citizen, technologically equipped to work in the knowledge or creative economy; this can be compared and contrasted with the earlier hackers for whom governments generally developed a negative discourse in terms of illegality and criminality. In many ways, these two forms of subjectivity are different sides of the same coin, but the implication is that the hackers lie somehow outside legality in their use of technology, whereas the creative citizen lies within the boundaries of the law.

Here I would like to review some of the key thinkers to have explored the question of hackers and code. One of the first books to draw attention to the commodification of ideas and the

importance of software (and by extension hackers and users) as a focus of study was James Boyle's *Shamans, Software and Spleens* (1996), which came from a legal tradition of academic work that focused on the nature of the public domain. The public domain is defined legally as a space of creative works whose copyrights have expired and thus are available for public use, dissemination and republication without restrictions. Drawing on Lange (1981), an influential legal studies paper, Boyle noted that contributions to the public domain were decreasing as intellectual property rights increased in their term limits and breadth of coverage. Boyle analysed this shrinkage of the public domain in conjunction with a rise in the 'information society' and speculated on how our notions of what could and could not be property were slowly being changed. One of the most interesting claims in the book was the distinction drawn between the legal definition of the 'source' and the 'author'; in contrast to many who proclaimed the death of the author, Boyle believed that the author had become a necessary fiction in order for the legal system to institute a workable notion of intellectual property. Thus, he argued that the power of the author (or the owner of the subsequent protected work) would be a site of contestation in the coming information society. The public domain, with its reliance on expired copyrights, which fed back creatively into the production of new cultural texts and information, would then offer a fundamental contradiction to the requirement for an autonomous subject of authorship, since its offer of 'free' culture possibly would undermine an economy based on the consumption of commodity texts. Boyle argued for greater awareness and a political contestation of intellectual property rights through an 'environmentalism' of culture (Boyle 1996, 2003). Unfortunately, it was never clear who might populate this 'empty' subject or movement.

Lawrence Lessig has become the most widely known legal scholar for both his theoretical and practical interventions in FLOSS and free culture (Lessig 1999, 2002b, 2004). His tireless criss-crossing of the world, attending conferences and even making legal representations to the Supreme Court, have given him and his work an extremely high profile in the online world and FLOSS. His first major work on the issue of FLOSS and the

challenges it presents to the Internet was *Code and Other Laws of Cyberspace* (1999). In this book, Lessig attempted to unpick the relatively new phenomena of cyberspace and the structural features that constituted and supported the online environment. It is here that Lessig made his most incisive and widely discussed connections between Code and Law, stating baldly that on the Internet 'Code is Law' (Lessig 1999: 3–8). Although perhaps not as original a connection as is often assumed – it was anticipated in 1978 by Winner (2001: 317) in the phrase 'Technology as Legislation' – Lessig here made a more literal homology between the US constitution and code.

Using the analogy between code and law, Lessig has attempted to show how, due to a widespread misunderstanding about the code-based structural mechanisms of the Internet, online groups, software developers and others were gradually being co-opted by governments and corporate interests into making the Internet *less* flexible and *less* amenable to an anarchic cultural democracy. The book also connected to a wider cultural shift in its attention on the role of technology and code in our everyday lives. Lessig argued that cyberspace was not autonomous, nor could it ignore the offline world, despite the techno-utopians proclaiming the freedom of cyberspace where they could 'reject kings, presidents and voting [in favour of] rough consensus and running code' (Clark, quoted in Lessig 1999: 4).

Lessig foresaw, correctly, that a rising tide of regulation from both governmental pressures for regulation and corporate pressures for the protection of copyright and other intellectual property would have profound effects on the 'freedom' of cyberspace. He argued that a threefold resistance was needed via (1) the judiciary, through legal precedents to translate constitutional remedies into this new landscape; (2) code and computer programmers, through a commitment to what he quaintly referred to as 'open code'; and lastly (3) to politics, or more accurately a more deliberative moment in politics (Lessig 1999: 221–30). So far these remedies have not been widely adopted. The Supreme Court firmly rejected Lessig's *Eldred v. Ashcroft* case that tried to hold back the tide of continual copyright extensions; FLOSS programmers have

mostly failed to create a defined political subjectivity or even attach themselves to a concerted political programme. Finally, with the election of George W. Bush in 2000 and the terrorist attacks in 2001, a liberal political attitude towards the Internet was firmly wiped off the agenda.

With *The Future of Ideas* (2002b), Lessig began to link his ideas to a concept of the commons and social creativity, highlighting the importance of 'the freedom to tinker' (Lessig 2002b: 61) in order that innovation and new forms of social and technical practices emerged. Perhaps due to his legal background, or his instinctive liberal desire to achieve a consensual 'middle-ground' position, Lessig used a somewhat idiosyncratic term 'open code' – presumably to avoid offending either the free software or the open source movement (Lessig 2002b: 61–2). In the end he was forced to revise his ideas with *Free Culture* (2004) where he admitted that a return to the focus on 'free culture' was important:

> The inspiration for the title of and for much of the argument of this book comes from the work of Richard Stallman and the Free Software Foundation. Indeed, as I reread Stallman's own work, especially the essays in *Free Software, Free Society*, I realise that all of the theoretical insights I develop here are insights that Stallman described decades ago.
>
> (Lessig 2004: xv)

Lessig launched the Creative Commons organisation in 2001 to provide a unified front for a number of different free culture movements, through a variety of Creative Commons licences. However, this project has been fraught with infighting, institutionalisation and controversial relationships with industry that have been widely criticised.[23] He now positions himself as a spokesperson for the 'free culture' movement, although increasingly it is defined within a narrow conception of remixing and reuse of culture, which he calls 'read/write culture' (Needle 2006).

Yochai Benkler is one of the most important critical legal scholars to think carefully about the implications of peer-to-peer[24] and the network. In his early work he concentrated on economics-based approaches to explain how the network production model was revolutionary for information production (Benkler 2002, 2003a,

2003b). The economic foundation of his approach has served to give a theoretical foundation to the problem of motivation for public goods within a network environment. Nonetheless, Benkler's call for a 'core common infrastructure', or a space of non-owned cultural production, echoes many of the other critical legal scholars who have seen a key project as the establishing of an infrastructure to support commons-based production. Importantly, this is not defined as a political project but rather as a technical correction to a policy problem. It also goes some way to explaining the early germination of the thinking behind the Creative Commons, which is directed towards addressing the legal problem of a lack of commons-supporting structures (see Chapter 6 for a more detailed examination of this topic). In *The Wealth of Networks* (2006) Benkler identifies the importance of 'social production' as a key factor in the development of a network society, using concepts such as autonomy, social justice and freedom as key values that he feels are emphasised in a commons-based production model. This is in contrast to the old industrial models of proprietary ownership, secrets, and state monopolies controlling and owning information through intellectual property law. Most usefully though, Benkler outlines what he calls an 'Institutional Ecology' which clearly differentiates the different layers that can be analysed in regard to their implementation as either open structures or enclosures. These include (1) physical transport, such as the wires that carry the network; (2) physical devices, such as computers; (3) logical transmission protocols, such as TCP/IP; (4) logical software, such as Microsoft Word; and (5) content, such as movies, music and books (Benkler 2006: 395). Throughout this book, I will be using this layer model[25] (which is not exclusive to Benkler) as a useful way of thinking about the differing layers of commons in discussions of the nature of the political economy of FLOSS.

Critical legal studies have also contributed to rethinking the way in which software is conceptualised; they have successfully argued that it be re-thought as a category of speech by contributing labour to legal cases and practices (for example, the Electronic Freedom Foundation). In the important case *MPAA v. 2600*, an action

was bought by the Motion Picture Alliance of America (MPAA) against the hacker magazine *2600*[26] and Emmanuel Goldstein[27] (otherwise known as Eric Corley, the proprietor of the website). In this case *2600* had carried copies of the DeCSS algorithm, which had been hacked by the Norwegian teenager Jon Lech Johansen, and which allowed DVDs to be played on any computer platform. More controversially, it could be used to 'rip' or extract films from DVDs to be stored on computer hard disks. DVDs contain an encrypted layer that prevents them being played in regions that are not authorised by the use of special codings (for the motion picture industry this has the advantage of allowing them to control the release of films worldwide).[28] Unfortunately, the CSS encryption also prevented DVDs from playing on open platforms such as GNU/Linux. The DeCSS software essentially circumvented this protection and allowed DVDs to play on GNU/Linux and other operating systems.

The magazine's defence team argued that reproducing code on their magazine website was protected by freedom of speech as defined in the American Constitution. As their lawyer, Martin Garbus, commented, 'what this case basically deals with is balancing First Amendment values – the right to an open Internet, the right to free speech' (*Feed* Magazine 2000). In the end *2600* lost their case regarding the hosting of the DeCSS code, and also were told to remove the hyperlinks to other sites holding the DeCSS code. However, rather bizarrely, having the links as static text (that is, not hyperlinked) was considered an expression of free speech in text. This demonstrated some of the interesting contradictions faced in trying to understand and define what code is; in the case of the HTML URL links on the *2600* website, the live links were considered by the court to be mechanical devices whereas unlinked HTML URLs were considered text, and therefore protected. The court also reaffirmed the relationship between code and free speech:

> If someone chose to write a novel entirely in computer object code by using strings of 1s and 0s for each letter of each word, the resulting work would be no different for constitutional purposes than if it had been written in

> English. The 'object code' version would be incomprehensible to readers outside the programming community (and tedious to read even for most within the community), but it would be no more incomprehensible than a work written in Sanskrit for those unversed in that language… For all of these reasons, we join the other courts that have concluded that computer code, and computer programs constructed from code can merit First Amendment protection.
>
> (*MPAA v. 2600* 2001)

In another key case defendants in the US argued that source code is a linguistic expression and should therefore have full First Amendment protection as free speech[29], whereas the plaintiffs argued conversely that as code is understood by a minority of the public, and is a device or circuitry, it is therefore not expressive (Mendels 1999). On 4 April 2000, the United States Court of Appeals for the Sixth Circuit held in *Junger v. Daley*:

> Though unquestionably expressive, these things identified by the Court are not traditional speech. Particularly, a musical score cannot be read by the majority of the public but can be used as a means of communication among musicians. Likewise, computer source code, though unintelligible to many, is the preferred method of communication among computer programmers [sic].
>
> (*Junger v. Daley* 2000)

Further they held that:

> Because computer source code is an expressive means for the exchange of information and ideas about computer programming, we hold that it is protected by the First Amendment.
>
> (*Junger v. Daley* 2000)

Clearly then, software is important as both a structural architecture governing action – what Lessig (1999) has termed code as law – and also as an element in the agency of the developers – as free speech. This makes software an interesting and pertinent focal point for understanding the claims of the new economy and the changing relationships between property rights, the commons, political rights and economic structures.

The Commons

The common is that material space we share and the precondition for our commonness, our shared understandings as human beings (Silverstone 2007: 35). Arendt (1989) argues that without commonality we are condemned to a 'private world of self-interest and political impotence'. Arendt uses the metaphor of a table, around which we can sit and which provides a material environment for us to communicate and share a common world – it both connects and separates us. Silverstone (2003) argues that we should understand the media as a contemporary location of shared space giving us the ability to be present with other human beings. In this public world we have plurality but still are able to reflect on our commonness:

> For though the common world is the common meeting ground of all, those who are present have different locations in it… being seen and being heard by others derive their significance from the fact that everybody sees and hears from a different position … Only where things can be seen by many in a variety of aspects without changing their identity, so that those who are gathered around them know they see sameness in utter diversity, can worldly reality truly and reliably appear.
>
> (Arendt 1989: 57)

The emergence in recent years of debates about the loss or decline of this common is also a key issue related to the questions asked in this book. It is here that the discourse produced by the social practices of the free software (and open source) movements may be representative of an underlying change that has taken place at the level of creativity and production. In other words, what is the perceived threat to the common? Indeed, how can we conceptualise and understand what the common is and how it has changed in recent years? How do the commons-based production of open source groups and their dispersed and collective notions of ownership and control contest or support an information-based society?

It is the discursively constituted subject of the 'creative' citizen that a discourse analysis of free software and open source

documentation will explore later in this book. The free software community intentionally and unintentionally constructs shared meanings (their common world) for understanding hacker identity in their discursive practices. This analysis will help to deconstruct and reveal the meanings inherent in code, 'creativity' and the politics of open source and will look at the ways in which these movements use identity as a means of establishing authenticity. It will also examine how free software and, to a lesser extent, open source, is imagined as a movement or community (Anderson 1983).

The discussion of the citizen in the space of the public or 'common' is by no means a modern phenomenon. These questions are made up of multiple historical, philosophical, cultural, political and economic understandings of the citizen and public identity vis-à-vis the rest of society. However, what is a relatively new phenomenon is the association with and appropriation of a 'creative' citizen that inhabits a common world of technology within the process of the emergence of an information-based economy and, more particularly, new intellectual property laws. These efforts have been led by attempts by the US and the European Union (both net exporters of intellectual property) to reposition their economies (EU 2005b) as producers (and owners) of information, knowledge and communications. The laws they have introduced emerged from policy documents as early as the 1970s in the US and in 1999 with the EU Declaration on the Information Society.[30]

The discourses of the information economy attempt to obscure the contradictions inherent in the expansion of intellectual property rights (such as access to knowledge vs. private ownership; monopoly vs. free flow of information; abundance vs. a legally constructed scarcity). But as May (2002) argues, the knowledge structure is an important possible site of resistance in itself reinforced through the diffusion of technologies of the information society. This can clearly be seen in the practices of the hackers and crackers of the free software and open source movements and their critical practices and discourses.

New technologies and new ways of using information are continually being developed and these serve to question our assumptions about copyright and creativity. The current criminalisation of piracy, data 'theft' and hacking are the latest salvos by industries trying to restrict the flow and use of their creative work. It is interesting to note that the owners of these creative works are seldom the creators and pressure for the extension and strengthening of copyright comes almost exclusively from the multinational corporations. This alone should raise questions as to who is benefiting from the rise in intellectual property protection (see Drahos and Braithwaite 2002: 15).

Previously there has been a lack of critical enquiry into the concept of 'commons'. Recent moves to try to protect some ideal of the commons in liberal discourse have, I believe, resulted in the hollowing out of the concept of the commons. The commons has been transformed into a 'Creative Commons',[31] which is a distorted and diminished concept of the commons, predicated on a system of private rights codified in intellectual property rights (IPRs) (I discuss these issues in more detail in Chapter 3).

Code

Code usually refers to the digital collection of 0s and 1s that can be used to store functions for operating a computer (as machine code) or alternatively for storing information (as binary data). However to all intents and purposes the encoding of either machine code or binary data is exactly the same; to the viewer the representations cannot be discerned, and indeed a computer will attempt to execute binary data or read machine code as a file (but the consequences may be a crashed computer). The digital encoding of analogue information (such as in the ripping of an old vinyl LP) is the transfer from one medium of storage (grooves in vinyl) to another (digital bits that can represent waveforms).[32] In any digital transmission the information is broken down to its most basic level as a string of 0s and 1s and chopped into neat packets of data and transmitted through a network, rather like little parcels sent through the post. The flexibility of being

able to render information, whether audio-visual or textual, into this standardised digital form allows the incredible manipulation and transformation of information that computers allow. It also facilitates the access, storage and relational connections between vast quantities of data located in different places, such as is demonstrated through search engines like Google. Thus code must be understood as a medium, if not a *super-medium*.

For computer programmers, code is a shortening of the term 'source code', usually understood as the concretisation of general algorithms instantiated into particular programming languages. For example, the mechanical process required to move a unit of data from point A to point B could be written in English. However, for the computer to execute this command, this algorithm needs to be translated into machine code (Stallman 2002a: 3). At this level the code is represented digitally and numerically and is very difficult to write directly. Indeed, the mythology surrounding expert programmers and hackers dates back to the times when this was the only means of programming computers (Williams 2002; Post 2003). The production of computer code at this low level is complex and time-consuming. To enable programs to be written more quickly and remove complexity, an abstraction of the underlying machine code is used instead. These are the contemporary programming languages, often known as third-generation languages (3GL), in which the human programmer or coder is usually required to write; examples are C++, Java and Basic.[33] These highly abstracted languages use a formalised syntax and are usually constructed around simplified English keywords. Together with symbols and punctuation, programs are written in a structured syntactical style made up of statements, loops and conditionals by the programmer to construct the logical operation of the program.

Programs are written in preliminary documents, usually plain text files, which contain the logic of program operation and are known as source code files. In addition to the concrete controlling logic of the program flow, the source code will often contain a commentary by the programmer in a special textual area usually delimitated by special characters, 'REM' in Basic for example.

These comments assist both the programmer and others wishing to understand the programming code. Additionally, these textual areas are used to demonstrate authorship, list collaborators and document changes.

Code has a number of connotations that are continually contested and reinvested with meaning. The Free Software Foundation (FSF) and open source movement (OSM) both argue that the original source code should always be made available. For different reasons they hold that the ability for an end-user to be able to view the source code is important. However, even though the OSM and FSF use similar arguments for the benefits of freely available source code, they differ radically in their underlying philosophies,[34] an issue that will be explored throughout this book.

Software itself can be analysed using three analytical categories which this chapter introduces:

1. *use functions*: the ability and freedom to perform a specific or general computer-based task using the software;
2. *prescriptive functions*: restrictions on what can be done with the software, usually by architecting into the software the delegation of explicit control and prescriptions on the user.
3. *external functions*: actions outside the scope of the software; this latter category is used to distinguish between prescriptive functions and actions that cannot be performed because the software was not designed for the purpose, for example playing music on a word processor.

Proprietary software is sold on the value of its use functions. A word processor, for example, has the functionality to produce a wide variety of documents and letters. In contrast, software manufacturers often conceal the prescriptive functions built into the software. Due to the pressure from content providers, copyright owners and patent holders, manufacturers increasingly control usage of the software by restricting the actions of the user through the use of prescriptive functions.

A contemporary and highly contentious example of these techniques is digital rights management (DRM).[35] DRM prevents users from carrying out unauthorised actions on copyrighted works

often irrespective of the ownership or rights of the individual user (Lessig 1999). Adobe Acrobat and E-paper, for instance, have the ability to prevent the user from copying, changing and even quoting from a protected document when using the particular DRM-protected software in which the document is delivered to the user. The software is *delegated* the legal restrictions of the copyrighted work and then *prescribes* these restrictions back on the user. The user is thus unable to perform activities that break the terms of the legal copyright.

Prescriptive functions also raise privacy concerns,[36] especially with increased use of monitoring software, such as that increasingly installed in the workplace, which monitors illicit actions by the user and reports them to the employer. The ability to read the source and prevent the prescriptive potentials of software that might, for instance, constitute an unlawful infringement of the user's liberty (like, for example, access to public domain works or fair-use) is one of the justifications for viewable source code made by the FSF. For example, the Trusted Computing Platform is the latest in a long line of approaches by the hardware and software industry to allow it to tackle end-users' security problems with digital computers, but also opens the possibility of enhanced DRM linked to this security model.[37]

Structure of this Book

The free/libre and open source[38] movements developed out of debates and practices in hacking, programming and technical groups in the 1980s and 1990s. This book explores discourse and debates between these groups and the nature of their production practices. In particular, I want to explore the role of new technologies created and maintained by these dispersed online networks that have problematised the moves made by governments and multinational corporations toward creating a global intellectual property regime. Following the growth in digital technology, with its capacity to copy, store and reproduce audio, visual and textual information effortlessly, intellectual property rights (IPRs) around the world have been under continual

pressure. Governments, the creative industries' spokespeople and major corporations argue that much is at stake when the ease with which digital copying takes place removes any obstacles to the 'pirating' of products. They maintain that without stronger intellectual property protection there can be neither creativity nor any incentive to create. This disparate group of computer programmers, academics, legal thinkers and online groups argue that, on the contrary, extending and strengthening intellectual property protection has serious repercussions for democratic debate and also will also make it harder to reuse culture and ideas in new ways – a critical part of the creative process – in effect undermining the very creative economy that governments argue for. They argue that the right to turn cultural artefacts and meanings into property (such as those granted through copyright) and the ability for a public to use and articulate 'meaning' (through criticism and debate in quoting and reuse) manifest a critical contradiction within post-modern[39] liberal societies. If people cannot 'speak' without buying the rights to the underlying property, then the needs of democratic citizens are necessarily silenced. Therefore the new terrain of creative authorship is a cultural question and here I examine it through the discourses and political economy of FLOSS production.

With an approach drawn from political economy, I am interested in the control of computer code. I argue that the ownership of copyrights and other intellectual property rights will be a site of power *and* resistance in the 21st century. Computer code will be an important key resource, both as a form of property in itself, and as a means to engender control (through rights management), enforce ownership rights (through surveillance, access control and as digital locks) and to contest them (through hacking, cracking and cultural jamming). Clearly, with FLOSS the ownership question is dispersed across networks that make analysis complicated by the difficulties of tracing ownership in a commons-based project built on top of the intellectual property system (and discussed in more depth in later chapters). The ownership of code raises interesting questions about the locus and mechanisms of ownership and also offers ways of critiquing the current acceptance of property-rights

based systems that attempt to 'incentivise' creativity. FLOSS is a practice that subverts common-sense notions of property rights, and in doing so offers a means through which larger structural changes in property rights can be demonstrated and contested. It also shows how knowledge is transformed into a form of information that can be controlled and made scarce through computer-code-controlled locks and fences.

Today, knowledge is no longer thought of as a public good that contributes to an informed citizenry, but rather is considered in terms of making money, as a key source of profit in itself, in contrast to the ideals of the Enlightenment. In the US there have been concrete moves towards the further commodification of knowledge through expansion of the legal recognition of patents and DRM protection but this is also taking place globally in nations that are signatories to the WTO-sponsored Trade-Related Aspects of Intellectual Property Rights (TRIPS) agreement.[40] These moves have been seen to create or develop a 'creative' or 'knowledge-based' economy, and they are a key policy programme of many Western governments, supported by the creative industries[41] (European Council 2000; May 2000; Barroso 2005). Information, knowledge, concepts and ideas are subject to an expansion in intellectual property protection that in the past has attracted little critical debate regarding its pernicious effects on a democratic society (Boyle 2003; RSA 2005). Even when there is discussion of the balance between IPRs and the public sphere, the former is understood through 'hard' facts based on jobs and economic growth and the latter based on the 'soft' issues of culture and democracy (see particularly Withers 2006).

One of the most important digital 'goods' is computer programming source code.[42] Computer code is a vital part of the infrastructure of our modern technological societies, whether mechanical, informational or aesthetic. Information and communication technology (ICT) relies on the production of computer code to drive and control the processes that handle data control, circulation and collection to meet the computing requirements of other industries. There are economic benefits to be gained from the production and ownership of this source code

and the legal recognition of computer code as a form of 'property' has become important. But increasingly, wealth that is channelled through the online environment has also become reliant on the code as a *super-medium* for controlling and delivering all forms of media content, from music to feature films. Consequently, corporations demand further property protection of the computer code medium itself and also seek to protect it by using technological locks, legislation and re-education programmes.

In the US, this has led to the extension of patents to software (which is already protected by copyright) through 'business process' protection. The arguments in favour of this expansion are extremely contentious. Broad patenting could lead to the propertisation of the basic building blocks of computer programming itself. An example is the Amazon One-Click patent, which gives Amazon a 20-year monopoly on the single-click method of buying goods online (and, remember, clicking is one of the most basic web 'actions' possible). Computer programmers would not be able to program computers freely if the basic structures were owned and controlled – unless, of course, they had a large corporate employer who could defend the numerous legal property claims and insulate them;[43] there is a curious paradox in the idea of keeping knowledge and information *restricted* in order to stimulate wider software creativity, which requires *openness*.

Equally, there have been moves to introduce similar patent legislation in Europe through the Computer Implemented Inventions Directive in 2005 that would have expanded patent protection to software. In the end this was decisively rejected by the European Parliament and withdrawn from debate (mainly as a tactical measure by those *supporting* the legislation). These issues raise the question as to whether democracy can slow down the expansion of intellectual property rights, particularly when claims for economic growth and corporate profits are involved.

In the UK, the Labour government's commitment to a review of intellectual property protection in its 2005 election manifesto stated they '[would] modernise copyright and other forms of protection of intellectual property rights so that they are appropriate for the

digital age' (Labour Party 2005: 99). Since then the government has instituted a number of committees and working groups to look into the issue and recommend changes.[44] In particular, the Department for Culture, Media and Sport (DCMS) began to develop policy in this area and, unsurprisingly, spokespeople for the 'creative' industries sought to increase their lobbying efforts aiming to extend and expand intellectual property protection (Rowan 2005).

Although the forecasts of a 'new' economy built on knowledge and information are largely drawn from the European Union's 2000 Lisbon Agenda, the continual expansion of Intellectual Property Rights (IPRs) can be traced back over 100 years, linked mainly to the needs of copyright holders to hold on to their lucrative monopoly protection and large profits guaranteed by the state. Today, the advocates of intellectual property point out that IPRs are central to the technologically developed societies' defence against the threat of the export of manufacturing and industrial work to cheaper labour locations (through outsourcing, for example) and to protect innovations from 'piracy'. This, they explain, can be seen in the growth in the importance of international world trade agreements for the protection of copyright, patents and trademarks and their corresponding value to national economies. Indeed, measures of research and development work, innovation and intellectual capital have become some of the key indicators of the health and productivity of national economies – sidelining population, health and suicide rates (Barry 2001: 104). The circularity of this justificatory discourse then inevitably leads to calls for extended and more comprehensive IPR legislation that in turn criminalises previously unregulated or legal activities.

Thus arguments about the threats from the 'globalisation' of capital (or indeed of piracy) are used as a stick with which to beat back any attempt to limit or even discuss the dangers of granting ever-longer monopoly rights to information and knowledge. Indeed, often these debates over intellectual property may not appear public or even political due to the technical or legal form of the debate that takes place (Bowrey 2005: 78). However, it is precisely in debates over technical detail – of the administrative

or legal status of rights – and the moral and ethical legitimacy of claims to these rights that intellectual property conflicts are now being played out.

The voice of the public, whether viewed as citizens or, more depressingly, as consumers/customers, has been increasingly drowned out of this discussion (see Couldry 2004: 3–4). The corporations continually make the case for 'immaterial' products to be realised as profit, arguing that protection for digital goods should be clear, legally unambiguous and fully alienable. This has to date been implemented through a number of different means: (1) the use and extension of existing intellectual property rights (IPRs), such as copyright and patents; (2) through the implementation of digital rights management technologies (DRM), such as CD copy-protection; and (3) the End User Licence Agreements (EULAs) or 'click-through' contracts, the terms of which are increasingly weighted in favour of the corporations at the expense of the consumer (or citizen). In other words, three layers of protection are wrapped around the knowledge object – copyrights, code and contract – and consequently the carefully constructed balance between the public sphere and private interest is lost.

The debate over IPRs has become increasingly hysterical, with denunciations of 'pirates', 'thieves' and 'anarchists', rather than reasoned discussion over how the expansion of a private right at the expense of public access to information and knowledge is best balanced. Too much attention has been focused on the profit-related aspects of intellectual property and not enough on the dangers inherent in the commodification of our cultural common meanings, both those we build and those we hold as history, an issue that I hope to highlight throughout this book.

At the core of this problem is the creative industries' attempt to link copyright and creativity – they argue that without copyright there could not be creativity. Here, creativity is correlated with a nation's economic success that, it is argued, relies on the creative productivity of its population (see Florida 2004). Thus, on the one hand, it is argued that the maintenance or extension of intellectual property rights is an incentive for a plurality of individuals to be able to develop an innovative and creative

culture that is crucial for cultural vibrancy (that is, as a fruitful cultural democracy and public domain). On the other hand we are told that the acquisition and monopolisation of intellectual property has become vital to business strategy and profitability and to the generation of economic power (that is, those that produce creatively have a right to the fruits of their labour). The production of software challenges both these contentions and by limiting our creativity to acquisitive behaviour and profit (the 'rational' actor of economic theory[45]), undercuts and subverts alternative notions of what creativity could and – in my opinion – should be. The question of creativity also links to our understanding of what it is to be a creative producer of knowledge or information, especially when considered in light of the creativity required to produce computer code.[46] Should we expect creative producers to become workers on 'work-for-hire' freelance or short-term contracts, working on creative assembly lines under the division of labour of the mind that Adorno and Eisler (1994) famously described in their account of the production of Hollywood film music in the 1940s?

Corporations increasingly seek to own and profit from creative copyrights in perpetuity, in effect owning and controlling culture. They further attempt to enrol the state in enforcing their particular private interests in a way that raises important questions about the extent to which the cultural industries could 'capture' aspects of the state.[47] Somewhere between these arguments, the needs of citizens are being lost, particularly as knowledge becomes increasingly commodified and turned into digital forms. As language, ideas and concepts are slowly drained out of our public and common usage, our critical and democratic need to express ourselves, and to use and reuse culture in new and challenging ways, is blocked, foreclosed or only available at a price. By controlling the use of trademarked phrases, brands or copyright claims, code is able to act as a gate-keeper to knowledge and police the ability to use and disseminate it – through code-enforced intellectual property rights (such as digital rights management). Computer code has the potential to have a chilling effect on democratic debate.

Against the conventional wisdom that creativity requires strong intellectual property laws in order to spur innovation, free software and open source advocates rely on a non-commercial free exchange of information, software and expertise. Utilising what has been identified as a 'hacker ethic' (Himanen 2001), those involved in these practices and those who use FLOSS products challenge the proprietary computer software industry with an increasingly viable alternative to shrink-wrapped software products and traditional bespoke software production. Now IBM, Hewlett-Packard and other major computer corporations use open source software and sell services based upon it to customers. Microsoft too has admitted that it sees more of a strategic threat from GNU/Linux, a software/open-source operating system available freely on the Internet, than from other proprietary computer software vendors.[48] This isn't just affecting the producers of software 'tools'. Copyright-protected cultural production is also challenged through the use of open-source-type licences that make available freely shared content, the biggest example of which is the Creative Commons,[49] which is used by a number of ostensibly free-content websites such as ourmedia.org and archive.org.

The software that open-source groups write is owned collectively, peer-produced and shared freely between all groups (including programmers, users, artists, musicians and citizens). Open source certainly highlights questions regarding the 'common' production of public goods that have been lost from our contemporary vocabulary, especially through viewing the monopoly granted by intellectual property rights through a narrow public/private binary. One of the key justifications given by open source and free software proponents for the success of open source and free software is that 'it works'.[50] Not only do its proponents argue that it is more efficient and costs less to develop than traditional software development methodologies but additionally, because the source code is freely available on the Internet, the access to and editing of code means that the users and developers can avoid 'lock-in'[51] and customise and improve their software.[52]

With their emphasis on sharing, openness, freedom and community, these movements appear to collapse the distinction

between production and consumption that is a hallmark of capitalism. It is partly in this way and to broaden a sense of how 'it works' that this book investigates the social logic of FLOSS production and its important normative principle that everyone should have access to the source material for reuse and adaptation. This sharing and openness, it is argued, has an important democratic effect on cultural production and social practices (it might be described as a fledgling 'semiotic' democracy[53]) (Lessig 1999; Fisher 2004), by giving citizens the possibility of looking inside the so-called black box of software.[54] As more of our cultural production is locked away behind protected software containers, through mechanisms such as DRM, which is itself legitimated through intellectual property rights, FLOSS throws light on the mechanisms that are regulating our digital environment. Open source is also 'the canary in the mine' because it has been one of the first movements to raise the alarm about the commodification of knowledge and information[55], particularly through threats to its own practices of software production, but also by highlighting the dangers of rights management software, which it is impossible to implement in FLOSS code.[56] It has also offered important critiques of IPR in its online debates and discussions, as will be discussed throughout the book.

Starting with Chapter 2, I focus on the information society, intellectual property rights and a history of the software industry to give context to the political economy of FLOSS through the work of a number of key theorists.

In Chapter 3, I look at the concept of the commons drawing on history, Roman law and political theory to try to create an analytical tool for understanding FLOSS. Through these concepts I also hope to throw some light onto the way in which the concept of the commons becomes a crucial way of understanding informational modes of production.

These concepts then contribute to explanations developed in Chapter 4, where I undertake a detailed examination of the underlying structures and practices of FLOSS and outline how they are supported through private mechanisms of ownership (that is, copyright/copyleft). These, I argue, are critical to understanding

the way in which online groups such as FLOSS organise and direct their software projects.

In Chapter 5, I then look at the contestation of computer code, and discuss how competing claims about 'code' reflect important normative standpoints within free software and open source. Through a close reading of their discourses I examine how each side asserts ideological claims about their preferred form of intellectual property rights, immaterial labour and community more generally.

Finally, in Chapter 6, I outline the key findings of the book and try to highlight the way in which FLOSS gives us an important insight into understanding and critiquing the notion of the information society. FLOSS, through its practices, also underscores the ways in which the direction of technology remains open to human action. That is, that politics still matters in relation to technology, and the concept of the 'commons' is a critical tool for understanding the way in which technology remains radically open to contestation. The social practices of FLOSS groups make 'code' public, but also change it from a technical issue into a matter of concern. In doing so, they undercut notions of the autonomy of technology, placing it firmly back within the sphere of political and social control.

In this chapter I have introduced some of the basic concepts and themes that I will be exploring throughout this book. These include questions of the commons, new subjectivities such as the hacker/creative citizen and the growth in intellectual property rights. I have also discussed some of the background ideas, such as code, that are important for understanding how my approach will seek to uncover the structures and practices of the free software and open source movements. I have introduced the issue of the expansion in IPRs within a global framework that is linked to moves by the rich world to attempt to guarantee economic stability and profit in the twenty-first century by the instantiation of new international treaties, such as the TRIPS agreement, and also through discursive constructions of a 'new' economy based on information, knowledge and communications. I now turn to some of the important theoretical perspectives presented regarding informationalism.

2

THE INFORMATION SOCIETY

> The world of 1948 was vastly different from the world of 1996. The American economy, more then than now, was viewed as the ultimate in technology and productivity in virtually all fields of economic endeavor [sic]. The quintessential model of industrial might in those days was the array of vast, smoke-encased integrated steel mills in the Pittsburgh district and on the shores of Lake Michigan. Output was things, big physical things. Virtually unimaginable a half-century ago was the extent to which concepts and ideas would substitute for physical resources and human brawn in the production of goods and services.
>
> (Alan Greenspan, quoted in Perelman 2003)

This chapter examines contemporary developments across the spectrum of economics, politics and law that contribute to the 'new economy' of creativity and intellectual property production. This economy is closely linked to the cyclical rises and falls of the software industry that facilitates many of the changes taking place and highlights the importance of changes in the production of software to an information-based economy. Here, I aim to outline a critical and historical foundation for the political economy of FLOSS that I look at in later chapters. The limitations of space mean that by necessity a broad overview is given, but I intend that the following sections draw together the main themes that are present in this analysis of FLOSS culture. I will then spend the later chapters analysing these trends through the optic of the FLOSS movements.

I start by looking at more general changes in relation to shifts in corporate structures and government responses to manufacturing flight to cheaper labour locations in what Sennett (1999) calls the 'new capitalism', which he argues is identifiable by discontinuous reinvention of institutions, flexible specialisa-

tion, and concentration without centralisation (Sennett 1999: 47) together with Jameson's notion of 'late capitalism' (Jameson 1992) and particularly the concept of the 'information society' (Webster 1995; May 2002). Then I look to the history of the software industry to trace the developments over the past 50 or so years and explore the extent to which software has supported and enabled these wider structural changes. Next I look at changes in intellectual property rights worldwide and the way in which immaterial property, such as patent holdings and copyrights, have become increasingly important to corporate profitability and governments' conceptions of national success and wealth.

The New Capitalism

Over the past 40 years, 'creativity' has become the focus of an intensified interest by governments and capital. Claims are made that creativity is the key to the functioning of modern economies and as such creativity must be 'democratised' (Florida 2004), that is that we must all equally be able to 'be creative' and through this creativity, more productive.[1] This is a widening of the discourse of creativity that originally corresponded to the ideal of the Romantic 'creativity' of the lone artist, to describe instead the rather more mundane market production of goods and services (see Howkins 2001; Florida 2004). Indeed, governments, non-governmental organisations, multinational corporations and intergovernmental agencies now strive to build the infrastructure, legal environment and economic tools through treaties and agreements in international and national law to support 'creativity' (Drahos and Braithwaite 2002). Thus, the concept of creativity is being reconfigured to meet the needs of capital, no longer limited to the Romantic myths of authorship (Boyle 1996; Prichard 2001: 2–3; Boyle 2003), nor predicated on the concept of the 'Great Men' or the 'Genius'. Instead, creativity is understood as a 'floating signifier', that is it can correspond to a number of different meanings across society.[2] This ambiguity of the meaning of creativity is both positive, in that it opens the

possibility of new work forms that are stimulating, intellectually challenging and non-routine, and negative when used to rhetorical effect which can be used to mask routine work (e.g. call-centres and code-writing centres). Creativity, then, is generally linked to the concept of immaterial production or mental labour of all kinds which connects to computerisation; explicitly because computers rely on the work of human minds in order to produce, code, control and communicate using technology.

Today the discourse of creativity is also used widely to distinguish between an older industrial form of capitalism, and the new world of media, information and knowledge, often titled the 'creative' economy (Howkins 2001: xvi–xvii). Here the creation of new processes, inventions, works, ideas and productive knowledge across a wide spectrum of the economy are thought to be the key to modern (or perhaps, post-modern) economic expansion. However, this latest development in thinking should be placed within a historical context that begins with the emergence of theories of the information society.

The information society

The information society usually refers to a shift in Western economies from the production of goods to the production of innovation.[3] Sometimes referred to as 'post-Fordism', this identifies a shift from a Fordist mass industrial society that was epitomised by Henry Ford's huge mass-production factories, to that of an economy based on information and technology, and shifts in the patterns of consumption, flexible changes in the workplace and a higher intensity of profit-related growth by the move from a mass market to concentrating on a higher stratum of high-earning consumers. Generally, the shift is understood to have occurred in the 1960s and 1970s in response to various factors such as union activity, the high cost of the welfare state, strikes and the oil shock that led to a drop in competitiveness in the West.[4] These moves have increased the speed and reduced the cost of all types of commodities; this has produced a large surplus of goods and services. The push toward

'flexible production' and just-in-time manufacture has also led to changes in consumption and the growth of a new regime of accumulation based around information, communication and knowledge.[5] An example of new technology facilitating new forms of commodity and market conditions is that of the recent house market asset expansion across the world. This has been enabled by the growth in sophisticated hedging funds that diversify risk and the securitisation of mortgages (such as sub-prime loans) that use computer technology to handle the massive amount of mathematical and statistical calculation required to diversify risk. Unfortunately, when these assets turn 'toxic' when borrowers are unable to repay the loans, the computer-based calculations unwind and we see the dangers of technical complexity beyond the comprehension of any single actor, and the resultant panic and financial collapse of institutions such as Northern Rock in the UK and Bear Stearns in the US.

Within traditional information society theory, the US, Europe and Japan are usually identified as the exemplars of the information age (Witheford 1994: 88). These are identified as efficient, market-led and open to new ideas and creativity that allow them to move to new forms of production. Bell (1973), Brzezinski (1973) and Drucker (1968) were the first theorists of a new wave of economic development, variously titled a 'postindustrial', 'technetronic' or 'knowledge' society that also attempted to refute the claims of Marxism that capitalism was doomed to collapse due to its inherent contradictions. Instead, they argued that technology would offer solutions to capitalism's crisis and instead technological growth would lead to affluence and a stable economic system. Additionally, left critiques of traditional Marxism, such as Touraine's (1971) discussion of the 'programmed-society' also identified a new form of politics reflected in the growth of a computerised society through identity politics and new social movements.

Castells (2000b), exploring the rise of networks, has argued that we can see the emergence of an information society that is built around the growing importance of knowledge and information to the generation of profit. This is an important background to the

changing nature of production and the importance of intellectual property rights (IPRs).

> [There has been a] deepening of [the] capitalistic logic of profit seeking in capital-labour relationships; enhancing productivity of labour and capital; globalising production, circulation and markets, seizing the most advantageous conditions for profit making everywhere; and marshalling the state's support for productivity gains and competitiveness of national economies, often to the detriment of social protection and public interest regulations.
>
> (Castells 2000b: 19)

The information society remains a capitalist society, which Castells argues has grown from three major causes; the information technology revolution, the restructuring of capitalism in the 1980s and the long-term effect of social and political movements in the 1960s and 1970s. However, Castells stresses that the actual deployment of information technology is the result of conscious social action – not merely a deterministic result of technical change. Government action is important because capitalists and theorists have argued that these technical changes can be harnessed economically within national economies. Therefore, the establishing of robust intellectual property law becomes a key activity of the state in the information society, outlined as a crucial part of the 'Lisbon Strategy' in European Union policy documents (EU 2001, 2005).

The development of information and knowledge as important new economic resources differs from previous usages of information and knowledge that were embedded within the commodity. There has been a move away from the importance of material inputs (which previously were critical elements in production) to ideas and knowledge as contributing significant value to the product. But this information has also become dis-articulated from its carrier (that is, the commodity) and consequently has been accorded a separate value. Therefore value-added is increasingly reliant on non-material inputs into products and services – Apple Computer's reliance on branding being a notable example. Global flows of specialised analytical knowledge are becoming vital to

wealth creation, and consequently are increasingly influencing political affairs. But this is not just a technological or material change, it is manifested and supported through the operation of discursive structures, hegemonic economic imaginaries, that operate to render certain economic decisions more possible than others (Jessop 2004: 167–8).

Theory and research into the information society has been usefully categorised by Webster (1995: 6) into five analytical definitions: (1) technological, emphasising technological innovation and breakthroughs in information processing, storage and transmission; (2) economic, tracing the growth of information industries and their economic value; (3) occupational, which focuses on the changes in occupations within society when informational workers (such as symbolic workers or knowledge engineers) outnumber industrial workers (steelworkers, factory hands); (4) spatial, where the emphasis is on geographic space and the information networks which connect locations and their implications; and finally (5) cultural, where the effects of the growth in information impact our everyday lives. One of the key themes that links these differing definitions is the emptying of meaning from the concept of information, thus concentrating on quantitative measures of information growth to measure and explain the information society (Webster 1995: 28).

Additionally, May (2002) outlines a useful series of four claims that figure prominently in discourses of the information society, of which he is decidedly sceptical. He outlines them as: (1) claims to a social revolution, that is that the information society will usher in a new age broadly comparable to the agricultural and industrial revolutions; (2) claims for a new economy, whereby transformations in the workforce develops into a weightless economy; (3) claims for a new kind of information politics, usually based around the use of ICT to mobilise and perform political actions; and finally (4) claims for the decline of the state, whereby the challenge of technology compromises the ability of the state to intervene (such as through global flows of capital, multinational corporate actions etc). He uses these claims to analyse the way in which differing conceptions of the information society tend to draw on technologically deterministic arguments in order to

justify their predictions and prescriptions (May 2002: 12–17). As an alternative to this approach, he argues that empirical research tends to negate these claims and shows much more of a continuity with the previous historical development of society and less support for claims of a discontinuity or revolutionary break (May 2002: 19–21). This is an argument I have sympathy with, but nonetheless here I argue that these 'information society' claims and arguments have had a discursive or persuasive effect upon governments and corporations that has resulted in an increasing turn towards supporting an 'informationalisation' (Webster 1995) of the economy. These discourses are manifested in changes being made through government policy, legislation and the changing relationship of education, university research and private industry. Rather than a deterministic effect of the technology, I argue that an information-based society arises as a material formation through conscious political action and government commitment on the one hand and corporate activities on the other (see below).

The reasons why this movement towards building an information society has been undertaken are complex, including problems in staying competitive with other manufacturing locations, self-reinforcing discourses about an information society and the intensification and expansion of capitalism into new areas (such as into social life itself). The 'information society' can be used as a justificatory discourse to convince voters of the need for changes in society that may have been difficult to implement otherwise (see Fairclough (2000) on the language of New Labour). The disjuncture that the information society is supposed to represent indicates that something urgent needs to be done, and makes that sense of urgency a precondition for change and upheaval even though the empirical evidence for such change is scanty. This can be seen in the strong current of individual responsibility that runs through information society discourses about the 'portfolio worker' and knowledge analyst (May 2002: 51). This in turn leads to policy that justifies dismantling social provision such as welfare systems and societal insurance through pensions, and creating individualised private-sector services. This is justified in the name of competitiveness and the imagined needs of a new type

of worker who no longer wants to rely on the state or society and instead is responsible for her own provision. The discourses of a shrunken state associated with this move are largely illusory, and paradoxically create a regulatory state of equal size in its place (such as Blairite centralism).[6]

The claims of a disjuncture apparent in a new 'information' society have been legitimated through the actions of UN-sponsored summits such as the World Summit on the Information Society (WSIS) which has brought together intergovernmental, corporate and civil-society participants into a framework which has been used to justify legislative changes in response to a perceived information society. Proposals such as the EU Directive on the Information Society and the US National Information System have also added to the inevitability of legislative change. Within the EU the directive has been seen as the precursor of a new economy based around high technology, knowledge creation and creativity. Across the EU, national governments have been keen to implement their own information society plans, usually involving the extension and hardening of existing intellectual property legislation but also the informatisation of government itself through programmes such as e-democracy and e-government. These can have controversial unintended effects, as witnessed in the UK, in an online e-petition system set up by the government which had a politically embarrassing petition submitted and signed by 2 million people against government policy over road charging.[7]

The project of the information society is also being actively constructed through government and corporate planning and organisation and through the enrolment of other actors and networks (such as NGOs). Even though there is little empirical research to support the economic growth hypothesis behind this legislative activity, parliaments around the world are continuing to pass 'technical' legislation to facilitate its emergence. The ambitious proposals of information society policies of the developed nations seem intent on shifting the economic base of their societies and the reconfiguration of the world economy away from goods and services, towards a model built on information, knowledge and communication (Boyle 1996; Dyer-Witheford 1999; Hardt

and Negri 2000; May 2000; WSIS 2003).[8] This is policy-driven (that is, via endogenous or exogenous pressure on the state) and still a matter of debate. The exemplar of this process of policy-based change is the European Union's Lisbon Strategy, which proposes that Europe become 'the most competitive and dynamic knowledge-based economy in the world, capable of sustainable economic growth with more and better jobs and greater social cohesion' (quoted in European Council 2000; Barroso 2005).

These types of legal and political changes will have lasting and important consequences for the world economy. It is argued that the information economy will transform the global balance of power and the dominance by Western powers (Drahos and Braithwaite 2002); this kind of technological and economic determinism runs deeply through policy and economic reports on the subject and needs to be treated with some scepticism. But clearly, the ownership and control of knowledge, which is the condition of possibility for creativity and economic productivity, will undoubtedly have some affect on the global economy (May 2000). For instance, through the use of Trade-Related Aspects of Intellectual Property Rights (TRIPS) the US is ensuring that a global intellectual property regime is emerging. Postcolonial economies are being strong-armed into accepting the developed world's (that is, mainly the US's) intellectual property regime. This will generate huge profits for the owners of the intellectual property rights (Drahos and Braithwaite 2002). Yet, while these structures are being built, and although there are some important areas under analysis (Mansell 2004), there is little attempt to connect them up to a larger political economy which would help explain how shifts in IPRs are linked to changes in the way in which computerisation is producing new possibilities for the valourisation of labour.

It has been argued that the expansion of capital into the informational sphere has been intensifying in the last decade (Boyle 1996; Coombe 1998; Drahos and Braithwaite 2002; Lessig 1999, 2004; Litman 2001; Vaidhyanathan 2001) in the North partly due to manufacturing flight to low-cost countries (and hence a governmental concern with informational added value

to occupy the potential masses of unemployed), but also due to the imminent collapse of many multinational media corporations' 'inventories' (see Bassett 2004: 352–4)[9] as they pass into the public domain. Together with the expansion of technology, especially into the realm of personal consumption, the synergetic connection of content and technology has resulted in a dual fear within some sections of the corporate world – of (1) dangers of uncontrolled (and unprofitable) digital copyright infringement and the resultant collapse in profit and (2) the sudden realisation that technologically enabled delivery of cultural products could actually result in massive profits, provided that sufficient legal redress is available. Both of these lead to a claim by corporations for the reconfiguration of intellectual property rights from a limited right granted as a bargain between public and private to that of a fully alienable property right (fully identical to physical property rights).

Together with a discourse of modernity and progress that surrounds the continued informatisation of society, there is also an attendant conception of greater risk. The discourse of risk begins where trust in security and the belief in progress end. Greater reliance on technology and a corresponding greater reliance on statistics and technology to generate justifications for decision-making rely on a 'mathematised morality' (Beck 2000). Thus risk analysis becomes an increasing component in our society, as decisions have to be taken regarding the dangers of greater reliance on technology and information (for example, computer-controlled nuclear plants, CCTV systems, or automatic defence systems). Risks, however, are not the same as destruction; they are statistical probabilities that may threaten destruction or disaster, and so should be understood in actuarial terms, similar to the way in which insurance companies estimate risk. Technology and science offer a strong rationalised discourse of predictability and empirically tested decisions that are objective and value-free, but this is itself value-laden, a point that is often forgotten by those charged with assembling technology analysis and risk analysis (Slovic 2000: 42). Additionally, there is the danger identified by Beck (2000) that risk becomes a concept that reverses the relationships between past, present and future and we begin to

determine action based on what 'could happen'. At this moment a risk society becomes 'reflexive', in other words, the foundations for its activity and its objectives become the object of scientific and political controversies. Politics becomes a risk-assessment exercise that is in danger of becoming technocratic and expert-led, rather than democratic. Here are the beginnings of the reconceptualisation of the citizen as a 'creative citizen', oriented towards goals and projects that are linked to economic and technical efficiencies (profit),[10] able to weigh the various risk assessments and act to individually manage their own life history in light of the risk that is threatened (through, for example, lifelong learning, personal provision of insurance and the like). For example, a contemporary dominant discourse is that of the risk of pension failure and the transfer from the state of the responsibilities for managing this risk to the individual. Yet this is a transfer that could only have taken place with the requisite technologies in place to allow individualised management of pension accounts, and the corresponding informatised environment that allows real-time feedback on pension accounts (and stock-holdings) and sophisticated actuarial and statistical analysis of risk.

Thus the more general challenge for governments and economies in the North who are attempting to shape themselves to an informational base is to develop a framework that fosters the growth of an economy increasingly dominated by conceptual products (Greenspan 2003). These processes are being driven by corporate lobbying, especially by media corporations, some of whom have huge quantities of copyrighted works that are on the verge of passing out of copyright, by a rhetoric of globalisation, post-industrial promises of wealth and job creation and the worry caused by large-scale deindustrialisation, outsourcing and transfer of industrial manufacturing capacity. Additionally, there has been a new global intellectual property system created through the World Intellectual Property Organisation (WIPO) and the Uruguay round of international agreements (now the World Trade Organisation) together with the pressure exerted through the TRIPS agreement, in effect forcing recalcitrant nations to strengthen their intellectual property law protections or face

serious trade-based consequences (Drahos and Braithwaite 2002). It is argued that these structural and political changes are taking place globally and that states are attempting to construct an information economy that can support a market-based economic system, a hegemonic tendency within late capitalism (Hardt and Negri 2000: 291).

Led by the US and the European Union (net exporters of intellectual property) there is an attempt to reposition the 'post-industrial' world as a producer (and owner) of information products. These institutional factors can be traced through institutional policy documents as early as 1999 with the Declaration on the Information Society (EU 2005b). Indeed, the European Union explicitly links information communication technology (ICT) to the question of citizenship, stating that 'ICT provides a boost to citizenship and to the quality of life' (EU 2005: 4).

The debates around the constitutive role of intellectual property laws in commercial and popular culture and its monopoly on the use and reuse of meaning, interpretation and transformation is increasingly excluded from the political sphere and articulated through the market. Yet in recent decades the fraction of the total output of our economy that is essentially conceptual (that is, immaterial) rather than physical or material has been steadily rising (Greenspan 2003). This trend has, of necessity, shifted the emphasis in asset valuation from the ownership of physical property to the control and ownership of intellectual property and to the legal rights that inhere in the latter. Though the shift may appear glacial, in fact, its impact on economics, on law, on culture and on democracy over the past 30 years has been remarkable and in the twenty-first century is intensifying and speeding up.

Many of the legal changes that have facilitated increased intellectual property laws have been accepted largely without empirical evidence or with any concerted research as to the consequences of large-scale expansion of the intellectual property regime (it is often in fact passed as 'technical' legislation with little or no debate, as is the case in the UK). The consequences for democracy and creativity have largely been ignored in the

rapid enclosure of ideas and expressions that has intensified in the past decade. With a world economic system being created on the basis of a global monopoly on property rights in intellectual and immaterial production, nations and corporations are increasingly focusing on the collection of intellectual property portfolios and securing rights in knowledge. These monopoly rights could have major economic and democratic consequences and some writers have warned of a new feudalism and the emergence of a rentier class who own the rights to all access, use and exploitation of ideas and expressions (Drahos and Braithwaite 2002). Others have warned of a world of 'sources' and 'authors', where the ownership and control of ideas and concepts are allocated on the basis of extremely problematic concepts of creativity, authorship and originality (Boyle 1996).

The form of social and cultural power that law increasingly grants to the holders of copyright, patents, trademarks and design rights, the commodification of citizenship and the public sphere and the molecular challenges to this regime through the activities of the free software and open source movements are the key reference points in the chapters that follow.

I have attempted to show that the 'information society' is at once imaginary, but that this imaginary discursive formation has real effects and has had effects in the past. As the claims of the information society theorists become instantiated within intellectual property laws, employment laws and so forth, these then change the behaviour of companies, employees and other actors in the economy. We only need to look at the way in which patent applications from universities, particularly in the US, have generated income streams that have changed research behaviour (sometimes called 'third-stream activities'). They have also made companies increasingly wary of universities as complementary sites of research and development and instead treat them as competitors. Within the realm of computer software, the ability to copyright and patent software invention is leading to a new focus on proprietary ownership and sub-licensing of novel algorithms and of course to an increasingly legalistic software development environment.

Contemporary knowledge and creativity

In this section, I want to look at the changes taking place under the term the 'information society' though an understanding of the juncture of the antagonism between capital and labour, in the interstices that mark the privatisation of the common and the social (Hardt and Negri 2004: 148–9). This highlights the importance of keeping in mind the fact that labour is a necessary and important factor in the production of information, knowledge and communications technologies. Additionally, it points to the importance of an ability to alienate this informational production and be able to reinforce principles of scarcity and ownership in order to construct and support a market. This is where intellectual property rights become critically important. However, as with many of the themes that run through this book, the subject of changes in relations of capital and labour warrant a complete research project in its own right and the quantity of theory associated with it is huge albeit very interesting (for a good overview I refer the reader particularly to the work of Hardt and Negri 2000, 2004; Virno 2004; Lazzarato 1996; Jameson 1992; Dyer-Witheford 1999). However, here I can give only a broad overview of some of the key issues and questions that are raised.

To help distinguish new modes of alienation of labour-produced goods, I use the analytical category of *immaterial goods*, that is, the form of goods and services within an informational milieu. This category is aimed at helping to distinguish a particularly informational mode of exploitation, whereby through the use of enclosing techniques, including IPRs and digital locks, the ownership and control of information can be organised and manipulated. The concept of immaterial goods raises questions about a new class of ownership that is supported through the monopolisation of information, data or knowledge (Drahos and Braithwaite (2002) talk about a new feudalism). This is not to introduce a notion of a 'power-elite' that is single-handedly driving the information society. Rather, understanding this new class of ownership in terms of immaterial production allows me to separate informational modes of capitalist exploitation from the agricultural

and industrial and focus on the challenges and problems facing those who wish to extract profit from informational goods and services, a process that might be different from that which takes place in an era of scarce resources. For example, the cost of digital reproduction of a file is very low (it has a low marginal cost) and is non-exclusive (that is, more than one person may hold a copy of it) which means that, unlike physical goods, infinite numbers of copies may be made of a digital book, without diminishing or degrading the original. This presents very different challenges for a corporation operating in the informational world, compared to those faced by a corporation operating in the scarcity-ridden physical world.[11]

Increasingly, companies are moving from operating in an environment where they assert permanent property rights over scarce material and compete on price to one where resources are potentially infinite, protected by limited intellectual property rights and where they do not compete primarily on price. This opens the door to the possibility of a new kind of economy, a biopolitical economy (Hardt and Negri 2004), or the imposition of the old economy onto the new through the introduction of strong intellectual property rights (Howkins 2001: 123–4). But both types of economy are predicated on the tension introduced by the private gain versus public good, in other words, the contradiction between intellectual property and democratic freedom.

This also raises questions about the problems caused by a fragmentation of ownership of information, such that the ability to bring the owners of immaterial goods together to sell, transform or produce more goods becomes impossible, a tragedy of the anti-commons (Heller 1998; Hunter 2003). But even if companies or the owners of immaterial goods do not agree with the increase in intellectual property laws, the changes made in the legal and economic environment are creating a logic that cannot be ignored, as Bill Gates bluntly explained:

> If people had understood how patents would be granted when most of today's ideas were invented and had taken out patents, the [computer] industry would be at a complete standstill today ... The solution is patenting

> as much as we can. A future startup with no patents of its own will be forced to pay whatever price the giants choose to impose. That price might be high. Established companies have an interest in excluding future competitors.
> (quoted in Lessig 2002a: 2)

Non-material inputs (such as design, marketing, advertising, quality-control and technical ideas), or what Marx called 'universal labour', are now the subject of furious competition amongst states and corporations. The priority placed on the ability to control, direct and profit from the flows of vast amounts of information and knowledge is changing the economic organisation of states by making the monopoly ownership of information, processes and knowledge a source of profit in its own right and an increasingly important one. Information has always been important to capitalism, of course, but the relative weight of the ownership of information is increasing as the potential for monopoly control of its creation, dissemination and use has been strengthened. The control and codification of knowledge has become increasingly subject to the attention of capitalists as '[a]ccordingly, while the weight of current economic output is probably only modestly higher than it was a half-century ago, value added [i.e. knowledge and non-material additions], adjusted for price change, has risen well over threefold' (Greenspan, quoted in Perelman 2003).

These questions are not new, but they point to issues over the changing structure of creativity referred to earlier, and how it is being configured to serve as a locus for profit in late capitalism. Certainly, contemporary management theory, to take one extreme example, has taken up creativity within its discourse and as a source of potential profit with great excitement (see Prichard 2001). It is argued that creativity has become the key source of value in late capitalism, which draws on the source of value that can be extracted from living labour. That is, the productive labour to form immaterial objects through intellectual and affective endeavour is a growing and important source of value in capitalism (see Florida 2004). This is also becoming an increasing source of conflict between the need for capital to develop the

tools to expropriate social production and the possibility of living within the social itself (Virno 2004: 35–7).[12]

In parallel, there has been a corresponding attempt to legitimise the ownership and control of knowledge and information. Heilbronner comments that 'much of what is called "growth" in capitalist societies consists in this commodification of life ... the continuous search of business for areas of social activity that can be subsumed within the capital-generating circuit' (quoted in May 2000: 28–9). But, the deployment of laws and norms that legitimise informational ownership requires conscious social and political action resulting in institutional change (Dyer-Witheford 1999; May 2000). This leaves traces that can be critically examined and opens the possibility of political contestation and debate (Barber 1984; Feenberg and Hannay 1995).

The expansion of capital into information, knowledge and communicational arenas raises particular questions about the extent to which capitalism intervenes in our social life to a different degree than in previous modes of production. This is where the concept of the 'social factory' gives a useful analytical means of discussing the way in which certain common or shared parts of our social existence are currently being targeted for profit. It is particularly interesting in relation to the emergence of unpaid free labour, in the form of digital workers who through their use of an online site or resource contribute towards a final product, examples being Wikipedia, the free online encyclopaedia, and AOL chatrooms (Terranova 2004: 73–5) which are populated by people discussing their interests. It is also an important category in helping to understand the turn to Web 2.0 and the emergence of the commodification of online social life, such as through the online game SecondLife[13] or through web social networking sites such as MySpace[14] and Facebook[15], although I note this began with Web 1.0, or actually with the very early, pre-web online spaces (see Hardt and Negri 2000: 289–97).

By seeking to extend property rights to intellectual artefacts (immaterial products) and social relationships (business processes and methods and the like) these interests are strengthening and extending the concept of informational or immaterial proprietor-

ship (May 2000; Bettig 1996; Boyle 1996). The intensification of ownership of the cultural and intellectual sphere is justified with reference to a number of justificatory schema, such as self-expression, 'natural' rights or economic necessity. Historically, intellectual property rights such as copyright, trademarks, patents, design rights and so on have been subject to a public goods balance, both in civil and common law traditions, and in the political sphere at least in the UK and the US. In any democratic capitalist society, the tension between free access to information, and an economic right to exploit informational know-how or cultural products has been a shifting and contested area of ownership and control (Jefferson 1999; Habermas 1992; Dyer-Witheford 1999). However, the degree to which ownership of the immaterial has expanded in recent times is unprecedented – from the ownership and control of cultural texts (text, images, film and music) to the human genome and even life itself (e.g. the OncoMouse, a patented life-form under US law). These discourses of rights have correspondingly shifted from a balance between a public interest in the availability of knowledge and private reward for creativity – through a limited right to exploit the creation of cultural expression and knowledge – to a discourse of a fully alienable property right analogous to physical property, that which May (2002: 21) identifies as the move from property rights of 'holding' to that of 'withholding'; in other words, seeking the rights to control and restrict use of the immaterial in perpetuity.

Having examined some of the justifications and theories about the possibility of 'creative' labour, I want to turn to the software industry (which itself widely uses intellectual property laws) and examine the ways in which this industry was directly shaped by, and in turn shaped, the growth in value of intellectual property artefacts, such as software, and how the technologies developed have come to transform wide sections of the economy.

A Brief History of the Software Industry

Rapid technological developments in computer technology grew out of a perceived need by US President Eisenhower to have an

alternative technological development path to corporate and military research programmes, which would be civilian/scientist managed, would allow blue-sky research to be undertaken and would encourage the vibrancy of basic and applied research in technology (Naughton 2005: 79). The so-called military-industrial complex that Eisenhower warned against in 1961 showed the close links that lay between military procurement and industrial suppliers and subcontractors. This was the reason that the Advanced Research Projects Agency (ARPA) was formed in 1958, immediately after the threat of a technological lead demonstrated by the Soviet Union with its launch of Sputnik I and II. ARPA began with a budget of $2 billion and a staff of around 70 and was initially given direction over all US space programmes and advanced strategic missile research (Hafner and Lyon 1998: 21). By the end of the year, though, the National Aeronautics and Space Administration (NASA) had been formed and space and missile projects were transferred over from ARPA. This left ARPA with a $150 million budget and no clear future. However, soon it had realigned its mission with a new basic research and special-project direction that would allow it to work with universities, which had not previously directly collaborated on projects, in order to develop high-risk, cutting-edge research (Hafner and Lyon 1998: 21; Markoff 2005: 50–1). Thus the military was able directly to influence and tap into the research work at top universities across the US.

As the technologies were so new, and the requirements of the military were locked into a command-and-control mentality, ARPA was able to concentrate on research outside of this sphere, such as computer networking or human augmentation projects (human-computer interaction) (Hafner and Lyon 1998: 40–2).

ARPA's projects developed in the highly politically charged years of the 1960s, with public disquiet over the Vietnam war, the emergent sixties counterculture, the civil rights movement and the Berkeley Free Speech Movement, all of which influenced many of the key actors involved in computer design.[16] This pulled the researchers in contradictory directions, as much of the work was classified and funded by the US government and military even

as the hackers and technicians were 'turning-on' to experiments with psychedelic drugs such as LSD and the counterculture. Remarkably, out of this strange mixture of ideas and experiences emerged many of the critical breakthroughs in technology that went on to drive the computer industry over the next 30 years, including multi-user operating systems, video games, multiple overlapping windows, the first mouse, a screen pointer (or cursor), video-conferencing technology, hyperlinks and cut-and-paste (Markoff 2005: 174–85).

Following historians of the software industry, here I concentrate on the early history of software from 1969 to the present day (Campbell-Kelly 2004: 109–14; Lowood 2001). The computer historian Paul Edwards refers to the tradition of writings about the development of the computer pre-1969 as based around 'machine calculation' (computer hardware) which differs from that post-1969 where 'machine logic' (software) became increasingly dominant (quoted in Lowood 2001: 146). In this section and throughout this book, the focus will be on the 'machine logic' and the developments and challenges that the software industry has faced in particular.

The choice of 1969 as a key date for a disjuncture defining a new stage in the history of computer technology follows the announcement of IBM in June 1969 that it was 'unbundling' supplied software from the sale or lease of its corporate computer systems. There is a strong argument that before 1967 there was not a software industry as such, as most software was either supplied 'free' as part of the purchasing or leasing package, or else was written to order by software contracting professionals (Campbell-Kelly 2004: 29).

In 1967 the antitrust division of the US Department of Justice started an investigation into IBM and its 70 per cent control of the domestic computer market. IBM had stood accused of 'bundling', which was defined in court as 'the offering of a number of elements that are considered to be interrelated and necessary from a customer's point of view, in the computer field, under a single pricing plan, without detailing the pricing of the component elements themselves' (Campbell-Kelly 2004: 109). This, it was

argued, allowed IBM to undercut other suppliers of software and services in order to win orders. In other words, IBM was selling goods below cost price – a direct antitrust violation. IBM announced in December 1968 that it intended to unbundle its prices and charge separately for different parts of the supplied system and by the end of June 1969 had put in place these procedures. IBM itself never admitted any connection between the antitrust action and its decision to unbundle, but the end result was the creation of a completely new market for software that could be installed and run on IBM computers.

Thus, during the 1970s there was rapid change in the business, culture and technologies associated with the software industry, together with rapid growth in the value and size of the software market. At about this time, the term software engineer came into widespread usage and computer science departments began to focus on the practical problems involved in the implementation of software projects from a technical point of view. This was a transformation from a predominantly hardware-focused industry (where the software was supplied 'free' and poorly supported) to one where the requirements and needs of software changed rapidly to fit a new type of software-buying computer owner.

The unbundling has been described as the 'crucial inflection point', as the value of software products expanded from around $20 million in 1969 to $400 million by 1975 (Campbell-Kelly 2004: 114). Although its effects are largely difficult to discern because it was overtaken by the collapse in software stocks in 1970 and the resulting computer recession of 1970–71, it remains an important decision in helping to cement the foundations of a software products market. The 1970s saw the creation of many of the hallmarks in the software products industry that are still with us, such as the notion of a 'captive operating system' which allowed the monopoly platform to be maximised for a firm's own software products and profitability (something that IBM was to move away from in the late 1970s, while it was still under antitrust investigation). It also is an interesting precursor of the desktop monopoly of Microsoft that also resulted in an antitrust action by the US government.

The 1970s also saw the beginnings of the Unix operating system that was written by Ken Thompson and Dennis Ritchie at Bell Labs. Unix was originally conceived of as a simple, reliable operating system for a single user as opposed to the 'software disaster' of MULTICS[17] (Multiplexed Information and Computing Service), which conceptually was path-breaking but in practice slow to develop and iron out problems (Campbell-Kelly 2004: 144). Thompson and Ritchie worked in a research culture that was partly academic and partly corporate and so they happily made the source code freely available to fellow scholars. This openness to the sharing of their labours meant that universities in particular were very keen on the use of Unix in their computer systems. The history of Unix is well documented[18] so will be only glossed here, but needless to say its technologies and the open development processes formed an important precursor to the free Unix clone operating systems which were to develop into Free BSD and GNU/Linux.

The 1970s were also the decade in which the ARPANET project began as part of DARPA's experimental network of networks. It was under this project that the crucial protocols of what was later to be called the Internet were designed and implemented as TCP/IP (Weber 2004: 33). It was also here that Unix came under greater scrutiny as it was adopted as a common platform, in particular being open for others to view and change the source code, which allowed researchers in academic and business environments to experiment with improving the code. It also allowed the project to avoid the huge costs associated with using a proprietary platform, particularly DEC's VAX VMS, which could lock in users and programmers and could be difficult to transfer software from (Weber 2004: 36).

Following the software crash of the 1970s, it took nearly a decade for the software industry to begin to grow again. This time, however, software was entering a new phase so that it was claimed that by the mid-1980s 'few users write their own software anymore' (Campbell-Kelly 2004: 166) due to the rising complexity of computer systems and the specialisation of the role of the computer programmer. The United States dominated the software

industry with over two-thirds of the world's software market and 95 per cent of the software products supplied. By 1982, IBM was much mocked for the bug-ridden and slow software that it was supplying as 'bundles' of inter-working component software products. Nonetheless, IBM still dominated the server software market, although it never produced good applications software itself and so it continued to sell other software products.[19]

Meanwhile, in 1971 Intel Corporation introduced the integrated microprocessor. This chip contained all the necessary components for a computer processor on a single chip and was capable of functioning as a stand-alone computer. In contrast to the large-scale corporate focus of IBM's computer line, this processor was initially developed for the embedded computer market, controlling small specific applications and machinery; hence IBM and many of the other major players in the corporate software products industry ignored the fledgling personal computer industry at first. The first microprocessor-based computer was the Altair 8800, manufactured by Micro Instrumentation Telemetry Systems (MITS). This computer was supplied and sold as a kit for assembly by computer hobbyists and priced at $400. Hundreds were sold in the first six months. Although this self-assembly computer market was very limited, such as members of the infamous Homebrew Computer Club[20] which started in 1975, some of the people developing software for these machines went on to form major software companies in their own right; they included Steve Jobs[21], Bill Gates, Paul Allen[22] and Gary Kildall[23].

The major event for the personal computer *industry* and one viewed as launching it was the introduction of the Apple II computer in April 1977. Steve Jobs and Steve Wozniak had formed Apple Computer in 1976 and their first machine, the Apple, had been a raw computer board for hobbyists. The Apple II, however, was a complete computer system, with keyboard, a monitor and the CPU all in one package, ready assembled. It encouraged the launch of the Commodore PET, the Tandy TRS-80 and others. Although much loved by home users and computer hobbyists, these machines did not at first seem to be business machines, not least because corporate processing was already handled by large

mainframe and mini computer systems and organisation structures had created separate data-processing departments.

In 1980, IBM had decided to enter the PC market but in order to avoid the company's slow development processes it decided to outsource nearly all the components and subsystems, including hardware and software. This was to be a decision with far-reaching consequences for the software market as the IBM PC had been designed as an 'open' platform to encourage non-IBM suppliers to create add-on hardware and software. But it also had the unforeseen consequence of unleashing a massive IBM-clone market. Microsoft, having cleverly signed a non-exclusive contract to supply the MS-DOS operating system[24] to IBM, was thus able to sell licences for this software to any hardware manufacturer.

The years from 1979 to 1983 set the scene for thousands of new computer software firms such as Digital Research, Broderbund and Lotus to write applications software for a new type of computer user. However before this software gold rush could begin there needed to be a stable operating system platform and the programming tools and languages to create the software in the first place. Microsoft and Digital Research were key companies in the development of this early software and they saw clearly the potential of the personal computer long before the larger mainframe manufacturers noticed or could react to this new market.

The history of Microsoft has now passed into computer folklore so I will only summarise the key elements that played into the company's success and leave the reader to examine the more detailed accounts elsewhere[25]. The most widely known moment in Microsoft's success was the deal it struck with IBM (over and above its competitor Digital Research) to supply the operating system for the IBM PC computer.

By 1983, nearly a million IBM-compatible PCs had been sold and Microsoft had succeeded in capturing 90 per cent of this market, giving it sales of $70 million a year. The launch of the IBM PC allowed the creation of new types of software, most notably the spreadsheet, Visicalc[26], which is often considered to be the 'killer app' that unleashed the personal computer

industry. The 'killer app' hypothesis argues that a new and novel application that changes an activity in an innovative way, or allows something to be done that was previously impossible causes that new technology to be widely adopted (Campbell-Kelly 2004: 212).[27] With the launch of Visicalc, a software package that reconfigured the way in which people thought about computers as tools of business, many businesspeople now thought seriously about purchasing and using one in their everyday lives. Two other examples of 'killer apps' include the adoption of email (usually credited with popularising the Internet) and the Web browser (credited with enabling Internet commerce). Although an appealing way of explaining the way in which technology is taken up and popularised it is a simplification of the processes involved in technology. Technology usually goes under a number of complex and contradictory developments which may or may not become successful due to political, economic or social pressures (Bijker, Hughes and Pinch 2001), not to mention the lack of technical support or the immaturity of the technology. For example the emergence of WIMPs (Windows, Icons, Mouse, Pointer) operating systems is often thought to have started with the 'revolutionary' arrival of the Apple Macintosh[28], in 1984, or perhaps Windows 95, but actually was based on a previous research project run by Xerox. Windows itself went through a number of very unsuccessful iterations before it could claim to be a success and was taken up by users.

From 1983 to 1995, the computer software industry went through a period of maturation that is sometimes skewed in interpretation by an unwarranted focus on Microsoft – in fact there are more books about Microsoft than about the rest of the software industry combined. Partly this is due to the wealth of the founders, particularly Bill Gates, but also due to the company's aggressiveness and the profits that Microsoft made from its sale of computer software. Often, in fact, Microsoft is seen as a latter-day IBM, monopolising the computer software industry and having a huge amount of control over it. Nonetheless, to put Microsoft in perspective, IBM in the 1960s controlled 75 per cent of the computer industry worldwide, whereas Microsoft has

never controlled more than 10 per cent of the computer software industry. So although by 1990 Microsoft was arguably the best-known software firm, its sales of $1.18 billion represented only 3 per cent of the $35 billion worldwide market for computer software products, and this was still only one eighth of IBM's software sales of $9.95 billion (Campbell-Kelly 2004: 232). Only in 1998 did Microsoft's software sales exceed those of IBM's, and even though it was valued as the most valuable company in the world in 1999 (at $84.4 billion) its total revenues continued to be dwarfed by IBM's, which were $19.7 billion. It is only within the narrow sector of the personal computer software market that Microsoft has anything like the dominance that is often attributed to it. According to the market research company OneStat.com, Microsoft Windows now has 97.4 per cent of the global desktop operating system market, compared to just 1.4 per cent for the Apple Macintosh and 0.3 per cent for GNU/Linux.[29]

During the 1970s and 1980s there were, in parallel, huge changes in the market for games, video games and consoles. Although starting from simple foundations, such as Pong (a tennis game), these were an important precursor to introducing a wider public to computer technology and proved an important social driver of technology. By 1982 the video game market was worth up to $5 billion and extremely profitable, but following saturation the market collapsed in 1983. In 1986 Nintendo almost single-handedly kick-started the video-console market with the NES. Today video games are considered an important part of the software market (although previously games were considered consumer goods or entertainment and never figured in statistics about the computer software industry, which was considered business-related). Today we have a situation where many of the key players in the business sector now see the future as multimedia and home-centred: both Apple and Microsoft are aiming to take a share of the 'home entertainment' market, Microsoft with its Xbox game console and Apple through a focus on personal lifestyle software and the recently launched 'Apple TV' hardware and software appliance. However, Microsoft, at least, remains focused on business-to-business applications software and has recently launched its latest

operating system, Vista, which is intended to cement its control of the business desktop (through Vista Ultimate) whilst also remaining relevant in the home (Vista Home Edition).

Today, most of the focus is on the Internet and mobile platforms, particularly the way in which they too might be commercialised. After the dotcom boom of the 1990s, which drew on the hype surrounding then little understood websites, today we are on the cusp of the Web 2.0 boom (represented by YouTube, Facebook and the other social networking sites). This brings to the fore a focus on content as well as software, and ironically it is enabled through the kinds of sharing and intertextual interchange of files that started the computer industry in the first place. The computer industry can be seen as a historical movement from power manifested within hardware (the hardware platform), moving to a period in which software became dominant (the software platform) and now to one in which software *and content* are licensed to enable the networking and remixing to take place (the network platform). Many of the key Web 2.0 technologies are built on free/libre and open source software (FLOSS) technologies and also utilise content that is licensed in a similar way (for example, Creative Commons licences). In the next section I will bring this forward by looking at intellectual property and how it has shaped and is being shaped by changes in technology and the wider economic possibilities in licensing.

Intellectual Property Rights in the Digital Age

The 1709 British Statute of Anne marked the beginning of the modern concept of copyright that accorded exclusive rights to authors and their publishers[30] (Litman 2001: 15). By law the duration was limited to 28 years, initially 14 years and then extendable by another 14. Once that period ended, the work would pass into the public domain, and would be made available for citizens freely to draw on, publish or distribute without copyright restrictions (except that the entire work could not be re-copyrighted without a transformation having taken place). Previously, governments in Europe had granted monopoly rights

to publishers, mainly in order to censor through licensing the rights to print books. When considered as part of the history of intellectual property, printing is arguably one of the most significant technical advances in history, transforming the spread of knowledge, and introducing the possibility of non-physical forms of property (such as the text, play or sheet-music). Gutenberg's development of movable type caused a revolution in the process of book production (McLuhan 1967). Up until this point, books were copied by hand and reproduction was only possible through a large investment of materials and physical labour. With the ability to reproduce large quantities of texts and no control over its distribution, publishers soon began to worry about others reprinting or copying their books. It was also assumed that the publishers owned printed works in perpetuity; this was a source of constant complaint by Enlightenment philosophers about the restrictions on the flow of knowledge and the temptation of book publishers to set their prices as high as possible.

Copyright was intended to guarantee the profits of the stationers and publishers. It was thought that the initial 14 + 14 years would give enough incentive to publish but avoid a perpetual monopoly. 'The Statute of Anne was an elaborate attempt to regulate publishers, a way to balance the interests of the book-publishing industry with the concerns that monopolies were growing too powerful in England' (Vaidhyanathan 2001: 40). The British justification for copyright legislation was fundamentally an anti-monopoly position, balancing competition to bring down the price of classics and the like, and the legislation was clearly in the interests of a particular reading public who would buy these books, rather than a concern necessarily with any democratic project or suchlike. Of course, the Enlightenment justification for the universal education and betterment of all men was invoked, but this seemed to be distilled through a particular aristocratic understanding of the role of education and knowledge, rather than a perceived need for democratic education, perhaps reflecting the particular class views of the ruling aristocracy. In the British parliament, for example, during the debates there was much derision over the status of mere 'scribblers' against that of

the disinterested aristocrats who published out of an aesthetic sensibility. Indeed, to counter these arguments petitioners used Locke's arguments (Vaidhyanathan 2001: 42) about the rights of producers who should be rewarded for their labour, which carried some rhetorical force. This is perhaps why the concept of the 'author' became such a useful tool for the publishing industry to hide behind even as it was lobbying to prevent copyright becoming overly focused on the author. As Thomas Babington Macaulay argued in 1841 to the British parliament:

> Copyright is monopoly, and produces all the effects which the general voice of mankind attributes to monopoly ... the effect of monopoly generally is to make articles scarce, to make them dear, and to make them bad ... It is good that authors should be remunerated; and the least exceptionable way of remunerating them is by a monopoly. Yet monopoly is an evil. For the sake of the good we must submit to the evil; but the evil ought not to last a day longer than is necessary for the purpose of securing the good.
>
> (Macaulay, quoted in Boyle 2003)

Once the period of copyright ends, the work enters the public domain and becomes freely available for anyone to use and draw ideas from. The public domain becomes an increasingly valuable but little appreciated source of inspiration and material. Folk music and blues are perhaps the most obvious sources of creativity that unashamedly reuse old songs in new ways. The musicians who wrote this type of music never gave much thought to 'locking up' their works and restricting their reproduction; indeed it would have undermined their ability to write, as such music is collaborative and oral. Today, however, music has become a vast industry, and as the profit from the ownership of songs and recordings has enriched musicians and labels, so sadly have the ethos and values associated with making music changed accordingly. We see this in the examples of rock musicians, who having happily drawn on this cultural commons to produce their own work in the 1950s and 1960s, proceeded to take legal action against proponents of new forms of music such as hip-hop, rapping, sampling and mashing in the 1980s and 1990s. This

change in mentality has seemingly occurred wherever creativity that can make money and increasingly today art, music and design are seen in terms of their monetary value. Here the technology of *storage* and reproduction are crucial to understanding how changes in technology can create a new market in works (such as recorded music) and where lobbying acts to broaden copyright and other intellectual property rights.[31]

Nowadays it is hard to imagine a time before copyright and copyright restrictions. Indeed, copyright is so wrapped up in the language of 'property' that we often find it difficult to distinguish between physical property and the rights accorded by intellectual property law. Many people confuse copyright with physical property (Drahos and Braithwaite 2002: 61–2). Although it is referred to as an intellectual property right, it is not actually property in the same sense as the ownership of a house or car. This is because essentially copyright is a right to copy the *expression* of an idea rather than an unlimited property right – a copy-*right* (Litman 2001: 15–21). The reason for the distinction between copyright and property rights in general is that after the work has been put into producing a piece of artwork or a manuscript it can be infinitely reproduced at little extra cost. This is very different to physical property, which will slowly wear out. Additionally, if I give you a copy of the work, it does not diminish my use of the artwork; therefore we can all have a copy without anybody losing out. Again, contrasted with physical property, if I own a car, only I can drive it – naturally only one person at a time can use that car (Boyle 1996: 18–24).

Essentially, copyright is understood as a monopoly, a bundle of rights that applies to the 'expression' of an idea (Bettig 1996). It establishes the author as the creator of an intellectual work and creates exclusive legal rights for the author to control derivatives, duplication, performance or distribution of their creative works (May 2000: 8–9). So, for example, the writer of a book can transfer the specific narrative of a novel into a written manuscript. Copyright protects this manuscript; it does not protect the 'idea'

or general plot. Similarly, for a designer, the artwork itself would be protected under copyright which confers rights over its use. The right of alienation of the copyright allows the owner of the work to transfer the copyright to a third party (which enables artists to sell their works to corporations who can manufacture and sell it). Copyright is one of a number of intellectual property rights, which also include patents, trademarks and design rights, that allow the creator to exploit the work by licensing others to use it (May 2000: 6–11). Unlike a patent (which requires patent registration in a patent office), the moment a work is created, it automatically becomes copyrighted. Examples of such works include literary pieces, drawings, paintings, photographs, film, music and sound recordings and, more recently, computer software programs.

Copyright and other intellectual property rights are often defended using claims for the superiority of the 'rationality' of private property by using an economic model of the market. This justifies private property by the claim that only by allocating value to a particular resource can it efficiently be used and its use maximised (May 2000: 13). By fostering progress in economic organisation and increasing efficiency, it is argued that society as a whole will benefit from increased wealth and greater quantities of culture and information. Allocating a price in the market means that users are constantly required to assess the return on their use of a particular informational product and think about how it can be maximised. The argument is that through a market mechanism, a more efficient use of 'creative' resources as well as innovations is possible. Scarcity is critical to the operation of markets and property law has been shaped by and indeed can function only due to the fact that scarcity is a hallmark of physical property. For example, if you are eating an apple, it is impossible for me to eat it too. This allows markets to operate on the basis of exchange of limited amounts of goods that are priced according to the 'laws' of supply and demand (Boyle 1996: 6; Greenspan 2003; Olson 1971). However, as we move into the so-called 'information society', it soon becomes apparent that intellectual property does not operate in the same way that physical property does. In fact, the idea of

scarcity or the fact that information can have only one owner does not necessarily apply. Without scarcity and without any 'wearing out' or consumption of the works, it is argued that it becomes difficult, if not impossible, for the market to function rationally (Vaidhyanathan 2004; Castells 2000; Hardt and Negri 2000).

Claims to intellectual property involve categorisation of both an object, invented or created, and a subject, the owner, inventor or author. They imply both a connection and distinction between the two. Those who make claims to intellectual property must demonstrate a point of origin, yet it must be possible to use and to replicate the object in other places. Indeed, objects can be exchanged and treated as property and in some cases as a commodity. But when it comes to creative work, claims to intellectual property become contestable and problematic. For these types of objects it is also necessary to create new subjects of rights; thus the classification of the object and the subject become vitally important. How is it possible to deal with the contradiction between the economic and social justification for intellectual property – namely that it is a public good – with the moral justification that authors have 'natural rights' to the products of their creative activity? For example, in software, is writing code a form of authorship (that is, something created) or does it involve design and invention? How is it possible to define subject rights in the division of labour that exists in the creation of computer code?

Thus legal limits on immaterial property have traditionally been limited compared to physical property. In the case of intellectual property rights, legal benefits include the right to charge rent for use, to receive compensation for loss and to be able to collect payment for transfer or sale of the rights in the immaterial goods.

Loss of control in the digital environment is a unifying theme in contemporary debates over intellectual property. In other words, discussion often revolves around a perceived inability to 'monetise' assets (that is, to turn the ownership of intellectual property into profit) by the loss caused by peer-to-peer file-sharing

on the Internet, for example. Software that handles and transforms content is causing corporations to concentrate on shaping the legal regime to ensure their revenue is not lost.

Consequently, copyright constricts the flow of immaterial goods and thereby increases the scarcity. But it is an artificial scarcity and is only held in place through the operation of copyright law, which was not intended to operate as a restriction on the flow of knowledge indefinitely. Copyright is understood as a bargain between creators or authors and the public (and enforced by the state through legislation) to provide recompense to the creator but also to increase the amount of knowledge, music and art in our society for the benefit of all. It is therefore clear that copyright is being transformed from its original intention to that of meeting the needs of corporations wishing to safeguard existing profits and artificially create new markets (May 2000: 6–7). If this is to the detriment of our ability to find and use knowledge and information, and to debate and deliberate, it could have dire consequences for both democracy and creativity.

The growing importance of intellectual property rights and their contestation points to a cultural politics involving possession and transformation of a proliferation of mass and user-produced culture. Citizens in modern societies are differentially empowered social actors active in a public sphere, which increasingly calls on the use of commodified cultural texts to create, improvise and transform meaning and communication. This also calls upon them to present themselves in an active engagement with both commodified culture (such as through decoding through consumption) and their own user-created transformation (Coombe 1998: 69). With the strengthening and widening of intellectual property laws, these citizens' agency will increasingly be mediated and structured through a juridico-technical apparatus designed to legitimate and prohibit certain cultural usage (Lessig 2002b: 181–99). In some cases, this social meaning will be 'frozen' and controlled irrespective of the communicational needs of the citizens (for example, branding is often used by actors to communicate particular meanings; however the

brand remains owned by a corporation that can prevent certain uses or associations). In effect, this power structure serves to legitimate and delegitimise particular voices within a public space (Coombe 1998: 70) but also to structure subsequent meanings and practices. Even within a private sphere or moral economy there is an increasing penetration of surveillance to ensure that cultural texts are consumed 'appropriately' particularly with the use of DRM technologies, 'trusted' computer systems (which essentially allow only trusted actions on cultural texts by 'certified' users) (Lessig 2002b: 140) and other monitoring technologies (such as online registration, licensing, IP address logging, and so on).

As Coombe (1998) argues persuasively, the legal tradition within liberal philosophy is the protection of the individual from state intervention. Thus the importance of the public sphere is predicated on the ability of citizens to act in the public sphere as a check on the power of the state (Habermas 1988). But in post-modern societies, it is increasingly the private property rights of companies (as legal entities) who are disrupting and threatening one of the essential preconditions for an effective democracy, namely free speech.

This legal exclusion is being supported by technological means such as technical protection measures which are themselves now subject to legal protection (see the Digital Millennium Copyright Act in the US). Corporations and governments are currently developing and configuring ever more closed disciplinary technologies such as DRM music CDs,[32] and Blu-ray to replace the current weakly protected film DVDs.[33] These technical devices act as electronic fences, regulating access to those who have paid, those who are approved of and those who consume. Digital rights management software sequesters and locks creative works, preventing their copying, modification and reuse. For instance, Adobe e-Books can restrict to a fine level of granularity how you can use the text; the publisher can even mandate how many times you can print pages from the book, whether you can copy it, or if you can copy and paste sections into other texts. They can also set an expiry date for the book, so that after a certain

date the book will self-destruct and delete itself from the system (Gillespie 2004).

These legislative changes have been raised in reaction to the digital transformations of our use of culture, which have facilitated widespread cultural participation and interaction that previously was not possible (Castells 2000, 2001; Lessig 2004; Benkler 2006). At the same time, it has allowed the creation of new technologies that potentially limit and control these forms of cultural participation and interaction (Gillespie 2004; Lessig 1999). The 'expression' of ideas and concepts, such as books and music, can be encoded into digital information so that it can be transferred through databases and web pages. The production and distribution of this information is a key source of wealth in the digital age and creates a new set of conflicts over capital and property rights that concern the right to distribute and gain access to information. With these restrictions on the access and use of information it can be argued that there is a danger of a corresponding restriction on the use of ideas and concepts.

Therefore, the informational or creative economy is seen as a viable economic model provided full property rights are extended in this way to the intellectual, informational and immaterial (Greenspan 2003). Today the new 'creative economy' is identified as one that is 'largely based on selling novelty, variety and customization' (Florida 2004: 148) and the shift from the 'consumption of goods to the consumption of experiences' (Florida 2004: 162). In the US, this has resulted in changes in copyright; in response to corporate pressure, copyright terms have been extended (The Sonny Bono Copyright Extension Act 1998), Internet laws of property rights developed (Anti-Cybersquatting Act 1999), user licences of information products strengthened in the corporations' favour (see the Uniform Computer Information Transaction Act 2004 (UCITA) which has been passed in Virginia and Maryland in the US) and the legalisation of technical protection methods such as digital rights management (DRM) technologies and the criminalisation of their circumvention – the Digital Millennium Copyright Act 1998 (DMCA). Additionally, database legislation is planned which will criminalise the

duplication of databases which contain collections of facts; these were up to now considered outside the scope of legal protection. In the EU, many of these areas have also been covered by new directives and subsequent legislation, although criticism from civil society in regard to software patents (The Computer Implemented Inventions Directive 2002) has highlighted the cynicism and disregard that the political process has for democratic calls to slow down the process of legislative implementation.[34]

Capitalists increasingly wish to capture knowledge and information for their exclusive control and here it is useful to distinguish between capital in general and the expansion of property law into new areas to generate profit, and the propertisation of knowledge in particular. Although knowledge and information have always been fed into the productive processes there are increasing efforts to render it as a property right in itself. Therefore it is in the interests of the knowledge and information industries (which include the software industry) to push for further legal categories of property in knowledge. Against this trend, a new global movement of networked groups is now emerging, such as the free/libre and open source (FLOSS), which operates across a variety of creative media (music, art, design and software), often under the banner of 'free culture'. Here, equity demands a balance to be struck between public good and private benefit allowing a wider access to knowledge rather than closing down democratic debate or contestation.

Meanwhile, corporations are constructing the means to control ideas and concepts at a level of pay-per-view, whether watching, reading or listening. We all use and reuse ideas and concepts that are shared and non-owned without realising it. Changes are taking place due to the lobbying of multinational media corporations and governments, particularly through the American use of TRIPS (Trade-Related Aspects of Intellectual Property Rights agreement) and other international bodies such as the World Trade Organisation (WTO) – changes which are sadly lacking in democratic debate and deliberation. Thus the more general challenge for governments and economies attempting to shape themselves to an informational base is to develop a framework

that fosters the growth of an economy increasingly dominated by conceptual or creative products that have scarcity created through the use of intellectual property rights.

Although there are a number of different methods of invoking intellectual property rights from copyrights and patents, to trademarks, industrial designs and trade secrets, in this book I focus particularly on the debates, struggles and movements that have contested copyrights and patents. The sheer breadth of complexity in covering the other intellectual property rights and the issues they raise would be a large undertaking in itself (for example property rights in genetic information),[35] but more importantly copyrights and patents are the two key intellectual property rights contested by the FLOSS groups. Viewed historically, copyrights came into focus before patents began to be raised as a particular focus of the FLOSS movements. However, following changes in the law that strengthened the patentability of all sorts of non-traditional techniques and processes, patent law, which has a much wider ramification for software production than copyrights, has become equally important to these movements. Within the US, patents have been expanded to cover software and business processes. These are widely seen as a direct threat to the existence of the free software and open source projects that rely on freely contributed work. Stallman (2005) identified 283 different patent violations within the Linux kernel, which would be enough to cause a major reprogramming headache, and at worst a catastrophic patent infringement case (Stallman 2005). Signs of aggressive use of software patents in order to close open source projects have already begun to appear, with the LIBDCA project, which is an encoding software package, receiving a threatening letter from Digital Theater Systems Inc. which claimed that its patent was being infringed (Smyth, Smyth and Smith 1998; FFII 2004). These discussions will be covered in more depth in the following chapters.

In this chapter I presented a broad historical and theoretical background to how contemporary society is often conceptualised in government policy and more generally in corporate and management thinking. First, I looked at arguments about the more

general changes in relation to shifts in corporate structures and government responses to manufacturing flight to cheaper labour locations together with the concepts of the 'creative economy' and the 'information society'. Then I placed these wider claims in context of a particularly technological understanding of economic growth by concentrating on the history of the software industry. Lastly I briefly examined changes in intellectual property rights worldwide and the way in which immaterial property, such as patents and copyrights, are becoming increasingly important. This chapter was intended to place into context the FLOSS developments discussed in later chapters, and the way in which contestation of software points to important debates that are taking place in wider society about property, technology and economic development. To further bring these questions forward, I will now turn to a genealogy of property rights in relation to the commons, to highlight historical change in our understanding of what can be the subject of property, and how these form an important relationship with our conception of politics, the public and common ownership.

3

THE CONCEPT OF THE COMMONS

In this chapter I would like to change focus slightly by looking at the more abstract and general level of the conceptualisation of property, particularly in relation to the individual and the state. I want to outline a genealogy of property rights and highlight both the contingent nature of their development and the way in which previous ages had a wider notion of property, different from the modern 'binary' focus on either public or private ownership.

I particularly aim to highlight the way in which the intellectual property regime peculiar to immaterial goods, such as software, is increasingly justified (if not legitimated) through a discourse drawing on a concept of the 'common'. FLOSS, for example, is often described in this way. Here I wish to 'uncover' the conceptual and historical background of the common and property rights through the use of topological categories that can help explore different forms of ownership rights and collective control of *things*.[1]

This is important in regard to debates and contradictions found in discussing the concept of the 'common' in general, but useful to explore the case of the 'intellectual commons'. The commons itself is an essentially contested concept, and categorical slippage is frequent in the literature between (i) 'common'; (ii) 'commons'; (iii) 'common-wealth'; (iv) 'public domain'; (v) 'public sphere'; (vi) 'freedom'; (vii) 'commonalty' (or 'commonality'); (viii) 'copyleft'; (ix) 'sharing'; and (x) 'anti-copyright' (Hardin 1968; Coquille 1979; Lessig 1999; Benkler 2002; Lessig 2002a; Boyle 2003; Creative Commons 2003; Rose 2003; Hunter 2004). This topology is intended to foreground the relational status of

property rights and unpack the contemporary discussions around 'digital commons', such as Creative Commons or GNU GPL, to see how they might be understood, and more importantly to see whether they are commons at all.

This chapter is intended to highlight the contingent and contestable nature of property rights, both physical and immaterial, and contribute to further research on understanding the development of the 'information' society. This can be seen as contributing to a critique of essentialist conceptions of property rights in the digital age, rejecting claims such as 'information wants to be free'. The argument is instead predicated on emphasising the political construction of property rights, avoiding naturalistic discourses or explanations.

I argue in this section that law has been a privileged domain for understanding control and ownership of property. As a result, the 'common' has been hidden behind the concepts of 'private' and 'public' within discussions monopolised through this legal binary (Hardt and Negri 2004: 202). By drawing on a variety of diverse and often contradictory work (e.g. Smith 1875; Fenn 1925; Coquille 1979; Rose 2003; Hardt and Negri 2004), my intention is to develop a topology which can cast light on the concept of the commons through the lens of the Roman law classifications[2] of things (*res*). This is in contrast to their use by Rose (2003), for example, who wishes to derive the historical fountainhead of our modern property classifications through an archeological approach to the basis of modern property rights. It is also meant to go further than the 'taxonomy' introduced by theorists such as Bollier (2003), who try to identify common 'assets' and understand the way in which they are governed and used and suggest alternative ways of structuring their ownership and control (Bollier 2003: 178–88).

The intention is to highlight the expansion of property rights to code[3] (and other digital objects) within a claimed 'information society' and to see how the actions of the open source and free culture movements problematise and introduce conflict into attempts to build a knowledge or creative economy legislatively. The concepts introduced here are not meant to be either legal

or historical classifications.[4] It should also be noted that the Romans themselves had a very fluid understanding about how these concepts could be used and applied, the boundaries between them shifting and contradictory.[5] Additionally, I stress that these concepts should not be understood as being entirely independent of each other. The relations are not of *negotiation* but of *contamination*. In other words, the articulation of a particular constellation of property rights and obligations in a concept such as *res privatae* can challenge and change the identity of the others. This is partly what is seen today in the extension and transformation of the meaning and scope of intellectual property law.

The inclusion of a discussion of Roman law is intended to contribute towards a genealogy of the common (Foucault 1989, 1990). That is, I wish to demonstrate that our modern conceptions of property rights and 'commons'-based ownership are the result of contingent moments in history rather than the outcome of rational, planned or natural trends. This contingency highlights the fact that our present conception of property rights, and indeed the way in which we organise our economic system more generally, are open to contestation and change. With current debates, particularly in the realm of FLOSS and commons-based organisations such as Creative Commons, there is a tendency to reify particular property concepts (such as contract). By critically examining how these concepts and legal structures have been the result of conflict and contestation throughout history, the claims sometimes made by FLOSS actors of access to or knowledge of a mythical 'sharing' age, are opened to critical debate. Equally actors' claims to a more 'normal' or 'natural' commons-based approach to economic production are also problematised.

Gaius, a Roman jurist, tried to categorise things that are capable of private dominion through *ius* (the law) (Fenn 1925: 720). In Roman law, to distinguish between the different forms of *dominium*, the legal categories of *res*, or 'things', were used to represent different forms of property ownership, both collective and individual. *Res humani juris* were things that can be subject to human ownership and control; and *res divini juris* consisted of things that belong to the gods (Coquille 1979; Rose

2003). *Res humani juris* were further divided between those under human law that could be either public or private (Fenn 1925: 720). The Romans distinguished between the concepts of *imperium* (jurisdiction) and the concept of *dominium* (ownership or sovereignty). The Roman laws of 'things' fall under the title *dominium,*[6] the general name of those laws concerned with the object of a legal act (that is, which may by owned) (Coquille 1979). In other words, *dominium* is the classification of things, which are capable of becoming the object of rights (Fenn 1925: 726). Within this chapter, the following concepts, which are sub-categories drawn from Gaius, will be outlined and extended to serve as useful 'tools' for thinking about FLOSS in later chapters: *res nullis* (things belonging to no-one), *res privatae* (private things), *res publicae* (public things), *res universitatis* (things belonging to a group), *res communes* (common things), and *res divini juris* (things that are under the jurisdiction of the gods).[7] I will also introduce a neologism, namely *res imperium*[8] (things owned in the international arena), and through the development of the concept of *res divini juris* the section will also attempt to ground the concept of a negative ontology for property, outlining a strictly *political* concept of property right.

Res Nullis (Things Belonging to No-one)

This category can be understood as a 'sink' into which all things currently unclaimed fall, perhaps somewhat similar to a 'standing reserve' of un-owned things.[9] *Res nullis*[10] thus defines the status of objects that belong to no-one and have not been claimed either by a human being or other legal entity (i.e. a corporation). Objects lie within this category until, under legal definition, they are transferred into a form of ownership. It can also be understood as a provisional category for the transfer between different forms of *res*, so it is a transitional form through which *res* can move. It has two senses of usage: (1) as things that have no owner; or (2) to denote a thing that is capable of becoming the object of private property.

When something is claimed it is transferred into one of the other categories (such as *res privatae*). For example, *res nullis* is also the category to which copyrighted works return when they fall outside of the protection of copyright, sometimes referred to as the 'public domain'.[11] This public domain is supported by the action of law, and it has been argued that the public domain should be conceptualised as *res publicae* (Rose 2003: 16). However, the issue is a little more complicated, because although the work *in toto* is now subjected to the rights and limitations of state law, it is also open to reuse and reincorporation into other works (that is, as *res nullis*). Perhaps a better conception of the public domain lies in the understanding that it is *res nullis* but subject to *imperium* (jurisdiction) of state law, as certain restrictions are put on its reuse. This would make it similar to *res communes* and the example of the relationship under Roman law between the sea (which is *res communes* and outside state control) and the shoreline (which is technically *res communes* but which was under the *imperium* of the Roman state)[12] is useful in thinking this through. In another context, Western companies often see no moral or ethical problem in the harvesting and commodification of a developing country's 'unowned' traditional knowledges and biological diversity. The taking and utilising of 'customary knowledge' is not considered 'stealing' as it is not yet seen as private property (that is, *res privatae*).

Res Privatae (Private Things)

Res privatae is the form of ownership specific to private individuals (or corporations as legal entities) that indicates an exclusive property right. These rights tend to be viewed as given by the state (i.e. positive law) or through customary rights (i.e. natural law). They include: (1) the right of access; (2) the right of management; (3) the right of extraction; (4) the right to exclude others; and finally (5) the right to alienation (transfer ownership). Many theorists argue that the right to exclude is the key right, particularly in the realm of intellectual property, as this creates the possibility of incentivising creators by granting a monopoly

right (Maskus 2000), and certainly the right to exclude has been instrumental in arguments regarding the construction and legal protection of technical protection measures such as digital rights management (DRM) technologies (Lessig 2004).

The concept of the 'private' in the Anglo-Saxon tradition usually confuses two distinct classifications, namely: (1) the rights and freedoms of social subjects; and (2) the right of private property (Hardt and Negri 2004: 203). This is due to the fact that legal theory defines all aspects of the subject including interests, feelings and even its soul, as 'properties' that are 'owned' by the individual. Within the concept of *res privatae*, the distinction is kept clearly separate, with *res privatae* explicitly meaning *dominium* over possessions (i.e. things), rather than 'civil rights'. This bundle of property rights, in legal theory, tends to be viewed in terms of *rights* over *things*, or sometimes a 'complex web of legally enforceable relationships' (Sprankling 2000: 7).

There are three major philosophical justifications for property rights: (i) following Locke, in his *Two Treatises on Government*, the essence of the Lockean argument is that property is 'just desert' (Locke 2002). Consequently, property is justified in reference to it being considered a suitable reward for the expenditure of labour (Maskus 2000; Vaidhyanathan 2001; Goldstein 2003); (ii) secondly in the *Philosophy of Right*, Hegel outlines his opinion that property is necessary for the inter-subjective recognition of the self. Individuals define themselves through their control, possession and use of property (Schroder 2004). For Hegel, it is this 'abstract right' of property that establishes the empty form of subjectivity, rather than personality, which is actually added as 'content' by the spheres of morality and ethics. Property is therefore necessary to create a form of subjectivity by creating a legal fiction of a 'person' created through the intersubjective activity; it forms an empty and formal notion of subjectivity. As Lacan uses his metaphor of the mask to explain subjectivity, Zizek similarly argues 'this nothingness behind the mask is the very absolute negativity ... [which] is the subject par excellence, not a limited object opposed to the force of subjectivity!' (Zizek, quoted in Schroder, 2004: 15); and lastly (iii) the final justificatory schema

argues property rights are important through their emphasis on the efficient allocation of resources. In a market economy, particular items will be transferred to those who value them most highly and they will be most likely to utilise them efficiently in order to produce more goods and therefore maximise their return on their investment (May 2000: 92).

It is usually argued that intellectual property rights demand a balance between public good and private benefit. This is a tension between: (1) public access to information and knowledge in a public sphere which is said to be a public good; and (2) private ownership of information and knowledge (Habermas 1992) which may or may not be in the 'public interest'. The risk is that when public goods are privatised, the condition of possibility for a critical democratic public is destroyed. This is the tension that is developing in the ability to access, use and reuse culture (see Coombe 1998). The legal rights granted by IPRs include the right to charge rent for use, to receive compensation for loss and to be able to collect payment for transfer or sale of the rights in the immaterial goods. But in principle, unless drawn from an essentialist position regarding the 'real' nature of things, either physical or immaterial, there is no theoretical limit to the creation or extension of property law within a sovereign state, although there may be practical ones (for example, policy set by the WTO or WIPO, democratic protests, business lobbying).

Res Publicae (Public Things)

The concept of *res publicae* has been used in contradictory ways within law and political philosophy,[13] namely as (1) state ownership, as public property; and (2) civil government or governance, as public government. This brief historical excursus demonstrates a number of different conceptions associated with the concept of *res publicae*.

First (state ownership), *res publicae* are things belonging to the public and open to the public by operation of the law (i.e. public things) (Rose 2003: 8). Some examples include public squares, ports, bridges, public buildings, waterways, roads and

even possibly the Internet itself. Within this usage the concept is associated with state[14] ownership, or public ownership. Those things which are the *dominium* of *res publicae* are of two classes: (i) those things which have a public character by reason of a public use, such as navigable rivers, roads, and ports; and (ii) things or goods belonging to the state, such as slaves, houses and territory, which have been expropriated through conquest of an enemy (Rose 2003: 8–9).

Secondly (*res publicae* as civil government), as a famous speech, which Thucydides attributed to Pericles,[15] outlined as the basic features of the constitution of the polis. This is what Aristotle called *Politeiai* and Cicero called *res publicae*, namely 'love of freedom', 'respect for the law', 'equality before the law', 'rule through agreement' and 'government through the consent of the governed' (Maihoffer 2003: 283). In political tracts[16] such as Cicero's *De re publica*[17] the meaning is split between *re publica* (singular, species) referring to a 'republic', and *res publicae* (plural, genus) referring to 'states', 'republics' or 'constitutions'. It can also be used in relation to 'public business' or 'public affairs'. This points to the development of a political theory that seeks to link and justify the claims for public control of assets, either through the one, the few or the many (that is, democracy).

According to Cicero, the meaning of *res publicae* is *res populi* (the people's business) where 'the people' is 'a union of a number of men, acknowledging each other's rights and pursuing in common their advantage, utility or interest' (Coleman 2000: 230). Indeed, the English 'republic' is derived from the Latin *res publicae* meaning 'the public thing', 'the public concern' or the 'people's business'. Cicero had a particular interest in justice, and in the distinction between commons and private goods, and argued that one should treat common goods as common and private ones as one's own. He argued that no property is private by nature – instead it becomes private: (1) by long occupation (by custom); (2) by victory (the spoils of war); or (3) by law (settlement, contract or lot). Today, the term most often associated with this definition of *res publicae* is 'commonwealth'. For Cicero, private owners must 'contribute to the common stock of things that benefit everyone together and,

by giving and receiving, effort and means, they bind the fellowship of society together' (Coleman 2000: 257).

Later *res publicae* became more associated with the concept of civil communities, especially through developments in medieval and Renaissance political philosophy (Coleman 2000b: 199–230). The early Florentines distinguished a republic from a monarchy as the 'contrast between the absolute and arbitrary exercise of government and its limitation by law and the will of the people' (Rubinstein 1993: 4). This is the development of a political vocabulary of republicanism during the Renaissance and a political concept of *res publicae*, that would have important influence on later political thought (Skinner 2003: 133–4), most notably through Machiavelli, who argued that a republic could guarantee liberty that acts to promote the common good of its citizens who are represented in collective assemblies (Coleman 2000b: 273).[18]

These ideas, together with previous thinkers' articulation of the question of the state and power, came together in Hobbes' *Leviathan* (1998).[19] Hobbes equated the arguments between different theorists such as *civitas*, 'commonwealth' and the 'state' in the concept of the 'Leviathan'.[20] Hobbes drew the distinction between a 'State of Nature' and a 'Civil State', and the importance of avoiding the *bellum omnium contra omnes* or 'war of all against all' in a state of nature. This is the source of his legitimation of the existence of a strong sovereign to whom all members of the commonwealth should give their allegiance – a shared commonwealth or *res publicae*. With Locke (2002), particularly in the *Two Treatises on Government*,[21] we see the development of the concept of the social contract and the ideal of 'civil society'.[22] Civil society rather than the State was the arena of the politically active citizen. He also was concerned with a 'civilised' society, one that is ordered under the 'rule of law' rather than a despot.[23]

Marx and Engels (1998) famously argued in *The Communist Manifesto*,[24] that the state as *res publicae* was the organised use of force by one class in order to bring another into subjection, and that it was a form of organisation set up by the bourgeoisie in order to protect their property and interests (Marx and Engels 1998). Their

belief was that the state was an ideological screen that hid the machinations of the ruling class and ruling ideas (Marx and Engels 1999). It could not then be considered a commonwealth, as it was the creation of a particular class in society; the commonwealth could only exist when a socialist society was born which truly shared the fruits of society in common.

Today, the term *res publicae* is polysemic with regard to political philosophy. More recent writers, such as Marquand (2004), argue that *res publicae* is the 'public domain'[25] which, in his definition, appears to be both separate from the state (that is, similar to a public sphere or civil society) and also a set of norms and values (that is, public ethics, civic duty and the like) (Marquand 2004: 38). This follows the work of Sennett (1978), who argues that a *res publicae* 'stands in general for those bonds of association and mutual commitment which exist between people who are not joined together by ties of family ... it is the bond of a crowd, of a "people," of a polity' (Sennett 1978: 4). Skinner (2003) argues that the republican tradition is referring to the *res* as the government, which should reflect the will of the *publicae* (that is, the community as a whole) (Skinner 2003: 302). This category has also tended to be confused with *res communes*, which may be due to questions of ownership and/or protection of *res communes* by the state through law (see *res communes* below).

In sum, *res publicae* is usually understood as either: (1) state control or ownership; and (2) things held and managed in common (Hardt and Negri 2004: 203). Here again the key aspect of this classification is that *res publicae* points towards the *dominium* of that *res* or 'thing' owned by a *public*. Of course, the definition centres particularly on the conception of 'what is a public?', a position that Hardt and Negri (2004) argue is tied explicitly to an idea of the nation state and the identities that it articulates.[26]

Res Communes (Common Things)

The 'common' is often understood with regard to commonalty (the common body of man) or commonality (that which we share *in* common). For example, Hardt and Negri (2004) argue that it

should be considered with regard to the *common interest*, that is the general interest that is not made abstract in the control of the state (that is, the *public interest*) but rather is held in common by 'the singularities that co-operate in social, biopolitical production ... managed by the multitude' (Hardt and Negri 2004: 206). This, they believe, marks the passage from *res publica* to *res communis*, a form of democratic sovereignty based on control of biopolitical production by the multitude (i.e. a form of social organisation that displaces sovereignty)[27] (Hardt and Negri 2004: 206).[28]

Res communes define things that are capable of non-exclusive ownership or incapable of ownership. In other words, things that are open to all by their nature. The Roman examples of *res communes* were the oceans and the air, although Rose (2003) adds stocks of wild fish and animals (rather than individual animals, which can be captured). These are large and diffuse things and therefore difficult to capture. Nonetheless, they remain finite, and they can be extinguished or used up, as the examples of overfished cod stocks or rainforest clearing demonstrate.

UK law (excluding Scotland)[29] which derives its legal conception of the 'common' from Roman law, did claim a form of ownership,[30] or perhaps more accurately, a kind of supervision of the commons. Commons are usually identified by legal Acts of Parliament, such as in an Act of 1829 that was required to transfer Hampstead Heath into the protective custody of the Metropolitan Board of Works after it was threatened by Sir Thomas Maryon-Wilson's building plans (OSS 2005).[31]

A similar logic can perhaps be seen in the decision by states in the twentieth century to impose a 200-nautical-mile (370.4 km) Exclusive Economic Zone (EEZ) on top of the 12 nautical miles (22.224 km) territorial boundary of the state (Rose 2003: 6).[32] This was justified at the 1973 Third United Nations Conference on the Law of the Sea to protect fish stocks, but it seems that oil and mineral drilling rights were an additional incentive. Interestingly, the relevant Convention defines this deep seabed area and its resources as 'the common heritage of mankind' (UN 1970: Part XI, Section 2, Article 136) regulated by an International Seabed Authority (an autonomous organisation having a relationship

with the UN)[33] (UN 1970: Part XI, Section 4). There are also movements by states to experiment with creating ownership in things that were previously thought not possible to be owned, such as the air, for example through Tradable Environmental Allowances (TEAs) (Rose 2003: 7). The terms of the Outer Space Treaty 1967 Article 1, designate 'outer' space as *res communes*, and in the Moon Treaty 1984 limits are placed on national (i.e. property) ownership of the moon and the exploitation that can take place there (UN 1967; UN 1979).[34] Similarly, in the US, the concepts of common land through 'state land trusts' have been around since 1787, when Congress required western territories to set aside land. Today, the US states hold more than 150 million acres in trust; some of this land is leased for timber, grazing or oil extraction (Bollier 2003: 85–97).

One of the key problems with *res communes* controlled by a state (that is, through law) is that it requires the courts to be responsible for holding the state to account. This puts that state in a position of both legislating and executing decisions with regard to *res communes*. This can most easily be demonstrated with regard to the position of State/Public Trust Doctrine in America. Analogous to *res communes*, these lands have been encroached upon time and time again by the state or corporations when they wish to use the land for almost any purpose (Bollier 2003: 85–97). The legal uncertainty of public trust, especially when the state or local government is democratically elected and therefore instilled with a sovereignty, causes a conflict in attempting to protect misuse of the commons and common ownership. This highlights an important distinction between public ownership and common ownership.

Today, the basic category of *res communes* extends to or has been drawn upon by theorists and commentators to refer to the space of intellectual thought, conceptualising ideas and concepts within a digital arena as an 'ideas commons', an 'innovation commons' (Lessig 2002a: 23; 2005), an 'intellectual commons', a 'digital commons', inevitably an 'e-commons' (Boyle 2003), 'the public domain' (Boyle 1996) or 'intellectual space' (Rose 2003). Many theorists appear to have a broad notion of what the

'commons' is. However, there are often confusions between the public domain (perhaps more accurately *res nullis*), the notion of the commons (*res communes*), and that of state ownership (*res publicae*).

Res Universitatis (Things Belonging to a Group)

Res universitatis are things that are owned by a group in its corporate capacity. These corporate bodies, given by a royal decree, were often municipalities or guilds which would often own property such as racetracks and theatres. To be a *res universitatis* required the authorisation of the state and the *res universitatis* would own and control the groups' property in common for its members. Interestingly, this is where the name 'university' is derived, as a corporate body formed of students and academics dedicated to education. The *res universitatis* form of property ownership typically exists where a resource is too large for a single member to administer, yet is still bounded enough to be able to be conceptualised as a property right. It has been described as property on the outside (i.e. to non-members) and a commons on the inside (Hyde 2006: 83) and hence shares many of the characteristics of *res communes*. Examples include guilds, monasteries, merchant groups, clubs and common land communities (see Rose 2003: 17; Sennett 2008).

Res Imperium (Things Owned in the International Arena)

Within the Roman legal tradition, the laws between nations would have fallen under the *ius gentium* as the laws common to all men.[35] However, *res imperium* is introduced here to bring in the modern sense of international law and its construction and a definition of new forms of *dominium* beyond the boundaries of the nation state but yet still owned (examples include telecommunication satellites, broadcast information and airspace). Much of the ownership of extraterritorial property is subject to international treaty agreement and the jurisdiction of the relevant state (as *res publicae*). Their status as objects of international treaty makes

their enforcement difficult, force only being available through international organisations like the World Trade Organisation (WTO) or through the mobilisation for war. Additionally, there have been moves to create international law based on international courts, and these lie outside the particular jurisdiction of an individual state through supranational jurisdictional practices. Hardt and Negri identify these as global or imperial forms of law (Hardt and Negri 2004: 29). One has no doubt that in the race to space there will be inevitable attempts to change their status under the Outer Space Treaty and the Moon Treaty from *res communes* to another form of *dominium* that is more amenable to exploitation. Perhaps this will be instantiated through a *sui generis* category like *res imperium*.

Theorists are beginning to argue that the development of intellectual property law, particularly under the aegis of the WTO and TRIPS, gives us the first truly global legal system (Bettig 1996; May 2000; Drahos and Braithwaite 2002). In terms of the standardisation of intellectual property regimes, there is some uniformity, although as many people have pointed out, there can still be major differences between states as the TRIPS agreement only specifies minimum term lengths and types which individual states can extend (Maskus 2000). This category becomes important when it is applied to the question of the location of things (particularly code-based) between nation states, which is increasingly the case with satellite systems, international way stations and other technologies that extend beyond the state. Although these can be understood with reference to *res publicae* (public ownership), it might be useful to be able analytically to distinguish things which are owned in a private or public capacity, but which for particular reasons lie anchored within international commons or similar spaces.

Res Divini Juris (Things Under the Jurisdiction of the Gods)

Res divini juris[36] was originally a category defining those things that were not subject to the ownership of humans, but were within the control of the gods (sacred, holy or religious things).

Examples might be temples, sacred places, altars and places that had been consecrated. Things within this category were considered to belong to no-one because they were dedicated to the service of the gods or their use might be offensive to the gods (Rose 2003: 21).

The key issue with the category of *res divini juris* is that it is outside the boundaries of individual ownership and the nation state; here I want to link it instead to the political possibilities of human singularities acting together outside of a geographically bound political community. This inevitably means that the boundaries of *res divini juris* are in a constant state of exception, open to the political contestation of the multitude. In essence, this concept will be developed here in a two-fold direction: (1) as a form of *dominium* which is outside of human ownership, but in common collectively (rather like the concept of the common heritage of mankind); and (2) as a space of international political interaction beyond the *public space* represented inside the nation state (a social ontology, perhaps as a form of global civil society[37] or global/networked democracy).

Therefore here I tentatively suggest that *res divini juris* could be radicalised if it were reconceptualised as beyond the ownership or control of any *single* individual, corporation or state. Rather, it would be the proper domain of the *res communes humanitatus* (Bollier 2003: 177), the 'common heritage of mankind' (UN 1970: Part XI, Section 2, Article 136). This concept is useful as it designates a domain beyond state sovereign control and raises an interesting political and legal challenge to the assumption that private ownership is the only way to protect things. The common heritage of mankind has been usefully glossed by Payne (1978) as:

> (1) no state can appropriate the area; (2) exploitation of resources in this area should be regulated by an international body, and not dominated by a few states; and, most importantly, (3) revenue . . . should be distributed in a manner designed to reduce the economic disparity between developing and developed countries.
>
> (Payne 1978: 946)

This suggests interesting possibilities for supranational forms of commonalty, if taken together with a form of 'common heritage of mankind' and 'sacred trust' such as in the Treaty of Peace with Germany (28 June 1919, Art. 22) (Fenn 1925: see footnote 49).

Here I wish to link the concept of *res divini juris* to the radical critiques of commonality found in the work of Marx, Hardt and Negri, Virno and others, in order to conceptualise a new kind of shared outlook or common interest (Marx and Engels 1999; Hardt and Negri 2000; Virno 2004). This form of *dominium* (or perhaps more accurately *imperium*), could then be manifested for the good of humankind as a whole. Examples could be the human genome, the natural world, life-forms, space, the moon, and ideas and concepts. The concept could be instantiated through some form of international treaty or through the development of a concept of *species being* (such as the 'General Intellect')[38], or perhaps life itself, such as in the concept of *élan vital*[39] developed by Bergson or alternatively through *sui generis* international law.[40] That is, a concept developed through the practices of non-owned, freely shared commons-based peer production (Benkler 2002).

In such a form, one of the key questions is who will defend *res divini juris*. This is because, as outlined with the concept of *res publicae* and *res communes* (particularly in the state-centric versions of the common), there is a danger of co-option or alienation by a political class (or even a bureaucratic class). As *res divini juris* is intended to be an international concept, there would clearly be a problem with legitimacy. If an Amazonian rainforest were declared *res divini juris,* for example, this alone would not necessarily stop its exploitation, seizure or confiscation (one can imagine a later government rescinding the decision, for example). It is sometimes argued that a global public (or collective will) as a political subjectivity might help to give legitimacy to some form of international collective action (Hardt and Negri 2004; Virno 2004). Here global struggle over values and sharing would be important because this is the practice of a *defence* of a collective right as an immanent value within humankind for the benefit of all, such as demonstrated by the concept of cultural property for the good of humankind (Hardt and Negri 2000: 294–300).

Instead I argue here that the idea of a negative ontology seems to offer some potential to help invest *res divini juris* with an open and contestable meaning.[41] This is not intended to be a common space or public sphere that is formed through the creation of a common purpose or good (that is, it is not based around concepts of commonality or commonalty, for example). The organising principle of this space is a negative or 'lack' and this lack mediates between the individual, the common, the particular and the universal (Laclau and Mouffe 2001) – that is, an ontology that uses 'lack', 'negation', or any concept that requires 'nothingness' for its existence. The political nature of property would therefore remain open and contestable (Mouffe 2000; Laclau and Mouffe 2001; Mouffe 2005). Here the possibility of the 'real' is rejected as an impossible *jouissance* (an enjoyment beyond any barrier or limit), because it tries to move beyond an ethical identification with the universal concept of 'good' through the idea of sublimation. Through sublimation a public space is created, a unifying field or habitus, which although tied to the individual, offers a public arena or space.[42] For example, a work of art is often identified with the body of an individual artist, or perhaps a singer's voice. However, art is also addressed to a public (or audience), entailing the connection to and creation of a public space without ever losing its individuality or singularity.

> The public of sublimation is not, in this sense, a public of common denominator, of communality. Sublimation is rather the public space in which our singular perverse bodies may make contact with one another through the creation of beautiful objects that stand for them.
>
> (Rajchman, quoted in Stavrakakis 1999: 132)

This represents an attempt at the political praxis of institutionalising a negation within political reality so that, rather than relying on a unifying commonality (nationality, interest, ethnicity), instead the disharmony, contingency and conflict that is constitutive of the political is brought to the fore (Mouffe 2000, 2005). This is not to say that democracy would therefore end in chaos, as clearly democracy is a form of order (Stavrakakis 1999: 138). In this democratic vision of society, the lack is made manifest, such that

Table 1: Summary of the genealogy of property forms

Node	Translation	Description
res nullis	Things belonging to no-one	Unowned/unclaimed objects not defined as any other category.
res privatae	Private things	Privately owned objects such as privately owned cars, houses and clothes.
res publicae	Public things	Publicly owned objects such as army barracks, roads or state buildings.
res universitatis	Things belonging to a group	Things owned in a group such as a local council, guild or society.
res communes	Commons things	Things held in common, such as common land, the sky or the sea.
res imperium	Things owned in the international arena	Objects subject to ownership but outside of national jurisdiction, such as satellites or deep-sea probes.
res divini juris	Things that are under the jurisdiction of the gods	Things that under the jurisdiction of the gods, for Roman law, but here redefined as international objects held in common by political action.

'politics loses the possibility of [total] representation. It cannot presume to be – or even to represent – the whole within the whole' (Luhmann, quoted in Stavrakakis 1999: 138). Thus, the radical openness of *res divini juris* would point to its inclusiveness (by the impossibility of closure) and constant rejuvenation through political action and the productive capacities of the multitude. This suggests new forms of democratic control and communal ownership that stand counter to state-centred or traditional public ownership of resources, all forms examined by Hardt and Negri (2004), Virno (2004) and Brown and Szeman (2005).

This concept then, adapted from its original Roman definition, stands for a form of common ownership beyond the boundaries of the nation state and suggests the possibility of a global collective ownership. This is meant in terms of a form of 'common heritage of mankind', which would presumably require a form

of international agreement or treaty obligation to enforce it, but here I wanted to focus on the political instantiation of the concept (rather than a rational/bureaucratic moment).

In this chapter I have introduced a genealogy of property rights to contribute to understanding the way in which the property relationships within society are organised (summarised in Table 1 above). Most relevant to FLOSS is the concept of the common that has been used by a number of theorists to explain the practices of FLOSS developers. I will be using these concepts in later chapters to try to unpick the way in which practitioners as well as theorists of FLOSS use different property forms to understand and explain the activities of FLOSS development. In the next chapter, I return to the subject of computer code and look at the particular case of FLOSS and its historical trajectory.

4

FROM FREE SOFTWARE TO OPEN SOURCE?

The congress shall have the power … To promote the Progress of Science and the useful Arts, by securing for limited times to Authors and Inventors the exclusive Right to their respective Writings and Discoveries.

The United States Constitution (1787)

Most directly, the thing you do is theft.

Bill Gates (1976)

In this chapter I want to pick up some of the themes that have been discussed in the previous chapters and examine them through the optic of the free/libre and open source software movement (FLOSS). In particular I am interested in the way in which FLOSS groups have come to represent collectively a certain post-Fordist model of production, christened variously; the *Wealth of Networks* by Benkler (2006) or the *Hi-Tech Gift Economy* by Richard Barbrook (1998). In these models, the production and distributed creativity of the FLOSS groups symbolise either: (i) a new form of collective production that undermines/re-writes capitalism (see Benkler 2004); or alternatively (ii) a deeper, more profound model of human productive capabilities (often pre-capitalist, or previously held back by the fetters of capitalism). These two strands of explanation have a discontinuity hypothesis that informs their respective logics; that is, that we are somehow witnessing the birth of a new radical replacement for capitalism, or certainly a replacement for capitalism in its Fordist form. This clearly reflects the information society discourses that were discussed in the previous chapters and their claims for a new form of society or economy.

In light of these claims the free software and open source movements are an interesting and contradictory constellation of groups. Indeed, under the rubric of commons-based production and influenced by private ownership, libertarian values and republican ideals, they are creating an important collection of artefacts, products and technical devices on the Internet that contribute to what might be called *technologies of the commons*.[1] The social practices of these groups challenge traditional notions of production processes within the software industry (i.e. top-down, waterfall-model software development)[2] and offer inspiration to other groups operating in specific content segments of the media, such as music and film.[3] The democratic and reflexive nature of their activities, which are undertaken in transparent online forums and email lists, offer some important challenges to the 'common-sense' notions that innovation can only take place within narrow market-based contexts and that without management directing the process, innovations are difficult, if not impossible, to realise.

Within a proprietary paradigm of software production, dominant capitalist principles, values and perceptions of power are embedded in technology quite deliberately through the use of certain programming structures and processes, such as the separation of source and executable code and the use of intellectual property rights and trade secrets to institutionalise and monopolise knowledge. For example, the Google PageRank algorithm is a mathematical formula used to rank web-pages in Google's search rankings which is patented and now has a substantial market value.[4] Indeed, this is a logic that is difficult to avoid, otherwise the very conditions for the software market are undermined either through the easy imitation of the design, functionality or form of the software artefact or through the digital reproduction (i.e. copying) of the software. In other words, the principle of scarcity has to be constructed by the market actors and continually enforced through intellectual property law-based lawsuits or technologically enforced measures. One of the clearest examples of these two alternative logics of development[5] for the computer software industry was represented in an early argument at the Homebrew Computer Club, which in the 1970s had as

members Bill Gates, Steve Jobs and other computer hackers and hobbyists. In this case, the early hacker conventions and norms of freely sharing software and ideas were directly attacked in the famous 'Open Letter to Hobbyists' from Bill Gates and published in 1976.

> Will quality software be written for the hobby market? As the majority of hobbyists must be aware, most of you steal your software. Hardware must be paid for, but software is something to share. Who cares if the people who worked on it get paid? Is this fair? One thing you don't do by stealing software is get back at MITS for some problem you may have had. MITS doesn't make money selling software. The royalty paid to us, the manual, the tape and the overhead make it a break-even operation. One thing you do do is prevent good software from being written. Who can afford to do professional work for nothing? What hobbyist can put 3-man years into programming, finding all bugs, documenting his product and distributing for free? … Most directly, the thing you do is theft.
>
> (Gates 1976)

The alternative demonstrated through open source and free software involves a network of actors who form into loose coalitions to produce software based on a commons-based production model. That is, that the code produced is held either by proxy within a *res universitatis* type structure (such as the Free Software Foundation) or offered up freely in a more loosely defined but legally backed *res communes* (through the use of 'copyleft' licences, for example). In addition to social activities and norms there are a number of technical practices and artefacts that assist in the production of FLOSS projects, and of equal importance, have wide applicability in serving to structure and reinforce these commons-based practices, many of which have been taken up in other spheres (particularly the cultural, discussed below).

Here I want to use a political economy approach to unpick these arguments and critically examine these claims to stress the continuity of FLOSS movements (for a discussion of the major debates within the movements see the next chapter) within existing capitalistic logics. In particular I want to look at the extension of intellectual property rights to facilitate market exchange and

the exchange of goods and services for software. I think it is of crucial importance that the FLOSS practices are understood to be taking place within an intensification of capitalism rather than signalling its replacement. Indeed, FLOSS points to a new level of cost-free labour that is outside the traditional labour/capital nexus because it is conducted largely in the free time of the computer programmers concerned. That is not to say that FLOSS is a wholly negative development, even for those who are critical of the market; rather I believe that FLOSS performs an important function as a signal of the boundaries of market expansion, presenting a form of collective action that has crystallised in *response* to capital's quest to commodify ideas, knowledge and information. In a certain sense the progressive development of a *res universitatis* through the GNU project (discussed below in detail) can also be understood as a form of collective organisation, rather like a union or guild-like structure for computer programmers, that seeks to protect its members from the worst effects of property relations in specific types of information (in this case, computer code).

What FLOSS projects share, though, is a linking of their social practices around a concept of the common (*res communes*). As outlined in the previous chapter, the common is a polysemous concept and within the communities active in commons-based production the concept is given little critical thought. This is partly due to a pragmatic tendency within technical cultures to stick to something that 'works', but also perhaps reflects a realisation that the multiplicity of 'commons' (or the essential contestability of the concept of the common) means that it would be very difficult to reach any kind of consensus on a definition which members could agree on. More particularly, members come from across the political spectrum, from right-wing libertarians to left-wing Marxists. Nonetheless, the fact that such different actors have effectively drawn into their practices what is, in effect, a non-capitalistic form of exchange (a public good, sharing-knowledge economy) raises pertinent questions about the viability and trajectory of co-operative modes of production in these delimited spaces. Additionally, the mythologies that surround FLOSS production, often seen as superior to the traditional *capitalistic*

practices of proprietary software firms, point to a strange contradiction within FLOSS groups that seek both to transcend capital and yet remain discursively (and in practice) anchored to it. This perhaps betrays a political/technical cleavage that is often reflected in discourse and discussion conducted within FLOSS groups (and discussed in more detail in the next chapter) but that nonetheless raises interesting challenges to a researcher who has to grapple with a certain FLOSS 'newspeak' when teasing out the way in which social practices are legitimated and understood within respective groups.

In order to examine these issues in greater depth, in the following sections I concentrate on the contestation of copyright and the strategies and social practices that FLOSS groups used in their engagement with this subset of IPR law. Most significant was the early techno-utopians' hardline 'a-political' stance and insistence on adopting a technocratic approach to solving societal problems and to bypassing ('hacking') legislative approaches,[6] the well-known 'Californian ideology' identified by Barbrook and Cameron (1995). That is, groups such as the Free Software Foundation (FSF) and the open source movement (OSM) are involved in sharing ideas and concepts in a common discursive sphere (usually online) that is both an input and an output to the productive process. Free software and open source also offer a contestation of what it is to be creative, creating a politics of creativity, through their discursive output manifested in a number of online texts and key foundational documents.[7] By using technical 'hacks' it was widely thought that hackers could circumvent legal and administrative attempts to control what happened in the digital realm. We will come back to the question of the political in technical practices in the following sections but for now it is important to note that even the key proponents of FLOSS were wary of the cut-and-thrust of 'real-life' politics and advocated an engineering philosophy to 'make things work',[8] epitomised by the famous words of the Internet pioneer Dave Clark and adopted by the Internet Engineering Task Force: 'We reject kings, presidents and voting. We believe in rough consensus and running code' (Borsook 1995).

This brings to the fore the question of the political economy of FLOSS, and this will be investigated throughout the chapter. To write software requires that certain basic human needs be met, such as food and warmth, and that requires some engagement with the wider capitalist (and usually proprietary) economy. Contrary to the views expressed in *A Declaration of the Independence of Cyberspace* by John Perry Barlow (1996), that 'our identities have no bodies', it is clear that the computer programmers who write free software need to make a living and support themselves economically. This usually means employment in the computer sector, working for large corporations such as IBM, Intel or even Microsoft. It is also ironic that FLOSS groups utilise the structures put in place to support copyright to protect FLOSS software from being commodified or 'closed'. These contradictions in the social practices of FLOSS production point to interesting questions about the extent to which FLOSS is parasitic, or at least reliant upon, the very intellectual property rights (IPR) laws that it purports to attack or subvert.

Nonetheless, the key site of contestation and the source of unity for FLOSS developers has been their rejection of the use of IPRs to protect or monopolise certain aspects of the common use of coding conventions, routines or algorithms. More accurately FLOSS groups use copyright to create a space whereby the limitations and private ordering aspects of copyright law (authorial rights to distribution, derivatives and so forth) are expressly renounced. This of course, raises its own problems in terms of the performative contradiction of a movement that seeks to transcend copyright, but which rather than expressly fight politically for changes in copyright law, prefers a 'technical' course of action by 'hacking' law to produce a similar result. I think that this also serves to demonstrate the weak foundations of this space, which relies on state-controlled monopoly rights in information and knowledge to create a commons, but one that, with a change in copyright law, could easily come crashing down.

In this chapter I look specifically at the way in which copyright, which traditionally protects the expressive dimension to intellectual property, has been constructed as a threat to private computer

programming practices and then widened through political contestation into a public issue. In particular, I want to focus on the way in which copyright policy, and in particular tendencies in contemporary policy towards increasing term-lengths and breadth of cover, has had major effects on the perceived ability of computer programmers to carry out their work. I then want to look at how this has been extrapolated through critiques of the intellectual property protection of culture and ideas through a notion of a 'cultural commons' – that is, the artefacts and practices of social life that are shared freely and are generally considered to be outside the formal economic property system.

First I present a brief history of free software and open source that is intended as a gloss on a very complex and developing subject (especially with regard to copyright law).[9] It is not meant to be exhaustive, rather it is intended to contextualise the debate and practices that underpin FLOSS and show how capitalism is not external to FLOSS but a critical source of funding and support. I then turn to the specific means by which free software germinated a 'politics of code' which led to copyright being used in unexpected ways to secure sharing freedoms through the principle of 'copyleft' for writing of computer code.[10] Finally, I turn to look at the institutional support and networks of co-operation that serve to ensure the economic base of FLOSS, particularly through institutional links and state support, albeit ones that demonstrate unintended consequences borne of their reliance on the unquestioned romantic notions of authorship derived from copyright and the broader logic of capital.

A Brief History of Free Software and Open Source

The origins of free software can be traced back with that of the software industry to the late 1950s when computer technology was a fledgling industry, funded by the US government. The difficulties faced by computer scientists in a new knowledge sphere and their early links to academic and scholarly norms of research (such as publishing results, sharing software programs and presenting at conferences), contributed to sharing practices that later fed

directly into the ethics of the FLOSS movement. For example, in the early days of computer research at Stanford, Berkeley, MIT and other university institutes, and due to the small number of participants involved, it made sense to freely share the software they wrote (Markoff 2005; Levy 1994). Additionally, of course, this was shortly before the important decision to unbundle IBM software from computer systems, where software was largely given away free with the source code (see Chapter 2). These early experiences by programmers and developers tended to reinforce the notion that software was a public 'informational' good that should be freely shared, and indeed the concept of property or ownership of software was anathema to the ethics of the early hackers who proved their skills precisely by *showing* and sharing how cleverly they could program.

Generally it is argued that the free software and open source software movement proper began in the 'hacker' culture of US computer science laboratories at Stanford, Berkeley, Carnegie Mellon and MIT in the 1960s and 1970s. Software was originally closely related to the manufacture of hardware technology and partially developed by the academic institutions such as UCLA, Stanford, MIT, Berkeley and others (Hafner and Lyon 1998: 38). Funded mainly by the US Department of Defense, the largest buyer of computer technology in the world at the time, technology and technical practices were shaped by the needs of the military and government and software was custom written for each machine and application (Hafner and Lyon 1998: 42). Here there was no meaningful differentiation between user and programmer (or indeed between software and hardware).

During the early days of the computer industry, people identified themselves as craftspeople. Their culture was very much one of artisanship rather than pure engineering (Levy 1984; Thomas 2002), a view that is still evident today in highly technical discussions on mailing lists and websites such as Slashdot (see Sennett 2008). Groups working on computer projects would stay together throughout the entire life-cycle of computer systems or software artefacts. This collective team-based unity later caused problems when more conventional Fordist methods of production,

such as the division of labour and management hierarchies, were slowly installed to routinise the processes of software production (still rather unsuccessfully as it turned out). In fact the anti-corporate and anti-managerial feeling of much of the free software and open source movement discourse can be traced back to these early freedoms and to the experimental practices of the first software coders being contained and blocked by employers. As one developer commented when new management techniques were first used in a programming team, 'those of us in the field remember feeling that a firm division of labor had been introduced almost overnight' (quoted in Weber 2003: 25).

During the late 1960s, Unix, one of the key technologies that has had a huge influence on FLOSS, was developed by Dennis Ritchie and Ken Thompson, for general mainframe processing, working for Bell Labs, owned by AT&T. This operating system used a clever way of sharing the computer processing time amongst a number of users, as a single processor was such an expensive piece of hardware. This early experience indoctrinated the early programmers with the principle of sharing resources amongst themselves – the 'commons' was understood as the amount of processor time and software that was available that had to be shared equitably between different users. It was also built around the design philosophy of 'build small neat things instead of grandiose ones' (Weber 2003: 26). This doctrine of simplicity and smallness would become crucial to its success as it led to a modular design philosophy that is still in evidence today. The Unix philosophy became more widely known as it spread, identified through three main tenets:

> [(1)] Write programs that do one thing and do it well; [(2)] write programs that work together; [and (3)] write programs that handle text streams because that is a universal interface.
>
> (Weber 2003: 28)

In fact these rules turned out not only to be handy heuristics for solving complex and sophisticated programming problems (such as those evidenced in operating systems technologies). They also had the unintended consequences of making the distributed nature

of the Internet possible: (i) enabling the sharing of the source and binaries of the development of a large software project (that is, modularity through small programs which eases the division of labour and the complexity of the task); (ii) making it possible to communicate with other programs as a constellation of networked components without reference to geographic proximity; (iii) enabling the exchange of data and information (that is, using the *Esperanto* of text streams to communicate between the modular objects that made up the computer system); and (iv) allowing the creation of a toolbox of modular software objects which could be assembled by developers in a number of different configurations that permitted novel and innovative uses, in contrast to the failed attempts of the open systems' followers to mandate top-down control.

With the introduction of software copyright in 1992,[11] which defined code as an 'expression' similar to that contained within a novel, came the materialisation of the contradiction between the social activities of programmers who shared knowledge and methods amongst colleagues, and the requirements of corporations using law to fix copyrighted works to exploit their use. It also raised the related questions of the 'fixing' of a piece of software that is continually in flux, of who the 'authors' might be, and indeed focused attention on the requirement for its fixation in a physical medium, for as Kittler (1995) observed, 'there is no software', there was only a computer-readable digital encoding in 0s and 1s on the magnetic surface of a computer disk drive or in the tiny capacitor arrays of computer memory. In fact, much legal argumentation and debate has taken place over how to understand code – whether as a form of writing (e.g. an expressive work) or as a tool (e.g. a machine).[12]

Particularly where shrink-wrapped software was to be sold there was a need to strip away the 'sharable' aspect of software to support a proprietary software model to keep secret the knowledge embedded within the source code. This shipped product then required the full force of legal, technical and social appropriation to try to force computer users to purchase (rather than copy) the software. But even within the corporations, there was a concern

that the ease of copying on digital computers made the removal of intellectual property incredibly easy for competitors and also led to the loss of trade secrets with staff that left the firm. This was therefore the beginning of the focus by companies on the use of contract, particularly non-disclosure, agreements, to prevent employees from transferring knowledge skills to other companies. Additionally, firms looked to the use of software licensing, using the end-user licensing agreement (EULA) to prevent unauthorised copying, disassembling or distribution of their copy of the software. Many users remain unaware that their purchased software is only a licence to use, and not actually ownership of the software; the licence comes with a restricted set of rights that are unsurprisingly weighted in favour of the interests of the software corporation. This monopolisation of the extensibility of the software in effect reinforces a consumer/producer binary predicated on the idea that the software is supplied as a finished product to be bought and sold.[13]

In trying to understand the micropolitics of the FLOSS groups it is important to understand the first challenges that they faced with software sharing and with the realities of its rising economic importance. These were particularly manifested in the corporate belief in containing the spread of code as software *knowledge* (internally through copyright and non-disclosure agreements) and stabilising the product as finished and sellable as exchange value by closing down *software re-appropriation* (externally by preventing software being used through EULA and Terms of Service agreements). As software increased in value and it became a key profit source for corporations, procedures and processes were introduced to protect the leakage or loss of commercially sensitive information (IBM is perhaps the most famous example of a company previously taking profits through hardware and now increasingly profiting through software services, patents and consulting). Often forgotten, however, is the fact that, historically, computer software has always involved a hidden economy of software exchange, mostly ignored or unrecognised by the employer or manager. Writing software requires almost continuous efforts to 're-invent the wheel', implementing the same

kinds of algorithms, functions and techniques on similar data and information. Computer programmers have always phoned friends, emailed for advice, talked to other developers, drawn from shared knowledge resources (such as programming textbooks) or logged on to collective knowledge repositories (such as Usenet-type mailing lists and their previous incarnations as computer message boards and old code repositories). Indeed these practices have been commodified through the use of developer 'clubs' and associations, usually formed by the software manufacturer; these include Apple Developer Connection[14] (ADC) and Microsoft Developer Network[15] (MSDN), both of which cost over $1,000 to join and are aimed at corporate clients.

Nonetheless, developers are sympathetic to, if not always openly supportive of, the capitalist economy and to a large extent have accepted the proprietary software companies' practices (although if these developers are compliant formally, their practices suggest something else, as the widespread use of copied or 'pirated' software demonstrates). Indeed, these proprietary companies directly employed many computer developers and programmers in the first place and they generally accept that the exchange of their labour for wages legitimates the fact that the intellectual property rights of the created code belong to the employer (although that wouldn't preclude using a copied version of the software at home). Nonetheless, it is critical to understanding the importance of labour in the construction of computer code when assessing the way in which it is politicised by FLOSS. As companies have sought to protect their perceived intellectual property, so they have steadily increased the employment contractual agreements, patents, copyrights and non-disclosure arrangements in order to intimidate staff from 'leaking' information (for example, Microsoft employees now have to sign a weighty non-disclosure contract upon taking up employment).[16] Unsurprisingly when the company begins to claim the contents of your knowledge and skill this can lead to worrying problems in terms of being able to seek alternative employment. After all, it becomes difficult to differentiate between the IPR claims of the corporation and

the individual's right to use their skill and knowledge in their work.[17]

So the fact that software is an incredibly labour-intensive activity that requires a high degree of skill and training[18] means that efforts to control the flow of information and knowledge across the boundary of the firm are difficult to police. Moreover computer programming depends heavily on craft skills and key individuals to produce the majority of the systems' work; this was immortalised in the hard-won lessons of *The Mythical Man Month* written by Frederick Brooks Jr (1995) and fictionalised as overworked exploitation in *The Soul of a New Machine* by Tracy Kidder (2000). Efforts across the computer industry to standardise, rationalise and manage technology projects are still, relatively speaking, in the early stages of development and many new programming technologies that assist developers should be understood with reference to the needs of economic efficiency and intellectual property protection; these include modelling languages,[19] Agile Programming,[20] Object Oriented techniques[21] and project management tools.[22] Nonetheless, managing a technology project continues to remain very much an art of balancing often eccentric and individualistic programmers (DeMarco and Lister 1999), who will seek exit with their knowledge and skills, with the collective needs of a corporation, which needs to centralise and control knowledge and information. This unarticulated battle between two opposing interests, which did find voice from time to time within the major companies, was increasingly brought into public view through the activities of the free software movement.

From Free Software to Open Source

The free software movement traces its roots back to the 1970s, and the activities of an eccentric American software developer named Richard M. Stallman (the founder).[23] Stallman was among the last of the 'true hackers' identified by Levy (1994) mainly due to his programming skills, and at the time working for MIT as a programmer.[24] He had a history in the open shared-programming

environments that were the norm in early computer science labs, where he began to envisage a computer system that was not held in proprietary hands. In an example of the apocryphal stories that are often used in hacker circles, Stallman was apparently incensed when a fellow programmer refused to share the source code to a laser printer 'driver'[25] that had an irritating and easily rectified fault. This was due to the fact that the driver programmer had signed a non-disclosure agreement with the printer manufacturer that prevented him from distributing the source code. For Stallman it was anathema to programming ethics to hide or refuse to share source code, particularly when it would help another person fix a problem. For Stallman this was proof that computer engineering was taking a turn away from the hacker ethics of sharing software and ideas towards that of privatised knowledge and emasculated users (Williams 2002: 8–10).[26] Additionally, at about the same time, MIT, his employer since 1971, began to implement various technical and management processes (such as password-protected user accounts) to improve productivity in the lab where Stallman worked. In disgust he resigned in 1984 stating '[f]rom that day forward, I decided this was something I could never participate in ... I decided never to make other people victims just like I had been a victim' (Williams 2002: 12).

Stallman recognised that the principles and ideals behind sharing software developed out of the practices within the computer science laboratories of Stanford, Berkeley, Carnegie Mellon and MIT universities and were widely shared by hackers and expert computer programmers (Markoff 2005). These laboratories shared the collegial methods of working across academia, namely sharing information, peer-review, debate and criticism, the principles of academic freedom and the research ethic, sometimes subsumed under the phrase 'hacker ethic' (Himanen 2001). In order to safeguard these values, Stallman penned a 'Gnu manifesto' in which, as he explained:

> GNU, which stands for Gnu's not Unix, is the name for the complete Unix compatible operating system which I am writing so that I can give it away for free to everyone who can use it. Several other volunteers are

> helping me. Contributions of time, money, programs and equipment are greatly needed.
>
> (Stallman 2002: 31)

The call for assistance was posted to the Usenet newsgroup *net.unix.wizards* on 27 September 1983 at 12.30am and was taken as an idealistic call to arms[27] (Williams 2002: 89). The project of writing a completely new operating system from scratch, for free, by volunteers was completely unheard of. Given that operating systems are one of the most complex assemblages of software imaginable (and running into millions of lines of code, perhaps therefore more accurately described as *unimaginable*), Stallman's breadth of ambition in seeking to implement the project would have impressed any programmer reading the list. However, even at this point, despite the call to respect the hacker ethic of sharing source code and giving users the fullest freedom to use the software possible, the manifesto explained:

> GNU is not public domain. Everyone will be permitted to modify and redistribute GNU, but no distributor will be allowed to restrict its further redistribution. That is to say, proprietary modifications will not be allowed. I want to make sure that all versions of GNU remain free.
>
> (Stallman 2002: 32)

In October 1985 the newly founded Free Software Foundation (FSF), which was committed to the project of supporting free software, launched the GNU project. After much consideration Stallman had accepted that in order to prevent proprietary companies from 'closing' the source (that is, preventing the distribution of their modifications to the source code by only distributing binaries) some mechanism would need to be devised to enforce the principles of software sharing. This pointed to the use of some form of copyright licensing, as the public domain (*res nullis*) would allow anybody to take the source code and use it however they liked. Stallman was at first critical of the use of copyright for software, which enforced a notion of individual property right in software at the expense of the collective/communal.[28] Nonetheless, in 1985, with assistance from the legal

counsel to the FSF, Eben Moglen, himself a professor of law, the first version of the GNU *Emacs* licence was created.[29] The principle used was that of a copyright notice that declared that the software was owned, and then a contract which permitted certain uses of the software source code, namely that it could be distributed, copied and modified providing these modifications were released openly, and also that any subsequent or derivative works were also licensed under the same licence. The 'viral' nature of this licence was not seen to be particularly radical at the time, especially as the licence only applied to the *Emacs* text editor. In a spirit of hacking, Stallman called for suggestions and modifications for the licence to improve its usage. Immediately others began to see the value in 'porting' the licence to other software and so the specific software name mentioned in the copyright notice, *Emacs*, was dropped in favour of the generic 'software'. It took four years for the Free Software Foundation to release the first version of the General Public License (GPL), as it was named. Stallman versioned the licence in the same way that software was tracked in software projects by a major and minor numbering system yielding version 1 of the GPL. The preamble was explicit in stating its purpose:

> The license agreements of most software companies try to keep users at the mercy of those companies. By contrast, our General Public License is intended to guarantee your freedom to share and change free software – to make sure the software is free for all its users … When we speak of free software, we are referring to freedom, not price. Specifically, the General Public License is designed to make sure that you have the freedom to give away or sell copies of free software, that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new programs; and that you know you can do these things … To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate certain responsibilities for you if you distribute copies of the software, or if you modify it.
>
> (Free Software Foundation 1989)

In other words, the GPL created a communal system of software that, although formally owned (usually by each developer), actually enabled the co-operation and sharing of software. However, to enable the FSF to guarantee its assignment of copyright it required that contributions to the GNU operating system in particular were transferred to the FSF. Nonetheless, other software written to contribute to software sharing could still be licensed freely using the GPL and many people took up the opportunity to do so. The legal space that is created by the GPL is generally held to be a 'commons' that is brought into existence through a clever legal 'hack'. GNU is held to be owned in common (that is, all are equally free to draw from it), but actually it functions rather like a *res universitatis* as it bears the structure of commons on the inside and of property on the outside. It might therefore be more accurate to view the GNU project as a guild-like or co-operative structure that gives the right to share and use to others but retains ownership within the FSF (this point will be discussed later). This move centralises the copyrights in the non-profit FSF organisation and clearly designates both ownership and intention should legal problems arise. Stallman has explained that this is necessary to be able to fight inevitable challenges to the legitimacy of the GPL and also to protect against the submission of 'bogus' or 'tainted' code that might jeopardise the entire GNU project due to copyright infringement (either from malicious spoilers, or due to accidents or ignorance). It also allows control of the project to remain vested in the Free Software Foundation and helps prevent forking, a move criticised by Raymond (2001), the founder of the 'other' FLOSS movement, open source, as 'cathedral building'. However, there is no requirement for all GPL-licensed code to be transferred to the Free Software Foundation; indeed there is a huge quantity of software that is licensed in this way which is not explicitly controlled by the FSF. Work that is licensed separately from the FSF, and where copyright has not been transferred, is more accurately defined as *res privatae*, that is as privately owned code that is licensed for others to use (see Creative Commons below). Stallman described it more colourfully in 1986 as '… a form of intellectual jujitsu, using the legal system that software

hoarders have set up against [hackers]' (Williams 2002: 127) and later in 2002 as:

> The job was that of legislating for a new society, but since I wasn't a government, I couldn't actually change any laws. I had to try to do this by building on top of the existing legal system, which had not been designed for anything like this.
>
> (Williams 2002: 129)

This new form of software sharing was soon christened 'copyleft', which Stallman claimed as 'all rights reversed', and it has proved to be a remarkably stable base for this form of peer-produced software. The numerous free software and open source projects that came into existence were produced, sold, distributed and used by others through the use of this copyright licence (and others). Open source and free software are released under the terms of what are now known as public licences. The GNU General Public License (GPL) gives the user the rights to have copies of the human-readable source code along with the functional binaries and ensures that all future derivatives of the work must also be released under the terms of the licence. This is the infamous 'viral'[30] characteristic of the public licences which safeguard the appropriation of the source code into commercial proprietary products that are not themselves released with the source code.[31]

Indeed, part of copyleft's success has been down to the fact that the GPL licence was not intended to prevent commercial usage or distribution. In fact it leaves the choice about how the free software is delivered and whether it costs money to the distribution company (Kesan and Shah 2002; Moody 2002).[32]

One of the key components for any operating system is the kernel, and in free software this was conspicuously missing. The kernel is arguably the most important part of the operating system as it schedules tasks, manages key communications between parts of the system and the underlying hardware and provides a crucial abstraction layer and communication circuit. It is also one of the most complicated and difficult parts of the operating system to write. The GNU project had libraries, compilers, text editors and a Unix shell, but no kernel; after a failed attempt to convert freely

available academic kernels (first, *Trix* developed by Professor Steve Ward at MIT, and later the *Mach* microkernel developed at Carnegie Mellon University) into a workable GNU version in 1990, the FSF began work on the GNU Hurd kernel. In this case, the GNU Hurd turned out to be the Achilles heel of the free software movement: delivery dates slipped time and again as unexpected problems continually cropped up during development. Of course that did not stop the GNU tools (such as *Emacs*) being widely used in other implementations of operating systems such as FreeBSD, and as utilities that were widely available.

The free software movement also had a problem with its economic base. Its members were volunteers who funded themselves through consulting work (including Stallman). This meant that there was only limited time for campaigning and very little support for developing the free-software operating system, especially the kernel. However, in 1990, the John D. and Catherine T. MacArthur Foundation granted Stallman a 'genius grant' as a $240,000 award.[33] This helped place the organisation, and particularly Stallman, on a firm financial footing. It also meant that now he could take more time out to evangelise around the world to increase knowledge and awareness of the GNU project. One of the unforeseen consequences of this was that Stallman visited Finland as part of his tour of universities and spoke about the importance of free software to computer science students, including Linus Torvalds, the creator of Linux. Torvalds recalled that although the political and ethical call to arms didn't really inspire him, he saw the 'underlying logic: no programmer can write error-free code. By sharing software, hackers put a program's improvement ahead of individual motivations such as greed or ego protection' (Williams 2002: 136).

Torvalds, then a young Finnish computer science student, was studying at Helsinki University. As part of a project for his computer science degree, and in order to enable him to run Unix on his home computer, he began work, rather ambitiously, on a simple Unix kernel operating system. This was after his request to the developer Professor Andrew S. Tanenbaum at the University of Amsterdam to use an academic teaching aid, *Minix*, to form the

basis of his project was rejected because Tanenbaum considered it a teaching tool. Instead Torvalds wrote the software as a very basic kernel and then did something rather radical. In a similar fashion to Stallman, Torvalds posted the source code as version 0.01 onto an FTP site and posted an appeal to others to help develop the software:

> Hello everybody out there using minix – I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones. This has been brewing since April, and is starting to get ready. I'd like any feedback on things people like/dislike in minix, as my OS resembles it somewhat (same physical layout of the file-system (due to practical reasons) among other things).
>
> (Torvalds 1995)

This version was barely usable and was more a call for help and interest than a serious attempt to distribute the operating system. But within six months a workable system (although still requiring *Minix* to get it going) got a number of people very interested. After all, the promise of a free-software kernel had been sitting on the table for many years without any successful releases; without it there could be no true free-software operating system. On 5 October 1991, Torvalds emailed a newsgroup:

> Do you pine for the nice days of minix-1.1, when men were men and wrote their own device drivers? Are you without a nice project and just dying to cut your teeth on an OS you can try to modify for your needs? Are you finding it frustrating when everything works on minix? No more all-nighters to get a nifty program working? Then this post might be just for you :-).
>
> (Torvalds 1995)

> As I mentioned a month(?) ago, I'm working on a free version of a minix-lookalike for AT-386 computers. It has finally reached the stage where it's even usable (though may not be depending on what you want), and I am willing to put out the sources for wider distribution. It is just version 0.02 (+1 (very small) patch already), but I've successfully run bash/gcc/gnu-make/gnu-sed/compress etc under it.
>
> (Torvalds 1995).

Version 0.02 was released as the 'official' version of Linux (as Torvalds' operating system kernel came to be named)[34] and posted to an FTP site. In January 1992, it was licensed under the GNU GPL and released as version 0.9. But it was version 0.95, released in March 1992, that is considered the first usable version of the operating system. It excited developers and users alike – although Tanenbaum remained critical: 'I still maintain the point that designing a monolithic kernel in 1991 is a fundamental error. Be thankful you are not my student. You would not get a high grade for such a design :-)' (Tanenbaum 1992).

The rise of Linux was swift when combined with the distributed contributions of a number of other amateur hackers and computer programmers. However the key was the merging of the Linux kernel with the GNU tools developed by the Free Software Foundation and the fact that the other Unix systems, such as FreeBSD were mired in legal uncertainty.[35] This package, sometimes referred to as GNU/Linux, enabled the operating system to be launched, distributed and freely available across the Internet. This was despite the fact that Stallman has considered the Linux kernel a 'cuckoo' in the GNU operating system, partially due to Torvalds' lack of interest in the general philosophy of the free software movement – as demonstrated in the recent debates over upgrading the GNU GPL to a world of Software as a Service (SaaS):

> Well, you do have to realise that Linux has never been an FSF project, and in fact has never even been a 'Free Software' project … I personally have always been very clear about this: Linux is 'Open source'. It was never a FSF project, and it was always about giving source code back and keeping it open, not about anything else … Linux from the very beginning was not about the FSF ideals, but about 'Full source must be available'.
>
> (Torvalds 2006)

At issue here is the distinction between two different views as to the best way of producing and licensing FLOSS software. On the one hand, Stallman is committed to a community of artisan producers and is anxious that free riders are not able to take the community-created code and 'close it' into proprietary versions. On the other, represented here by Torvalds, is the argument that the inherent

efficiency of an open-source approach means that it would always beat proprietary techniques, and therefore the protection afforded by a GPL licence through copyleft is a distraction.

This dispute centres on the difference between free software and open source and the best way to maintain code as both a shared resource and as a set of social practices. Code in Linux, although licensed under GPL, is in fact a distributed form of copyright materials. This is very different from the conception of a *res universitatis* where the ownership of the code is held in trust by a corporate body (in the Free Software Foundation Gnu model). In other words, each individual contributor can claim ownership of a small portion of the Linux code but freely licenses others to use it. This has a potentially double danger: (1) relicensing of the code is extremely problematic, for example when the GPL is changed or a flaw is found in the existing licence;[36] (2) the code is open to hacking or tainting by others. This could be the case where copyrighted code is included in the kernel build (that is, the source code) without the actual permission of the copyright holder. This would leave the Linux kernel open to attack for reasons of copyright infringement. This is the reason that the FSF requires the author of the code to sign over the copyright in order that it can claim copyright and hence litigate infringement cases, but also avoid this kind of problem.[37]

The Commercialisation of Floss

Here I would like to spend a little time outlining some of the key movements and connections between the FLOSS groups and the commercial sectors. In particular I wish to draw attention to the economic reality of FLOSS development and its reliance on venture capital, and external funding from large multinationals and the stock market. To do this I give a broad overview of the historical development of FLOSS, particularly connected with the Linux system, and show that not only was funding offered, it was actively courted.

It began in June 1992, as commercial distributors began to get interested in Linux as a software commodity and Torvalds

confirmed that he wanted them to distribute the software. By November 1992 the first distribution Yggdrasil[38] was made commercially available both as a sellable physical product (the first CD-ROM version distributed) and as a download.[39] The early versions of Linux were very much 'homebrew' and amateur affairs. Indeed, this distribution was noted for the inability of this 'plug-and-play' Linux to run from the CD-ROM as claimed. Nonetheless it was an important move in the creation of a sellable product and served to demonstrate the potential for an easy-to-install and -use Linux-based operating system. By March 1993, however, in response to comments from the free software movement about the commercial turn in Linux, Torvalds wrote an email about taking 'advantage of the GPL to make a quick buck' stating 'please, don't bitch about commercial uses' (Torvalds 1993). But by March of the following year Linux became the most popular software CD sold by the distributors, selling 50,000 copies a month and encouraging other companies into the market. This led to a growth in distribution companies that bundle together the components of the GNU system onto discs and provide support and training for the fledgling operating system. This commercial turn in the Linux community was new in free software production which had until this point been largely ignored by companies. Mainly used in small start-ups and in the non-profit or education/academic environment free software had garnered a lot of support, improvements and documentation over the term of its development. It also had remained very much a community-led project with a quite distinct meritocratic attitude to the development and management of the source code 'tree'.[40] It is no surprise that many of the older members of the free software community were concerned about Linux's commercialisation and what it might signify for free software.[41]

Between 1994 and 1996, things began to change rapidly as large corporations such as Digital Equipment Corporation (DEC), a US computer company, began to invest money in Linux development. For example, they supplied funding capital for two porting projects to move Linux to the DEC Alpha chip, giving Torvalds free technical training, support and a free Alpha workstation. Marc

Andreessen and Jim Clark founded Netscape Communications at the same time to try to create a successful software corporation based on the World Wide Web. Its product, the highly successful Netscape Navigator, gave a crucial friendly (if proprietary) face to the Internet. By October and November 1994, the 'distro' companies Caldera (funded by Ray Noorda, the former CEO of Novell) and Red Hat were launched. In June 1995, Red Hat was bought by ACC Corporation and in December 1996, Torvalds released Linux version 2.0. During this time, sales of the GNU/Linux operating system grew rapidly, with Linux offering a real challenge to Microsoft Windows, particularly on infrastructural projects (such as networks, email servers and web services). One of the most potent advantages was of course the lack of licence fees, meaning that corporations, schools and universities could make huge cost savings. However Linux suffered from a heavily technical interface, usually command line, and an arcane and complicated setup process not conducive to desktop use.

In 1997, Torvalds was awarded the Nokia Foundation Award of FIM 50,000 (worth approx US$10,000) and the Uniforum Lifetime achievement award. He was also offered a job with Transmeta (funded by Paul Allen, co-founder of Microsoft) in which he would be free to work on Linux and support the project – and so he moved from Finland to the US.[42] During that year, Intel and Netscape invested millions of dollars in the company Red Hat, which now specialised in supplying GNU/Linux-based services, customisation and support. Money was now pouring from the various Silicon Valley venture capital funds and large corporations into a market bubble that signalled the start of the dotcom era.

During 1997, a group of industry insiders, advocates and eccentrics sat round the table of the CEO of VA Linux, Larry Augustin, and declared that free software should henceforth be known as 'open source'.[43] This was part of the process that led to the Linux IPO goldrush and the creation of 'open source pigs' as Metcalfe (2004) called them.[44] Bruce Perens, Eric Raymond, Ian Murdock, and Tim Sailer formed the non-profit organisation the Open Source Initiative (OSI), to formalise the 'open source'

term in early 1998. Eric Raymond then began a campaign against the Free Software Foundation, stating in a Salon.com interview:

> The name 'free software' has to go. The problem is nobody knows what 'free' means, and to the extent that they do think they know, it's tied in with a whole bunch of ideology and that crazy guy from Boston, Richard Stallman ... We need to be making arguments based on economics and development processes and expected return.
>
> (Leonard 2000)

The break between the free software movement and the new open source group was not just about strategy, it also had political and practical motivations (and is looked at in depth in the next chapter). One of the chief objectives was to cut the success of free software from the Free Software Foundation and particularly the idiosyncratic leadership of Richard Stallman (who viewed the project of free software as being principally about human rights and political emancipation). Instead, the open source movement actively sought to court business leaders and companies and began proclaiming the money-saving technical efficiencies of open source as a development methodology with low labour costs. Raymond (2001)[45] advocated a more explicitly business-friendly narrative for free software (now christened open source) arguing that many working together and checking each other's work makes it easier to avoid error, or 'given enough eyeballs, all bugs are shallow' which he termed Linus' Law (Raymond 2001: 30).

Meanwhile, Netscape Communications was under intense pressure from the success of Microsoft's recently launched Internet Explorer browser software announced in 1998. Netscape announced that it would be making its browser available under an open source licence, a huge coup for the open source movement (and Raymond in particular). Netscape had been in severe trouble from the so-called browser wars with Microsoft (who gave their new browser away for free) and was desperately trying to save its profitable server market. This open source codebase was known as the Mozilla project, and was released initially under the Netscape Public License (NPL). Netscape was later swallowed by AOL

for $2.4 billion, and AOL successfully prosecuted a legal case against Microsoft in an antitrust case that paid out $750 million to AOL. As part of this agreement AOL was required to install Microsoft Explorer, and so AOL disbanded Netscape. Nonetheless Mozilla was re-organised as a non-profit foundation in 2003 and continued to release the now renamed Firefox browser under the GPL licence (up to the present day). Additionally, a major change to technology and copyright law was inaugurated by the passing of the Digital Millennium Copyright Act 1998 (DMCA). This law criminalised the distribution of technology that can be used to 'circumvent' technical measures that prevent copyright infringement. The term of copyright was also extended in the Copyright Term Extension Act 1998 (also known as the Sonny Bono Copyright Term Extension Act) which increased the term of copyright from life of the author + 50 years (75 years for corporate owners) to life + 70 years (95 years for corporate owners) and also retrospectively increased the copyright of works published before 1 January 1978 by 20 years.

In 1999, IBM began to take note of the FLOSS phenomenon and also started investing in Red Hat and other distributors, and Tim O'Reilly began talking about the profitability of selling books for the technical open source market through his publishing company, O'Reilly Associates, which has made millions since it was founded. Infighting and arguing began to plague the open source movement when Bruce Perens resigned from the Open Source Initiative (OSI), after declaring that book publisher O'Reilly is 'one of the leading parasites [sic] of the free software community' (Shankland 1999). Bruce Perens was a lead developer on the Debian Linux Distribution and contributed to the Open Source Definition in 1998. A year after co-founding the Open Source Initiative with Eric Raymond, Perens left citing differences of opinion and argued that 'It's Time to Talk about Free Software Again' (Perens 1999, original capitalisation). Soon afterwards, Red Hat floated in the first Linux-related initial public offering (IPO) which gave Red Hat a market capitalisation of $4.8 billion and granted Torvalds windfall shares, which he sold, becoming an overnight millionaire. In an attempt to include many of the developers and

contributors to the Linux software, Red Hat offered all code contributors the option to buy shares at a special reserved offer price. Unfortunately many of the hackers did not have the kind of credit records that allowed them to register (Ananian 1999). The fact that credit checks disenfranchised these developers from the Linux IPOs – even though their freely contributed labour was recognised as having made the Linux system possible in the first place – showed how Linux and FLOSS were beginning to move from the needs of the developers to those of the corporate world. This was the end of the purely volunteer-based Linux project and the beginning of a more corporate-supported and -funded project. It was also the time of mergers and acquisitions across the software industry associated with Linux and FLOSS software. For example in November 1999, Red Hat announced it was buying Cygnus Solutions for $700 million and in December VA Linux floated on an IPO, a $7 billion market capitalisation, setting the record for the biggest IPO rise in the history of NASDAQ. It led Raymond to exclaim in an email: 'A few hours ago, I learned that I am now (at least in theory) absurdly rich' (Raymond 1999). He was now worth $36 million on paper.[46]

Microsoft too had also begun privately to acknowledge Linux, albeit as a threat:

> Open source software poses a direct, short-term revenue and platform threat to Microsoft – particularly in server space. Additionally, the intrinsic parallelism and free idea exchange in open source software has benefits that are not replicable with our current licensing model and therefore present a long-term developer mindshare threat.
>
> (Valloppillil, quoted in Shankland 1999)

By February 2000, VA Linux had bought Slashdot and Freshmeat, two popular hacker websites. Slashdot in particular was a very important website which set the news agenda for the technorati of developers, hackers and other interested parties. Caldera Systems also launched its IPO, which doubled in value from its initial offer price of $14, even though the SEC filing stated 'Caldera knows of no company that has built a profitable business based in whole or in part on open-source software'. Reminiscent of

the South Sea Bubble market boom in 1771 that included many bizarre proposals for share issues including a 'Company for the carrying on an understanding of great advantage, but nobody knows what it is' (MacKay 1995: 55). By now there were over 140 Linux distribution companies across the world.

In August 2000, HP, IBM, Intel and NEC, with additional support from Caldera, Dell, Linuxcare, LynuxWorks, Red Hat, SGI, SuSE, TurboLinux, and VA Linux Systems, announced the formation of the 'Open Source Development Lab' that would make expensive hardware available to Linux developers for free. These companies realised that by employing key individuals and supplying free hardware it would be easy to guide the development of open source projects. It also meant that the priorities for development could be made to fit the requirements of the software market rather than the democratic mass of developers. In 2000 the RSA patent allowing the secure transfer of security information on free software platforms expired. The RSA is a computer algorithm for the encryption of data and communications. The letters RSA are the initials of the inventors' surnames: Ron Rivest, Adi Shamir and Len Adleman, who all worked at MIT. The patent expired on 21 September 2000 and was a key technology for the commercialisation of the web, providing the ability to secure data on free software based computers.

By December, IBM announced its plans to invest $1 billion in Linux-based outsourcing in 2001. The Transmeta IPO signalled the end of the Linux IPO wave in 2000 and Torvalds was awarded an 'undisclosed amount' of Transmeta shares in the flotation. The European Union's 'Framework Programme 5', was then launched; this was a €3.6 billion effort to improve the competitiveness of the European software industry. The programme recognised the importance of open source software, and money was made available for projects that advanced the programme's objectives. Siegmar Mosdorf, the German Secretary of State in the Federal Ministry for Economy and Technology, voiced the German government's support for open source, claiming 'I am convinced that open source development can form the European base model in the information age' (Gillespie 2000). IBM also announced

plans to spend $200 million over four years to make it easier for European companies to bring its software to Linux. Forrester Research estimated that more than 55 per cent of the world's 2,500 biggest firms used open source software, with almost a quarter using the software in production systems (Connor 2000). Free software had come a long way from its hacker origins.

A Microsoft vice president, Jim Allchin, argued that government-sponsored software should not be released under GPL and that the GPL might stifle innovation. This was later described as FUD ('fear, uncertainty, and doubt') as Microsoft simultaneously launched 'shared source'.[47] Microsoft began to publicly attack the threat to its business interests from Linux, and Steve Ballmer, the corporation's CEO, was quoted in an interview in the *Chicago Sun-Times* as saying 'Linux is a cancer that attaches itself in an intellectual property sense to everything it touches' (Greene 2001).

In 2002, Peru began to consider legislation to require the use of free software within the government citing the importance of: (1) free access to information; (2) permanence of public data; and (3) security of the state and its citizens. Responding to Microsoft's formal complaint about this move, Villanueva Nuñez, a congressman in Peru, noted:

> In addition, a reading of [Microsoft's] opinion would lead to the conclusion that the State market is crucial and essential for the proprietary software industry, to such a point that the choice made by the State in this bill would completely eliminate the market for these firms. If that is true, we can deduce that the State must be subsidizing the proprietary software industry. In the unlikely event that this were true, the State would have the right to apply the subsidies in the area it considered of greatest social value; it is undeniable, in this improbable hypothesis, that if the State decided to subsidize software, it would have to do so choosing the free over the proprietary, considering its social effect and the rational use of taxpayers' money.
>
> (Villanueva Nuñez 2002)

The EU also released its sixth Framework Programme announcing the funding of scientific research and development that would

succeed in 'enabling the Union, within the next ten years, to become the world's most competitive and dynamic knowledge economy' (EU 2006). At the same time, the Motion Picture Association of America (MPAA) filed a case in Norway against Jon Johansen, co-author of the DeCSS source code that allowed Linux-based systems to play DVDs.[48]

This brief overview of the market dotcom bubble in the late 1990s demonstrates that the FLOSS groups generally are closely connected both to government and to corporate bodies. Indeed, as a commercial operating system, Linux is hardly 'free' in costing little or no money. Red Hat, SuSE and other distributors have built very profitable businesses based on offering Linux service and support. Indeed IBM has been incredibly successful in using Linux to offset the development costs of producing software. For example, IBM claimed more than $2 billion in Linux-related revenue in 2003 (Ness 2004). In 2005, IBM announced that the company's business model was shifting from goods and products to software and services (worth over $6 billion in investment). And in February 2005, IBM pledged $100 million in Linux-related investment and support (Cowley 2005).

Although it might have looked like the 'free software' moniker was attractive a few years ago when many firms were slashing their IT budgets and saving costs by avoiding licensing fees, it has become very clear that 'free software' involves many significant costs that are often hidden behind the discourse of FLOSS software. Indeed, free software has made many of its adherents fabulously wealthy and generated the foundations for the Web 2.0 companies now proliferating across the world. Furthermore, the technology that makes Linux useful for business applications has come largely from IBM, HP, Sun and other large corporations who fund the research and development by selling other products and services. Indeed the success of Linux is largely due to the decision by IBM and HP and others early on not to port their own Unix implementations over to Intel Xeon-based servers. It has produced a constellation of software tools that, if it were rebuilt from scratch using proprietary methods, would cost more than US$1 billion (Wheeler 2002). Many of the leading 'hackers' and

developers associated with Linux now have jobs at companies like IBM, Red Hat, Sun and the like where their salaries are now directly dependent on their ability to extract profits from intellectual property.

The Politics of Code

Copyright was long considered a rather esoteric backwater of legal theory, and it is only in the past couple of decades that it has risen up the political agenda. Through a political economy approach it is clear that the economic imperative to both protect existing intellectual assets (many of which were due to fall out of copyright) together with the rush to an informational economy have changed the way in which both governments and corporations understand economic life (as discussed in Chapter 2). Further, the expansion in intellectual property rights across the entire field of law and the diminution of the rights of the public against that of private interest has raised a number of conflictual situations (mostly concerned with patents and drug corporations) but haven't become serious public issues. In this section I want to trace the way in which the politics of copyright (and other intellectual property rights in general) have moved from the technical realm (as consensual issues of a technical or legal nature) to the broader political sphere. For this, the ontology of copyright has been forced into the open, and the taken-for-granted nature of the legal field has become a site of contestation and competing interests.

I would now like to look at some of the conditions under which FLOSS came into existence, and understand why these forms and not others became popular and successful. It is clear from the preceding section that FLOSS grew from many practices that were developed before the commercialisation of the software industry. In this it shares a particularly academic and non-monetary origin in the US Defense Department-funded public-sector projects that were based mainly on university campuses. However, it is also drawn from the particularly social nature of the programming

endeavour, and the way in which computer software is a uniquely collaborative engineering effort.

One of the most important conditions for programming is that one must first be taught the principle of computer programming, that is, the methods for combining a set of programming abstractions that give the general procedures for producing logical algorithmic instructions. In schools and universities these skills are taught with general cases and often abstracted from a particular programming language (sometimes using so-called modelling languages such as Z or UML). When these skills are understood, they may then be implemented into any particular programming language (C++, Java, Pascal) but to do so, the language itself has to be taught too.

Here the use of previous examples, common routines and structures and shared libraries are commonly distributed in order that the programmers can see the way in which a particular language functions. These routines and structures are often very widely used, with some of the routines having been implemented over and over again as teaching and learning devices. Here we will look in particular at the classic algorithm, the bubble sort.[49] The 'bubble sort algorithm' is interesting because it is both a universal teaching tool, often one of the first things taught on computer science and programming courses. However, although its history is unclear and original authorship has been difficult to trace (Astrachan 2003: 1), this has not stopped some companies from seeking to close the code through copyrights and patents.[50]

The bubble sort is a simple sorting mechanism that takes a list and puts the elements that make it up in order by running in a loop and comparing each item in turn with the one above and swapping them around in the list if one is bigger than the other. Essentially the smaller numbers 'float' to the top and the larger ones 'sink' to the bottom (bubbles float to the top). Although it is not the most efficient sorting algorithm, it is relatively straightforward for teaching purposes and is small enough to give students a grounding in an important part of programming processes – namely, using lists, datatypes, loops and conditionals to process data. Consequently, it is still widely used.

To avoid talking about a particular instantiation of a programming language, algorithms are often written out in 'pseudo-code', that is in a non-computer, non-compilable language that is computer-like but still contains enough English to be readable. This is the pseudo-code for a bubble sort, of the type that would be given to students to implement in a particular language.[51]

```
function bubblesort (A : list[1..n]) {
    var int i, j;
    for i from n downto 1 {
        for j from 1 to i-1 {
            if (A[j] > A[j+1])
                swap(A[j], A[j+1])
        }
    }
}
```

This is the generic programming algorithm, the abstract non-instantiated idea of the process to be undertaken in software (or hardware). This is how the algorithm would often be taught in class, and reproduced in literature surrounding programming that appealed across different computer language communities. However it is here that the problem of using copyright for code begins. For in the above pseudo-code, due to copyright legislating a *mechanical* process as a *literary* production, the *machine becomes literature*. That is, the abstract idea of an informational process becomes a fixed text associated with an original author. For example, an analogy might be an attempt to copyright the workings of an internal combustion engine, or better, the *ideal* process of the workings of an internal combustion engine.

Yet it is of crucial importance to the production of computer software that one may continue to implement this Ur-code, and why the copyrighting and patenting of computer software is becoming increasingly a problem to these FLOSS groups.[52] To take our example further, the pseudo-code would then be implemented in specific computer languages, which could include Java[53] or Perl.[54]

A Java implementation of bubble sort

```
public static void bubbleSort(int[] data)
{ boolean isSorted;
   int tempVariable;
   int numberOfTimesLooped = 0;
   do
   { isSorted = true;
      for (int i = 1; i < data.length - numberOfTimesLooped; i++)
      {
         if (data[i] < data[i - 1])
         { tempVariable = data[i];
            data[i] = data[i - 1];
            data[i - 1] = tempVariable;
            isSorted = false;
         }
      } numberOfTimesLooped++;
   } while (!isSorted);
}
```

A Perl implementation of bubble sort

```
sub swap {
   ($_[0], $_[1]) = ($_[1], $_[0]);
}

sub bubble_sort {
   for ($i=$[; $i < $#_; ++$i) {
      for ($j=$[; $j < $#_; ++$j) {
         ($_[$j] > $ _[$j+1]) && swap($_[$j], $_[$j+1]);
      }
   }
}
```

In both these cases, the pseudo-code or algorithm has been converted into the precise requirements for mechanical functioning of the language syntax. That is, the computer can execute this function on a series of data presented as a list to the routine, returning the list in a sorted format. Yet each of these examples

would be considered copyrighted as *literary works* under current legislation, as indeed is the binary executable (which is in reality a long list of numbers which are not even human-readable let alone expressive). Additionally, it raises the question about interpretation and implementation. After all, as computer scientists and hackers soon realised, there is not much point in continually reinventing the wheel by writing new bubble-sort routines. Once a well-written routine has been produced it is sensible and more efficient to *share* that version, allowing others to reuse it as necessary, rather than claim ownership in it. This is where the social practices of software developers begin to diverge from the commercial logic of software firms.

For the developers involved in the production of large-scale software systems, this logic of sharing becomes even more stark against the logic of capital, as the writing of code introduces inevitable bugs and glitches into the computer code. So a way of minimising these errors is by reusing code that has already been through the write-rewrite-test-release cycle and hence can be assumed to be fairly free of problems. Indeed, this is the reason that computer software libraries are widely available to developers to use. So, in a million-line software system, every bug is a huge nightmare of tracing the multiple branching complexity of modern software systems and attempting to correct them. In other words, *machine originality* is never considered to be a good unto itself, in contrast to *literary originality*.

Here we arrive at the crux of the conflict between FLOSS advocates and the proprietary software companies. Without a legal mechanism to protect software and enforce scarcity there could be no market for the software in the first place – people would just copy it. But for the FLOSS groups it is clear that previously the proprietary software companies were aware of the problematic legal issues surrounding copyright and code, and merely papered over the divisions by enforcing copyright in mainly finished products and client software. For their own developers, and to encourage other third-party developers, there would be large sharing communities with little in terms of legal restrictions on the code that was being spread. Indeed, when it is considered

that much software is the reimplementation of classic computer science problems taught in university courses and joined into larger assemblages, then the originality of the software becomes increasingly problematic.

It also became, perhaps counter-intuitively, *less* pressing as the software industry became increasingly monopolistic, with major firms able to control the environment in which developers undertook their programming assignments. Providing they remained within the walls of the operating system or application environment, copyright issues could stay largely out of the picture.[55] However, the bargain was one that would steadily break down as software rose in value and alternative ways to monetarise the software were discovered. This is often referred to as a tragedy of the anti-commons (Heller 1998), where the privatisation of tiny fragments of the code makes it impossible to implement a larger system without getting copyright clearance for every fragment. It is here that the Free Software Foundation's role as a rights-holder for the distributed project of GNU becomes significant as a means of preventing this tragedy from undermining the free software project.

Combined with moves to lock up source code by corporations, there was an increasing movement to prevent source from escaping the boundary of the firm in the first place. Sometimes referred to as the 'crown jewels', most memorably by Barlow in *In Search of NuPrometheus* (1990) when discussing the case of the illicit copying and distribution of the QuickDraw source code for the Apple Macintosh operating system in 1989 (but actually an industry-wide problem) (see Selznak 2004), source code allows the user to modify, improve and 'fork' the code in whatever direction is desired. By stripping this code from the distributed binary, the user is restricted to using the software only in ways that are predetermined through the development of *prescribed functions*. This both prevents the appropriation of the algorithms encoded within the source code, giving the distributed software a 'fixed' nature which is easier to identify (which is handy for copyright infringement cases and the like), but which also allows the control of 'downstream' development; in effect it can lock in the users by

proscribing certain ways of doing and acting with the software. This is exactly the restriction of freedom that Stallman (2004) refers to when bemoaning the loss of 'free as in freedom' for the user to choose what he or she may want to do. In the case of the NuPrometheus, a hacker had deliberately leaked the Apple source code out of the boundaries of the corporation and Apple had called in a technologically illiterate FBI to investigate – albeit with no criminal case nor a clear suspicion as to who was responsible (Sterling 1992: 220–2).[56] It did, however, serve as a marker of the beginnings of activities that take place within the confines of the computer (on the terrain of electrical transmission and storage of information) becoming both economically important and legislatively protected through civil and criminal law.[57]

The Ethics of Code

The GNU project is a network that has ethical force as well as legal protection and it is in terms of its ethical standing that it is often justified. Copyleft is ethical[58] in as much as it institutes a value of sharing within the software code community by giving a protection mechanism against others who might seek to act as free riders; any attempt to remove these rights would meet resistance in terms of both legal and community responses.[59] That is, it acts as a bounded guild, with the external use of its software guarded through the use of the copyright licence set up in the General Public License. However, it is also institutional because it is a non-profit organisation based in the US that requires that the source code is explicitly transferred into its repositories (as noted above), making it possible to enforce the requirements of the GPL.[60]

The BSD licence is the other major competitor to the GNU GPL but it has a notable difference.[61] It does not require the further development of the software to also be licensed under the same licence conditions. In other words, software developed under a BSD licence may be rolled into a proprietary piece of software. A notable example of this is the BSD implementation of the TCP/IP stack code that was copied into the Microsoft Windows operating system, although it is probable that little of

the original code remains within the operating system today, as improvements in technology would have required some rewriting of the code. Nonetheless, it remains clear that here Microsoft would have been building upon the work of others and actively acknowledging their contribution.

There has been a flurry of new types of open source licences, both Open Source Initiative-approved and not: they particularly come from the need for corporations to continue to control projects which they decide to 'open source'.[62] This has led many to argue that these licences are an attempt to recruit the army of volunteer programmers that contribute to projects as free labour (see Terranova 2004; Hardt and Negri 2004), while being able to control and own the underlying technology. This has also stimulated the FLOSS community to construct a set of values or distinct philosophy by which a true 'free culture' licence can be identified.[63]

One interesting effect of the GPL has been to alter the balance of forces in work-related negotiations. If there is a system to be created and there already exists a GPL program capable of fulfilling 80 per cent of what is demanded and which requires just a couple of months' work to complete, then this becomes more attractive for an employer rather than paying for 12 months' labour to write a proprietary version from scratch. The means of production, in brief, are not completely alienated by the employer as they are simultaneously placed within the wider FLOSS community as *res universitatis*. If information is not locked in to one workplace, but can be brought with you, then knowledge as a force for stabilisation in a labour environment otherwise characterised by uncertainty and exploitation might be less antagonistic. On the other hand, by relying on free labour (as *unpaid* labour) the firm might merely outsource the production of code – in effect paying nobody. In this case, the attraction of certain corporations to FLOSS code becomes part of a parcel of measures within post-Fordism to cut costs by not directly employing computer programmers – the current experiment with piece-work 'code bounties' which are money payments (usually in the region of between $1,000–10,000) for fixing a particular

code problem in free software (thus saving the employer the cost of pensions, holiday, illness cover, salary and so forth) is one example.

The types of user that have been identified in FLOSS projects are generally the hackers themselves and where the users are not FLOSS developers, the consequences of using these open source products have changed the way the users act toward software (particularly in regard to the disempowerment of proprietary software). Examples include alternative media (which is a large user and promoter of FLOSS software),[64] countries from the South (such as Brazil[65] and South Africa,[66] which have been lobbied by FLOSS users and developers (see May 2006)) and cultural groups that have taken up the ideas and values of FLOSS and applied them to the cultural sphere (such as Creative Commons,[67] Free Culture,[68] anti-DRM groups,[69] and No2IDCard[70] groups).

The kinds of feelings, beliefs, values and dispositions to act in a certain way are embedded within the social practices surrounding open source culture and manifested though the licences that can be said to be a form of technology of the commons. This is especially apparent in the terminology of the 'free' (free software, free music). 'Free' in this context doesn't have anything to do with 'free of charge' but rather denotes the openness of the knowledge embodied in the digital good (*libre* rather than *gratis*).

That is, they help to preserve the social practices that structure the *res universitatis* of FLOSS, but also a wider sense of a commons or *res communes*. These are combined with the discourses which are important to the FLOSS community and present a kind of corpus of texts that help organise the way in which certain ideas (an ethic of sharing, the commons, free distribution of software and culture) are articulated and legitimated (see the next chapter).

The Economic Base of FLOSS

Generally, the field of political economy extends a neo-Marxist economic approach to cultural production, informed by Marx's distinctions between the ruling class and the ruling ideas and

the base/superstructural model. The aim has been to understand how the economic structures of society limit and shape the production of culture. Here though, I examine how software, and in particular, free/libre and open source software (FLOSS) is constrained by certain economic structures, and most importantly to see how the perceived agency of the FLOSS actors' milieu is limited by larger structural features of the entire computing/software industry (see the history of the software industry above for more information). Here, for example, the economic structure of software is increasingly shaped by those who own copyrights and patents as opposed to those who do not. It is not that the FLOSS developers are not making choices but that these choices are made within the contexts set by wider structures (usually copyright) and often contexts that they attempt to bypass through 'hacks' such as copyleft. Crucially, though, this ownership of software copyright is being critiqued through the activities of FLOSS when the question of such ownership is problematised through their practices. Copyleft clearly offers an alternative to the system of private property as *res privatae* for the developers, an alternative that they perceive as a communal system of ownership that is either shared through an organisational structure such as FSF as *res universitatis* or through a web of interconnected private property ownerships and mediated through copyleft to create a form of commons, *res communes*.

FLOSS is often conceived as being independent of or floating above the system of capital, but in fact, as shown in previous chapters, it is hugely reliant on private capital for its funding (and often for its skills, knowledge and organisational structure). It is also an unresolved legal morass, with probably millions of overlapping, contradictory and impossible-to-resolve copyright and infringement issues. Clearly, this copyleft system does work, but looking beyond the discursive justificatory and often mythical ideas about how the 'impossible' can function, it is clear that a particular logic of sharing is in operation. The technical needs of the developers, users and consumers of the information systems for which it is implemented support and drive this logic. It is also clear that this logic of sharing is at the mercy of the

state, whereby if a change in law were to make copyleft clauses unenforceable then the effects on the FLOSS projects would be immense.

In the last few chapters I have attempted to draw a rough cartography of the FLOSS movements, and to describe some commonalities and shared norms and values that inform the outlook and attitudes of those involved in these movements. Firstly, and perhaps obviously, these are highly skilled and often highly educated individuals who have a marked technical ability and a strong connection to computer-based work. Secondly, they are often young, keen to impress and to try new approaches to technical problems, often with little real-world experience outside of their technical sphere. Thirdly, they are often problem-solvers, seeking to resolve a particular issue directly affecting themselves, whether as a programming challenge, a software bug or an irritation that they would like to circumvent or 'hack'. Fourthly, they draw directly on notions of meritocracy, talent, technical ability and the rewards of a Protestant work ethic. Lastly, they are realists, believers in the project of science and rationality and extremely critical of (and sometimes downright reactionary towards) notions that problematise the common-sense or obviousness of the natural and social world.

Most of the FLOSS members are computer programmers, employed or studying in subjects related to computer science or infomatics. There are students in universities or working for corporations, big public-sector organisations, or consultancy firms – usually well paid and having stable work lives. Some are also self-employed, preferring the higher wages and constantly changing work environment that this offers. However, there are enormous variations in the amount of money that FLOSS developers earn. Indeed, this can run from student developers working on handbuilt boxes to the superstar consultants parachuted into ailing software projects on supercomputers and mainframes. Most are graduate developers, with a computer science degree not uncommon, but a university qualification does not have the same high status within computing circles as in other

industries; indeed experience and technical fluency are key skills that are keenly sought by corporations.

Computer programmers are in perhaps one of the most rapidly changing industries, with knowledge and skills continually having to be updated. This is key to understanding the impetus for FLOSS developments, as programmers must continually update, rehearse and relearn skills in a variety of different computer environments, programming languages and technical approaches with a bewildering number of interconnections and cross-pollinations. They must also, of course, meet the demands of their employers, their customers and their peers in terms of the technical solutions and software products that they produce day to day. This points to a continual learning environment that is encouraged by employers, both for the intrinsic value of keeping up with competitors, but also to stimulate the creation of new and innovative software products that can be marketed and sold.

Two major empirical studies show that these FLOSS developers are drawn from a very narrow segment of the population – generally highly educated young males with partners, rather than women, who are under-represented (Infonomics 2002: Part IV; FLOSSPOLS 2005). The Infonomics report (2002) found that only 1.1 per cent of FLOSS developers were female (1.7 per cent in a more recent FLOSSPOLS report) and that there is a clear predominance of people aged between 16 and 36 years. The study shows that only 25 per cent are older than 30, and only 10 per cent are older than 35. FLOSS developers also tend to have a high educational level with 70 per cent of developers having a university degree (either bachelors 33 per cent, masters 28 per cent or PhD 9 per cent). Unsurprisingly, both professional and university disciplines that are related to IT are the chief professional background for FLOSS developers (83 per cent); more than 65 per cent are employed, 14 per cent self-employed and 17 per cent are students. A surprisingly large proportion (71 per cent) came from EU countries, with France (16.5 per cent) and Germany (12.4 per cent) having the largest groups (the US made up 10.3 per cent, the UK only 6.5 per cent). However, when the

mobility of the developers is factored in, the US acts as a magnet for FLOSS developers from across the EU.

Interestingly, across the FLOSS community a number of different tasks and roles are carried out that are indirectly related to coding software which include (1) convincing people to use FLOSS software; (2) raising public awareness of FLOSS; (3) fixing bugs/patches/testing software and reading reports; (4) translating texts; (5) providing creative elements (graphics/designs); (6) organising workshops; (7) documenting software; (8) providing ideas for software; (9) participating in workshops; (10) translating software and manuals; (11) providing tutorials; and (12) writing how-tos and replying to online forum discussions (FLOSSPOLS 1995: 17). All of these activities can be considered important to the success of a software project as they are the support infrastructure that ensures take-up of technology and provides general support.

One of the key findings of the FLOSS report (Infonomics 2002) was that the majority of respondents (70.5 per cent) stated that the reason they used and contributed to FLOSS projects was because they wanted to learn and develop new skills, although social factors were also important, with 30.6 per cent keen to be involved in the FLOSS environment and 67.2 per cent wanting to share their knowledge with others. Interestingly, 37.2 per cent explicitly stated that part of the attraction of FLOSS was being involved in a new form of co-operation and improving the work of others (39.8 per cent). Only 12.3 per cent wanted to make money from the endeavour, although 29.8 per cent wanted to improve their employment prospects. Questions asked about the expectations they thought others might have of them showed that 72.2 per cent believed that others expected them to share knowledge and skills and 41.4 per cent also thought that they were expected to help improve other people's projects. This shows an interesting norm of sharing knowledge that is manifested within the social practices of the FLOSS groups and which perhaps helps explain how the weak bonds of internet-based development have continued to remain in place within FLOSS projects over a considerable time (for example, Linux, started in 1991, is now 17 years old and still going strong). The drive to exchange knowledge is striking, with

57.4 per cent of respondents believing this was a key purpose of FLOSS, and 64.5 per cent agreeing that FLOSS encourages greater freedoms in software development. Only 4.3 per cent highlighted career development as a purpose of the FLOSS movement and only 8.3 per cent thought that it was the aim of offering an alternative to proprietary software. When asked about the balance between their input into FLOSS projects and the amount they received in return, 56 per cent of developers thought that they took more from the community than they gave and believed others were more generous in contributing to the projects.

The difference between free software and open source is a complicated issue, particularly when explored in relation to discursive and ideological constructions contrasted with the practices of developers in their normal environment. However, it is interesting to note that 48 per cent of respondents aligned themselves with free software and over 32.6 per cent with the open source community (19 per cent did not care). However when asked about the difference between free software and open source, 52.9 per cent thought that the work was exactly the same but it was the principles that were an important difference, whereas 29.7 per cent thought that the ways in which they thought and lived differed substantially (17.3 per cent did not care). From this and the other data the following six types of FLOSS developer were abstracted into a software developer typology:

1. The first type consists of those developers who assign themselves to the free-software community and who see fundamental differences between the two communities (18 per cent);

2. The second type consists of those developers who consider themselves as part of the open-source community and who perceive fundamental differences between the two communities (9 per cent);

3. The third type is made up by those developers who assign themselves to the free-software community and who perceive only [differences in principles] between the two communities, but consider work in the two communities the same (26 per cent);

> 4. Accordingly, those developers who assign themselves to the open-source community and see principal, but no fundamental differences between the two communities provide the fourth type (17 per cent);
>
> 5. The fifth type consists of developers who assign themselves to either the free-software or the open-source software community, but are not bothered by differences between the two communities (9 per cent);
>
> 6. And finally, those developers who do not care to which community they belong provide the sixth type (20 per cent).
>
> (Infonomics 2002)

These types seem to indicate that the polarisation between the free software and open source movements is much more complicated than a simple binary dichotomy; developers take a number of contradictory positions in relation to the 'official' discourses presented by the free software and open source movements. It could be argued that the movement's organisational representatives and publications offer poles of attraction for the developers. These 'strong attractors' can be thought to lie towards the extremes of a spectrum of positions taken with regard to the ethics of software production and sharing.

When cross-referenced against age the statistics from the research showed that there was a slight tendency for those in types 1 and 2 to be younger but in the older developers the distinction becomes less important. However, the Infonomics report declares that neither age nor length of time involved in the FLOSS projects has a strong impact on the ideological orientation of the developers. An interesting finding was that 92.3 per cent of developers thought that money was less a concern in FLOSS development than in proprietary development, and 44.7 per cent expressed the concern that the drive for money in proprietary software development was a 'bad' thing (Infonomics 2002: 15). Nonetheless, 58 per cent of developers considered that their contribution to FLOSS projects was their property and that this was important, whereas 35.6 per cent also considered their contributions as property but unimportant (so 94 per cent thought that their source code contributions were their private property).

Only 6.4 per cent of respondents stated that their contributions were not private property.

Lakhani and Wolf (2005) undertook an empirical web-survey of 684 developers to understand motivation and effort in FLOSS projects.[71] The developers were primarily male (97.5 per cent) with an average age of 30 years and lived primarily in the West – 45 per cent from the US and 38 per cent from western Europe. They reported that the main reason for contributing to software projects was that the project was intellectually stimulating (44.9 per cent) and that improving programming skills came second (41.8 per cent). Approximately 30 per cent believed that source code should be 'open' and 28.6 per cent felt an obligation to contribute back to the FLOSS community. Extrinsic factors included access to the code for work-related reasons (86 per cent) and for reasons of obligation to a community (19 per cent). One of the most interesting findings in this work was that participants believed that personal creativity was the biggest determinant of their effort in FLOSS projects, a finding which tallied with previously cited research (Lakhani and Wolf 2005: 10–15).

Recently, researchers have undertaken case studies to contribute to a better understanding of FLOSS development. Hertel, Niedner and Herrmann (2003) tried to identify what determined the motivations of 141 developers in the GNU/Linux project by analysing FLOSS documentation. By examining documentation and mailing lists they argued that there are two main motives:

> (a) intrinsic motivation ('fun to program') and personal challenges to improve existing software for own needs, and (b) social comparison motives such as competition with other developers (either within OSS projects or between OSS projects and commercial software projects) and/or the interest to build a reputation that might be helpful for their occupational career.
>
> (Hertel, Niedner and Herrmann 2003: 1162)

As outlined throughout this chapter it is clear that FLOSS draws on intellectual property rights for its legal guarantee for the functioning of the kind of commons that it creates through copyleft. FLOSS is not an anti-copyright movement, although it

is often cast in that light and its advocates are often critical of copyright itself. In other words FLOSS uses *res privatae* (private property rights) to create a *res communes* and then uses the copyleft clause to prevent the system becoming unstable through a free-rider effect. There are, of course, particular instantiations of this whereby free software operates like a *res universitatis* (for example, the Free Software Foundation) and open source operates like a *res privatae* that allows sharing (for example, Apache, the webserver software).

At a more practical level, FLOSS is currently precariously balanced between the need for a common public form in which innovation and creativity can blossom and the reliance, to a large extent, on private corporations and the skills and donated time of employees and individuals that operate within the market. Although, it seems clear that public intervention could help prevent the free software and open source movements drifting into crisis, I argue that to understand FLOSS we should also think outside of our existing binary categories of public and private and develop new notions of the commons to understand the way in which these groups operate. If these new commons become increasingly important economically and politically the question of ownership will become sharper and may require action by the state. This means relying on *res publicae* to act as some form of control or watchdog on these new forms of *res communes* may be a better safeguard of the production of either *res communes* or even a potential *res divini juris*.[72]

Today with knowledge becoming a key resource for capitalism, the question remains as to what lengths capital may go to control the raw materials of creative or informational production. This has been achieved in the past through what some theorists refer to as the 'enclosure of the common' or the privatisation of knowledge objects (see Drahos and Braithwaite 2002). As we have seen, knowledge and information function differently from physical property and consequently seriously problematise the ability to create a market and enforce payment for their use. However by the use of law, technology and education, the real-world constraints of physical property are mapped back onto the immaterial. This

creates three benefits for the owners of knowledge objects: (1) that an artificial scarcity can be created; (2) that the products derived from these knowledge objects can be segmented into different markets (streaming, download and physical commodities, for example); and (3) that the user or consumer will be unable to copy, reuse or transfer these knowledge objects. To date we have seen a two-fold movement through the criminalisation and extension of intellectual property regimes and the use of technological protection measures (TPMs) such as digital rights management to enforce these new property rights. Naturally, this means that TPMs require that the underlying code that is used to apply these restrictions back on the user remain both secret and proprietary, to prevent the immaterial leaking out. Thus, public access to the free flow of information and knowledge are threatened and the freedom to use ideas is restricted through legally based but technologically constructed enclosures.

FLOSS, with its emphasis on the free availability and openness and transparency of source code, completely undermines, often unconsciously, the structural and institutional constraints that are being assembled by capital to construct an information society. As capital expands it seeks ownership of more of the common intellectual space in order to control the production of immaterial goods and services. This tendency can contradict some of the requirements for creative production (such as access to knowledge), but more particularly prevents technologists involved in open source and free software products from writing software due to a variety of constraints, including patents, criminal use of encryption technologies or even disassembly required to make open source and free software compatible with certain media (e.g. DeCSS for the CSS DVD encryption system to work on Linux). This means that open source and free software developers have suddenly found themselves at the centre of a brewing political storm, despite their varied attempts to keep the technical and political spheres firmly separate (see the next chapter). This much-derided 'politicisation of open source' (O'Reilly 2002) has contributed to a rising awareness both inside and outside of the technical world of the importance of a commons to creative and

innovative work, but the technologists remain wedded to the idea that the technical can be kept separate from the political. However, despite their efforts this FLOSS political consciousness has gradually begun to pass over into the cultural sphere with the emergence of 'open source' politics, music, design, art and culture flowering as an interesting new counter-cultural moment.[73]

In this chapter I presented a history of FLOSS that attempted to contextualise the debates and practices that underpin FLOSS and show how capitalism is not external to FLOSS but a critical source of funding and support. I then outlined how a 'politics of code' has developed which has led to copyright being used in unexpected ways to secure the sharing of freedoms through the principle of 'copyleft' for the writing of computer code. Finally, I turned to look briefly at the institutional support and networks of co-operation that serve to ensure the economic base of FLOSS and argued that this economic base is an important site for research into theories of the information society or technology in general. In the next chapter I move from the structural approach taken in this chapter and consider the qualitative dimension of particular debates between FLOSS practitioners. This will present the positions of the two main camps represented in the FLOSS movement, the ways in which wider societal discourses are located within their texts, and how they are utilised to expound a particular technological politics.

5

THE CONTESTATION OF CODE

While free software by any other name would give you the same freedom, it makes a big difference which name we use: different words convey different ideas.

(Stallman 2003b)

The term 'free' software is very ambiguous (something the Free Software Foundation's propaganda had to wrestle with constantly). Does 'free' mean 'no money charged?' or does it mean 'free to be modified by anyone', or something else?

(Raymond 2001c)

In this chapter, I look at the way in which FLOSS actors articulate the discourses that represent computer-programming code and their social practices and norms. Throughout this chapter, the focus will be on how the underlying philosophical and technical questions of the Free Software Foundation and the open source movement are legitimated through discourses that surround the code. For this I have used the cases of two particular exemplars, Richard M. Stallman and Eric S. Raymond, partly because they are self-appointed leaders of their respective movements but also because they are theorists, practitioners and advocates of FLOSS. As such, they also embody and articulate certain conceptual arguments that have become deeply emblematic of their respective positions. In addition, I look to other actors within these camps to flesh out and support the arguments I make in this chapter, particularly in relation to the way in which their contextualised arguments about code *qua* code have been taken up in the broader discussions about culture in debates between the Free Culture and Creative Commons movements.

Although the subject of discourse for computer code may seem somewhat esoteric, a broader political project appears to be manifested within the debates between the two movements (Feenberg 1995; Lessig 1999, 2002a; Moglen 1999; Raymond 2003; Stallman 2002a). In short, these movements crystallise discursively a more substantive challenge for wider society, namely issues surrounding the legitimacy of technocratic society, reflexive modernisation, the democratisation of technology and the public deliberation of technology policy. The respective positions of the two groups provide a unique case study for theories about modernity and technology. Here, though, I concentrate particularly on the extent to which they attempt to universalise particular positions and how they draw from wider debates and discussions.

I argue that the Free Software Foundation (FSF) and the open source movement (OSM) are therefore engaged in a struggle at the level of discourse, that is, that the code itself cannot and does not 'speak for itself', and instead is represented by actors who draw on other discourses or explanations to legitimate their position. In order to establish closure of meaning, these two FLOSS movements attempt to fix the polysemic elements within an order of discourse[1] surrounding the production and interpretation of computer-based programming code. This is done through a number of techniques that I analyse in the following chapter. For example, I look at the ways in which the ontology of code is manifested within their respective discourses, which of the two movements is more likely to achieve discursive closure, why it might be successful, and the larger social and political implications of this struggle. Generally speaking this chapter is concerned with the textual artefacts of programming, including source code, documentation, programming models, formal languages and related discursive artefacts.

The discourses discussed in this chapter concentrate on the texts of two key individuals, Richard M. Stallman, the founder of the Free Software Foundation, and Eric S. Raymond, the founder of the Open Source Initiative (OSI), an organisation for controlling the 'open source' mark. These texts were chosen

for their foundational value to the two movements' respective constituents and their importance as perceived founding fathers of the free software and open source movements (see for instance Bonaccorsi and Rossi 2003: 9; DiBona, Ockman, and Stone 1999; Moody 2002; Scoville 1999).

As previously discussed, Richard Stallman was the founder of the GNU project to create a completely free software operating system and the author of the GNU General Public License (although with important help from Eben Moglen, a professor of law at Columbia University). His life's work has been committed to asking questions about the nexus between intellectual property rights and software. He also is involved in practical action in the form of writing software and licences, and more general left-of-centre political activism. As a prolific writer and programmer he has been instrumental in making the technical imaginary of a free software operating system possible and his contributions to the debate have been important interventions (often setting the very terms of the debate).

In contrast, Eric Raymond is a hacker of a different calibre (he supported an application called *fetchmail*), who nonetheless had a gift for explaining the activities of FLOSS that was widely influential. He also joined in the activities of the free software movement, developing free software and contributing to debates. It was with the publication of his essay *The Cathedral and the Bazaar* (2001), originally given at the Linux Kongress on 27 May 1997, that brought him public attention and gave voice to an existing if inchoate argument about the problems with the utopian aspects of free software. This eventually led to a split into open source and free software camps that divides free software advocates to the present day. As a self-described anarchist and capitalist Raymond's politics are libertarian and sometimes quite extreme, ranging from advocating gun ownership for all (including children) to an assertion of American military force for an 'imperialist' project to subdue and 'civilise' the 'Arab/Muslim world'.[2] He has nonetheless contributed a number of interesting articles to the debate about the ends of free software and FLOSS and has been very active in convincing the corporate world to

move towards open source licences (Netscape being a notable example discussed in the last chapter).

In seeking to understand the contrasting positions of free software and open source, a preliminary coding was made of a number of OSM and FSF documents and interviews (Stallman 1992, 1994, 1999b, 1999c, 2001, 2002b, 2003b, 2004; Kuhn and Stallman 2001; Raymond 1998, 1999b, 1999c, 2001a, 2001b, 2003, 2007).[3] By analysing text and interviews a coding was outlined for 19 discourses from the two respective movements, a total of 52,088 words.[4] Key signifiers were then identified in the texts and examined and compared together to show how concepts concerned with identity, such as representation and group identity, and concepts concerned with conflict, such as antagonism and hegemony, were ordered discursively. These were then examined using a comparative approach to uncover meaning and contested concepts by contrasting the way the two movements use key terms. Both the OSM and the FSF use textual elements to articulate various aspects of discourse. They attempt to fix the polysemic elements that are shared between their discourses and create 'chains of equivalence' (Phillips and Jørgensen 2002: 43). Additionally, they seek to place alternative articulations from each other within the field of discursivity and exclude them from the order of discourse (Fairclough 1992: 98; Phillips and Jørgensen 2002: 27). Although this temporary closure can never be definitive and fixed, nevertheless the antagonistic discursive struggle can be dissolved through a hegemonic intervention (Phillips and Jørgensen 2002: 48). This will be explored further in the chapter.

First, I undertake an examination of elements and key signifiers in discourses from the FSF and OSM respectively. Subject positions will also be presented from each discourse. I then offer a comparative discursive analysis, which is applied to the results of the discourses' key signifiers. Finally, I draw these streams together and outline possible issues for the FSF and the reasons for a likely naturalisation of the OSM's discourse (Fairclough 1992: 94).

Codings

Below I examine the implications of the positions of the two movements and attempt to point to their theoretical and philosophical origin. Both utilise a model for the production of knowledge, both in terms of a support for the claims to 'true' knowledge and in terms of their understanding of the relationship to the external world. These philosophical positions imply an underlying conception of agency and epistemology (Linstone and Murray 2002: 17–19) and will be examined in turn.

Two major strands in the discourse of the OSM and FSF converge. Firstly, they are based firmly within the community of technologists and are committed to the social good that open or free software can provide, but differ radically in their respective assumptions about how this good is to be achieved (Scoville 1999).[5] Secondly, they reflect wider societal questions about technological determinism, efficiency and the democratisation of technology. Each movement condenses these debates into strongly differing approaches to technological progress and the legitimacy of technocracy. As these issues are important contemporary questions, this chapter seeks to place these arguments within a broader framework and offer some conclusions and recommendations as to their wider application.

Both of the movements are uncomfortable with a general societal development towards technology that is 'closed', that is, technology that appears 'magical' or autonomous because the underlying complexity of the technology has become overwhelming or hidden from the user. In this situation human users of a technology become increasingly ignorant of the underlying processes and functions within technology and they worry that consequently the possibilities for human freedom are diminished. Winner describes this as 'the gap between complex phenomena that are part of our everyday experience and the ability to make such phenomena intelligible and coherent' (Winner 2001: 282, emphasis removed). He argues that as the sum total of human knowledge, particularly scientific and technical, increases, mastery of this knowledge becomes increasingly difficult so that

any particular individual, group or person cannot comprehend the whole. Such that:

> Society is composed of persons who cannot design, build, repair, or even operate most of the devices upon which their lives depend … The technological society contains many parts and specialised activities with a myriad of interconnections. The totality of such interconnections – the relationships of the parts to each other and the parts to the whole – is something which is no longer comprehensible to anyone.
>
> (Winner 2001: 284)

This issue of human freedom runs deeply through the discourses of both the free software and open source movements, and reflects the resistance amongst members of a technological engineering elite to halting a process that has already affected the lives of non-technical people. Here the argument is brought to the fore in the contrasting positions of those who believe that to lose touch with the 'reality' of being able to change technology, and hence control it, is wrong; and those who argue that there is no necessity for members of technological society to understand *all* aspects of social and technical life (this, I would argue, is a 'consumerist' position). The consumerist position betrays a particular passive attitude to the growth in technology and its penetration of many aspects of our social lives and activities. For the FLOSS groups this docility is anathema; their concern is that they must continue to have a determining influence on the design, implementation and operation of complex technology. This worry is not new, as H. G. Wells wrote in 1945:

> Spread out and examine the pattern of events, and you will find yourself face-to-face with a new scheme of being, hitherto unimaginable by the human mind. This new cold glare mocks and dazzles the human intelligence, and yet, such is the obstinate vitality of the philosophic urge in the minds of that insatiable quality, that they can still under its cold urgency seek some way out or round or through the impasse … The writer is convinced that there is no way out or round or through the impasse. It is the end.
>
> (Wells 1945, quoted in Winner 2001: 290)

Here, FLOSS becomes an extremely interesting set of technical practices that allows people to understand and direct complex technical phenomena in a decentred way. It is remarkable that in 1978 Winner was unable to theorise a method for dealing with this extreme technical complexity that societies were facing, stating 'I must report that I have found no such tools in practice [to deal with extreme complexity] … [as for the] relief for the sociotechnical complexity of the modern age – none of these [tools] offers much help' (Winner 2001: 288). With this in mind it is no surprise that theorists and political policy experts, who in many cases have begun to theorise the political and social milieu as a machine-like structure (reminding one of Hobbes), see the possibility of a steering mechanism for our complex polity through open source methods (Osborne 2007; Sachs 2007).

Whether open-source-type methods can be used in such a way remains to be seen; however, what does become interesting, and I develop this through the course of this chapter, is the notion that there is a substantive political philosophy underlying the FLOSS groups' positions on the subject of computer code. There is certainly a taken-for-granted notion of technological progress and a strict linear development of technology as a social good that runs through both movements. Additionally, there is a commitment to a social concept of the 'network' as an organising principle of socio-technical ensembles. Perhaps here the differences between the movements begin to become clearer; free software treating the community practices of coding as a humanistic interpretative activity; the open source movement concentrating instead on notions of individuality, efficiency and utility maximisation by economic actors (profit, management of code, and the like). However, both share the concept that the members are interconnected and have a relationship of interdependency that is geared towards a wider social project. Of course, not all parts in a network are equal and certainly not all members of a network have the same status (although this is seldom articulated within the movements, which tend to universalise the subject position of the technical engineer). Equally, there is a strange blindness when discussing the steering and control mechanisms of these

projects; too often the claim is that there *is no leadership* even when it is clear that this is not correct. In the case of the Free Software Foundation there are centripetal forces through its central position as both code repository and moral leader on the issues of free software and a separation into a core/periphery and levels of status (this was part of Raymond's critique of free software as a 'cathedral building'). For the open source movement there is a centrifugal force pushing the members away into their own respective 'homesteads' which they develop using their own labour together with a moral 'anarchy', that is, that privately the methods, licences and organisation are not centrally mandated.

The Free Software Foundation (FSF)

I now want to focus on the key discourses of the Free Software Foundation and in particular the way in which the main position of the free software community is constructed and defended. Within this movement, there is a strong moral dimension to the practices of coding. I would argue that Kant's notion of a categorical imperative seems to underlie the philosophical foundations of the FSF – at least in the sense that what is ethical for the individual must be generalisable. The FSF uses discourses that draw from Enlightenment philosophy, communitarianism and the collegiate ideals of academic and scientific communities (Bezroukov 2003; Kelty 2001) both intertextually and interdiscursively to present a strong moral position (Stallman 1993, 2003c, 2003d). The FSF appears to take a deontological position in regard to personal ethics and a Kantian flavour is readily seen in the calls to abide by the general moral laws of the FSF. These 'laws' can serve both as a guide to individual action (Stallman 1993) and a means of conflict resolution within the movement by making manifest a shared ethical outlook (Elliott and Scacchi 2002: 2).

In terms of writing computer code, the ethic of sharing all code with others within the project is unambiguously Kantian in principle because the moral laws developed are universalised to all computer programmers. Some writers have characterised Stallman as being driven by aesthetic rather than ethical reasoning (Harvey

2003). Here the idea is that programmers strive to produce beautiful computer code and that this beauty will aid human development (with free software's transparency and openness contributing to that). However, this seems to be based on a misunderstanding of Stallman's position regarding efficiency and utility: he says 'we in the Free Software movement recognise these practical benefits, and they are nice, but they are not the most important issue. More important are the ethical and political aspects' (Stallman 2003c). That is, that while 'good code' is important (and usually by definition 'good code' is clearly organised and structured – beautiful – and provides the best efficiency), it is the freedom of the programmer to use code (whether 'beautiful' or not) that is at the centre of the debate.

The FSF articulates what I will call a discourse of ethics and a discourse of freedom (see below for the coding of the texts). The discourse of ethics outlines a basic philosophical position whereby access to the underlying source code of a software object is understood as a human right. As Stallman outlines, 'I consider [Free Software] a human right, and thus a moral norm' (personal correspondence 2002). Additionally, Stallman believes that freedom is intrinsically linked to the FSF's aims and that freedom of the individual is the freedom from the tyranny of technology (or those that control it) (Stallman 2003b). Much of the language that Stallman and fellow advocates use in their discussions of the rights and wrongs of practices within their communities is constructed around moral imperatives. This often draws particularly on notions of good and evil. For example, when talking about digital rights management software Stallman states that 'DRM is evil and shouldn't be allowed … DRM imposes unacceptable restrictions on users, it is simply evil' (quoted in Poynder 2006: 22). This moral system is also constructed around notions of right and wrong that draw on a civil rights tradition connected closely with the Free Speech movement that arose at Berkeley in the 1960s (see Markoff 2005). Within this tradition, free speech is an unconditional good that maintains freedom against tyranny and this can be seen in the way in which a complex spectrum of positions on the question of the sharing of code is represented as a binary opposition by

Stallman:[6] 'It is a struggle between systems that are essentially fascist – in that they give businesses power over people and systems that respect freedom' (quoted in Poynder 2006).

Stallman's highlighting of the importance of freedom to code is reminiscent of Marcuse's critique of 'one-dimensional man' in modern society (Marcuse 1991) pointing to a collapse in critical approaches to the world and a flattening conformism caused by the rise of the consumer society and the growth in social administration. For Stallman, the ability to use code in an informational society is connected to a particular critical faculty of human beings – within hacker communities sometimes referred to as the 'freedom to tinker'. That is, the freedom to take things apart, see how they work, and discover whether their actions perform processing functions of which you do not approve or wish to change. With the decline in the ability to undertake these critical investigations into code, through proprietary software, digital locks (through DRM) or merely through ignorance caused by a disinclination to bother with the underlying complexities of modern technology, there is a concomitant reduction in human freedom. Again, the importance of a critical approach to the world connects Stallman's approach with Kant, and particularly with humanistic conceptions of the importance of the ability to undertake autonomous reasoning (in this case with computer code). In each case the autonomy of the critical coder is highlighted against a form of heteronomy (following the dictates of another). Interestingly, in a Kantian vein, Stallman also argues that 'the law should conform to ethics, not the other way around' (Stallman 1992).

Throughout Stallman's discussions of the threats to human freedom that are distributed across the FSF discourses examined in this chapter, one can identify two mechanisms that he argues present particular dangers: (1) an internalisation of the norms of proprietary approaches; and (2) that proprietary solutions might be embedded within social structures (such as through law, digital rights management, and the decline of a particular hacker community of peers – this is explored further below). Both of these threats are found in his texts, although in many instances a focus upon structural factors seems to be growing in importance

in his discourses (perhaps reflecting a new generation of computer programmers that have grown up within the 'walled gardens' of contemporary software development).

One of the idiosyncrasies of the FSF is that rather than reproduce information, it presents its general philosophy by hypertext referencing to other texts of which it approves.[7] This seems to represent not only a legitimation of knowledge by reference to sources, but also an application of a form of a centre-given blessing (much as the Pope would legitimate certain ideas in the world). This may also appeal to Stallman's ideals of personal efficiency (computer programmers are notorious for not remaking the wheel by copying and pasting code) (Williams 2002) and the importance of crediting and acknowledging others (Kelty 2001). This manifest interdiscursivity is demonstrated on the FSF webpages in the following examples: (1) the belief that motivation is not vulgar behaviourism (Kohn 1987), (2) that technical efficiency is not the only driver of technological progress (Stallman 2003c) and (3) that self-interest is not the only motivator for hackers (Fueston 1998).

I would therefore argue that the FSF takes a Kantian approach to the production of truth. This is a contributory theory of knowledge that encourages many different and competing judgments and solutions to be applied to the problem area (Linstone and Murray 2002: 25). This approach allows many 'informed' individuals from different disciplines and specialities to contribute information to the project and consequently allows a broader definition of the problem area and encourages a goal-oriented methodology. This is demonstrated both in the general nature of the FSF itself, which seeks to maximise freedom, and in its specific aim to produce a non-proprietary version of Unix that is completely unrelated to proprietary versions (see Stallman 1993).

The FSF actively calls for contributions and participation from interested parties without necessarily specifying a physical platform for implementation (see Stallman 1993). Indeed, the radically open modular nature of the design of FLOSS computer software has the potential to encourage a democratisation of technology due to the creation of different and competing imple-

mentations. This is one of the hallmarks of the free software and open source movements and many websites serve both to fork projects and discuss different projects' relative pros and cons (see examples at Slashdot 2003).

The preliminary key signifiers identified in this analysis within the FSF discourse are code, rights, community, freedom, power and progress.

Code. For free software advocates and for FLOSS developers more generally, code is a key object of attention, being at the centre of their social practices and their economic ability to earn an income (most FLOSS developers usually work full-time as coders in corporations or other organisations). Hence, 'code' is a nodal point, in other words it is a privileged sign around which the other signs are organised. Other signs acquire their meaning from their position in relation to this nodal point.

> Programmers normally work with the 'source code' for a program, which is written in a programming language such as Fortran or C … It is designed to help programmers read and change programs … Source code is useful (at least potentially) to every user of a program. But most users are not allowed to have copies of the source code … It leads to resignation and discouragement, which can spread to affect other aspects of one's life.
>
> (Stallman 1992)

> [W]ho should control the code you use – you, or an elite few? We believe you are entitled to control the software you use, and giving you that control is the goal of Free Software.
>
> (Kuhn and Stallman 2001b)

For the FSF, code is a public good that is a socially constructed phenomenon and should be freely shared. Stallman refers to the dangers to the self, manifested when a programmer does not have copies of the source code. The dangers of infection and resignation are linked directly to code and the principles of the health and vitality of the programmer. This psychologism is radically different to the selfish individual outlined in the OSM discourses; rather this is a self that must share in order to live a good life (see the

previous section about the mechanisms of harm passed through proprietary approaches):

> Suppose that both you and your neighbor would find it useful to run a certain program. In ethical concern for your neighbor, you should feel that proper handling of the situation will enable both of you to use it. A proposal to permit only one of you to use the program, while restraining the other, is divisive; neither you nor your neighbor should find it acceptable … This is psychosocial harm associated with the material harm of discouraging use of the program.
>
> (Stallman 1992)

Code is more strongly associated with the social practice of 'coding', in other words the production of code in a social network. Stallman draws conclusions about the ethics of coding, sharing, contributing, and the importance of the publicness of ideas. As Stallman is concentrating on social practice, it seems logical that this cannot be protected or withheld from the group that was instrumental in forming the ideas in the first place. Here again Stallman refers to the specifically individual harm to the self brought about by the failure to take part in social activities that he sees as necessary to promote a healthy self.

> In any intellectual field, one can reach greater heights by standing on the shoulders of others. But that is no longer generally allowed in the software field – you can only stand on the shoulders of the other people in your own company.
>
> (Stallman 1992)

Copyright is therefore problematic as it restricts others' ability to use information and further their own and society's progress.

> [T]he power to restrict changing or copying it – is obstructive. Its negative effects are widespread and important. It follows that society shouldn't have owners for programs.
>
> (Stallman 1992)

There are also notions of distinction (Bourdieu 1986) and status that permeate the discussion of who can and who cannot share code. The world becomes divided into 'hackers' (who produce

the code) and 'users' (who passively utilise it). In fact, this is a universalisation of a particular interest (of coders to be able to use, change and distribute the code) to that of the general population, many of whom would have little or no interest in the code itself. Code also becomes a crucial issue due to the fact that it is at the centre of the bundle of intellectual property rights that are being asserted or whose ownership lies under contestation – an important aspect of changes being made in relation to IPR in legislatures and government policies. Stallman also explicitly attacks the notion of patents due to the monopoly they create in the realm of ideas (DiBona et al. 1999; Garfinkel, Kapor, and Stallman 1991; Stallman 1991a).

Rights. These are at the heart of the Free Software Foundation's articulation of access to and use of computer source code. The concept of rights and particularly a liberal notion of human rights underlies the constructions of access to source code as a basic human right (Stallman 2002b). Stallman utilises a strong concept of rights drawn from the American constitution to justify this position:

> [T]he idea of inalienable rights embodied in the GNU GPL comes from the founders of the United States.
>
> (Stallman 2002b)

> The ethical response to this situation is to proclaim freedom for each user, just as the Bill of Rights was supposed to exercise government power by guaranteeing each citizen's freedoms.
>
> (Kuhn and Stallman 2001b)

However, Stallman is careful to delimit the potential for a conception of Natural Rights for property ownership from a Lockean tradition, not wanting to allow the conception of labour producing 'just desert' which would allow actors to claim rights to creative work, such as code. That is, he continues to highlight the rights of access over the rights of ownership. Instead he wishes to emphasise the rights within the realm of freedom, rather than in the realm of property-owning rights.

> The idea of natural rights of authors was proposed and decisively rejected when the US Constitution was drawn up. That's why the Constitution only permits a system of copyright and does not require one; that's why it says that copyright must be temporary.
>
> (Stallman 1994)

For instance, he states:

> The real established tradition of our society is that copyright cuts into the natural rights of the public – and that this can only be justified for the public's sake.
>
> (Stallman 1994)

This is an important distinction for Stallman who is keen to emphasise the corrosive effect of copyright and patents on freedom of *expression*. By this means, the argument for a stronger moral claim for freedom of access to the source code is constructed. This forms part of the justification for free software by linking to the wider discourse of freedom that Stallman uses.

Community. Within the discourse of ethics, Stallman identifies being an active member of a civic community and the act of sharing with a neighbour as highly important. The conception of the good life is linked to that active contribution to a shared community (in this it is similar to notions of academic community).

> We look at what permits a good way of life, and at how useful programs can foster a community of goodwill, cooperation, and collaboration. Our criteria for free software specify the freedoms that a program must offer its users so that they can cooperate in a community.
>
> (Kuhn and Stallman 2001b)

> Well they are not all hackers, some of them just use the software, but they are part of a community where people often help each other. Proprietary software forbids community.
>
> (Stallman, quoted in Poynder 2006)

For the FSF 'non-free software is a social problem and free software is the solution' (Stallman 2003b). The 'non-free' software would create barriers between the members of that sharing

community; *res privatae* fragments a community structure. Within the discourses Stallman tends to value a 'good life' that promotes positive values, for example:

> Our criteria for Free Software specify the freedoms that a program's users need so that they can cooperate in a community.
>
> (Kuhn and Stallman 2001b)

The conception of 'social good' for free software includes the importance of the social and the communicative experience of coding. This social sharing manifested within the free software movement is built on trust and a reliance on others to provide improvements and ideas freely for the project. No members of the community can assert *res privatae* rights, which would give them power of access over others. Rather, the ideal typical form of code ownership is *res communes*, that is, community ownership (or at least that all members of a community have guaranteed access to the code).

> 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.
>
> (Kuhn and Stallman 2001b)

This conception of the social good is strongly communitarian and privileges both a vision of a social order that assigns social rights and responsibilities, and one that is fair and equitable. Each contributes code to the project according to their ability and takes code according to their need: '… above all society needs to encourage the spirit of voluntary cooperation in its citizens' (Stallman 1994).

The spirit of voluntary co-operation which Stallman believes is part of the desire to 'help your neighbour' and engage in 'civic spirit', should be promoted and encouraged.

> Programmers also suffer psychosocial harm knowing that many users will not be allowed to use their work. This leads to an attitude of cynicism or denial.
>
> (Stallman 1992)

Again, Stallman refers to the 'psychosocial harm' that the self experiences if not allowed to act in accordance with the principles of software sharing and community. This appears to be linked to a form of anomie[8], whereby the members of a community become fragmented from each other and unable to communicate effectively with each other. Sharing in this instance is related to the sharing of language and the ability to reflect and draw upon the contributions of others.[9]

Stallman claims to have a concept of community that is able to facilitate conflict and disagreement and which involves political debate and disagreement: 'Communities often have political disagreements. That's normal with communities, and it's normal for human beings' (quoted in Poynder 2006). This is in contrast to the strong moral position that is taken against those who have sought to question or challenge the moral authority or absoluteness of the four freedoms (outlined below).

There is also an undercurrent of a friend/enemy distinction that is reminiscent of the Old Testament brother/stranger and Luther's conception of church/state that identifies those who should be subject to the moral law (brothers) of the community (i.e. the General Public License) and those that are not (strangers)[10] (see Hyde 2006: 132–8). Within the boundary of the free software community the principles of sharing and mutual support are emphasised, but when dealing with outside agencies and actors, who are presumed to be less likely to share in the morality implicit in these practices, then a proselytising approach is taken, to convince them to join the community. If they do not choose to abide by the guidelines of the free software community, they are considered to be outside the circle of friends, and therefore outside of the community.

Freedom. Within the discourse of freedom, Stallman outlines a number of reasons why software should be free, drawing on principles of natural rights and social rights. This is one of the principal differences between the open source and free software movements.

> 'Free software' is a matter of liberty, not price. To understand the concept, you should think of 'free' as in 'free speech', not as in 'free beer'.
>
> (Stallman 2003b)

This construction of freedom has been most clearly outlined in Stallman's 'four kinds of freedom' – which idiosyncratically start with freedom 0.[11] He explains that 'Free software is a matter of the users' freedom to run, copy, distribute, study, change and improve the software' (Stallman 2004). These freedoms are defined as an axiom of the practices of software programming and is cast as an iron law, as a moral imperative that all should abide by:

> Freedom 0: The freedom to run the program, for any purpose.
> Freedom 1: The freedom to study how the program works, and adapt it to your needs.
> Freedom 2: The freedom to redistribute copies so you can help your neighbor.
> Freedom 3: The freedom to improve the program, and release your improvements to the public, so that the whole community benefits.
>
> (Stallman 2004)

The presentation of the ideals of freedom and sharing imply an ethical choice for the software developer to take to his fellow developers (Fueston 1998). Note also the use of psychosocial categories such as 'discomfort' that are associated with individual alienation from a community-based project.

> That's true: talking about freedom, about ethical issues, about responsibilities as well as convenience is asking people to think about things they might rather ignore. This can trigger discomfort, and some people may reject the idea for that. It does not follow that society would be better off if we stop talking about these things.
>
> (Stallman 2003b)

> If you feel that freedom and community are important for their own sake – not just for the convenience they bring – please join us in using the term 'free software'.
>
> (Stallman 2003b)

The freedom that Stallman envisages is formed around the ideas of being able to shape and change both one's own destiny and also the tools that are used along the way. This strong conception of 'free' includes not just ideals of freedom of speech and freedom of assembly but also the dangers to freedom of thought if ideas themselves have restrictions on their usage. 'To stop using the word "free" now would be a mistake; we need more, not less, talk about freedom' (Stallman 2003b).

Stallman identifies a common engineering problem, namely the danger of being unable to question 'black boxes' that cannot be opened to check their contents. This is very similar to the concept of a 'technical code' (Feenberg 2002), a moral obligation imposed on humans by the delegation to machines.

> What does society need? It needs information that is truly available to its citizens – for example, programs that people can read, fix, adapt, and improve, not just operate. But what software owners typically deliver is a black box that we can't study or change.
>
> (Stallman 1994)

Quite simply, the black boxes not only present the possibility of surreptitious spying or monitoring, but they cannot be repaired or changed and, most importantly for Stallman, improved, so that progress is hindered.

> I am working to build a system where people are free to decide their own actions; in particular, free to help their neighbors, and free to alter and improve the tools which they use in their daily lives. A system based on voluntary cooperation and on decentralization.
>
> (Stallman 1992)

Power. The concept of 'power' is associated with a Weberian concept of power over the user. Where one can use a proprietary software licence to restrict the activities of the user or programmer, there is an act of power.

> Proprietary software is an exercise of power. Copyright law today grants software developers that power, so they and only they choose the rules to

> impose on everyone else – a relatively few people make the basic software decisions for everyone, typically by denying their freedom.
>
> (Kuhn and Stallman 2001)

Power to control draws particularly on the notion of *prescription*, where technology is enforcing a control over humans that has been *delegated* to technological artefacts. Ensuring the inability to check the existence of these prescriptive functions within the source code: 'Current copyright law places us in the position of power over users of our code, whether we like it or not' (Kuhn and Stallman 2001b).

> E.g. The [World Summit on the Information Society]'s declaration includes little that is bold or new. When it comes to the question of what people will be free to do with the internet, it responds to demands made by various governments to impose restrictions on citizens of cyberspace.
>
> (Stallman 2003d)

With 'black boxes', the user is forced to act under the control of the proprietary software manufacturers with no recourse to appeal. This is a fundamentally undemocratic moment and reflects a concern with technological power over humans (over and above human power over other humans), and forms part of the argument for the transparency of code as a desirable and democratic approach to social life: 'I shouldn't have the power to tell you not to do these things. No one should' (Stallman 1994).

In this conception of power, arguments are usually focused on ways to reduce the monopoly exercised by the mechanisms supplying this power, for example, by preventing copyright or patents, or making power relations transparent (see Kuhn and Stallman 2001b). The idea of the sovereignty of the individual naturally calls into question the implementation of prescriptive architectural properties within computer code because individuals should be able to see how code controls their actions. Restrictions on the natural freedoms of the user are understood as a restriction on freedom of choice. The FSF is therefore strongly opposed to the use of prescriptive functions in software (see Stallman 2003c). The ability to get inside the technical device means that the code can

be changed and controlled and that the freedom of the individual to choose is paramount (see Stallman 1992).

Progress. The FSF uses the signifier 'progress' to indicate that without the collective provision of free software an Enlightenment ideal of progress would be lost. Indeed, Stallman approvingly quotes the US Constitution, stating that copyright is designed to '... promote the progress of science and the useful arts ...' (Stallman 1992). Congress agreed that the rights of the public were temporarily suspended, but as Stallman explains 'It also states that the purpose of copyright is to promote progress – not to reward authors' (Stallman 1992). There are no authorial rights in the constitution, merely temporary copyrights. For the FSF these rights were predicated on a system of property that by its very nature was limited, material and not easily copied. These are contrasted with the virtual goods of the Internet that are claimed to be unlimited, non-material and easily and freely copied.

> Our ideas and intuitions about property for material objects are about whether it is right to take an object away from someone else. They don't directly apply to making a copy of something. But the owners ask us to apply them anyway.
>
> (Stallman 1994)

Subject positions. Readily using 'we' and 'they', the FSF utilises the concepts of in-group and out-group to identify friends and enemies. These subject positions are treated as a dichotomy and the reader is assumed to be supportive of the FSF objectives, a friend and colleague – or if not, an enemy. Most prominent is the attempt to set up a distinction between the free software and open source movements, when in fact they share many of the same authors and coders:

> We are not against the Open Source Movement, but we don't want to be lumped in with them. We acknowledge that they have contributed to our community, but we created this community, and we want people to know this. We want people to associate our achievements with our values and our philosophy, not with theirs. We want to be heard, not obscured behind

> a group with different views. To prevent people from thinking we are part of them, we take pains to avoid using the word 'open' to describe free software, or its contrary, 'closed', in talking about non-free software.
>
> (Stallman 2003b)

> Open source advocates do contribute to our community – not all of them, but many of them. There are people who develop free software that were motivated by the Open Source Movement rhetoric, for instance. These programs are good, so that is a good thing. The bad aspect is that it is weak: it doesn't teach people to see a freedom to defend, so they don't defend their freedom, and they won't defend our freedom
>
> (Stallman, quoted in Poynder 2006: 18)

Unsurprisingly, because open source has been enormously successful in capturing a concept of free software in a way that is business-friendly and open to interpretation in other contexts such as music, art and film, it has been widely used. This has led to a purist counter-movement by Stallman to clearly differentiate between their different positions and to explain the damage he believes has been done to the entire movement:

> [The agenda mapped out by the Free Software Movement has] been very badly subverted. In fact, it has nearly been lost. The distribution I told you about in 1995 [unnamed] that had 28% of programs released by the GNU Project was an entirely free distribution, and in those days it was not hard to find entirely free distributions. Today it is hard to find distributions that are entirely free.
>
> (Stallman, quoted in Poynder 2006: 18)

> Some people do believe in freedom as a goal but they emphasise practical things because they think that is the way that people will listen to them. But they don't realise that by giving a message that people listen to more, they are actually teaching a different thing. Open Source supporters didn't realise that, although immediately successful, this message is weakening in the long term.
>
> (Stallman, quoted in Poynder 2006: 18)

Additionally, a strong distinction is made between a concept of free culture and proprietary culture, often cast within a libratory

discourse. For instance, Stallman states '[m]y aim is to be the liberator of cyberspace. That is my public mission' (Poynder 2006: 30). The freedom he advocates is held up against the purported fascism represented by business and government working in tandem against the individuals involved in coding communities:

> Fascism, remember, is the convergence of government and business disrespecting people's freedom. So what they do is prohibit Free Software and they impose software patents.
>
> (Stallman, quoted in Poynder 2006: 31)

Within this scenario, it is the hackers who fought for freedom, and Stallman argues that '[i]t was they who opposed security [in the computer system] and prevented fascism' (Poynder 2006: 14).

The strong first-person modality of the text and the use of the collective 'we' implies an attempt to seek closure within the order of discourse and thereby excludes alternative or conflicting definitions or interpretations. The 'other' that one might have identified oneself with is therefore excluded and the potential for conflict and over-determination of the subject is avoided. Additionally, the text seeks to speak both for and to the group and in the process defines key signifiers such as 'code' and 'freedom'. Consequently, the FSF struggles to divide the social space of computer programmers, technologists and coders into groups along lines that would further the FSF's aims and objectives and fill the different master signifiers with their content. This results in a hegemonic struggle with the open source movement for the contestation of the key terms and signifiers that are shared between the two movements.

The open source movement (OSM)

The open source movement uses what I will identify as a discourse of technical efficiency and a discourse of neoliberalism. These are used intertextually and interdiscursively to legitimise and position their arguments as rational, natural and common-sensical. Eric S. Raymond is one the founders of the open source movement

and he clearly differentiates the OSM position from that of the FSF, stating that:

> Open Source is not particularly a moral or a legal issue. It's an engineering issue. I advocate Open Source, because very pragmatically, I think it leads to better engineering results and better economic results.
>
> (Raymond 2003b)

> Today, Open Source is a leader in sharing knowledge to everyone's benefit. We offer one of the most effective methods yet tried to achieve the goals of this [World Summit on the Information Society]. Please do not allow abuses of law to un-do the progress we have made. We ask the United Nations to take the lead in helping to unite the world in a productive Open Source partnership that helps liberate the poor and increases the freedom, knowledge, and well-being of every person.
>
> (Perens 2005)

The OSM differentiates sharply between the technical, rational and objective sphere of software development, and a political sphere (see O'Reilly 2002). The problem of freedom is seen as one of 'freedom to choose' within a system of market relations. In other words, the developer has the right to choose the licensing model and the user the right to choose to use the software. This highlights a consumerist notion of 'economic freedom within a marketplace' (O'Reilly 2001) rather than that of the more politicised notion of the essential human right to freedom of choice within all spheres.

Raymond follows a similar argument in the management of complexity to that advocated by Charles Lindblom, who argued that the attempt to comprehend the entire system is unnecessary and instead 'disjointed incrementalism' could be used. Here disjointed incrementalism refers to:

> societies and organisations [which] are too large and complex to allow decision makers a synoptic vision of all factors that are relevant to policy choices ... [however] following the incrementalist strategy, this makes no difference.
>
> (Winner 2001: 291)

Under this technique, organisations follow small steps which are radically decentralised and which are limited to the immediate vicinity and responsibilities (reminding one of the way in which technological approaches divide the world into distinct entities). Here, knowledge of the complete system is no longer relevant or necessary (in contrast to 'cathedral builders' in the free software movement) and any broader perspective on the situation is considered to be pointless. This also reflects Raymond's libertarian politics.

In order to claim an epistemological foundation (and by extension a scientific basis to his approach), Raymond outlines a crucial philosophical method to justify the OSM's methodology. This is called the 'Delphi Effect' (Raymond 2001).

> Delphi may be characterized as a method for structuring a group communication process so that the process is effective in allowing a group of individuals, as a whole, to deal with a complex problem.
>
> (Linstone and Murray 2002: 3)

First pioneered by Helmer and Rescher at the Rand Corporation, the Delphi technique is an example of Lockean inquiry (Linstone and Murray 2002: 15). The key aspect of this theory is that truth is experiential. In other words, the truth content of a statement is associated entirely with its empirical content. Every complex proposition is reduced to simple observations and the validity of these is ensured by the freely obtained agreement between different observers. This is an experimental, consensual system. This consensual system is best suited to an already informed and specialised knowledge community which shares a 'core' body of knowledge. This characterises the members of the OSM.

To build the system Raymond starts from a set of elementary empirical judgments, raw data, observations or sensations. From these is built a network of expanding, increasingly more general networks of factual propositions. The final system is subjected to agreement by a group of experts. The raw data is granted a prior existential status. This is demonstrated by the argument Raymond presents about a possible property-based system on the Internet in *Homesteading the Noosphere* (Raymond 2003).

He explains his approach to studying the FLOSS phenomena as 'alien anthropologist mode':

> Since I had to live in so many cultures and speak so many languages when I was younger, I developed a set of reflexes for adapting to social situations that I don't understand at all well. It was a mindset where I thought: 'Here I am, the anthropologist from Mars trying to be completely analytical and objective about what I am seeing.' In doing so I tend to notice what the behavioural patterns around me are, and what the social patterns are. And I tend to apply that perspective everywhere I go. So it was natural that I should study the behaviour of hackers around me in the same way.
>
> (Raymond, quoted in Poynder 2006b: 12)

Raymond argues that in a certain sense he and Richard Stallman are both interested in ethics – it is just that the ethics in Raymond's case are constructed around economic imperatives.

> Like Stallman, I'm interested in the ethical issues around software. Unlike him, I tend to answer the question 'What is ethical?' in terms of economics and game theory. That is, by asking 'What is economically sustainable? What kinds of behaviour lead to productive long-term cooperation among selfish agents?' Until I understood how the economics of Open Source worked, I could only consider talk of the 'ethics' around it to be wishful thinking or ungrounded idealism. Once I did understand the economics, I could tie various ethical claims to the self-interest of software developers, consumers, and third parties, and make sense of them.
>
> (Raymond, quoted in Poynder 2006b: 18)

Although Raymond tries to offer a neutral and politics-free position on FLOSS he has admitted that he sees connections between open source and his politics (Poynder 2006b: 29). He argues that:

> [D]emocracy isn't very interesting to me. It is only relevant as a transition stage: as a way of limiting the power of the state. I do not think it is anything like the final answer. What I would want to see is emergent anarchy.
>
> (Raymond, quoted in Poynder 2006b: 28)

> It [market anarchism] means I want to live in a society that is legally ordered but without government; with no one having a monopoly on force.
>
> (Raymond, quoted in Poynder 2006b: 28)

> In a functioning anarchy you can have influence leaders who are followed voluntarily because they are respected; but you can't have people who have the ability to put a gun to somebody's head and say jump. So you can have people who have influence, but you can't have people who give orders.
>
> (Raymond, quoted in Poynder 2006b: 28)

He also outlines a project for open source which is a form of entryism in which, through an infiltration of the corporate structures that govern software and, by implication, the fact that corporations increasingly need software to run, there is the possibility of subversion by hackers:

> Yes [I do see it as entryism]. And I say that business people do too! I don't believe in hidden agendas. The most effective way to change people's behaviour is to use manipulation techniques even when the person knows you are using them. Not only are those the most effective way to change people's behaviour, but they are the only ones that I consider ethical … And I hope that this demonstration will have long-term political consequences.
>
> (Raymond, quoted in Poynder 2006b: 31)

Raymond's politics are structured around a second-amendment mistrust of government and the citizen's rights to protection from tyranny. He says: 'Love your country, but never, *ever* trust your government' (Poynder 2006b: 31, original emphasis). When asked whether guns should be a deterrent to burglars or government, he replied:

> Well, to me the two cases aren't really distinguishable. I don't draw much distinction between criminals and governments – both groups engage in violence that is not defensible.
>
> (Raymond, quoted in Poynder 2006b: 31)

I now look at how the OSM uses the key signifiers: code, efficiency, freedom, property, the market and the individual, within its discourses.

Code. For the OSM, 'code' is also a key nodal point around which the other signs are organised. This sign refers to an empirical

object, the source code, which, for Raymond, exhibits clear property rights. However, for code to be allocated property rights Raymond identities a form of 'homesteading' in the realm of ideas that is undertaken by the programmer (similar to claiming land in a frontier). This is because the sharing aspect is the important innovative aspect to FLOSS, and yet the nature of sharing would appear to problematise property rights in software.

> The 'noosphere' … is the territory of ideas, the space of all possible thoughts … What we see implied in hacker ownership customs is a Lockean theory of property rights in one subset of the noosphere, the space of all programs. Hence 'homesteading the noosphere', which is what every founder of a new open-source project does.
>
> (Raymond 2003)

Raymond constructs a rational-choice model to explain the desire to produce code but this conception of property runs into difficulty when taking into account the infinitely copyable nature of code. This seems to undermine the scarcity requirement for the normal functioning of a property-related market and so he seeks to change the property value of code from the actual 'copy' to that of the ownership of the 'project' and therefore its history, direction and future.

Indeed, this enables an explanation of the single open source project and also allows the discussion of the taboos of 'forking', or appropriating a project without permission of the previous project owner. As he explains: 'If use were the only issue, there would be no taboo against forking, and open-source ownership would not resemble land tenure at all' (Raymond 2003).

For Raymond, code is hence a slightly more complex concept, taking in not only the source code itself but also the structure, control and direction of the entire open source project. The construction of a web-page thus begins to represent the marking of territory for Raymond because it is a location on the Internet where the project is managed and users congregate to seek copies and information. This also allows a form of property right to be developed which facilitates sharing.

Efficiency. Using a discourse of technical efficiency, Raymond identifies the following key ideas: (1) technical efficiency is derived from many people working simultaneously on a project – 'many eyeballs make bugs shallow' (Raymond 2001); (2) the technocratic belief that the best technical solution is the most efficient; (3) the inefficiency of centralised control systems and big social projects; and (4) the fact that the market is a superior mechanism for delivering goods and services (see Raymond 1999c).

> While a minority of hackers does indeed remain hostile to the profit motive, the general willingness of the community to cooperate with for-profit Linux packagers like Red Hat, SUSE, and Caldera demonstrates that most hackers will happily work with the corporate world when it serves their ends.
>
> (Raymond 1999c)

For Raymond, the profit motive is the greatest source of technical efficiency (see Raymond 1999c, 2003) and this explains his desire to construct a property system on the Internet which could prevent the so-called 'tragedy of the commons' (Hardin 1968).

Freedom. Using a discourse of neoliberalism, Raymond identifies several important concepts related to his ideas of freedom. He appears to use a form of ethical egoism and ideas drawn from psychological egoism (the selfish individual) as a justification for a normative stance stating that because we are *actually* acting in selfish ways, we therefore *should* act selfishly. Or perhaps put more accurately, that we are only kidding ourselves if we attempt to act in altruistic ways.

For Raymond, open source software projects are started because of the needs of an individual, whether they include fixing a bug or writing a function: 'Every good work of software starts by scratching a developer's personal itch' (Raymond 2001).

He dismisses the idea that people may wish to write software altruistically for others. For Raymond, the 'truth' is that altruism does not exist – 'One may call their motivation "altruistic", but this ignores the fact that altruism is itself a form of ego satisfaction for the altruist' (Raymond 2001).

For Raymond, the most important type of freedom is economic freedom. Consequently, rational choice and economics are used to explain how the uncoordinated action of many programmers working on a project mirrors that of the invisible hand of the market (Raymond 1999c). In fact, Raymond explains that the term bazaar[12] is synonymous with the market (Raymond 2001). Hence Raymond highlights the importance of the 'privatization' of the source code and the fact that every project should have an 'owner' (Raymond 1999c, 2001, 2003).

> [Richard Stallman's] four freedoms are a statement of intention, and have the character of an ethical program. The Open Source definition is an attempt to implement that program, and defines in a legalistic way the constraints that Open Source software licenses must satisfy in order for those four freedoms to be fulfilled. So you could say that the four freedoms are strategy and the Open Source definition is tactics… I'll add though, that I believe in a Freedom Zero that Richard [Stallman] doesn't. Freedom Zero is this: Programmers (and creative people in general) have a right to issue their work under any license they choose – closed, open, or purple with pink polka-dots – and have that choice respected. If you don't like the license a programmer is offering, you are free not to use his code, but not to attack his right to choose his own license.
>
> (Raymond, quoted in Poynder 2006b: 20)

As the OSM regards freedom entirely within the realm of an individual's economic freedom and their freedom to work on projects, Raymond does not question the possibility or dangers of prescriptive functions. For example, the Open Source Definition states that 'the license shall not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources' (Perens 2003). This is in marked contrast to the FSF's absolutist prohibition on the mixing of different forms of software and the resulting viral nature of the GNU General Public License (see Stallman 1991c, 2002a: 167).

Property. The intertextual and interdiscursive elements within the discourse of the OSM borrow heavily from Lockean philosophy,

Adam Smith, Ayn Rand and other libertarian and rational-choice theorists such as Mancur Olson. Raymond has a strong notion of utilitarianism and it appears that rational-choice theory forms a basis for his world view (for examples see Raymond 1999c, 2001a, 2003).

However, Raymond also has a strongly social Darwinist thread that seeks to give deterministic causes, for example:

> It is sometimes fashionable to describe human property as an arbitrary social convention, but this is dead wrong. Anybody who has ever owned a dog who barked when strangers came near its owner's property has experienced the essential continuity between animal territoriality and human property.
>
> (Raymond 2003)

Additionally, Raymond rejects altruism and seeks to present co-operative behaviour as an accidental byproduct of the interactions of free agents in a competitive market. Clearly this ignores many of the troublesome theoretical accounts of the inability of rational-choice individuals actually to work together at all due to the so-called free-rider problem (Barnes 1995: 20; Olson 1971). Drawing approvingly on the work of Locke and Adam Smith, Raymond explains the success of OSM by drawing an analogy with Smith's 'invisible hand' of the market, thus seeking to naturalise the process. What is most interesting is the necessity for a property system (*res privatae*) in his convoluted explanation, even when one isn't forthcoming (keeping in mind that the FLOSS functions much more like a commons than a private-property system):

> We have examined the customs which regulate the ownership and control of open-source software. We have seen how they imply an underlying theory of property rights homologous to the Lockean theory of land tenure.
>
> (Raymond 2003)

For Raymond, open source addresses the problems of large-scale computer programming and provides a technically efficient means of bypassing the problems and inefficiencies of highly centralised bureaucratic structures (Raymond 2001).

> I wanted to show the way in which the implicit norms that have developed in the [Open Source] movement are very similar to common law and land tenure, both of which are quasi-instinctive, or wired in the systems of property rights, that human beings develop anywhere where they have rights to commodities worth defending.
>
> (Raymond, quoted in Poynder 2006b: 29)

The market. The conception of social good for open source software is derived from the importance of the technical advantages it is understood to offer, both in terms of software quality and efficiency and in the provision of public goods through the invisible hand of the market (Raymond 1999a). This social sharing manifested within the OSM is built on trust and reliance on others to bring improvements and ideas selfishly but freely into a project (Raymond 2001a). Of course, here is an obvious reification of the market, as markets are wholly dependent on state regulation for their existence and hence involve a tacit trust in government processes and an expectation that contracts will be enforced. Raymond's position on the state clearly is paradoxical, particularly as open source licences rely on contract and copyright law for their very existence, not to mention being dependent on the complex financial and logistical institutions of the state to guarantee certain claims in the final instance (e.g. contract law disputes). Nonetheless, he elevates the market to the status of a quasi-mystical force in the production and circulation of commodities.

The individual. The OSM conception of the social good is strongly neoliberal and libertarian. It privileges both a vision of a highly individualistic social order, a vision strongly influenced by Darwin and the theory of the survival of the fittest, and also holds that collective goods can be produced through the selfish action of individuals.

> [T]he closed-source world cannot win an evolutionary arms race with open source communities that can put orders of magnitude more skilled time into a problem.
>
> (Raymond 2001)

The individual owner is a key figure in OSM (Raymond 2001). Raymond conceives this figure in terms of almost absolutist monarchy and it is clear that democracy is far from the heart of the hierarchical structure that he identifies as most effective at providing leadership:

> The owner makes all decisions and collects all credit and blame. The only possible conflicts are over succession issues – who gets to be the new owner if the old one disappears or loses interest.
>
> (Raymond 2003)

Subject positions. Using nominalisation, passivisation and objective modalities, the OSM's discourse seeks to appeal to the technologically and scientifically trained developer community.[13] Presenting opinions as facts through the removal of subjective modality allows the OSM to outline and draw on a discourse of natural law. The laws themselves are constructed by the systemic mystification of agency. For example, markets are presented as subject to laws of nature, Lockean individuals precede society and the evolutionist law of the jungle is not only an explanatory device but presented as a prescription to societies' economic ills.

Interestingly, Raymond avoids divisive terms such as 'us' and 'them'; there is no group or collective that he claims to be a spokesperson for. Instead, he privileges the individual, and by the use of 'scientific' reasoning and pseudo-anthropological methods, attempts to uncover the truth about hackers and coders in their open source endeavours. This rationalist approach tends to find favour among the audience of FLOSS developers.

Comparative Discourse Analysis

Although not all key master signifiers are shared between the discourses in the Free Software Foundation and the open source movement, there is a struggle at the level of discourse and both movements are conscious of the other when they present their explanations and interpretations (Kuhn and Stallman 2001b; O'Reilly 2001, 2003; Raymond 2001b, 2001c; Stallman 2003d).

In particular: contestation of the concepts of the individual; property's status, as being either collective or private; and the best way of maximising freedom and prosperity, seem to be of particular concern.

For the FSF, 'code' is constructed as a public or collective good that is akin to a utility or law (see Lessig 2002a for a development of this idea). The development of GNU/Linux is the key working example, a collective project that has been shared and worked on freely and remains firmly within the copyleft principles of the FSF. The collective system of GNU/Linux support and development represents the best example of this approach. Some individuals donate 'tools', for example Stallman creating huge amounts of GNU tools and libraries, others supply key operating system modules, for example the Linux kernel or the desktop interface. The Unix system is a highly modular design that allows this kind of collective effort to be easily organised by combining multiple software sources.

In the OSM, 'code' is property owned by an individual who has the right to control and develop it, but this is strictly associated with the notion of the project itself rather than just the underlying source code. Linus Torvalds, the creator of Linux, is the exemplar of the vision held by Raymond and the OSM. Within the OSM literature, Torvalds is the epitome of the individual programmer creating from scratch a Unix system, Linux. He supplied the skills, the vision and he remains a key figure in directing and managing the project; he is often described as a 'benevolent dictator'.[14]

In terms of 'freedom', it can be noted that the freedom to use, modify, read and copy software, designated as a collective good, is key to the FSF. 'Free as in Freedom, not Beer' (Stallman 2003b) is the slogan that the FSF has made famous. The OSM, however, is concerned with the freedom of the individual to work on a project that is of particular selfish interest (hence 'an itch that needs to be scratched', see above). If this is useful to others, then that is the free market working, not a collective, centrally planned project. The individual, the lead developer, is of key importance here, a strong Ayn Randian character that pulls everybody else along by the sheer force of will and power (Rand 1992). Comparisons

between this stereotype of the OSM developer, or benevolent dictator, and the characters within *Atlas Shrugged* (Rand 1992)[15] do not appear to be accidental.

In terms of 'progress', both movements have a strong modernist technocratic model of a linear progress, although they differ in the conception of the ends of the project.[16] The FSF appears to have an Enlightenment ideal of progress as a light to shine on the darkness of tradition, a collective good for all humanity. Comparisons have been made between the FSF philosophy and the principles of academic and scientific research publishing, both strongly influenced by Enlightenment philosophy. In contrast, the OSM has a more brutal ideal of capitalist progress and technical efficiency – to achieve a more efficient and profitable solution without regard to values is optimal.[17]

It is interesting to note that some early open source founders have returned to the FSF due to their discomfort with the direction of the OSM (for example Perens 1999). Indeed, Raymond's eccentric and often strongly libertarian positions on issues from gun-control to terrorism[18] have alienated many potential supporters (Raymond 1999b, 2002a, 2002b). The thread of libertarianism runs deeply through all of his writings and it is clear that this has informed his disguised attack on Stallman in *The Cathedral and the Bazaar* (Raymond 2001). This book has become a major and influential text for open source, particular as the manner in which it is written highlights the technical rather than the social or political achievements of FLOSS.[19] In his work, Raymond's belief in the power of rational choice, namely uncoordinated selfish action to produce collective goods, utilises an American anti-government, anti-centralist rhetoric. His call for minimal government, and sometimes for no government at all (Raymond 1999b), borrows ideas about the sovereignty of the individual drawn from Locke's ideas of pre-governmental life (Raymond 2003). For someone keen to avoid values and ethics and concentrate on the technical and rational, his ideas are permeated with his particular ideological position.

Conclusions

This chapter has examined the discursive struggle taking place between the open source movement and the Free Software Foundation. Using discourse analysis, it has demonstrated that there is an attempt to achieve hegemony by fixing the elements within the discourses surrounding the production of code. This could have wider ideological implications for the Internet community and, indeed, society at large.

Through an analysis of the discourse produced by these two movements, I argue that it appears that the OSM is providing a more convincing order of discourse to the technologist community of FLOSS developers (and increasingly to wider publics that include free culture groups such as Creative Commons). OSM rhetoric appeals to wider arguments from neoliberal economics and technocratic discourses and the more overt interdiscursivity from neoliberal texts (Raymond 2003). It also appeals to a wider societal suspicion of organised politics and the political sphere, a development that has grown in the last century in the West. Here, an appeal to 'openness' can sometimes be read through a lens of freedom and transparency that privileges an individualist notion of how society should be organised.[20] This is opposed to the FSF's 'appealingly utopian – and perhaps quixotic – notion that all information should be shared' (Scoville 1999).

Indeed, the importance of hegemony within this order of discourse has greater political and philosophical implications when considered in the context of the growth in popularity of the ideas surrounding anti-copyright, copyleft,[21] the public domain and issues of freedom and democratisation. This is demonstrated in the growth of free software and open source projects ranging from hardware designs, to record labels, personal music players, books and online discussion sites.[22] It has even been considered an issue of national security, as demonstrated by the governments of Brazil, India, China and South Africa (Weber 2003) seeking to avoid dependence on Western, mainly American, proprietary

computer software products, particularly for governmental, educational and military usage.

Raymond's open source methods thus have implications beyond the narrow domain of computer software engineering. Raymond and others have been quick to use open source as a potential vision of social reality to explain the fragmented, complex and highly technological nature of contemporary society (see Osborne 2007; Sachs 2007). The OSM also argues that grand general theories are unnecessary and therefore concern for these wider shared values is not useful. One of the most alarming political implications of an open source ideology is that it can be used as a justification for ignoring the concept of the 'public'. Thus the public good is best achieved in a manner akin to the 'invisible hand' described by Adam Smith:

> [The individual] intends only his own security; and by directing that industry in such a manner as its produce may be of the greatest value, he intends only his own gain, and he is in this … led by an invisible hand to promote an end which was no part of his intention. Nor is it always the worse for society that it was no part of his intention. By pursuing his own interest he frequently promotes that of society more effectively than when he really intends to promote it.
>
> (Smith, quoted in Winner 2001: 293)

Here, each small fragment of the system is responsible for itself and its relationship with those directly related to it (rather like the interfaces used in software). This chimes with populist neoliberal and libertarian political arguments that are commonly utilised by politicians of all persuasions to some degree.

Many engineers attempt to take a consequentialist position in relation to technology. They are trained in rational positivistic approaches to solving problems and concentrate on mean-end rationality (Feenberg 2002).[23] This instrumental approach informs their training and influences their personal standpoint (Barbrook and Cameron 1995; Slashdot 2001a, 2001b) and has been described as the 'engineering philosophy of technology' (Mitcham 1994). This philosophy

> begins with a justification of technology or an analysis of the nature of technology itself … it then proceeds to find that nature manifested throughout human affairs and, indeed, even seeks to explain both nonhuman and human worlds in technological terms. Culture is a form of technology (Kapp); the state and economy should be organised according to technological principles (Engelmeier and Veblen); religious experience is united with technological creativity (Dessauer and García Bacca) … Engineering philosophy of technology might even be termed a technological philosophy, one that uses technological criteria and paradigms to question and to judge other aspects of human affairs, and thus deepen or extend technological consciousness.
>
> (Mitcham 1994: 62)

For engineers, the OSM offers a common-sense approach both in its language and philosophy, by seeking 'obvious' provable solutions and explicitly positioning itself apart from political debate. For instance 'OSS is not about politics, it's about software. Don't lose sight of that' (Slashdot 2001). The OSM and its followers generally view politics within the technical sphere with misgivings (O'Reilly 2002). However, they are quite happy to view the political sphere as technological. By positioning technologists' best practice as apolitical they seek to create an opposition to the FSF's value-based discourses even if they agree with its other technical discursive elements.

The FSF, with its emphasis on the subject positions 'us' and 'them', requires the reader to take a deontological ethical position. It also highlights the values of the 'commons' (in some senses this is akin to the concept of the public). To many, this approach seems old-fashioned and unscientific (Scoville 1999). Without a convincing call to politicise this technical sphere more generally, the FSF will thus continue to be seen as 'wildly utopian' (Scoville 1999). In order to contest this hegemonic discourse, the FSF will need to make a more concerted attempt to deal with the strongly objective modality of the discourse of the open source movement and the implication that technology is a purely value-free activity. The Free Software Foundation could address the deeply divisive nature of its discourse by positioning itself as having both a moral

and ethical position, *and* a scientific and technical one. However, this would be to concede the possibility of a non-political technical activity and a value-free technical discipline and risk undermining its moral and ethical position.

Indeed, it is interesting that the Free Software Foundation has not sought to widen its discourse from that of deontological ethics and community-shared processes for the production of social goods to that of a wider discourse of democracy.[24] This could take the form of a concentration on particular 'democratic freedoms' that the free/libre software movements offer in terms of participation, value-sensitive design and transparency (and see Nissenbaum and Howe (2003) for an example of how this could be achieved). It could also draw on discursive elements from wider democratic debates in society (Barber 1984; Habermas 1988, 1992, 1997), theories about the democratisation of technology (Feenberg 2002; Sclove 1995), issues surrounding public debate for steering technological policy-making (Margetts 1996; Winner 1986) and the need for public involvement in the production of far-reaching and highly invasive technologies, for example, genetically modified (GM) foods, and in the environmental movement (Feenberg 1995; Winner 1986).

The fact that free software has failed to do this may betray its own underpinnings within an 'engineering philosophy of technology'. Even when it wishes to highlight community and the common it presents them as of benefit to particular technological understandings of human experience. An attempt to appreciate the non-technical aspects of the development of computer code could have raised awareness of the nontechnological. However, Stallman's position is similar to Veblen's in *The Instinct of Workmanship*, where he wished to argue for the reorganisation of economic and social life so as to free engineering principles from commercial and political control. Thus the problems of technology are thought of as resolvable through the use of more technology and not less (Mitcham 1994: 38).

An important undercurrent to this analysis is that as technology increasingly colonises and structures more aspects of our lives it is becoming increasingly important that the constitutive nature of

technology as socially shaped is recognised (Kesan and Shah 2002; MacKenzie and Wajcman 1999). If computer code is analogous to law (Kesan and Shah 2002; Lessig 1999), then it is clear that without some form of democratic accountability the code-based regulation of human behaviour will continue to lack legitimacy (Habermas 1988; Sclove 1995). Therefore the discursive struggle between the FSF and the OSM is an important challenge for wider society to recognise that values are being instantiated within technological forms that can and should be contested before they become ingrained. Introducing democratic accountability to code may well be *the* democratic challenge of the twenty-first century and steering the implementation of technological artefacts will increasingly contribute to our ability to keep our future open and democratic. Indeed, the debates between these FLOSS movements are beginning to stimulate a much broader interest in these key issues.

In this chapter I have undertaken a discourse analysis of the key texts of FLOSS, particularly the work of Richard M. Stallman and Eric S. Raymond, who stand as theorists, advocates and practitioners involved in FLOSS development. This is interesting for a political economy of FLOSS because throughout their contestation these individuals, and their followers, seek to locate some justification for their positions through links to wider discourses and questions raised about the restructuring of the North's economies around information and knowledge (that is, towards the creative economy). These debates point to deeper disquiet within a particular technical community about property rights, legitimated through particular discourses of an 'information society'. By subjecting FLOSS discourses to a detailed analysis, I have tried to show that the underlying ideological battles over the legitimacy of the 'right to code' are being fought through a surface veneer of technical skill, efficiency and software development methodology. However, FLOSS texts also contain encoded sedimented discourses from the broader society which contest claims regarding the expansion and range of changing property rights; the proper ends of man's labour; and questions over the form of human freedom. FLOSS also

highlights important and recurring issues about human control over technological forms through questions about the control of future code production. Although a seemingly esoteric and technical arena of discussion, these questions are timely for wider society, and indeed are now being taken up in discussions outside of FLOSS (see Benkler 2006; Osborne 2007; Sachs 2007). In the next chapter, I bring these themes together to understand how FLOSS ideas contribute towards a potential liberatory moment for FLOSS, and the implications of this move.

6

THE POETICS OF CODE

Throughout this book, I have argued that FLOSS is a unique laboratory for the study of intellectual property rights, for observing the rising importance of the commons, and for examining the wider ramifications of the informationalisation of society. Using a political economy combined with discourse analysis, I have tried to show that FLOSS discourse and practices give the researcher an important vantage point. This can be used to understand wider public issues through a lens that casts useful light on the implications of the growth in IPRs, the commercialisation of information and knowledge and the effects on social interaction and production.

Firstly, I outlined the general themes that are a useful context for understanding FLOSS, including: the World Wide Web (www); the Internet technologies underlying it; hackers and users; the commons; computer code; and the politics of code. I argued that each of these is an important contributor to the way in which FLOSS history and culture has developed.

I then looked at the way in which other discourses and theories of the 'creative economy' and the 'information society' also tended to universalise claims for all workers, focusing on issues relating to claims over a 'new' capitalism, the claims for the move to an information-based society, questions of contemporary knowledge and creativity and a brief history of the software industry. Within FLOSS many of these informational discourses, particularly the importance of new economy and technology as a site of freedom draw from a more mythological rather than actual history yet serve to legitimate particular FLOSS approaches. I discussed the arguments surrounding the creation of new knowledges and how

it is necessary that ideas and concepts may be freely exchanged. Hence, if ideas and concepts can be digitally locked and controlled, our ability to build on the past would be diminished. Thus a non-owned public domain, or commons, of freely shared concepts and ideas, on which anyone may draw, without diminishing the availability of ideas and concepts for others, is crucial for democracy and creativity.

Linked to this is the universalisation of the experience of FLOSS to wider society. There is some argument that as 'early adopters' of new technology FLOSS points to the direction of changes in intellectual property rights and their effect on users, although these claims are not always substantiated. Here, though, I also examined the background history of the software industry, placing in context Microsoft's dominance of the desktop market, and how the changes from a focus on 'machine calculation' to a focus on 'machine logic' neatly capture the emphasis that is made by FLOSS developers on the importance of computer code over access to hardware.

I then presented an excursus on the concept of the common. I outlined a genealogy that includes *res nullis* (things belonging to no-one), *res privatae* (private things), *res publicae* (public things), *res universitatis* (things belonging to a group), *res communes* (common things), *res divini juris* (things that are under the jurisdiction of the gods) and *res imperium* (things owned in the international arena). This genealogy highlighted how these property forms are based on contingent turns in history, rather than rational or planned outcomes. These concepts were intended as an ongoing contribution to research into the way in which we understand our current formation of property rights, and more importantly, to bring back to our attention older forms of communal and shared rights (in many cases the concept of ownership would not apply here).

It is important to remember that the question of the commons is not just a legal matter. For an institution such as copyright or copyleft to be effective, it has to be confirmed as a structure by actors. If the state, or any other governing force, is to protect the commons it will take political action and need to be legitimated

as an institution *by* actors. Here questions over existing political institutions become pertinent. Do we need new political structures? Certainly some theorists have seen the promise of a new form of politics in online forms of sociality and new 'precarious' networks and ways of living (see Dyer-Witherford 1999; Hardt and Negri 2000; Benkler 2006). They argue that the activity of people within social networking technologies opens communicative dimensions that promote a new form of sociality and the potential for a new form of politics – called the *multitude* by Hardt and Negri (2000). Additionally, they argue that these online social formations have begun to change social norms and attitudes, examples being the problematising of certain forms of property right, such as copyright (see Coombe 1998; Lessig 2004). Despite current attempts by the content industries to criminalise sharing at the discursive level and close down these spaces, in many respects people now view sharing as legitimate (Vaidhyanathan 2004). This appears to be an example of the transfer of the norms of FLOSS culture into more mainstream thinking.

I applied these ideas in a political economy of the 'limited totality' of FLOSS, where I looked at the history of free software and open source, the politics of code, the ethics of code and the economic base of FLOSS. This was intended to draw attention to the ideas, material capabilities and institutions that serve to reinforce the norms and practices of FLOSS. It also highlighted the problem of claiming to be autonomous of capitalism, or indeed of being some radically new form of social organisation. Indeed, FLOSS is both critically dependent on capitalist firms and individuals to fund its development, but it also forms part of a larger informational-industrial-military complex from which it draws its organisational and logistical roots. Through an examination of the networks of financial support and corporate interventions in the development of FLOSS, both as a technical and as a discursive activity, it is clear that the current shape of FLOSS groups and their debates have been shaped by this sometimes overlooked history. Additionally, throughout its development, FLOSS has been influenced by previous discussions regarding the shift in Northern economies and the impact this

would have on information, knowledge and creativity. Even though this book has concentrated particularly on the US/EU dimension of FLOSS, it is clear that these debates resonate across the world. Additionally, it has been shown that rather than external or discontinuous to the history of software, FLOSS forms a continuation of the development of software development practices and management, demonstrated by the way in which FLOSS has been easily absorbed into the mainstream software industry (particularly with the move to an 'open source' language of articulation seen in Web 2.0).

Finally, I undertook a close reading of the discourses of the key theorists/advocates/practitioners within FLOSS, namely Richard M. Stallman and Eric. S. Raymond, and showed how the legitimations and justifications presented by these two important actors sedimented particular ideological views on the history and future direction of FLOSS. Using this analysis, I drew out the ideological struggle and showed how these two subject-positions point to the 'poles of attraction' within the FLOSS community that inform the discursive justifications and self-understandings of FLOSS practices. These readings demonstrated the extent to which FLOSS discourses have incorporated many of the debates that have been prevalent in information society theory. Perhaps also they serve to represent deeper concerns about the loss of 'craft' skills in computer programming and the move towards an industrial informational economy. Rather than pointing towards an anti-capitalist economy or an alternative conception of technology, both movements share a broadly pro-technology linear model of the development of human knowledge. Although the question of control is highlighted, particularly in the discourses of the Free Software Foundation, both tend to view the workings of technology as unproblematic and further technical development as a human good. This is interesting because many alternative movements and charities have had sympathy with the arguments of Stallman, who roots his arguments within a particular conception of human freedom. Nonetheless, neither movement offers an *alternative* idea of human means, beyond that provided by the market (although in Stallman's discourses a re-embedded market

is made secondary to human needs). Indeed, if, as I argue, we should view Stallman's interventions as less radical then they are sometimes made out to be, we can understand them as a form of *res universitatis*, a kind of trade union or guild for computer programmers. This essentially seems to boil down to a social democratic vision of society, where the market is required but needs to be tamed through unions and civil society. However, that should not distract from the kernel of radicalism that FLOSS still contains, and which is sometimes articulated in its discourses. By highlighting the communicative dimension of social development, the need for technical education (a kind of *critical code literacy*), and the importance of the human at the centre of these struggles, they contribute to a humanistic turn in engineering philosophy. In many ways, the belief in an optimistic outcome of Enlightenment rationality and in the positive values of human freedom and progress are critical to the self-understandings manifested within FLOSS culture. These discourses have helped to alert the public and politicians to the dangers posed to liberal democratic society if warnings about knowledge privatisation are not heeded. By viewing FLOSS practices and discourses as a problematisation of the privatisation of knowledge, FLOSS draws political, economic and social attention to the often forgotten nature of a commons, especially one that is culturally or linguistically created (and here, technologically reinforced).

My research has attempted to place FLOSS developers' private concerns over restrictive property-based code within the larger public shift to knowledge and information as key areas of economic growth (the information society) and also the rise in intellectual property law and the commodification of knowledge, information and culture (i.e. the creative economy).[1] The particular implications of the expansion of property rights to the realm of computer code and its interpretation by the programming community can then be generalised to questions regarding the larger implications of a purported 'semiotic democracy' in which capital attempts to own and control meaning and culture (and mediated through code). As I have shown, the existence of free software and open source is precariously balanced between the need for a common public form

in which community and creativity can blossom and the reliance, to a large extent, on private corporations and the skills and donated time of employees and individuals operating within the market. This reality of FLOSS, which is surprisingly ignored in many accounts that laud the networked, non-hierarchical structures of FLOSS, was explored using the concepts I introduced, such as *res communes* and *res universitatis*, which help to explain how FLOSS might become subject to privatisation or undermined by changes in intellectual property laws. In contrast, I suggest that supporting the vibrancy of the discourses and practices of FLOSS might best be achieved by acknowledging the precarious nature of its reliance on the market and exploring the democratic potentials through an experimentation instead with state organisations, such as universities, government departments and schools, providing financial security or support – which in many ways is already beginning to happen.

This book argues that FLOSS represents *technologies of the commons* that will need to be developed both in theory and practice. Here I would like to suggest that we might re-examine the concept of the commons with a view to a *re-enchantment* of the commons. I suggest that this enchanted form of common-ownership may be the only safe repository for a commons that will constantly be under threat from cooption and privatisation. An international body, such as the UN, may provide one means of protecting it. Crucially though, it would have to be supported by political action, for the creative citizen will be required continually to defend the space of the commons and a concept of the political is crucial to ensure the animation of the concept and practice of the commons. Whether this calls for a new form of active state, reinvigorated by a vibrant 'open source' civil society that can also act to prevent these transgressions, is also an important question.

Secondly, I would like tentatively to suggest that free software points toward a different relationship with the 'work' of our hands and therefore with our ability to flourish as human beings. Free software is not directly linked to necessity, and is in many ways similar to the creation of an artist – whom Arendt (1989) identified

as the only real 'worker' left in society. She argued that we would not be free until we realise that we are subject to necessity and liberate ourselves from it. Whilst we are forced to 'make a living' we will always be caught in a never-ending spiral of labouring and consuming. As technology creates more 'spare time', the shaping of our desires by the advertising industry makes us crave for more – a hunger which is only satisfied by consuming more. Free software is created here, in the space of consumption; however, it differs from consumption, as it is productive and creative. To create free culture is to contribute toward culture rather than consume (that is, destroy it). Further, the relationship between free culture, craftsmanship and humanistic ideals is similar to calls for a humanised technology made by writers early in the twentieth century (cf. Mumford 1963; Giedion 1969; Heidegger 2000). It also is an active strategy against fears of an autonomous technology and suggests alternatives to technocratic solutions to political issues (see Winner 2001). This suggests that FLOSS ideals may contribute towards more radical critiques of the information society and changing relationships with digital technology, an issue I wish to discuss briefly below.

The Rise of the Cultural Commons

There are two main issues relating to the privatised nature of the commons in the twenty-first century. (1) Where previously the commons was a non-owned community resource, today we see many dispersed private projects opened up ostensibly for the commons (such as the Creative Commons project) but which still reserve copyright and intellectual property rights to the private licenser. This book has examined whether a new concept of the commons is beginning to develop through conflicts over intellectual property rights, perhaps as a form of community-owned intellectual property (such as a *res universitatis*, as a state-backed utility or collecting society that protects and licenses materials freely) or through a market-based system based on contract and private property (that is, *res privatae*); and (2) The concept of the creative citizen as a new subjectivity is being contested by hackers within

FLOSS, in practices like the distribution/creation of open source material/software and discursive practices that reject the policies, advocacy and governmentality (Rose 1999) of state interests that promote corporate ownership and copyright culture.

By concentrating on a political economy analysis, this book has demonstrated that a historical and political contextualisation is critical for the understanding of FLOSS and the free culture movement. Although possibilities are opened through the radicalisation of the ideas behind these movements[2] (see Hardt and Negri 2004; Berry and Moss 2003), ultimately FLOSS's location within capitalism explains their structural and institutional form, and it provides their condition of possibility (such as technology-mediated intellectual property laws). Indeed, more generally, the method of political economy has much to recommend it when trying to critique the argument that societies are moving towards an informational economy. If informational products are taking up larger slices of the North's GDP and the value of intangibles is becoming greater and greater (often backed by the code mediation of asset bubbles in physical property like housing), it is important that we appreciate technology's role in facilitating these developments. Understanding the role of technology and computer code in mediating our experience of the world through models of reality also becomes more important (such as the complex software-mediated division of time and space into smaller and smaller units impractical to undertake on paper) – particularly with the increased introduction of digital artefacts into our lives.[3] FLOSS, which gives access to source code, allows the reader to see *inside* the code and begin to understand how this code-mediation takes place.[4]

Together these issues are bringing a great pressure to bear both on existing firms and on the way in which information is both used and understood in our societies – for example, in the latent contradiction between a liberal democratic need for the 'public' to have access to information and knowledge in order to participate in a public sphere, and the view that information is property that needs to be controlled, packaged and sold to a 'customer'. Indeed, this also problematises calls by governments for a more

culturally and creatively equipped citizen who can negotiate the informational landscape and generate and build new knowledge and ideas, with the paradoxical commitment to an incentivisation of this creativity through property rights.

In this book I have shown how multinational corporations are able to use their vast resources of people, skills, knowledge and money to neutralise threats from FLOSS through their ability to generate profit and also through their structures of power. I have also tried to show how the debates are managed, controlled and steered by careful use of discourse and language. Nonetheless, the potentiality of commons-based production has continued to spread into other areas, and its discourses of openness and transparency have even fed into the political process (such as calls for open source politics (Sachs 2007)). As a developing form of production, FLOSS is interesting where the possibilities exist for it to become radicalised and politicised (much to the chagrin of some open source advocates) and this points to the importance of understanding FLOSS not just as a technical project, but as an ongoing political project. It is contingent and may be developed in both progressive and reactionary directions, but through political action it could become a critical tool against the prevailing ideas promoted by private capital in organising and structuring the world.

The Commons as Political Imaginary

In my discussions of the structure and possible scenarios for organising FLOSS (see Chapter 4), the question arose as to whether a single, completely common legally defensible contract has been proposed (for example, a unified copyleft system). This could take the form of a single copyright licence (such as the GPL) that might be shared and used right across the spectrum of FLOSS groups. I certainly see one of the challenges to FLOSS coming directly from attempts by organisations such as the Creative Commons to present a unified, simplified and rationalised licensing model that will have the ultimate consequence of encouraging a depoliticisation of the issues raised by FLOSS, and more generally of free

culture. There are substantial benefits to this idea, in terms of the current incompatibility of various FLOSS licences, the restrictions on use some licences require and the need to form a community based on shared principles. Nonetheless, I believe that this would be a mistake. One of the most notable aspects of FLOSS culture is the vibrancy that the political contestation of its underlying principles brings to the surface through the agonistic struggle of each group within FLOSS (and I think that it is important to stress the multitude of different FLOSS groups beyond those discussed in this book). This gives FLOSS both an important technical advantage (such as an innovative software development environment), but also a political advantage in that it encourages debate and multiplicity in the contestation of the technical.

A further question that is brought to the fore by this book is whether the concept of contract and commons are intractably antithetical. Usually the contradiction between them would be assumed. However, FLOSS problematises the assumption that private right (contract) and a common good (sharing) are completely antithetical. FLOSS functions as a commons *because* it is based on private right due to the way it (mis)uses copyright to give away specific rights to users. The fact that FLOSS has taken the legal structures of copyright and used them in a similar way to methods used in computer software – as a function that can be bent to human will – shows a fascinating approach to law that again suggests an emphasis on a new form of humanism (centred on human freedom). Equally, as society increases its reliance on technology, the question of 'expert' decision-making becomes more important. Here the scepticism FLOSS shows towards the bureaucratic or technocratic organisation will, I think, become increasingly important (although a decentred hacker-elite might not be an acceptable solution to all). The *freedom to tinker* that hacker groups emphasise also has the crucial function of bringing human control of technology to the fore. In an age of purported post-humanism, I think that this new form of *code*-humanism is a necessary political imaginary that will have huge consequences, providing it is not crushed under the weight of vested technological interests (such as the software industry, government indifference

or attempts to centralise or unify the disparate FLOSS activities). Winner (2001: 323) outlined the beginnings of a similar 'humanist technology' as:

> … the crucial awareness that technology in a true sense *is legislation*. It recognises that technical forms do, to a large extent, shape the basic pattern and content of human activity in our time. Thus, politics becomes (amongst other things) an active encounter with the specific forms and processes contained in technology … when one is able to acknowledge that modern technics, much more than politics as conventionally understood, now legislates human existence. New technologies are institutional structures within an evolving constitution that give shape to a new polity, the technopolis in which we increasingly live … *Different ideas of social and political life entail different technologies for their realisation* … The notion that technical forms are merely neutral and that 'one size fits all' is a myth that no longer merits the least respect.
>
> (Winner 2001: 323–5, original emphasis)

However, the future directions of this response to technology seem to me to point in one of two directions: (1) an individual will to control (which could still be considered a technological politics); or (2) a form of collective decision-making through a democratisation of technology. Although FLOSS shows how the structures that surround us can be made responsive to the needs of citizens (even when, on the surface, it appears that law operates only *upon* us), it will be interesting to see how its political maturity evolves and whether it will move from the focus on individual human freedom to a more collective conception. So even though FLOSS acts as a political imaginary – showing that human beings *can* change the system in ways that transform the way in which we work together, and moreover, in a direction that is in the hands of citizens rather than through lobbyists, lawyers and legislators (copyleft licences notwithstanding) – it does not necessarily tell us how we should organise our response to the status quo. However, the way in which FLOSS groups, and increasingly free culture movements, favour the 'network' as an alternative means of organising human action (against formal democracy and hierarchical structure) suggests interesting means for developing

a counter-power (whether against the state or capital). It also mirrors Winner's (2001: 326) call for 'the development of these forms [of technology that] proceed through the direct participation of those concerned with their everyday employment and effects'. That is to say that, 'technological systems ought to be intellectually as well as physically accessible to those they are likely to affect'.

The possibilities offered by free culture are not (yet) completely linked or mediated through the operation of corporations and necessity. People can still write code, blog and share their thoughts on the web, and this act of sharing is also one of communication. However, it is a fragile space, and the question remains as to whether those involved in free culture and FLOSS will be able to prevent it from being overcoded, controlled and channelled towards (mere) consumption. Political economy shows that the vast resources of corporations can shape, control and co-opt projects that are perceived to be valuable.

If we begin to view free software and open source no longer purely as necessity or as a form of instrumentality (that is, as not just a technical activity), I suggest we can reposition these practices within the realm of human creativity and freedom. If free software lies outside the sphere of labour,[5] then following Arendt (1989), we can, perhaps, begin to understand FLOSS as a possible prerequisite for the beginning of Arendt's conception of political activity.[6] FLOSS is interesting in that it seems to contribute towards the conditions of possibility that Arendt argued for, namely: 'work'[7] and 'action'[8] – necessary for humans to perform great deeds, and to create a trace or memory. For Arendt, the realm of economics and markets is the sphere of necessity – you do not have the freedom to act, creatively or politically, as an agent. Conversely, free software and free culture seems to be constituted communicatively (e.g. as a conversation between volunteers),[9] and could, therefore, offer the political imaginary of a decentralised, transparent, non-market commons-based production, within which may lie the seeds of a new politics – the *politics of the commons*. For example, Fuller's (2003) concept of 'critical software' aims to act politically, to subvert existing codes and to give agency and freedom to the usually passive user – by,

for instance, avoiding the way in which a word processor can control and steer the user by 'suggesting' spellings and grammar that potentially shape the meaning or direction of human thought. Similarly, FLOSS, by giving away the source code, simultaneously gives agency to the producer and to the user, creating the scope for action rather than merely directing and controlling the user.

To develop theory in this direction we might borrow the method of Heidegger (2000), and uncover the original meaning of poetics: *Poeisis*, which is to bring forth, to lead or to bring out – in other words, to produce. He argues that, for the Greeks, poetry is a process whereby something that was not there, not present, is now created, brought forth and produced in harmony with the world. The opposite to *poeisis* is a *challenging-forth* (Heidegger 2000: 320), to rip out of nature, to enframe and order, put into hierarchy and system, to exploit and to turn into resources (or *standing reserve*). The latter, of course, is the very basis of global capitalism today and of its devastating impact on the environment and our social world.

In relation to FLOSS, this suggests we should turn towards *poeisis* as a project to form groups and social worlds that are committed to forming a common 'gathering', to bring forth and, through code production, offer another world – not a fantasy world that rejects technology or modernity itself – but an alternative to the world governed by private property, individualism and market exchange. This requires that we connect to the world of exploitation, not to represent it, but to transform it into another form of truth:

> Essential reflection upon technology, and decisive confrontation with it must happen in a realm that is, on the one hand, akin to the essence of technology and, on the other, fundamentally different from it … Such a realm is art. But certainly only if reflection upon art, for its part, does not shut its eyes to the constellation of truth, concerning which we are questioning. Thus questioning, we bear witness to the crisis that in our sheer preoccupation with technology we do not yet experience the essential unfolding of technology, that in our sheer aesthetic-mindedness we no longer guard and preserve the essential unfolding of art … The closer we come to the danger, the more brightly do the ways into the saving power

> begin to shine and the more questioning we become. For questioning is the piety of thought.
>
> (Heidegger 2000: 340–1).

Here, Heidegger looks to art to provide the possibilities of questioning technology. However FLOSS suggests that perhaps the 'saving power' lies not only in aesthetics as a critical realm, but also through the democratising paths suggested in the opening up of computer code and the *techné* of the hacker ethic. Throughout this book, I have undertaken a critique of FLOSS through political economy, which is the negative moment of this project, but perhaps *poeisis*, the positive constructive moment, can show other possibilities revealed through a new world picture. This is where FLOSS and the politics of the commons are relevant to a wider audience of active citizens. I believe that the wider implications of this rethinking of technology can take place on numerous levels, but the ones relevant to this book seem to me to be political action directed against autonomous black-box technology, and the other practising and realising the critical project of the *poetics of code* (as *poeisis*) through the possibilities suggested in and on the way to free culture.

NOTES

Preface

1. See for example Lessig (1999), Lessig (2004), Vaidhyanathan (2004), Weber (2004) and Ghosh (2005).
2. We 'stand on the shoulders of giants' to quote Newton (who himself 'borrowed' the term from John Salisbury).

1. The Canary in the Mine

1. The Indian government later challenged the patent arguing it was a case of blatant thievery (or bio-piracy) and showed research and pre-patent usage of the turmeric that convinced the US Patent Office to cancel the patents. However, this has not stopped the flow of patents on traditional knowledges and discoveries from non-US sources that produce profit for the corporations and patentees and nothing for the 'sources' (see Drahos 2000).
2. Paradoxically, the court also found that awarding property rights to the patient would somehow hinder the progress of science and the communal sharing of knowledge, and yet giving the same property rights to the doctors was considered necessary to give an economic incentive to future research (Boyle 2003: 24).
3. '[T]he [Creative Industries Intellectual Property Rights Forum] felt it was important to highlight the positive aspects of protecting intellectual property (IP), to highlight the economic contribution of the creative industries and to map that back on how IP touches the lives of people across a range of audiences … A series of copyright scenarios [should be] created in a format, that is easily accessible to the key audiences – children and young people in education; people in further and higher education; individual creators and creative businesses; and consumers' (CIIPRF 2005: 2). For more information about this programme see www.creative-partnerships.com
4. See http://news.bbc.co.uk/1/hi/entertainment/film/3881587.stm
5. See Lego's adult and child friendly advice at www.lego.com/eng/info/fairplay.asp

6. It was estimated in 2002 that the GNU/Linux operating system would have cost over $1 billion to have been commercially developed and has taken approximately 8,000 person-years (Wheeler 2002).
7. The only concrete linkage made between AHRC research in the humanities and arts was to the improvements that could be offered to computer game design and production (HC 310-I 2007: Q30)..
8. The ease of copying in the digital environment and the social acceptance associated with it threaten revenue streams for producers that seek to make a profit.
9. Whilst stressing a critical discourse analysis perspective to develop analysis within a coherent discursive *moment*, this chapter views critical discourse analysis as a constellation of approaches and methods to hermeneutically examine the radical incompleteness and contingency of hegemonic discourse (Laclau 2001; Laclau and Mouffe 2001; Phillips and Jørgensen 2002).
10. Hyde (2006: 71–3) discusses organ donation of one motivated by a sense of a gift rather than a market economy, for example.
11. A potlatch ceremony is one where the gifts of a tribe are consumed, often ostentatiously, to demonstrate status and rank.
12. The first sale doctrine essentially means that once you have purchased a book or record, you are free to give it away, sell it secondhand or throw it away. Many end user licence agreements (EULAs) now attempt to restrict or deny these legal rights with digital objects.
13. Web 2.0 emerged as a slogan after a conference organised by O'Reilly Associates, a publisher of open source technical manuals. The conference asked delegates and speakers to imagine that if today's Web was considered a version 1.0 (usually considered a buggy, unfinished and clunky version of software) then what would the second generation of the web, or 'Web 2.0', look like. This has proved to be an extremely influential idea and has excited many new software development companies to pursue new and challenging web technologies and platforms.
14. When Steve Jobs made the claim of 'open source for the rest of us' he was presumably referring to the ability of Apple to transform complex technology into an easily used form. Open source is notoriously technical and user-unfriendly and yet it offers a huge array of useful, clever and innovative technologies (such as virtual desktops). In doing so, Apple 'adds value' and charges for the slick front-end it places on these technologies. This leads directly to claims from some open source proponents that large companies harvest ideas from FLOSS without intending to put anything back into the commons. See www.apple.com/server/macosx/ for more claims to make open source easy, for instance: 'Tiger offers 200+ new features and builds on more than 100 of the latest open source projects'.

15. Here the term 'open source' can be thought of as an empty signifier (Laclau and Mouffe 2001), which has a cutting-edge ring to it without necessarily meaning anything in particular.
16. Many of these technologies were released under the Berkeley Software Distribution licence (BSD) which allows any derivatives, distribution and modifications to be made providing the regents of the University of California are attributed as original copyright owner.
17. Protocols are software 'abstractions' that simplify the underlying hardware to allow computers to communicate with each other easily. An example of this is TCP/IP (Transmission Control Protocol/Internet Protocol) which organises the passing of data across the Internet by creating a table of standardised addresses for data delivery and a method of chopping messages into pieces, called packets, which are sent across the Internet in a manner analogous to shipping containers for physical goods. That is, that you need not know the content of the message, as on the outside all packets look the same, with an address to send to.
18. Software is sold on the basis of a licence to use it. However this does not mean that the buyer owns the software, in fact the Microsoft licence explicitly states 'The OS Components are licensed, not sold' (Microsoft, 2003)
19. Throughout this book I discuss the contributions and actions of many of the participants in FLOSS who often, like Richard Stallman, act as theorists, advocates and practitioners of the FLOSS ethos. The reader should note the overlap between their roles as actors within FLOSS and the way in which they are discussed in this book.
20. That is to say that there is a plethora of FLOSS licences available under a number of different names and licence conditions (many of which are incompatible with each other) but generally speaking the GPL is considered the epitome of a *free software* licence and the BSD licence is the epitome of an *open source* licence.
21. This is interesting in relation to the fact that many arguments about FLOSS software are about the lack of a responsible entity for software failure (i.e. relying on networked globally dispersed groups). However it is noticeable that proprietary software firms make it extremely clear through their licence conditions that they cannot be held responsible for any malfunction or error that destroys data or causes a mishap. The Cisco (2006) Software License Agreement is typical in stating: DISCLAIMER. EXCEPT AS SPECIFIED IN THIS WARRANTY, ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS, AND WARRANTIES INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE,

NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE, ARE HEREBY EXCLUDED TO THE EXTENT ALLOWED BY APPLICABLE LAW… IN NO EVENT WILL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY LOST REVENUE, PROFIT, OR DATA, OR FOR SPECIAL, INDIRECT, CONSEQUENTIAL, INCIDENTAL, OR PUNITIVE DAMAGES HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY ARISING OUT OF THE USE OF OR INABILITY TO USE THE SOFTWARE EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES (Cisco 2006, original emphasis).

22. A hacker is traditionally considered to be somebody adept at computer programming in contrast to the more common usage as a term of abuse or having associations of criminality (Taylor 1999; Moody 2002: 16).
23. Creative Commons used BzzAgents, an online word-of-mouth advertiser, to publicise its work. However, there were criticisms about the use of a commercial company to, in effect, place advertising within private conversations people were having with each other. Lessig soon distanced himself from the decision and Creative Commons later withdrew from the arrangement (see Lessig 2005b).
24. Peer-to-Peer (P2P) is a methodology for networking computers together that advocates the logic of intelligence being held at the ends of the network. Thus the network itself is 'dumb' having only the simplest technical logic associated with it, often only the communications protocols necessary to pass data from A to B without any knowledge of different types of content. The Internet's underlying technical protocols TCP/IP are designed as peer-to-peer and this explains the multitude of different transmission implementations (e.g. HTML/HTTP, FTP, SendMail, VPN etc) that can be run across the Internet without having to re-engineer the underlying network each time.
25. A more technical model such as the Open System Interconnection (OSI) layer model is a seven-layer model developed by the International Standards Organization to improve the design of software and hardware and improve reliability for communications across the different layers.
26. Known as *MPAA v. 2600* or *Universal City Studios, Inc., Paramount Pictures Corporation, Metro-Goldwyn-Mayer Studios Inc., Tristar Pictures, Inc., Columbia Pictures Industries, Inc., Time Warner Entertainment Company, L.P., Disney Enterprises Inc., Twentieth Century Fox Film Corporation, V. Eric Corley, Also Known As Emmanuel Goldstein, And 2600 Enterprises Inc.*

27. The name Emmanuel Goldstein was, of course, taken from George Orwell's futuristic dystopia *1984*.
28. In fact, strictly speaking, the CSS algorithm was originally not intended to stop or limit DVD piracy, but rather to assist the film industry to maximise revenue through staggered releases of films around the world.
29. 'The Supreme Court has explained that "all ideas having even the slightest redeeming social importance," including those concerning "the advancement of truth, science, morality, and arts" have the full protection of the First Amendment. Roth v. United States, 354 U.S. 476, 484 (1957) (quoting 1 Journals of the Continental Congress 108 (1774)). This protection is not reserved for purely expressive communication. The Supreme Court has recognized First Amendment protection for symbolic conduct, such as draft-card burning, that has both functional and expressive features. See United States v. O'Brien, 391 U.S. 367 (1968)' (*Junger v. Daley* 2000).
30. Formally known as the Declaration on Human Rights and the Rule of Law in the Information Society (which was adopted by the Committee of Ministers of the Council of Europe on 13 May 2005).
31. See www.creativecommons.org
32. Sometimes referred to as de-materialisation, that is the transfer from a physical expression or container to a representation within binary data on a computer system. This is usually stored as 0s and 1s on a magnetic storage device such as a computer hard drive but can also by optically stored as binary pits on an optical storage device such as a CD or DVD.
33. Most languages now are Object Oriented Programming languages (OOPs). This means that they have a formal structure that is in contrast to the older Procedure/Data division (and one that still tends to inform more media scholarship on the subject) and instead relies on an Object/Method division. In practice this means that the software developer attempts to create a model of the world within the computer that is directly constructed from real objects – such that a car modelled in an OOPs language would create a Car object, with the same properties and functions of its real-life equivalent. The idea is to make it easier to model the complexity and multiple relationships of the real-world objects and to provide a means of testing that is more intuitive. So in our car example, the engine in the computer modelled object could not run at a negative speed, nor could it consume a negative quantity of petrol.
34. Although free software and open source have both been criticised for being 'communist' neither is explicitly anti-capitalist or anti-

commercial in their approach to source code, perhaps reflecting their North American biases.

35. Also known as Technical Protection Measures (TPM).
36. Especially when copyright owners increasingly demand the right to monitor the use of their copyrighted works by individuals to check for infringement and illegal copying.
37. See www.trustedcomputinggroup.org/home
38. Throughout the book, for stylistic convenience the free/libre and open source (FLOSS) movements will sometimes be represented by the general term *open source*. However, it is important to remain conscious of the fact that there are important political and philosophical differences in the free software and open source movements (see Chapter 5).
39. Here I use the term post-modern in terms of Fredric Jameson's (1992) notion of 'post-modern capitalism'. By post-modern economy, I am drawing on the notion developed by Jameson that post-modern capitalism is a development of capitalism that has moved from market capitalism, through monopoly capitalism to its current form.
40. The TRIPS agreement was negotiated at the end of the Uruguay Round of the General Agreement on Tariffs and Trade (GATT) treaty in 1994. It introduced intellectual property law into the international economic system for the first time, and is a comprehensive international agreement on many aspects of intellectual property. TRIPS contains requirements that signatories of national laws must meet the minimum requirements for: (i) copyright, including the rights of performers, producers of sound recordings and broadcasting organisations; (ii) geographical indications, including appellations of origin; (iii) industrial designs, including integrated circuit layout-designs; (iv) patents, including monopolies for the developers of new plant varieties; (v) trademarks, trade dress; and (vi) undisclosed or confidential information. TRIPS also outlines enforcement procedures, remedies, and dispute resolution mechanisms between signatories.
41. The creative industries is an all-inclusive term for companies, individuals and groups involved in work that is connected to cultural production (i.e. film making, music, television or computer software). It is a very loose and ill-defined term and as such within this book the stress will be on the corporate members of the cultural industries, particularly the large multinational corporations.
42. Estimated to be worth £21 billion for 2003, amounting to 12 per cent of UK investment (National Statistics, 2006: 1).
43. This the corporations achieve through patent pools, which allow corporations to agree not to sue each other by sharing patents, or

collecting their own patent portfolio through patenting everything they can (a policy that Microsoft is currently implementing)

44. In particular the *Gowers Review of Intellectual Property* for HM Treasury conducted in 2005/06. See www.hm-treasury.gov.uk./media/6/E/pbr06_gowers_report_755.pdf
45. The rational actor held in economics is the idea that as individuals we seek to maximise our utility selfishly (public choice theory). This allows numerous models of human behaviour to be built and tested mathematically, but it remains an essentially crippled view of human motivation and action.
46. Not to mention the economic value of computer code which is now estimated in the billions of dollars annually. For example, Windows Vista, the latest generation of the Microsoft operating system is estimated to have cost $5 billion to develop (Lewis 2007).
47. See http://news.bbc.co.uk/1/hi/entertainment/4740668.stm
48. See for example the Halloween documents at www.catb.org/~esr/halloween/
49. Creative Commons (CC) is a non-profit organisation based in the US, which was founded by Lawrence Lessig, that writes and manages general-purpose copyright licences or use by 'creators' to enable them to share their creativity. The licences range from essentially declaring the work to the public domain (and in which case no CC licence is required anyway) to licences that fall just short of copyright (such as CC-Attribution, Non-Commercial, No Derivative).
50. No small claim when you consider that in 2003 only 13 per cent of all IT projects in the UK, and less than 1 per cent of IT development projects, were successfully completed on time, to specification and to cost (Post 2003: 1).
51. 'Lock-in' is when it would be either too costly, socially difficult due to staff reluctance to retrain, or technically complex to move to another software product. Some vendors rely on these problems to minimise the ability of customers to swap to competitors by 'closing' software, using non-open standards, creating idiosyncratic user interfaces or copyright protecting 'familiar' user interface items (e.g. the Trashcan in MacOS). See www.publications.parliament.uk/pa/cm200607/cmhansrd/cm071009/halltext/71009h0008.htm#column_46WH
52. The claims that open source software is cheaper to produce and maintain than proprietary alternatives have led the Conservative Party to recently voice their support for open source. See www.conservatives.com/tile.do?def=news.story.page&obj_id=135394
53. Fisher (2004) argues that a 'semiotic' democracy is one in which the consumers of culture are able to shape and reconstruct meaning in

the often corporate cultural artefacts that surround them in their everyday lives.

54. The 'black-box' programming method is an attempt to decrease complexity by hiding superfluous information from the programmer within discrete programming methods or functions. This has greatly increased the power of computer software through abstraction, but has also had the unintended consequence of hiding the detail of particular ways of delegating power to technology (see Berry and Moss 2006).
55. Other notable movements, such as environmental groups and traditional knowledge advocates, have made similar criticisms about the pharmaceutical industry, agribusiness, and the export of production in manufacturing (see Boyle 1996; Shiva and Bedi 2002; Bollier 2003).
56. Clearly, if the Digital Rights Management code is implemented in FLOSS software, then the encryption algorithms would be open to public view, and therefore easily circumvented. This is the reason that there has been no official DVD player released for the Linux operating system

2. The Information Society

1. As Bloom (1987: 182) comments, even scientists claim to be creative and 'nothing could be more contrary to the spirit of science than the opinion that the scientist fabricates rather than discovers his results'.
2. For example, the term 'democracy' has a radical ambiguity, which under-determines the fixed meaning of the term. This ambiguity gives the context its openness and functions to allow debate and politics to take place through a 'floating' or 'empty' signifier. Floating signifiers are usually over-determined (i.e. full of meaning) whereas empty signifiers are under-determined (that is, have little intrinsic meaning) (Laclau and Mouffe 2001: 171).
3. It is important to note that in some senses theorists are discussing 'information societies' rather than a single universal form. That is useful when considering that 'information society' itself is a contested term, and that even across the EU in the form of the Lisbon Agenda, different countries are implementing information society objectives in differing ways.
4. Additionally, there are claims that the emphasis on intellectual property as an important national resource also emerges from the OPEC oil shocks in 1973–74 and the worries about other areas of the economy that might be threatened. For example, the US

Senate Committee on Foreign Relations held an emergency session and wondered 'If information and its communication represent a strategic resource in international affairs, whose value may approach or exceed that of energy, will appropriate US Government policies be formed only after this is an energy-type crisis?' (quoted in Howkins 2001: 74). Two years later US President Ford set up a Task Force on National Information Policy which stated:

> Property concepts have been central to legal theory and social and economic activity in our society, but concepts of property were formulated to deal with tangibles, primarily land and chattels. When information, ways of dealing with information, or information products are treated as property, issues arise which differ from those resulting from the application of property theories to tangible matter. (quoted in Howkins 2001: 74)

Clearly, foreign policy in the US was adapting to reflect a dawning realisation of the economic and social value of information to both national security and economic growth. It was also reflecting a desire to develop a policy toward information that benefited American interests. This also reflects the growing interest within the discipline of International Relations of the political economy of the informational economy (May 2000). This is mirrored in information policy documents and directives from the UK government and the European Union (EU 2001; EU 2005c; EU 2005a).

5. Dell computers is a good example of a company that doesn't make anything different from its competitors; rather it has invested in highly efficient computer controlled flexible and customisable production and manufacture processes that make it far more efficient than any other computer company.
6. This is because the regulatory state required to supervise the private-sector implementation of public services in effect extends the state even as it appears to be 'outsourcing'.
7. See http://politics.guardian.co.uk/publicservices/story/0,,2012405,00.html
8. Informing both economic strategies at the level of corporations and the world market, local and global territories and the personal narratives and discursive formations of private actors whose subjectivities are recast within an economic discourse of informationisation and knowledge (Jessop 2004: 168).
9. Here I use the concept of inventory drawing on Bassett (2004) where the inventory 'allows for the systematic collection and ordering of objects, but it also guarantees that the list so collected, will itself be productive…' (2004: 353).

10. This has interesting links to consequentialist methodologies: since they are based on no harm done, no harm will statistically have been done in the future.
11. One of the most interesting issues raised in terms of the digital is the lack of scarcity that applies to most digital forms and once dematerialised (i.e. within a digital form) they are easily copied and reproduced. Broadly speaking there are four categories of digital data (i) temporally expiring forms: where the data is worth more close to the event to which it refers (e.g. stock market prices, currency trades); (ii) digital information that has been costly to produce and therefore requires a recouping of the production costs (e.g. Microsoft Windows would have been expensive to originally produce); (iii) the content of digital information contained within the global inventory (that is production costs have already been recouped and therefore their scarcity is maintained through intellectual property rights), which would include music, TV programmes, historical documents, library contents and suchlike. Here the digital reproduction cost is negligible and as the production costs have been recouped there are few further reproduction barriers; and (iv) information in the public domain that is freely available for use and reuse.
12. The arguments over branding and advertising and the inability of the customer to use and reappropriate these symbols of their everyday milieu raises interesting questions about how language can be owned and controlled.
13. See www.secondlife.com
14. See www.myspace.com
15. See www.facebook.com
16. For example, Stuart Brand, who helped run Ken Kesey's Acid Tests and who created the Whole Earth Catalogue, was hired as an event manager by Doug Engelbart (the head of the Stanford research institute who famously ran the Augment research project in the 1960s – an ambitious and influential project to create hypertext graphic and text systems which was funded by the US government) (Markoff 2005: 148–9).
17. MULTICS was a mainframe timesharing operating system created in 1965 and continually used up until 2000. Timesharing was a way of dividing a computer processor between various uses so that the computer time could be used simultaneously by different actors.
18. Today UNIX refers to a class of operating systems rather than to a specific implementation of the operating system. Often the name will be a *NIX postfix, such as Linux, which indicates it is a similar operating system or uses the modular building block of Unix. Many of the different 'flavours' of Unix are the result of debate over the way

in which the operating system should develop, for example between FreeBSD Unix and NetBSD (over the networking and security capability), and also between the openness of implantations (e.g. GNU/Linux) and proprietary (e.g. IBM AIX). For more information on the history of Unix, see Salus, P. H. (1994) *A Quarter Century of Unix*, London: Addison-Wesley; Ritchie, D. M. (1984) 'The Evolution of the Unix Time-Sharing System.' *A&T Bell Laboratories Technical Journal* 63, October: 1577–93.

19. Server software usually consists of large-scale software packages, operating systems and database systems. It was left to others to write particular packaged software solutions such as banking application software that would often draw upon the server/operating system software to function.
20. The Homebrew Computer Club brought together hobbyists, computer enthusiasts and people who were interested in 'having fun' with computers. The Homebrew Club had an informal rule that if you took away a copy of software, you bought two back the following meeting to give away. This was famously attacked by member, Bill Gates, in his 'Open Letter to Hobbyists' which accused members of 'stealing software' and 'thieving' which he argued stifled innovation and hobbled computer software companies (Weber 2004: 37). It also served to demonstrate the 'two cultures' that existed in computer programming, those who believed software should be shared freely as a collective endeavour and those who saw it as a product that was privately produced and therefore could be sold by the owner.
21. Steve Jobs later founded Apple Computer with Steve Wozniak.
22. Bill Gates and Paul Allen founded Microsoft.
23. Gary Kildall was the creator of the CP/M operating system and the GEM Desktop graphical user interface, and founder of Digital Research.
24. Interestingly, rather than write an operating system to order, Gates and Allen bought the rights to a locally produced operating system developed by Seattle Computer Products (SCP) for $100,000 and then re-engineered it into the MicroSoft Disk Operating System (MS-DOS). This gave them a massive lead time advantage over Digital Research, who would have had to develop the operating system from scratch, and contributed to the fact that IBM signed the deal with Microsoft.
25. There are many books documenting the history of Microsoft some of the best accounts are given in Levy, S. (1995) *Insanely Great: The Life and Times of Macintosh, the Computer That Changed Everything*, London: Penguin; Wallace, J. & Erickson, J. (1992) *Hard Drive: Bill Gates and the Making of the Microsoft Empire*, London: Wiley;

Manes, S. & Andrews, P. (1994) *Gates: How Microsoft's Mogul Reinvented an Industry – and Made Himself the Richest Man in America*, London: Simon & Schuster; Ichniah, D. & Knepper, S. L. (1991) *The Making of Microsoft*, New York: Prima.

26. Visicalc was a spreadsheet package that enabled sophisticated financial forecasting and with the success of this application, word processors, databases and communications software were able to enter the business office.
27. Outside technology the killer app hypothesis has taken a similar form, for example when the film *The Jazz Singer* is credited to being the film that established sound cinema in the 1920s.
28. Apple's own success is, of course, more complex than the mere addition of a new computer user interface, including technical obstacles, compatibility, manufacturing, marketing and third-party support.
29. As with any statistics caution is advised when reading these figures, as they are notoriously open to error, industry pressure and double-counting (for example many GNU/Linux users have bought a Windows or Mac operating system 'bundled' with their computer over which they install Linux). Nonetheless the rule of thumb statistics for desktop computers is usually taken to be 90 per cent Windows, 5 per cent Mac, 5 per cent GNU/Linux.
30. See May and Sell (2006: 87–97) for an excellent and detailed discussion of the Statute of Anne. Of course there are important precursors to the Statute of Anne and the reader is advised to consult May and Sell (2006) for more information regarding antecedents.
31. A contemporary example of this is the new Webcasting, Netcasting and Simulcasting provision of the Broadcasting Treaty (see www.wipo.int/edocs/mdocs/sccr/en/sccr_15/sccr_15_2.pdf). This would create new protection rights but it has been very controversial and the technicalities have not been finalised.
32. An example of the dangers of these technologies is given by Sony's use of monitoring code which backfired dramatically, see www.theregister.co.uk/2005/11/10/sony_drm_trojan
33. HD-DVD (since early 2008 taken off the consumer market) used AACS LA (Advanced Access Content System License Authority). Blu-ray uses a digital rights management called BD+ together with AACS and a 'rom-mark', which is meant to guard against mass piracy. Both of these have allegedly been cracked, see http://arstechnica.com/news.ars/post/20070213-8837.html
34. Some examples of which include the Free Software Foundation (FSF), Free Software Foundation Europe (FSFE), Foundation for a Free

Information Infrastructure (FFII), Electronic Frontier Foundations (EFF), Free Culture groups and the Open Rights Group (ORG).

35. The patent office in 1889 in the US rejected a patent on trees and forests as 'product of nature doctrine', objects discovered are not inventions, although extraction processes could be. In 1972, a bioengineer called Ananda Mohan Chakrabarty developed a bacterium to consume oil-slicks and filed patent. The US Patent Office contested the claim and in *Diamond v. Chakrabarty* (16 June 1980), the court held that whether the product was living or not was irrelevant and that it should be able to be patented – ingenuity should be rewarded (i.e. invention) but also genetic technology and living organisms (i.e. discovery) should also be allowed. This seemed to leave no limit to the patenting of any life, including presumably humans, and because DNA is a chemical and changing it produces a new composition of matter, any life could therefore be patented. Following this ruling Harvard University developed the OncoMouse, developed by inserting an Oncogene (a foreign human gene) into a normal mouse embryo. As living organisms reproduce this adds a new and worrying dimension to intellectual property law, which has generally been concerned with the manufacture of physical objects and the monopoly of their manufacture (see Boyle 1996 for an extended discussion of this).

3. The Concept of the Commons

1. Here I use the term 'things' to refer to both physical and immaterial (i.e. digital) objects.
2. Although here they are used as conceptual categories. In the manner of Marx's method, they are 'rational abstractions' that help us to better understand than particular examples of commons production could alone (Marx and Engels 1999: 124–51).
3. Of course, remembering that in the digital encoding of information everything becomes 'code' as a stream of 1s and 0s.
4. Avoiding the problematic transference of civil law categories to understand common law, for example.
5. R.W. Lee quoted in Rose (2003) talking about classical legal distinctions states that 'all this is very confused' (Rose 2003: 3). It also reflects the work of a number of different legal jurists such as Gaius, Marcianus, Ulpian, Paulus, Scaevola, Neratius, Celsus, Pomponius, and Justinian (Fenn 1925: 727).
6. Under Roman law, *dominium* signifies quiritarian ownership of a thing (i.e. the highest form of right to the property, e.g. 'title'). A quiritarian right was asserted by *Res Mancipi* (i.e. literally 'thing

taken by the hand'), through a ceremony in which five witnesses would watch the incantation of certain words which would then lodge ownership in the buyer (or also give up the right). *Dominium* defines the circumstances under which a possessor may use, alienate, lose or acquire ownership, and of course, the durations therein (Smith 1875: 421–3)

7. Another useful concept is that of *Fructus/Usufructus* (leasehold) which is that which is produced out of things (i.e. *res*) by their productive power; as in grass in a field or fruit on a tree. This category is slightly different to the others in that it is used to explore the possibilities of differentiation between possession and ownership and is therefore not directly applicable to the subject of FLOSS although is included in this footnote for completeness. *Usufruct* describes the legal right to utilise and derive profit from property that belongs to another person (which could be the state), as long as the property is not damaged. In many legal systems of property, buyers of property may only purchase the usufruct of the property. *Usufructs* are of two kinds: (1) *Perfect usufruct*, which is of things which the *usufructuary* can enjoy without altering their substance, though their substance may be diminished or deteriorated naturally by time or by the use to which they are applied; as a house, a piece of land, animals, furniture and other movable effects; and (2) *Imperfect or quasi usufruct*, which is of things which would be useless to the *usufructuary* if they were not consumed, such as money, grain and liquors. *Usura* originally meant a charge for the use of a fungible property (that is, any perishable or good that is 'used up' in its consumption) under Roman law (see Hyde 2006: 112–13 for a discussion of the history of usury).
8. Drawing on the Latin *Imperium* meaning jurisdiction but also playing with the concept 'Empire' introduced by Hardt & Negri (2000).
9. 'Standing reserve' is drawn from Heidegger's use of the term to designate when 'the energy concealed in nature is unlocked, what is unlocked is transformed, what is transformed is stored up, what is stored up is, in turn, distributed, and what is distributed is switched about ever anew' (Dreyfus 2005).
10. More accurately in Roman law *res nullis humani juris* (without human ownership) to distinguish from *res nullis divini juris*.
11. Although, see *res publicae* for a discussion of the limited nature of the public domain as a *res nullis*.
12. Here the original Roman classification of *res communes* as unbounded makes it impossible to classify knowledge objects as *res communes*. Today, the situation is often reversed and our commons

are bounded by private property (*res privatae*) and state-controlled land (*res publicae*).

13. This has resulted in a certain amount of ambiguity in its definition and use and therefore confusion about which form of 'public things' we are discussing, whether public property or public government.
14. The concept of the 'state', as it emerged in the Renaissance, was derived from the Latin term *status*, a neutral expression meaning the condition or way of existence of a thing. It was evident in the Roman literature as *status rei publicae* where it indicated the legal structure of the community, but today it is used to indicate both: (1) the actual exercise of power by government; and (2) the people or territory over which power is exercised.
15. Pericles' Funeral Oration (after 490 BCE) from Thucydides, *The Peloponnesian War.*
16. Naturally, Plato's *Republic* comes to mind at this point, but his book is actually named *Politeia* (i.e. constitution). That we today call it the *Republic* is due to the extent to which our culture sees through the lens of a Roman-influenced culture. A republic, *re publica*, concerns itself with a strict distinction between public things (i.e. *res publicae*) and private things (*res privatae*). Plato does not, concentrating instead upon knowledge and education. The Romans posit a public sphere of the state against the private sphere of individual possessions by establishing laws of ownership and entitlement. This has led to misreadings of Plato's work from liberals such as Popper who have seen it as a totalitarian work (Coleman 2000: 81–2).
17. Cicero *De re publica*, a political treatise written in the 1st century BCE.
18. *Discourses on Livy* (1531).
19. *Leviathan* (1651).
20. *Non est potestas super terram quae comparetur ei* (i.e. There is no power on earth that compares with him): the words from the Bible (Job 41:33, Vulgate version) with which Hobbes inscribes the book, *Leviathan*.
21. Published in 1689.
22. An important difference between Hobbes and Locke, though, is that Locke believed that property existed before the institution of a state (i.e. through natural law), whereas for Hobbes the state must by necessity come first. Hobbes believed that the state of nature was a condition in which no property rights exist (Lopata 1973: 207).
23. Locke, though, made an important but often forgotten restriction on this initial acquisition of property, called the 'Lockean proviso'. This states that individuals have the right of acquisition only if 'enough and as good [is] left in common for others' (Locke 2002: 320).

24. First published in 1848.
25. In fact, Marquand equates the commonwealth with the public domain, writing that 'the public domain is also the domain of citizenship' (Marquand 2004: 77). His division between the state (i.e. public sphere), the market (i.e. economics) and the public domain is rather less clear although similar to Habermas' tripartite distinction between state, public sphere and private sphere (Habermas 1992).
26. The concept of the multitude against that of the 'people' raises interesting ideas related to the possibility of 'common ownership' over and above 'public ownership' (Hardt & Negri 2000, 2004; Virno 2004). For a detailed examination of the difference between 'multitude' and the 'people', see Virno (2004: 41–5).
27. Hardt & Negri draw a clear distinction between democracy and republicanism (Hardt & Negri 2004: 242).
28. Hardt & Negri have been developing a theory of ownership that transcends the state, and contrasts the *public* good with that of the *common* good. Although there are vague outlines of a politics informed through the act of production, or perhaps as a model for such politics, there is little or no explanation of *how* the multitude might rule itself or how this 'common interest' might be manifested (Hardt & Negri 2004: 339–40). This suggests an interesting and suggestive political conception of the 'common' although unfortunately rather vague on details. A political project connected to a notion of the common might be better able to articulate this through the concept of *res communes*.
29. Scotland entered a political union with England in 1707, but retained its own independent systems of law, banking and trade, and institutions of local government, education and established religion. Consequently, common land has been managed very differently to the rest of the UK, and very little traditional common land has survived (Wightman et al. 2004).
30. Common land in England is not state owned. In fact, commons always have a landowner, whether privately held or a public body and the public does not have an automatic right to walk on it (OSS 2005).
31. In the UK many of the public common lands were converted into enclosures by a number of Enclosure Acts; in total 21 per cent of England was enclosed by Act of Parliament from 1750 when private Acts were introduced in each particular case of enclosure, until the later Public General Act 1801 which attempted to simplify the process. The General Enclosure Acts of 1836 and 1840 allowed enclosure of common land to take place without reference to Parliament, and the final Enclosure Act of 1845 made amendments to protect smallholders and the interests of the public.

32. The Romans had no such *dominium* over the sea which was completely non-owned, in other words there was no extension of the state outwards from the shoreline (Fenn 1925: 717).
33. An interesting discussion of the political nature of the classification of 'common heritage of mankind' is given in Payne (1978) where he shows the political economy of mining manganese nodules in deep-sea areas (i.e. outside of territorial boundaries) is fraught with arguments about the status of things, notably between *res nullis* and *res communes*. The definitional construction has huge economic consequences for the tri-continental economies (Payne 1978).
34. There are, however, only ten parties and five signatories to the Moon Treaty and 97 parties and 27 signatories to the Outer Space Treaty (Monterey Institute 2005).
35. *Ius gentium* is the body of common law that was applied to foreigners in their dealings with Roman citizens.
36. More accurately in Roman law *res nullis divini juris*.
37. The expansion of the concept of global civil society beyond the nation state carries many normative and political consequences that are outside the scope of this section. (See Hardt & Negri 2000: 393–413; 2004.)
38. Marx wrote in the 'Fragment on Machines' in the *Grundrisse* about how labour moves away from being the centre of production (Virno 2004: 38). Instead production becomes focused on *mass intellectuality*, such that the production of affect, knowledge and information become the driving force of production.
39. Élan vital (i.e. 'creative impulse' or 'living energy') is an immaterial force that provides the vital impulse that continuously shapes all life (Bergson 1998).
40. These are meant as explanatory examples; it is the political dimension to *res divini juris* that is of crucial importance here.
41. In other words, *res divini juris* understood as an empty signifier that is not yet fully conceptualised (Laclau and Mouffe 2001).
42. This is similar to the concept of two-dimensional society that Marcuse (1991) outlines in *One-Dimensional Man* where he describes the dangers of a society whose 'novel feature is the flattening out of the antagonism between culture and social reality through the obliteration of the oppositional, alien and transcendental elements in the higher culture by virtue of which it constituted *another dimension* of reality' (1991: 57).

4. From Free Software to Open Source?

1. These are, of course, released as FLOSS software, examples include: (1) wikis – websites that can be edited online by users in browsers

and are instantly updated and allow a community of interest to form around a textual artefact; (2) content-management systems – which facilitate complex workflows and websites; (3) mailing lists – which are a focal point for discussion and debate; and (4) versioning systems – that enable the geographically dispersed and networked structured groups to work simultaneously on multiple versions of software, documentation and support without the complexity getting out of control for releasing and controlling the software project.

2. The Waterfall model of software development is designed to break the software development process into a linear and predictable path for project management. In its typical incarnation it is unidirectional towards a final product, travelling through the stages of Specification, Design, Coding, Testing and Delivery. Later versions added iterative loops to enable later stages to feed back to earlier ones, but generally speaking the process remains geared to building on previous stages and can be difficult to stop or change direction once in motion. This has contributed to the escalating costs associated with software development as the model encourages a single-minded determination to reach the final deliverable stages, and also can give the illusion to managers of progress, where in reality there might be none.
3. See Benkler (2006) for a detailed examination of the spread of the ideas from FLOSS into cultural production.
4. For a detailed explanation of the algorithm see www.iprcom.com/papers/pagerank
5. Although I refer to two competing logics, this is an analytical distinction as the reality of the software industry is one of hybrids, flows and fluid movements between each software production paradigm. An example of this is the fact that the earliest FLOSS software, GNU, was developed on proprietary platforms. Indeed, the Microsoft Windows operating system network code is actually software taken from the BSD Unix operating system under the terms of the open source BSD-licence. Nonetheless, discursively and reinforced through law, these logics are manifested in particular discursive and social practices and legal structures.
6. A hallmark of the early techno-utopians was their rejection of government and their call for libertarian anarchist approaches to organising society. That is, highly individualistic, anti-collective, anti-state and pro-capitalist (although with the important caveat that they were anti-monopolies) (see Barlow 1996 for the most widely known in technical circles).
7. See particularly the work of Richard M. Stallman and Eric S. Raymond who present themselves both as practitioners (i.e. by writing code), activists (i.e. in terms of political posturing and actions)

and theoreticians (i.e. by writing essays and discussion papers). This aspect is explored in greater depth in the next chapter.

8. There is even a magazine for hackers called *Make*. See www.makezine.com
9. In the words of many posts to forums and blogs on the Internet, IANAL (in the online parlance of 'I am not a Lawyer') and so the reader is advised to consult texts such as Cornish & Llewelyn (2003) for more detailed examination and explanation of copyright law.
10. The GPL has been found constitutional by a German court in 2004 (Shankland 2004) but is under considerable challenge in the US with the case against IBM currently being argued. To date Creative Commons licences have been found to be legal in Holland and Spain (Marson 2006; Garlick 2006) but no US or UK case has been brought.
11. The copyright rules covering literary works were extended to computer software in the Copyright (Computer Programs) Regulations 1992. For a comprehensive discussion of the development of copyright in relation to software see May and Sell (2006).
12. The clearest example of this is demonstrated with HTML links: these are written as text residing on a web-page, but are also URLs which are processed by the computer and act as a machine to relocate the browser to a new web-page.
13. Authorship and intellectual property rely on the idea that software is a completed, finished piece of work which can be claimed as the property of the publisher (Herman, Coombe & Kaye 2006: 192).
14. http://developer.apple.com
15. http://msdn2.microsoft.com/en-us/default.aspx
16. '**Non-Disclosure.** During my employment and at all times thereafter, I will not disclose to anyone outside MICROSOFT nor use for any purpose other than my work for MICROSOFT: a) any MICROSOFT confidential or proprietary information or trade secrets; or b) any information MICROSOFT has received from others that it is obligated to treat as confidential or proprietary. I will not disclose confidential or proprietary information or trade secrets to other MICROSOFT employees except on a "need-to-know" basis, and I will not disclose third party confidential or proprietary information except as permitted by any applicable agreement between MICROSOFT and the third party. "Confidential or proprietary information or trade secrets" means all data and information in whatever form, tangible or intangible, that is not generally known to the public and that relates to the business, technology, practices, products, marketing, sales, services, finances, or legal affairs of MICROSOFT or any third party doing business with or providing information

to MICROSOFT, including without limitation: information about actual or prospective customers, suppliers and business partners; business, sales, marketing, technical, financial and legal plans, proposals and projections; concepts, techniques, processes, methods, systems, designs, programs, code, formulas, research, experimental work and work in progress. If I have any questions as to what comprises such confidential or proprietary information or trade secrets, or to whom if anyone it may be disclosed, I will consult my manager.

Assignment of Inventions. I will make prompt and full disclosure to MICROSOFT, will hold in trust for the sole benefit of MICROSOFT, and will assign exclusively to MICROSOFT all my right, title, and interest in and to any and all inventions, discoveries, designs, developments, improvements, copyrightable material, and trade secrets (collectively herein "Inventions") that I solely or jointly may conceive, develop, author, reduce to practice or otherwise produce during my employment with MICROSOFT. I waive and quit claim to MICROSOFT any and all claims of any nature whatsoever that I now or hereafter may have for infringement of any patent application, patent, or other intellectual property right relating to any Inventions so assigned to MICROSOFT.' See www.cs.washington.edu/commercialization/ott/MSagreement.pdf

17. Microsoft, amongst others, is now preparing to use intellectual property to protect its existing market share and fight off competitors: 'Last year, Bill Gates told financial analysts that the firm would increase its patent filings to around 3,000 in 2005, up from 2,000 the year before and the low hundreds in the 1990s. The company currently holds over 6,000 patents, and has around 10,000 applications pending' (*Economist* 2005). These patents, of course, do not just control other corporations; they also prevent individuals from using the patent processes and methods in their own work, or taking it with them when they leave.
18. In the 1980s in the UK the perceived lack of trained engineers and technicians prompted a number of enquiries including *Software: A Vital Key to UK Competitiveness* (1986) conducted by the Advisory Council for Applied Research and Development, and *The UK Software Industry* (1988) by Peter C. Grindley of the London Business School (Campbell-Kelly 1995: 75).
19. Modelling languages such as Z and UML (Unified Modelling Language) attempt to specify the system characteristics in a non-algorithmic way which is mathematically rigorous. This, in theory, allows the system to be tested functionally before moving to implementation in a specific programming language. Unfortunately they

have not been very successful and have proved to be more often used in academic environments than in actual practice.

20. Agile programming is a set of techniques to improve both the quality and efficiency of software writing with practices such as continual peer-review, modular testing and constant builds of the software.
21. Object-oriented techniques are predicated on the logic of modelling the 'real' world through the use of software 'objects' which are classified into types (known as Classes). These Classes could be Car, Cat, Human, Dog etc, which are defined in the abstract and then instantiated into particular manifestations, so the Class Human could be used to form the specific objects Alice, Bob and Clive. The idea is to more closely model the real world by storing data and functions in the object itself, rather than applying the analytical distinction between functions and data that exists in procedural programming languages. In practice this means that should Alice and Bob wish to transfer money between each other's bank accounts, then the software objects Alice and Bob would replicate the process in software by transferring the money between them internally. In theory it should therefore be easier to follow and debug software errors by checking the contents of the objects against the state in which they should reflect the real world.
22. Many of the project management tools used in the software industry are adapted from the construction industry that has itself been remarkably successful in planning, managing and costing project plans. Unfortunately the greater complexity and the reliance on key programming individuals have meant that software planning is still a long way behind construction in terms of reliability and predictability.
23. For a full history of free software and Stallman's life see Williams (2002); for an overview of the free software movement and open source see Moody (2002); for a full list of the core texts central to the free software philosophy see Stallman (2002a).
24. Originally he attended MIT as a Masters student but later dropped out to become an employee of the lab and a full-time programmer.
25. A 'driver' is a small software program that translates the internal computer-generated text into the correct output format for a particular printer model. Usually the printer manufacturer wrote these drivers and as competition increased between manufacturers they increasingly sought to protect their perceived intellectual property, using methods to render documents in legible and efficient ways by supplying only a binary version of the driver.
26. Stallman is said to have said that 'the prospect of charging money for software was a crime against humanity' (Williams 2002: 85).

27. http://groups.google.com/group/net.unix-wizards/msg/4dadd63a976019d7?dmode=source&output=gplain
28. Stallman admits to being inspired by the notices in software that read 'verbatim copying permitted' on source code which essentially allowed software-sharing in spite of copyright law (Williams 2002: 123).
29. Emacs is a text editor that enables the programmer to edit plain text files on a computer in a relatively simple editing environment. It was widely adopted due to the fact that is was very extensible, and this combined with the source code being widely available meant that other programmers could easily write programs that improved its functionality.
30. The 'viral' description of the copyleft clause is something that Stallman does not approve of, stating: 'To compare something to a virus is very harsh… a spider plant is a more accurate comparison; it goes to another place if you actively take a cutting' (Williams 2002: 23, footnote 1).
31. GPL is a *quid pro quo* licence developed primarily to ensure the survival of a form of commons by the use of a clause that restricts the user of the code to agreeing to re-share any of their source code additions in a project under the same licence. This came to be known as 'copyleft' (all rights reversed) and acts as a defender of the code from 'enclosure' or closed proprietary usage.
32. GNU/Linux distribution companies are often referred to as 'distros'.
33. The foundation does not accept applications, nor does it accept suggestions or recommendations. The first an applicant hears that they have been nominated is when they are called with the congratulations from the committee.
34. Originally named Freax (a word created by merging freak, free, and x), Ari Lemmke, Torvalds' colleague at the Helsinki University of Technology, disliked the name Freax, preferring the name Linux, so renamed the file and placed it on the FTP server. The name Linux has remained till the present day although controversy still surrounds whether Linux corresponds to just the kernel (and hence the operating system is more correctly referred to as GNU/Linux) or to the complete operating system.
35. Berkeley Software Distribution (BSD) is a Unix operating system that is descended directly from the AT&T implementation which started in the 1970s. It has forked into a number of differing implementations due to various project limitations and differing ideas about project direction, for example 386BSD, FreeBSD, NetBSD and OpenBSD. The key difference between GNU Unix (i.e. Linux)

and the BSD family is that the BSD Unixes are licensed under the BSD licence, which has none of the copyleft clauses of the GPL. In fact, the only restriction on use is that attribution must be made when BSD software is included, whether in FLOSS or proprietary software. It has sometimes been referred to *copycentre* licensing to mark it apart from copyright and copyleft, because you can take it down to a copy centre and make as many copies as you want.

36. To move to a new licence would require that every contributor, who in theory could claim a portion of copyright on their addition, would need to authorise the move. In any case, Torvalds seems completely unwilling to consider such a move due to ideological differences with the FSF (and in particular Stallman). He also holds a large portion of the kernel copyrights and it would be very difficult to bypass him (or 'fork' the GPL 2.0 licensed code). There is no doubt, though, that this is likely to happen as the war of words escalates and a decision needs to be taken on the threat of SaaS to Linux as an operating system plays out in practice.
37. This is not an idle threat. SCO, a US corporation that claims ownership of the early Unix operating system, has sued IBM on the basis of copyright infringement in terms of (1) breaking an agreement not to develop Unix versions derived from the intellectual property of the original Unix; and (2) contributing copyright source code into the Linux operating system kernel. IBM is currently contesting these claims in court and it has so far not been clear that SCO has a case against IBM although this doesn't preclude this kind of lawsuit happening in future over the contents of the Linux source code. As a precautionary measure, Torvalds has attempted to segregate the offending code and ensure that 'clean-room' code is written to replace it.
38. Yggdrasil is the name of the 'World Tree' of Norse mythology.
39. Most distribution companies ('distros') using the GPL have found that it forces them to make available a 'free' version which is downloadable from their websites. As commercial pressures have intensified the location of the 'free' version has become harder to find, and indeed, the commercial version has taken to 'linking' with proprietary software which does not require this distribution limitation and hence can have 'added extras' that can be charged for.
40. The Source Code Tree (e.g. managed by CVS – Concurrent Versioning System, Bitkeeper and others) is so called because it documents and stores the forks and developmental paths of the software source code. This means that over the life of a software project (either as a project as a whole or the individual digital object) the source code can be recovered at any point in its development. This allows the ability

of rolling back time and compiling early versions of the software or tracing the development of the project (e.g. to see who did what, what has been added or removed and so on). The openness of the free software and open source storage of source code and binary files will no doubt be a valuable resource for future academic researchers into software or the history of computers.

41. This is not to imply that the free software movement was in any way anti-commercial. Rather the concerns were about the importance of keeping free software free – that is, that the source code would continue to be available.
42. Torvalds himself left Transmeta in June 2003 to devote himself full-time to the increasingly complex Linux kernel.
43. The 'open source' label itself came out of a strategy session held on 3 February 1998 in Palo Alto, California. The people present included Todd Anderson, Chris Peterson (of the Foresight Institute), John Hall and Larry Augustin (both of Linux International), Sam Ockman (of the Silicon Valley Linux User's Group), and Eric Raymond.
44. The full quote is: 'Orwell's farmhouse is full of open source pigs, which are now almost indistinguishable from the proprietary humans they recently overthrew' (Metcalfe 2004).
45. Raymond is a controversial commentator, describing himself as a market anarchist (Poynder 2006: 28) and 'observer-participant anthropologist in the Internet hacker culture' (Raymond 2006). He has variously advocated gun rights, no government, imperialistic projects to subdue Muslim countries and 'Ethics Out of the Barrel of a Gun' (Raymond 2006). He has also been extremely successful in making open source a widely used term both inside and outside of companies and, it could be argued, has helped to strengthen the links between open source and business and the resultant commercial successes. Nonetheless his interventions invariably divide the FLOSS movement.
46. Raymond later explained that he declined to make money off the share-price jump and did not sell them although he still continues to hold the shares.
47. Shared source was an attempt to offer key business clients restricted access to the Windows source code. Although designed as a PR move to sidestep FLOSS development, it actually served to highlight the importance of viewing the source code in a computing and software-development environment.
48. The DVDs are encoded to play within a certain 'region'. This encoding system is proprietary knowledge and the software to play it (i.e. the keys that unlock the video image) is protected under the terms of the Digital Millennium Copyright Act 2000 (in the US) as

a technical protection measure, or digital rights management (DRM) technology. By creating an open source version of the player, the hackers are effectively publicising the encoding system and doing so without the permission of the owner.

49. All code examples given in this section are taken from CodeCodex which is an online Wiki that has all code within it licensed under a free licence. CodeCodex is explicitly intended to be used to share code, particularly algorithms and functions that are commonly used in the programming community. See www.codecodex.com
50. See, for example, the 'routing element' patented successfully by Thomson Components Mostek Corporation (European Patent application number: EP19850402567 19851220); or 'Data sort method, data sort apparatus, and data sort program' United States Patent 7103596.
51. Not without controversy though, as Flannery, Teukolsky and Vetterling state 'we *will* draw the line, however at the inefficient N2 algorithm *bubble sort*. If you know what a bubble sort is, wipe it from your mind; if you don't, make a point of never finding out!' (quoted in Astrachan 2003: 2, original emphasis) while Kunth argues that the 'bubble sort has nothing to recommend it' (quoted in Astrachan 2003: 1).
52. Copyright would prevent the reproduction of this code example, and also prevent derivative works but not newly created ones, even if in similar pseudo-code. Patents, however, would prevent any attempt to implement this code in any implementation.
53. Java is a programming language which was developed by Sun Microsystems and is owned and controlled by the corporation. The patents, trademarks and copyrights of crucial parts of the Java system are asserted by Sun. Recently Sun has announced that it intends to re-license the Java language under the GNU GPL (in fact it intends to dual-license it for commercial works) (LaMonica 2006).
54. Perl was written by Larry Wall and dual-licensed under both the Artistic Licence (also written by Larry Wall) and the GNU GPL.
55. In programming terms this is often considered a 'sandbox', that is a walled area where you can play but not take your toys outside.
56. Cases involving legal or criminal action regarding source code have increased as the growth in software and its importance to profitability and planning have become more important. The reputed closeness of code to 'ideas' is also important when it is understood that code can contain the business processes that make a firm competitive – this explains the rise in interest with regard to so-called *business method* patents. The US case of *State Street Bank & Trust Company*

v. Signature Financial Group, Inc., 149 F.3d 1368 (Fed. Cir. 1998) was an important legal case that found that methods of doing business could be protected by a patent providing it met the criteria of producing a useful, concrete and tangible result. In this case it was a financial business that had a number of mutual funds ('spokes') that pooled their money together in a central location ('hub') to invest. This follows the principle that 'anything under the sun made by man is patentable', made in the Chakrabarty case, *Diamond v. Chakrabarty*, 447 U.S. 303 (1980), which found that genetically modified micro-organisms could be patented.

57. A more recent case was the unauthorised release of the Microsoft Windows 2000 and Windows NT 4.0 source code in 2004 (Lettice 2004). Although this was later traced to a partner company of Microsoft, rather than hackers or other actors, the release turned out embarrassing for an analysis revealed the extent to which the source code is patched, bugged, plays to special interests (i.e. Microsoft applications) and also reflects a closed source mentality of quick fixes and poor code management (Selznak 2004).
58. The ethics of the self reflected in the discourse of the free software movement remind one of the moral law of the Old Testament (i.e. a Mosaic Law that applies to all) rather than the open source ethics demonstrated by Luther and Calvin where the moral law is atomised (that is let each person decide on what they must do where the accent is on the self deciding). This is discussed in relation to the theological position of the Protestant versus the Catholic Church on Usury (i.e. interest) in Hyde (2006: 111–42).
59. The Free Software Foundation has raised enforcement proceedings against a number of individuals and companies where it has felt that the GPL has been ignored or the terms broken. This has, so far, been an extremely successful strategy as in the final instance, should the case come to court, it is a straightforward case of copyright infringement which the defendant would have difficulty fighting (i.e. they would not have the right to copy code). This is the basis of Moglen's claim (1999; 2003) that the GPL is not a contract which depends on the legal jurisdiction under which it is enforced.
60. One interesting question unfortunately beyond the scope of the book is the way in which FLOSS projects and particularly the licensing schemes are mediated through technical processes (such as search engines, source code repositories and library software (such as ccmixter.cc)). The plethora of actors required to keep the FLOSS community afloat, particularly the non-human ones, raises questions about the technological a priori required for commons-based projects.

61. The GNU GPL is the licence that particularly incenses Microsoft as it prevents them from using any of the code in their own projects due to the 'viral' nature of the licence. To such an extent that they eschew any GPL'd software within their organisation. Microsoft have in the past happily used BSD-licensed software such as TCP/IP stacks, as discussed above.
62. Examples include the Apple Public Source License, Eclipse Public License, IBM Public License, MIT License, Mozilla Public License 1.0 (MPL), NASA Open Source Agreement 1.3, Nokia Open Source License, RealNetworks Public Source License V1.0, Sun Public License, W3C License, Zope Public License. Microsoft in its own idiosyncratic way has also released a Microsoft Shared Source licence that allows only those who have signed a prior non-disclosure agreement to view the code. Perens (and others) described it as a 'look but don't touch – and we control everything' licence (see Perens 2001).
63. There is currently a great deal of debate about what it is to be free software or open source software and is demonstrated in particular within the arguments over the production of the GNU GPL version 3.0. See http://gplv3.fsf.org/gpl-draft-2007-03-28.html
64. The UK indymedia movement that is generally committed to a more open and radical democratic project, has embraced FLOSS software in all aspects of its media activities. From production, dissemination, and even in the commitment to open-access models (which draw from FLOSS adherence to open formats for data) FLOSS has had a great impact. Although the cheapness of FLOSS may also have been part of the motivating strategy, Indymedia has also taken up the label open source as a term identified as a more democratic language for talking about political action. See www.indymedia.org.uk/en/2003/12/283113.html
65. In 2005 Brazilian president Luiz Inácio Lula da Silva ordered that government ministries and state-run companies should run FLOSS software and has even rolled out an 'information society' campaign to improve media literacy of young Brazilians. See www.nytimes.com/2005/03/29/technology/29computer.html?ex=1269752400&en=9e12f51a80800820&ei=5088&partner=rssnyt
66. Ubuntu has become the darling of the FLOSS movement as a distributor that works from the global South, actually based in South Africa, and distributes very complete and easy to install Linux systems. On 22 February 2007, the South African government instituted a policy and strategy for FLOSS in government including both use and future licensing. See www.freesoftwaremagazine.com/blogs/the_government_of_south_africa_has_joined_the_movement

67. See www.creativecommons.org
68. See www.freeculture.cc
69. See defectivebydesign.org
70. Many of the members of the no2id campaign are FLOSS developers, some from the previous Foundation for a Free Information Infrastructure (FFII) that successfully fought the EU patent legislation. See www.no2id.net
71. Originally presented as part of the Boston Consulting Group Hacker Survey available at www.ostg.com/bcg/BCGHACKERSURVEY-0.73.pdf
72. Here, though, I would want to stress that I am not arguing for a libertarian model of support for FLOSS and related *res communes* projects. Rather, we need to think carefully about the changes required to the state in order for the duty of care towards commons-based projects to be guaranteed, and therefore not turned into bureaucratic empires or privatised into the hands of corporations. Nonetheless I am intrigued by the possibilities of a reinvigorated civil society drawing on the model of FLOSS and its copyleft model.
73. The list of open source and free culture websites promoting commons-based alternatives is staggering, with a simple search by Google returning 31 million sites containing the phrases. Most recently the Conservative party in the UK have laid claim to the open source idea, declaring they would like to see an 'open source politics' (Osborne 2007) and the 2007 Reith Lectures on the BBC by Jeremy Sachs calling for open-source-like methods to be applied to solving world poverty, political apathy and environmental crisis (Sachs 2007).

5. The Contestation of Code

1. This chapter uses the term 'order of discourse' to represent a specific area of discourse within the field of discursivity, following Fairclough (1992: 12) and Phillips and Jørgensen (2002: 27). Within the chapter, the order of discourse is the use, production and ethics surrounding computer code.
2. See http://armedndangerous.blogspot.com/2002_09_15_armedndangerous_archive.html#81815163
3. The texts were selected by reference to the importance that the respective movements had allocated them in terms of citation, refutation and also in relation to their prominence on Google searches (which indicates the counting of links to the documents and therefore, to a certain degree, the popularity with a readership). The searches were "Free software", "Open source", "Free libre open source", "Free software philosophy", "Open source philosophy", "Free libre

open source philosophy", "Free software licences", "Open source licences", and "Free libre open source licences". A snowball process was also used where an original document such as Stallman (1992) would link or be associated with other documents.

4. The texts in this chapter are analysed using the Text Analysis Markup System (TAMS), an open source discourse analysis software package (Weinstein 2003).
5. Both groups share a strong conception of linear progress and modernity (Raymond 2001a; Stallman 1993). However, the extent to which communicative concerns can override the technical code (Feenberg 1995: 87) is examined further in the section 'Comparative Discourse Analysis'.
6. Stallman says that he drew his political inspiration from the fact that 'in my first year of Harvard, in a Chinese History class, I read the story of the first revolt against the Chin Dynasty … the story is not reliable history, but it was very moving' (Williams 2002: 57, footnote 8).
7. See www.gnu.org and www.stallman.org for examples of this hypertext approach to interdiscursivity.
8. Emile Durkheim, a French sociologist, used 'anomie' to describe a condition where norms (i.e. expectations of behaviour) are confused, unclear or not present to a society's members.
9. Here one is reminded of an anecdote 'reported by Seneca from imperial Rome' regarding the dangers of seeing a commonness or publicness between members. 'At that time a proposition was laid before the senate to have slaves dress uniformly in public so that they could immediately be distinguished from free citizens. The proposition was turned down as far too dangerous, since the slaves would now be able to recognise each other and become aware of their potential power' (Arendt 1989: 218).
10. The striking similarity of Stallman's position on software sharing and the theological question of usury (as discussed by the scholastics) is very interesting in terms of a shared exchange within a brotherhood as opposed to a secular relationship with the outside (see Hyde 2006).
11. Starting from 0 is a standard computer programming assumption in a loop or repeating series of computer statements.
12. The term bazaar is important in software debates because the concept was used by Raymond in opposition to what he called 'Cathedral Building' (and which he associated with top-down control of the process of software writing and particularly with the centralised approach of Richard Stallman). It also has connections with a market economy of buying and selling (or perhaps a pre-capitalist economy)

that allows the interchange and interaction between individuals that is not mediated by the state (for a self-proclaimed libertarian anarchist like Raymond this would be a particularly attractive model).

13. For an examination of the extent to which persuasion plays a role in the building of technical systems see Bijker, Hughes and Pinch (2001) and MacKenzie and Wajcman (1999).
14. An example of this is the Benevolent Dictator for Life (BDFL) on the Python project (Python 2003).
15. Ayn Rand was a novelist-philosopher who founded a movement called Objectivism based on the novels *The Fountainhead*, *Atlas Shrugged* and *Anthem*. Broadly speaking she was a advocate of laissez-faire capitalism and the primacy of the individual (in fact much of her work is similar in argument to that of Friedrich August von Hayek). Alan Greenspan, the former chairman of the US Federal Reserve, was a follower of her work, as is Jimmy Wales, the co-founder of Wikipedia. See www.aynrand.org
16. The reliance on a modernist technocratic progress is, of course, interesting to contrast with Information Society theory where neither Stallman nor Raymond think explicitly about cultural production as a mode of production. Rather they still envision technology as machines to do things – usually within the real world – and less about the production of information, communication and knowledge. Web 2.0, of course, has explicitly identified the productive potential of using free culture (licensed through free licences) to produce content that can be wrapped in advertising, subscription and services. This point incensed many free culture advocates enough for them to suggest a GPL 3.0 which would curtail the exploitation of free culture in this way (most notably by forcing services that use free software to publish their source code – although this clause has now been removed from the latest draft). See http://gplv3.fsf.org/gpl-draft-2007-03-28.html
17. Interestingly, this corresponds more or less exactly to Jenkins' (2006) model of convergence – and which he describes as a new kind of cultural studies (often drawing an analogy with Web 2.0) – sooner or later, no doubt, to be termed 'cultural studies 2.0'.
18. 'Raymond represents Second Amendment libertarianism, and gun rights. Specifically, Raymond is a self-styled market anarchist who believes that citizens have the right to carry guns in order to protect themselves from the government' (Poynder 2006b: 5). He has also advocated giving children guns from the age of five to teach individual responsibility (Raymond 2002), and suggested countering Islamic terrorism by ensuring all American air passengers carry guns (Raymond 2002b).

19. In fact, Raymond has conceded that it was written directly to appeal to business and technical people who would have been alienated by Richard Stallman's ethical position on software sharing (see Poynder 2006b).
20. It also resonates with the notion of Popper's Open Society, and George Soros's Open Institute.
21. Copyleft provides certain freedoms to a user, stating that when redistributing an object of software, you cannot add restrictions to deny other people the ability to copy, use and modify it (Stallman 2003b).
22. See for example Creative Commons (2003), The Libre Society Manifesto (Berry & Moss 2003), LOCA records (Atton 2003; LOCA-Records 2003) and Stallman (2003a).
23. The Social Shaping of Technology and Social Construction of Technology literature can be a useful corrective to Technologists' own idealised justifications of how technology is developed (Bijker et al. 2001; MacKenzie and Wajcman 1999).
24. Although that is starting to happen with the rise in Free Culture and Creative Commons groups that have been inspired by the free software movement. See http://freeculture.org/ and http://freeculturefoundation.org/

6. The Poetics of Code

1. In terms of the 'sociological imagination' described by Mills (2000).
2. Hardt & Negri (2004) argue that one approach to understanding the democracy of the multitude is as an 'open source society'; this they explain is a society whose source code is revealed so that everyone can work collaboratively to 'create new, better social programs'. This is made even more interesting when placed alongside the calls for applying open source ideas to economics and politics by Sachs (2007) and Osborne (2007).
3. Digital artefacts are errors, blanks, clicks and other ephemera introduced into the computational results of digital processing. The computer software creates them rather than existing in experienced reality, hence they are generated by the medium of software code. They raise particular epistemological problems for a range of disciplines that have to learn to distinguish between the artefacts (i.e. as noise) and phenomena (i.e. as signal).
4. This research points to the fact that further scholarly work is needed urgently on understanding and translating code so that its political, sociological and cultural significance can be better understood.

Certainly work on methodology for researching code (such as a *semiotics of code*) would contribute to better understanding of an important aspect of our reliance on code and the meanings embedded within it.

5. Labour is that activity which corresponds to the biological processes and necessities of human existence (see Arendt 1989: 79–135).
6. An early version of this argument is discussed in Berry (2005).
7. Work is 'the activity which corresponds to the unnaturalness of human existence, which is not embedded in, and whose mortality is not compensated by, the species' ever-recurring life-cycle'. Work (as both *techné* and *poiesis*) corresponds to the fabrication of an artificial world of things that endure beyond their creation (i.e. lasting in time). Work thus creates an artificial world distinct from nature, a world noticeable by its durability, its permanence and relative independence from the individual actors and acts that called them into being (see Arendt 1989: 136–74).
8. 'Action, the only activity that goes on directly between men without the intermediary of things or matter, corresponds to the human condition of plurality, to the fact that men, not Man, live on the earth and inhabit the world . While all aspects of the human condition are somehow related to politics, this plurality is specifically the condition – not only the *conditio sine qua non*, but the *conditio per quam* – of all political life' (Arendt 1989: 7, see also 175–247).
9. Especially if we think of code as a conversation or speech act.

BIBLIOGRAPHY

Adorno, T. and Eisler, H. (1994) *Composing for the Films*. London: Athlone Press.

Ananian, C. S. (1999) 'A Linux Lament.' Retrieved 14 May 2008 from www.salon.com/tech/feature/1999/07/30/redhat_shares/index.html

Anderson, B. (1983) *Imagined Communities*. London: Verso.

Arendt, H. (1989) *The Human Condition*. Chicago, London: University of Chicago Press.

Astrachan, O. (2003) 'Bubble Sort: An Archaeological Algorithmic Analysis.' Presented at SIGCSE 03, 19–23 February, Reno, Nevada. Retrieved 14 May 2008 from www.cs.duke.edu/~ola/papers/bubble.pdf

Atton, C. (2003) 'Talking about a Quiet Revolution'. *Financial Times*, 4 November: 8.

Bakardjieva, M. and Feenberg, A. (2001) 'Involving the Virtual Subject: Conceptual, Methodological and Ethical Dimensions.' *Journal of Ethics and Information Technology* 2 (4): 233–40.

Barber, B. R. (1984) *Strong Democracy: Participatory Politics for a New Age*. California: University of California.

Barbrook, R. (1998) 'The Hi-Tech Gift Economy.' *First Monday*. Volume 3, Number 12, 7 December. Retrieved 14 May 2008 from www.firstmonday.org/issues/issue3_12/

Barbrook, R. and Cameron, A. (1995) 'The Californian Ideology.' Retrieved 14 May 2008 from www.hrc.wmin.ac.uk/theory-californianideology-main.html

Barlow, J. P. (1990) 'In Search of NuPrometheus.' In *Crime and Puzzlement*. Retrieved 14 May 2008 from http://w2.eff.org/legal/cases/SJG/?f=crime_and_puzzlement.1.txt

—— (1996) 'A Declaration of the Independence of Cyberspace.' Retrieved 14 May 2008 from http://homes.eff.org/~barlow/Declaration-Final.html

Barnes, B. (1995) *The Elements of Social Theory*. London: UCL Press.

Barroso, J. M. (2005) 'The Lisbon Strategy Time for Action.' Retrieved 14 May 2008 from http://europa.eu.int/rapid/pressReleasesAction.do?reference=SPEECH/05/188&format=HTML&aged=1&language=EN&guiLanguage=en

Barry, A. (2001) *Political Machines: Governing a Technological Society*. London: Athlone Press.

Bassett, C. (2004) 'How Many Movements' In L. Back and M. Bull (eds) *The Auditory Culture Reader*. London: Berg.

Beck, U. (2000) 'Risk Society Revisited: Theory, Politics and Research Programmes.' In B. Adam, U. Beck and J. van Loon (eds) *The Risk Society and Beyond. Critical Issues for Social Theory*. London: Sage: 211–29.

Bell, D. (1979) 'The Social Framework of the Information Society.' In M. Dertouzous and J. Moses (eds) *The Computer Age: A Twenty Year View.* Cambridge: MIT Press: 163–212.

Benhabib, S. (1992) *Situating the Self: Gender, Community and Postmodernism in Contemporary Ethics*. London: Routledge.

Benkler, Y. (2002) 'Coase's Penguin, or Linux and the Nature of the Firm.' *The Yale Law Journal* 112: 369–446.

—— (2003a) 'Freedom in the Commons, Towards a Political Economy of Information.' *Duke Law Journal*, 52 Duke L.J. 1245. Retrieved 14 May 2008 from www.law.duke.edu/shell/cite.pl?52+Duke+L.+J.+1245

—— (2003b) 'The Political Economy of Commons.' *Upgrade*, Vol. IV, No. 3, June.

—— (2004) 'Sharing Nicely: on Shareable Goods and the Emergence of Sharing as a Modality of Economic Production.' *The Yale Law Journal* 114 (2): 274–358.

—— (2006) *The Wealth of Networks*. London: Yale University Press.

Bergson, H. (1998) *Creative Evolution*. New York: Dover Publications.

Berry, D. M. (2005) 'Free as in "Free Speech" or Free as in "Free Labour"? A Philosophical Enquiry into Free Culture.' *Free Culture Magazine*. Issue 3, April. Retrieved 14 May 2008 from www.freesoftwaremagazine.com/articles/free_labour

Berry, D. M. and Moss, G. (2003) 'The Libre Culture Manifesto.' Retrieved 14 May 2008, from www.freesoftwaremagazine.com/free_issues/issue_02/libre_manifesto

—— (2006) 'Free and Open-Source Software: Opening and Democratising E-Government's Black Box.' *Information Polity* 11 (1): 21–34.

Bettig, R.V. (1996) *Copyrighting Culture: the Political Economy of Intellectual Property*. Oxford: Westview.

Bezrouka, N. (2006) 'Portraits of Open Source Pioneers.' Retrieved 14 May 2008 from www.softpanorama.org/People/Torvalds/index.shtml

Bezroukov, N. (2003) 'Open Source Software Development as a Special Type of Academic Research (Critique of Vulgar Raymondism).'

Retrieved 14 May 2008 from www. firstmonday.dk/issues/issue4_10/bezroukov/index.html.
Bijker, W. E., Hughes, T. P. and Pinch, T. (2001) *The Social Construction of Technological Systems*. Cambridge: MIT Press.
Bloom, A. (1987) *The Closing of the American Mind*. London: Simon and Schuster.
Bollier, D. (2003) *Silent Theft: The Private Plunder of our Common Wealth*. London: Routledge.
Bonaccorsi, A. and Rossi, C. (2003) 'Comparing Motivations of Individual Programmers and Firms to Take Part in the Open Source Movement. From Community to Business.' Retrieved 14 May 2008, from http://opensource.mit.edu/papers/bnaccorsirossimotivationlong.pdf
Borsook, P. (1995) 'How Anarchy Works.' In *Wired*. Issue 3.10. Retrieved 14 May 2008 from www.wired.com/wired/archive/3.10/ietf.html
Bourdieu, P. (1986) *Distinction*. London: Routledge.
Bowrey, K. (2005) *Law and Internet Cultures*. Cambridge: Cambridge University Press.
Boyd, C. (2005) ' "Free" Danish Beer Makes a Splash.' Retrieved 14 May 2008 from http://news.bbc.co.uk/1/hi/technology/4718719.stm
Boyle, J. (1996) *Shamans, Software, and Spleens: Law and the Construction of the Information Society*. Cambridge, MA: Harvard University Press.
—— (2003) 'The Second Enclosure Movement and the Construction of the Public Domain.' *Law and Contemporary Problems* 66: 33–74.
Brown, N. and Szeman, I. (2005) 'What is the Multitude? Questions for Michael Hardt and Antonion Negri.' *Cultural Studies* 19 (3): 372–87.
Brzezinski, Z. (1970) *Between Two Ages: America's Role in the Technetronic Age*. New York: Viking.
Campbell-Kelly, M. (1995) 'Development and Structure of the International Software Industry, 1950–1990.' *Business and Economic History* 24 (2): 73–110.
—— (2004) *From Airline Reservations to Sonic the Hedgehog: A History of the Software Industry*. London: MIT Press.
Castells, M. (2000a) *The Information Society: the Power of Identity*. Oxford: Blackwell.
—— (2000b) *The Information Society: the Rise of the Network Society*. Oxford: Blackwell.
—— (2001) *The Internet Galaxy*. Oxford: Oxford University Press.
Chopra, S. and Dexter, S. (2005a) 'The Political Economy of Open Source Software.' *The International Journal of Technology, Knowledge, and Society* 1 (1).
—— (2005b) 'A Comparative Ethical Assessment of Free Software Licensing Schemes.' *Proceedings of the Sixth International Conference*

of Computer Ethics: Philosophical Enquiry (CEPE2005), Enschede, The Netherlands, July.

CIIPRF (2005) The Creative Industries Intellectual Property Rights Forum. Meeting Minutes. 16 December 2005. Department of Culture, Media and Sport (UK). Retrieved 11 November 2006 from www.culture.gov.uk/creative_industries/ip_forum.htm

Cisco (2006) Software License Agreement. Retrieved 14 May 2008 from www.cisco.com/public/sw-license-agreement.html

Coleman, B. (2005) *The Social Construction of Freedom in Free and Open Source Software: Hackers, Ethics, and the Liberal Tradition*. PhD, Department of Anthropology: University of Chicago.

Coleman, J. (2000a) *A History of Political Thought: From Ancient Greece to Early Christianity*. London: Blackwell.

—— (2000b) *A History of Political Thought: From the Middle Ages to the Renaissance*. London: Blackwell.

Connor, D. (2000) 'Open Source Moves beyond the Server: Enterprise, Network Products Feel Presence of Software Movement.' *Network World*. Retrieved 28 April 2003 from www2.itworld.com/cma/ett_article_frame/0,2848,1_3548,00.html

Coombe, R. J. (1998) *The Cultural Life of Intellectual Properties: Authorship, Appropriation and the Law*. London: Duke University Press.

Coquille, D. R. (1979) 'Mosses from an Old Manse: Another Look at Some Historic Property Cases about the Environment. *Cornell Law Journal* 564: 761–821.

Cornish, W. and Llewelyn, D. (2003) *Intellectual Property: Patents, Copyrights, Trademarks and Allied Rights*. London: Sweet & Maxwell.

Couldry, N. (2004) 'The Productive "Consumer" and the Dispersed "Citizen".' *International Journal of Cultural Studies* 7 (1): 21–32.

Cowley, S. (2005) 'IBM Plans $100 Million Linux Investment.' Retrieved 14 May 2008 from www.infoworld.com/article/05/02/17/HNimblinuxinvest_1.html

Cox, R. W. (1996) *Approaches to World Order*. Cambridge University Press.

CreativeCommons (2003) Creative Commons. Retrieved 1 February 2003 from www.creativecommons.org

Cringely, R. X. (1999) 'This Time the Flavorade is Sweet: How Apple is Using Open Source Software to Defeat Microsoft Without Appearing to Do So.' Retrieved 14 May 2008 from www.pbs.org/cringely/pulpit/1999/pulpit_19990715_000617.html

DeMarco, T. and Lister, T. (1999) *Peopleware: Productive Projects and Teams*. London: Dorset House Publishing Co. Inc.

DiBona, C., Ockman, S. and Stone, M. (1999) *Open Sources: Voices of the Open Source Revolution*. Cambridge: O'Reilly.

DiBona, C., Cooper, D. and Stone, M. (2006) *Open Sources: The Continuing Evolution*. Cambridge: O'Reilly.

Drahos P. (2000) 'Indigenous Knowledge, Intellectual Property and Biopiracy: Is a Global Biocollecting Society the Answer?' *European Intellectual Property Review* 22 (6), 245–250.

Drahos, P. and Braithwaite, J. (2002) *Information Feudalism: Who Owns the Knowledge Economy?* London: Earthscan.

Dreyfus, H. (2005) 'Being and Power: Heidegger and Foucault.' Retrieved 14 May 2008, from http://ist-socrates.berkeley.edu/~hdreyfus/html/paper_being.html

Drucker, P. (1968) *The Age of Discontinuity*. New York: Harper and Row.

Dyer-Witheford, N. (1999) *Cyber-Marx: Cycles and Circuits of Struggle in High-technology Capitalism*. Urbana: University of Illinois Press.

Economist, The (2005) 'The Arms Race.' 20 October. Retrieved 22 May 2008 from www.economist.com/surveys/displaystory.cfm?story_id=E1_VDTVDVJ

Elliott, M. S. and Scacchi, W. (2002) 'Communicating and Mitigating Conflict in Open Source Software Development Projects.' Retrieved 14 May 2008 from www.ics.uci.edu/~wscacchi/

Ess, C. (2001) 'Internet Research Ethics.' Retrieved 12 January 2003 from www.nyu.edu/projects/nissenbaum/projects_ethics.html

EU (2001) Directive 2001/29/ec of the European Parliament and of the Council of 22 May 2001 on the Harmonisation of Certain Aspects of Copyright and Related Rights in the Information Society. Retrieved 14 May 2008 from http://eurlex.europa.eu/smartapi/cgi/sga_doc?smartapi!celexapi!prod!CELEXnumdoc&lg=EN&numdoc=32001L0029&model=guichett

—— (2005a) *Challenges for the European Information Society Beyond 2005*. Retrieved 14 May 2008 from http://europa.eu.int/information_society/eeurope/i2010/docs/communications/new_chall_en_adopted.pdf

—— (2005b) *Declaration on Information Society*. Retrieved 14 May 2008 from www.coe.int/t/e/integrated_projects/democracy/02_Activities/00_Declaration_on_Information_Society/

—— (2005c) *I2010 – a European Information Society for Growth and Employment*. Retrieved 14 May 2008 from http://europa.eu.int/information_society/eeurope/i2010/docs/communications/com_229_i2010_310505_fv_en.pdf

—— (2006) *Sixth Framework Programme*. Retrieved 14 May 2008, from http://cordis.europa.eu/fp6/dc/index.cfm?fuseaction=UserSite.FP6HomePage

European Council (2000) Presidency Conclusions: Lisbon European Council. 23 and 24 March 2000. Retrieved 14 May 2008 from http://ue.eu.int/ueDocs/cms_Data/docs/pressData/en/ec/00100-r1.en0.htm

Fairclough, N. (1992) *Discourse and Social Change*. Cambridge: Polity.

—— (2000) *New Labour, New Language?* London: Routledge.

Feed Magazine (2000) 'Re: Martin Garbus.' In *Feed Magazine*. Retrieved 14 May 2008 from http://web.archive.org/web/20000619102547/www.feedmag.com/re/re340_master.html

Feenberg, A. (1995) *Alternative Modernity: the Technical Turn in Philosophy and Social Theory*. London: University of California Press.

—— (2002) *Transforming Technology: A Critical Theory Revisited*. Oxford: Oxford University Press.

Feenberg, A. and Hannay, A. (1995) *Technology and the Politics of Knowledge*. Bloomington: Indiana University Press.

Feller, J., Fitzgerald, B., Hissan, S. and Lakhani, K. R. (2000) *Perspectives on Free and Open Source Software*. Cambridge: MIT Press.

Fenn, P. T. (1925) 'Justinian and the Freedom of the Sea.' *The American Journal of International Law* 19 (4): 716–27.

FFII (2004) Libdts Threatened By Dts Inc. and Ep 864 146 Software Patent. Retrieved 21 June 2005, from www.ffii.org/%7Ezoobab/libdts/

Fisher, W. (2004) *Promises to Keep: Technology, Law, and the Future of Entertainment*. Stanford: Stanford University Press.

Florida, R. (2004) *The Rise of the Creative Class: and How it's Transforming Work, Leisure, Community and Everyday Life*. New York: Basic Books.

FLOSSPOLS (2005) *Free/Libre and Open Source Software: Policy Report. Deliverable D10. Skills Survey Interim Report*. Retrieved 14 May 2008 from www.flosspols.org/deliverables/FLOSSPOLS-D10-skills%20survey_interim_report-revision-FINAL.pdf

Foucault, M. (1979) *Discipline and Punish*. London: Penguin.

—— (1989) *The Archaeology of Knowledge*. London: Routledge.

—— (1990) *History of Sexuality*. Harmondsworth: Penguin.

Free Software Foundation (1989). GNU General Public License Version 1. Retrieved on 14 May 2008 from www.gnu.org/copyleft/copying-1.0.html

Fueston, L. (1998) 'Self-interest.' Retrieved 14 May 2008 from www.fsf.org/philosophy/self-interest.html

Fuller, M. (2003) *Behind the Blip: Essays on the Culture of Software*. London: Autonomedia.

Garlick, M. (2006) 'Spanish Court Recognizes CC-Music.' Retrieved 14 May 2008 from http://creativecommons.org/weblog/entry/5830

Gates, III, W. H. (1976) 'An Open Letter to Hobbyists. Letter to Homebrew Computer Club.' Retrieved 14 May 2008 from www.blinkenlights.com/classiccmp/gateswhine.html

Ghosh, R. A. (1998) 'Cooking Pot Markets: An Economic Model for the Trade in Free Goods and Services on the Internet.' Retrieved 14 May 2008 from http://firstmonday.org/issues/issue3_3/ghosh/index.html#CONCLUSION

—— (ed.). (2005) *CODE: Collaborative Ownership and the Digital Economy*. London: MIT Press.

Gibbons, M., Limoges, C., Nowotny, H., Schwartzman, P. S. and Trow, M. (1994) *The New Production of Knowledge: The Dynamics of Science and Research in Contemporary Societies*. London: Sage.

Giedion, S. (1969). *Mechanization Takes Command*. London: WW Norton & Co Ltd.

Gillespie, R. (2000) 'German Federal Government to Support Open Source Software.' Retrieved 14 May 2008 from www.internetnews.com/busnews/article.php/6_408271

Gillespie, T. (2004) 'Copyright and Commerce: the DMCA, Trusted Systems, and the Stabilization of Distribution.' *New Media and Society* 20: 239–54.

Glass, R. L. (2000) 'The Sociology of Open Source: Of Cults and Cultures.' *IEEE Software*. May/June 17 (3): 104–5.

Golding, P. and Murdock, G. (2000) 'Culture, Communication, and Political Economy.' In J. Curran and M. Gurevitch (eds) *Mass Media and Society* (3rd edition). London: Edward Arnold.

Goldstein, P. (2003) *Copyright's Highway: From Gutenberg to the Celestial Jukebox*. Stanford: Stanford University Press.

Greene, T. (2001) 'Ballmer: "Linux is a Cancer".' *The Register*. Retrieved 14 May 2008 from www.theregister.co.uk/2001/06/02/ballmer_linux_is_a_cancer/

Greenspan, A. (2003) 'Remarks By Chairman Alan Greenspan at the 2003 Financial Markets Conference of the Federal Reserve Bank of Atlanta, Sea Island, Georgia.' Retrieved 14 May 2005, from www.federalreserve.gov/boarddocs/speeches/2003/20030404/default.htm

Groklaw (2006) 'Wells' Order Granting in Part IBM's Motion to Limit SCO's Claims.' Retrieved 14 May 2008 from www.groklaw.net/article.php?story=20060628203537917

Habermas, J. (1988) *Legitimation Crisis*. London: Polity.

—— (1992) *The Structural Transformation of the Public Sphere*. Cambridge: Polity.

—— (1997) *Between Facts and Norms*. Cambridge: Polity.

Hafner, K. and Lyon, M. (1998) *Where Wizards Stay Up Late: the Origins of the Internet*. New York: Touchstone.

Hardin, G. (1968) 'The Tragedy of the Commons.' *Science* 3859162: 1243–8.

Hardt, M. and Negri, A. (2000) *Empire*. London: Harvard.

——— (2004) *Multitude: War and Democracy in the Age of Empire*. New York: Penguin Press.

Harvey, B. (2003) 'What is a Hacker?' Retrieved 25 June 2003 from www.cs.berkeley.edu/~bh/hacker.html

HC 310-I (2007) 'UNCORRECTED TRANSCRIPT OF ORAL EVIDENCE to be published as HC 310-i. Minutes of Evidence taken before the Science and Technology Committee.' Retrieved 14 May 2008 from www.publications.parliament.uk/pa/cm200607/cmselect/cmsctech/uc310-i/uc31002.htm

Heidegger, M. (2000) 'The Question Concerning Technology.' In Krell, D. F. (ed.) *Martin Heidegger: Basic Writings*. London: Routledge.

Heller, M. A. (1998) 'The Tragedy of the Anticommons: Property in the Transition from Marx to Markets.' *Harvard Law Review* 1 (11): 621–88.

Heraclitus (2006) 139 Fragments. Retrieved 28 March 2007 from http://philoctetes.free.fr/heraclite.pdf

Herman, A., Coombe, R. and Kaye, L. (2006) 'Your Second Life: Goodwill and the Performativity of Intellectual Property in Online Gaming.' *Cultural Studies* 20 (2–3) March/May: 184–210.

Herring, S. (1996) 'Critical Analysis of Language Use in Computer-mediated Contexts: Some Ethical and Scholarly Considerations.' *The Information Society* 12 (2): 153–68.

Hertel, G., Niedner, S., and Herrmann, S. (2003) 'Motivation of Software Developers in Open Source Projects: an Internet-based Survey of Contributors to the Linux Kernel. *Research Policy* 32: 1159–77.

Himanen, P. (2001) *The Hacker Ethic and the Spirit of the Information Age*. London: Vintage.

Hobbes, T. (1998) *Leviathan*. Oxford: Oxford Paperbacks.

Howkins, J. (2001) *The Creative Economy: How People Make Money from Ideas*. London: Penguin.

Hunter, D. (2003) 'The Tragedy if the Digital Anticommons.' *California Law Review* 91 (2): 442–519

—— (2004) 'Marxist-Lessigism.' Retrieved 14 May 2008, from www.legalaffairs.org/issues/November-December-2004/feature_hunter_novdec04.html

Hyde, L. (2006). *The Gift: How the Creative Spirit Transforms the World*. London: Canongate.

Infonomics (2002) 'Free/libre and Open Source Software: Survey and Study: FLOSS Final Report.' Retrieved 14 May 2008 from www.infonomics.nl/FLOSS/report/index.htm

Jameson, F. (1992) *Postmodernism or the Cultural Logic of Late Capitalism*. London: Verso.

Jefferson, T. (1999) *Political Writings*. Cambridge: Cambridge University Press.

Jenkins, H. (2006) *Convergence Culture*. London: New York University Press.

Jessop, B. (2004) 'Critical Semiotic Analysis and Cultural Political Economy.' *Critical Discourse Studies* 1 (2): 159–74.

Johanson, J. (1999) 'The Truth about DVD CSS cracking by MoRE and [dEZZY/DoD].' Retrieved 14 May 2008 from www.lemuria.org/DeCSS/dvdtruth.txt

Jones, S. (1999) *Doing Internet Research: Critical Issues and Methods for Examining the Net*. London: Sage.

Junger v. Daley (2000) Peter D. Junger, Plaintiff-Appellant v. William Daley, United States Secretary of Commerce, et al., Defendants-Appellees. 2000 FED App. 0117P (6th Cir.). Retrieved 3 April 2005 from http://samsara.law.cwru.edu/victory.txt

Junnarker, S. (2001) 'IBM Makes $40 Million Open-source Offer.' Retrieved 14 May 2008 from www.news.com/IBM-makes-40-million-open-source-offer/2100-1001_3-275388.html

Kanellos, M. (2002) 'Anti-Unix Web Site on the Fritz?' Retrieved 14 May 2008 from http://news.com.com/2100-1001-874132.html

Kelty, C. M. (2001) 'Free Software/Free Science.' Retrieved 14 May 2008 from www.firstmonday.dk/issues/issue6_12/kelty/

Kesan, J. P. and Shah, R. (2002) 'Shaping Code.' Retrieved 14 May 2008 from http://papers.ssrn.com/sol3/papers.cfm?abstract_id=328920

Kittler, F. (1995) 'There Is No Software.' *CTheory*. Retrieved 14 May 2008 from www.ctheory.net/articles.aspx?id=74

Kuhn, B. M. and Stallman, R. M. (2001a) 'Freedom or Power?' Retrieved 14 May 2008, from www.gnu.org/philosophy/freedom-or-power.html

—— (2001b) 'Freedom or Power?' Retrieved 21 May 2008, from http://linux.oreillynet.com/pub/a/linux/2001/08/15/free_software.html

Labour Party (2005) The Labour Party Manifesto 2005. Retrieved 21 May 2008 from http://image.guardian.co.uk/sys-files/Politics/documents/ 2005/04/13/labourmanifesto.pdf

Laclau, E. (2001) 'Philosophical Roots of Discourse Theory.' Retrieved 21 May 2008 from www.diskurs.dk/skaermkursus/skaerm4eng.pdf

Laclau, E. and Mouffe, C. (2001) *Hegemony and Socialist Strategy: Towards a Radical Democratic Politics*. London: Verso.

Lakhani, K. R. and Wolf, R. G. (2005) 'Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects.' In Feller, J., Fitzgerald, B., Hissan, S. and Lakhani, K. R. (eds) *Perspectives on Free and Open Source Software*. Cambridge: MIT Press.

LaMonica, M. (2006) 'Sun picks GPL License for Java code.' Retrieved 21 May 2008 from http://news.com.com/Sun+picks+GPL+license+for+Java+code/2100-7344_3-6134584.html

Lancaster, D. (2001) 'Code, Culture and Cash: The Fading Altruism of Open Source Development.' *First Monday*. Issue 6, number 12. Retrieved 21 May 2008 from www.firstmonday.dk/issues/issue6_12/lancashire/#l3

Lange, D. (1981) 'Recognizing the Public Domain.' *Law and Contemp. Probs* 44: 147–78.

Lazzarato, M. (1996) 'Immaterial Labor.' In P. Virno and M. Hardt (eds), *Radical Thought in Italy: A Potential Politics*. Minnesota: University of Minnesota Press, 132–46.

Leonard, A. (2000) 'Let My Software Go! Interview with Eric S. Raymond.' Retrieved 21 May 2008 from http://archive.salon.com/21st/feature/1998/04/cov_14feature.html

—— (2000b) 'The Shape of Open Source to Come.' Retrieved 5 May 2004 from http://dir.salon.com/tech/log/2000/02/03/slashdot/index.html?CP=SAL&DN=110

Lessig, L. (1999) *Code and Other Laws of Cyberspace*. New York: Basic Books.

—— (2002a) 'Free Culture: Lawrence Lessig Keynote From Oscon 2002.' Retrieved 21 May 2008 from www.oreillynet.com/pub/a/policy/2002/08/15/lessig.html?page=2

—— (2002b) *The Future of Ideas: the Fate of the Commons in a Connected World*. New York: Vintage.

—— (2004) *Free Culture: How Big Media Uses Technology and the Law to Lock Down Culture and Control Creativity*. New York: Allen Lane.

—— (2005a) 'CC in Review: Lawrence Lessig on Supporting the Commons.' Retrieved 21 May 2008 from http://creativecommons.org/weblog/entry/5661

—— (2005b) 'Advice Taken.' Retrieved 21 May 2008 from www.lessig.org/blog/archives/002872.shtml

Lettice, J. (2004) 'MS Windows Source Code Escapes onto Internet.' *The Register*. 13 February. Retrieved 22 May 2008 from www.theregister.co.uk/2004/02/13/ms_windows_source_code_escapes/

Levy, S. (1994) *Hackers: Heroes of the Computer Revolution*. London: Penguin.

Lewis, P. (2007) 'Microsoft Vista: Should You Buy Now?' Retrieved 21 May 2008 from http://money.cnn.com/magazines/fortune/fortune_archive/2007/02/05/8399126/index.htm

Linstone, H. A. and Murray, T. (2002) *The Delphi Method: Techniques and Applications*. New York: NJIT.

Litman, J. (2001) *Digital Copyright*. New York: Prometheus Books.

LOCA-Records (2003) 'LOCA Records Mission.' Retrieved 21 May 2008 from www.locarecords.com

Locke, J. (2002) *The Second Treatise of Government; and, a Letter Concerning Toleration*. Mineola, NY: Dover Publications.

Lopata, B. B. (1973) 'Property Theory in Hobbes.' *Political Theory*, 1 (2): 203–18.

Lowood, H. (2001) 'The Hard Work of Software History.' *RBM: A Journal of Rare Books, Manuscripts and Culture*. Vol. 2, No. 2. Fall.

MacKay, C. (1995) *Extraordinary Popular Delusions and the Madness of Crowds*. London: Wordsworth Reference.

MacKenzie, D. and Wajcman, J. (1999) *The Social Shaping of Technology*. Buckingham: Open University Press.

McLuhan, M. (1995) *Understanding Media*. London: Routledge.

Maihoffer, W. (2003) 'The Ethos of the Republic and the Reality of Politics.' In G. Bock, Q. Skinner and M. Viroli (eds) *Machiavelli and Republicanism*. Cambridge: Cambridge University Press.

Mansell, R. (2004) 'Political Economy, Power and New Media.' *New Media and Society* 6 (1): 96–105.

Marcuse, H. (1991) *One Dimensional Man*. London: Routledge.

Margetts, H. (1996) 'The Implications for Democracy of Computerisation in Government.' In P. Hirst and S. Khilnani (eds) *Reinventing Democracy*. London: Blackwell.

Markoff, J. (2005) *What the Dormouse Said: How the 60s Counterculture Shaped the Personal Computer*. New York: Viking.

Marquand, D. (2004) *Decline of the Public*. London: Polity.

Marson, I. (2006) 'Creative Commons License Upheld by Court.' Retrieved 21 May 2008 from http://news.cnet.com/2100-1030_3-6052292.html

Marx, K. and Engels, F. (1998) *The Communist Manifesto*. Oxford: Oxford Paperbacks.

—— (1999) *The German Ideology*. London: Lawrence & Wishart.

Maskus, K. E. (2000) *Intellectual Property Rights in the Global Economy*. Washington: Institute for International Economics.

Mauss, M. (2002) *The Gift: The Form and Reason for Exchange in Archaic Societies*. London: Routledge.

May, C. (2000) *A Global Political Economy of Intellectual Property Rights: the New Enclosures?* London: Routledge.

—— (2002) *The Information Society: A Sceptical View*. London: Polity.

May, C. and Sell, S. K. (2006). *Intellectual Property Rights: A Critical History*. London: Lynne Rienner Publishers Ltd.

Mendels, P. (1999) 'Professor Argues for Free Speech in Computer Tongues.' In *The New York Times*. 5 March. Retrieved 21 May 2008 from www.nytimes.com/library/tech/99/03/cyber/cyberlaw/05law.html

Metcalfe, B. (2004) 'If Open-source Software is So Much Cooler, Why Isn't Transmeta Getting it?' Retrieved 21 May 2008 from www.infoworld.com/articles/op/xml/00/02/14/000214opmetcalfe.html

Microsoft (2003). Supplemental end user license agreement – Windows system resource manager for Microsoft Windows Server 2003, Enterprise edition and Datacenter edition. Retrieved 21 May 2008, from www.microsoft.com/windowsserver2003/downloads/wsrmeula.mspx

Mills, C. W. (2000) *The Sociological Imagination*. Oxford: Oxford University Press.

Mitra, A. and Cohen, E. (1999) 'Analysing the Web: Directions and Challenges.' In S. Jones (ed.) *Doing Internet Research: Critical Issues and Methods for Examining the Net*. Oxford: Sage.

Moglen, E. (1999) 'Anarchism Triumphant: Free Software and the Death of Copyright.' Retrieved 21 May 2008 from www.firstmonday.org/issues/issue4_8/moglen/index.html#author

—— (2003) 'The Dotcommunist Manifesto.' Retrieved 21 May 2008 from http://emoglen.law.columbia.edu/publications/dcm.html

Monterey Institute (2005) 'Current and Future Space Security: Major Treaties and Membership.' Retrieved 21 May 2008, from http://cns.miis.edu/research/space/treaties/

Moody, G. (2002) *Rebel Code: Inside Linux and the Open Source Revolution*. Oxford: Perseus Publishing.

Morris-Suzuki, T. (1988) *Beyond Computopia: Information, Automation and Democracy in Japan*. London: Kegan Paul International.

Mouffe, C. (2000) *The Democratic Paradox*. London: Verso.

—— (2005) *On the Political*. London: Routledge.

MPAA v. 2600 (2001) Universal City Studios, Inc., Paramount Pictures Corporation, Metro-Goldwyn-Mayer Studios Inc., Tristar Pictures, Inc., Columbia Pictures Industries, Inc., Time Warner Entertainment Company, L.P., Disney Enterprises Inc., Twentieth Century Fox Film Corporation, V. Eric Corley, Also Known As Emmanuel Goldstein, And

2600 Enterprises Inc. Retrieved 21 May 2008 from http://cryptome.org/mpaa-v-2600-cad.htm

Mumford, L. (1963) *Technics and Civilisation*. London: Harvest.

National Statistics (2006) 'ONS Improves Measure of "New Economy".' *National Statistics*. Retrieved 21 May 2008 from www.statistics.gov.uk/pdfdir/newec0206.pdf

Naughton, J. (2005) *A Brief History of the Future: the Origins of the Internet*. London: Phoenix.

Needle, D. (2006) 'Lessig Makes Plea For Read/Write Internet.' Retrieved 21 May 2008 from www.internetnews.com/bus-news/article.php/3585731

Negri, A. (1989) *The Politics of Subversion: A Manifesto for the Twenty-first Century*. London: Polity Press.

Negroponte, N. (1996) *Being Digital*. London: Hodder & Stoughton.

Ness, J. (2004) 'An Unlikely Champion: Why the Anticorporate Advocates of Free Software Have Come to Embrace One of the Industry's Biggest Players.' Retrieved 03 May 2004 from www.msnbc.msn.com/id/5973877/site/newsweek/

Niccolai, J. (2001) 'IBM's Graffiti Ads Run Afoul of City Officials.' Retrieved 21 May 2008 from http://archives.cnn.com/2001/TECH/industry/04/19/ibm.guerilla.idg/

Nissenbaum, H. and Howe, D. C. (2003) 'Values in design.' Retrieved 21 May 2008 from www.nyu.edu/projects/valuesindesign/index.html

Olson, M. (1971) *The Logic of Collective Action: Public Goods and the Theory of Groups*. London: Harvard University Press.

O'Reilly, T. (2001) 'My definition of Freedom Zero.' Retrieved 26 May 2003 from www.oreillynet.com/cs/weblog/view/wlg/526

—— (2002) 'The Growing Politicization of Open Source.' Retrieved 21 May 2008 from www.oreillynet.com/pub/wlg/1840

—— (2003) '"Freedom Zero" and "Freedom or Power": Tim O'Reilly and FSF Leaders Debate.' Retrieved 25 June 2003 from http://linuxtoday.com/news_story.php3?ltsn2001-08-17-011-20-OP-CY-MS

Osborne, G. (2007) 'Open Source Politics.' Retrieved 21 May 2008 from www.conservatives.com/tile.do?def=news.story.page&obj_id=135394

OSS (2005) 'Common Land in England and Wales.' Open Spaces Society. Retrieved 21 May 2008, from www.oss.org.uk/commons/commons.htm

Payne, R. J. (1978) 'Mining the Deep Seabed: the Political, Economic and Legal Struggle.' *The Journal of Politics* 40 (4): 933–55.

Perelman, M. (2003) 'The Political Economy of Intellectual Property.' *Monthly Review* 54 (8): 29–37.

Perens, B. (1999) 'It's Time to Talk about Free Software Again.' Retrieved 21 May 2008 from http://lists.debian.org/debian-devel/1999/02/msg01641.html

—— (2005) 'Friday November 18: My Speech at the UN World Summit on the Information Society.' Retrieved 15 October 2005 from http://swik.net/Bruce-Perens/Bruce+Perens'+Weblog/Friday+November+18:+My+Speech+at+the+UN+World+Summit+on+the+Information+Society/d0z5

Phillips, L. and Jørgensen, M. W. (2002) *Discourse Analysis as Theory and Method*. London: Sage.

Post (2003) 'Postnote: Government IT Projects.' Retrieved 12 April 2004, from www.parliament.uk/post

Poynder, R. (2006a) 'Freeing the Code.' *The Basement Interviews*. Retrieved 21 May 2008 from http://dialspace.dial.pipex.com/town/parade/df04/Richard_Stallman_Interview.pdf

—— (2006b) "Open Source." *The Basement Interviews*. Retrieved 21 May 2008 from http://dialspace.dial.pipex.com/town/parade/df04/Eric_Raymond_Interview.pdf

Prichard, C. (2001) 'Creative Bodies – Critically Reading "Creativity" in Management Discourse.' *Creativity and Innovation Management*, 11 (4): 265–76.

Python (2003) 'Python's Development Process.' Retrieved 2 May 2003 from www.python.org/dev/process.html

Rand, A. (1992) *Atlas Shrugged*. New York: Penguin.

Raymond, E. S. (1998) 'Goodbye, "Free Software"; Hello, "Open Source".' Retrieved 21 May 2008 from www.catb.org/~esr/open-source.html

—— (1999a) 'Surprised By Wealth.' Retrieved 21 May 2008 from http://lwn.net/1999/1216/a/esr-rich.html

—— (1999b) 'Shut Up and Show Them the Code.' Retrieved 21 May 2008 from www.linuxtoday.com/news_story.php3?ltsn=1999-06-28-023-10-NW-SM

—— (1999c) 'The Magic Cauldron.' Retrieved 21 May 2008 from www.catb.org/~esr/writings/magic-cauldron/

—— (2001a) *The Cathedral and the Bazaar: Musings on Linux and Open Source By an Accidental Revolutionary*. Cambridge: O'Reilly.

—— (2001b) 'Eric Raymond: Freedom, Power, or Confusion?' [ESR on debate between O'Reilly and FSF]. Retrieved 21 May 2008 from www.linuxtoday.com/news_story.php3?ltsn=2001-08-17-016-20-OP-CY

—— (2001c) 'Goodbye Free Software, Hello Open Source.' Retrieved 25 June 2003 from www.tuxedo.com/~esr

—— (2002a) 'Ethics From the Barrel of a Gun: What Bearing Weapons Teaches about the Good Life.' Retrieved 21 August 2008 from http://catb.org/esr/guns/gun-ethics.html
—— (2002b) 'Why We Fight – An Anti-idiotarian Manifesto.' Retrieved 21 May 2008 from http://catb.org/esr/aim/
—— (2003) 'Homesteading the Noosphere.' Retrieved 21 May 2008 from www.firstmonday.dk/issues/issue3_10/raymond/
—— (2006) Eric S. Raymond's Home Page. Retrieved 21 May 2008 from www.catb.org/~esr/
—— (2007) 'Why "Free" Software is too Ambiguous.' Retrieved 10 May 2007 from http://opensource.berlios.de/advocacy/free-notfree.php
Rose, C. (2003) 'Romans, Roads, and Romantic Creators: Traditions of Public Property in the Information Age.' Retrieved 1 October 2004, from www.boyle.com/ipconference
Rose, N. (1999) *Politics of Freedom: Reframing Political Thought*. Cambridge: Cambridge University Press.
Rothman, W. (2005) 'Napster Starts to Look Good.' *International Herald Tribune*, Review Section, 23 May: 16.
Rowan, D. (2005) 'Copyright Wrongs: We Can't Let the Music Industry Suits Stifle Creativity.' Retrieved 21 May 2008 from www.timesonline.co.uk/article/0,,1072-1657575,00.html
RSA (2005) 'The Adelphi Charter on Creativity, Innovation and Intellectual Property.' Retrieved 21 May 2008 from www.adelphi-charter.org
Rubinstein, N. (1993) 'Machiavelli and Florentine Republican Tradition.' In G. Bock, Q. Skinner and M. Viroli (eds) *Machiavelli and Republicanism*. Cambridge: Cambridge University Press.
Rushkoff, D. (2003) *Open Source Democracy: How Online Communication is Changing Offline Politics*. London: Demos.
Sachs, J. (2007) 'Lecture 1: Bursting at the Seams.' Retrieved 21 May 2008 from www.bbc.co.uk/radio4/reith2007/lecture1.shtml
Schroder, J. L. (2004) 'Unnatural Rights: Hegel and Intellectual Property.' Retrieved 21 May 2008 from http://ssrn.com/abstract=518182
Sclove, R. E. (1995) *Democracy and Technology*. London: Guildford Press.
Scoville, T. (1999) 'Whence the Source: Untangling the Open Source/Free Software Debate.' Retrieved 21 May 2008 from http://opensource.oreilly.com/news/scoville_0399.html
Selznak (2004) 'We Are Morons: a Quick Look at the Win2k Source.' Retrieved 21 May 2008 from www.kuro5hin.org/story/2004/2/15/71552/7795
Sennett, R. (1978) *The Fall of Public Man*. London: Penguin.

—— (1999) *The Corrosion of Character*. London: W. W. Norton & Company Ltd.
—— (2008) *The Craftsman*. London: Allen Lane.
Shankland, S. (1999) '"Open source" Infighting Grows.' Retrieved 21 May 2008 from http://news.cnet.com/2100-1001-221918.html
—— (2002) 'Unisys, Microsoft to Launch Anti-Unix Ads.' Retrieved 21 May 2008 http://news.cnet.com/2100-1001-870805.html
—— (2004) 'GPL Gains Clout in German Legal Case.' Retrieved 21 May 2008 from http://news.cnet.com/2100-7344-5198117.html
Sharf, B. F. (1999) 'Beyond Netiquette: the Ethics of Doing Naturalistic Discourse Research on the Internet.' In S. Jones (ed.) *Doing Internet Research: Critical Issues and Methods for Examining the Net*. London: Sage.
Shrader-Frechette, K. S. (1993) 'Reliance on Value Judgements in Repository Risk Assessment.' In K. S. Shrader-Frechette (ed.) *Burying Uncertainty. Risk and the Case against Geological Disposal of Nuclear Waste*. California: University of California Press: 27–38.
Silverstone, R. (2003) *Why Study the Media?* London: Sage.
—— (2007) *Media and Morality: On the Rise of the Mediapolis*. London: Polity Press.
Shiva, V. and Bedi, G. (2002). *Sustainable Agriculture and Food Security: The Impact of Globalization*. London: Sage.
Skinner, Q. (2003) 'Pre-humanist Origins of Republican Ideas.' In G. Bock, Q. Skinner and M. Viroli (eds) *Machiavelli and Republicanism*. Cambridge: Cambridge University Press.
Slashdot (2001a) 'Eric S. Raymond Answers.' Retrieved 21 May 2008 from http://slashdot.org/interviews/99/09/30/1418244.shtml
—— (2001b) 'Thus Spake Stallman.' Retrieved 21 May 2008 from http://slashdot.org/interviews/00/05/01/1052216.shtml
—— (2003) 'Slashdot: News for Nerds, Stuff that Matters.' Retrieved 21 May 2008 from www.slashdot.org
Slovic, P. (2000) 'Trust, Emotion, Sex, Politics and Science: Surveying the Risk Assessment Battlefield.' In *The Perception of Risk*. London: Earthscan: 390–412.
Smith, W. (1875) A Dictionary of Greek and Roman Antiquities. Retrieved 21 May 2008 from http://penelope.uchicago.edu/Thayer/E/Roman/Texts/secondary/SMIGRA*/Dominium.html
Smyth, S. M., Smyth, M. H. and Smith, W. P. (1998) *Multi-channel Predictive Subband Coder Using Psychoacoustic Adaptive Bit Allocation*. Digital Theatre Systems Inc, EU.
Sprankling, J. G. (2000) *Understanding Property Law*. Newark, NJ: LexisNexis.

Stallman, R. M. (1991a) 'Against Software Patents.' Retrieved 25 June 2003 from http://lpf.ai.mit.edu/Patents/against-software-patents.html
—— (1991b) 'The GNU General Purpose License.' Retrieved 21 May 2008 from www.gnu.org/copyleft/gpl.html
—— (1992) 'Why Software Should be Free.' Retrieved 21 May 2008 from www.gnu.org/philosophy/shouldbefree.html
—— (1993) 'The Gnu Manifesto.' Retrieved 21 May 2008 from www.gnu.org/gnu/manifesto.html
—— (1994) 'Why Software Should Not Have Owners.' Retrieved 21 May 2008 from www.gnu.org/philosophy/why-free.html
—— (1999a) 'The Gnu Operating System and the Free Software Movement.' In C. DiBona, S. Ockman and M. Stone (eds) *Open Sources: Voices from the Open Source Revolution*. Cambridge: O'Reilly.
—— (1999b) 'RMS Reponds.' Retrieved 21 May 2008 from http://slashdot.org/articles/99/06/28/1311232.shtml
—— (1999c) 'Licenses.' Retrieved 21 May 2008 from www.gnu.org/cgi-bin/estseek.cgi?phrase=licenses
—— (2001) 'Free Software is More Reliable!' Retrieved 21 May 2008 from www.gnu.org/software/reliability.html
—— (2002a) *Free Software, Free Society: Selected Essays of Richard M. Stallman*. Boston: GNU Press.
—— (2002b) 'Linux, GNU, and Freedom.' Retrieved 10 February 2003 from www.gnu.org/
—— (2003a) 'Gnu is Not Unix.' Retrieved 10 February 2003 from www.gnu.org
—— (2003b) 'Why "Free Software" is Better than "Open Source".' Retrieved 10 February 2003 from www.gnu.org
—— (2003c) 'Let's Share!' Retrieved 21 May 2008 from www.opendemocracy.net/media-copyrightlaw/article_31.jsp
—— (2003d) 'World Summit on the Information Society.' Retrieved 21 May 2008 from www.gnu.org/philosophy/wsis.html
—— (2004) 'The Free Software Definition.' Retrieved 21 May 2008 from www.gnu.org/philosophy/free-sw.html
—— (2005) 'Patent Absurdity.' Retrieved 21 May 2008 from www.guardian.co.uk/technology/2005/jun/23/onlinesupplement.insideit
Stavrakakis, Y. (1999) *Lacan and the Political*. London: Routledge.
Sterling, B. (1992) *The Hacker Crackdown*. London: Bantam Books.
Tanenbaum, A. (1992) 'Re: LINUX is Obsolete.' Email list. Retrieved 21 May 2008 from http://groups.google.com/group/comp.os.minix/browse_thread/thread/c25870d7a41696d2/f447530d082cd95d
Taylor, P. (1999) *Hackers: Crime and the Digital Sublime*. London: Routledge.

Terranova, T. (2004) *Network Culture: Politics for the Information Age*. London: Pluto.

Thomas, D. (2002) *Hacker Culture*. London: University of Minnesota Press.

Toffler, A. (1970) *Future Shock*. New York: Bantam.

—— (1980) *The Third Wave*. New York: Morrow.

Torvalds, L. (1993) 'Linux Commercial Use (was Re: Bbs Project – Linux?).' Retrieved 1 March 2003 from www.chrisshort.net/archives/open-source-software/reference/13-Mar-1992-linux-commercial-use.html

—— (1995) 'Birth of Linux.' Retrieved 10 February 2005 from http://yhosts.externet.hu/linux/linux-birth.html

—— (2006) 'Re: GPLv3 Position Statement.' Retrieved 21 May 2008 from http://lkml.org/lkml/2006/9/25/161

Touraine, A. (1971) *The Post-Industrial Society: Tomorrow's Social History: Classes, Conflicts and Culture in a Programmed Society*. London: Random House.

UN (1967) 'Treaty on Principles Governing the Activities of States in the Exploration and Use of Outer Space, Including the Moon and Other Celestial Bodies.' Retrieved 20 August 2004 from www.iasl.mcgill.ca/spacelaw/outerspace.html

—— (1970) 'United Nations Convention on the Law of the Sea: Agreement Relating to the Implementation of Part XI of the Convention.' Retrieved 21 May 2008 from www.un.org/Depts/los/convention_agreements/texts/unclos/closindx.htm

—— (1979) 'Agreement Governing the Activities of States on the Moon and Other Celestial Bodies.' Retrieved 21 May 2008 from www.unoosa.org/oosa/SpaceLaw/moon.html

Vaidhyanathan, S. (2001) *Copyrights and Copywrongs: the Rise of Intellectual Property and How it Threatens Creativity*. New York; London: New York University Press.

—— (2004) *The Anarchist in the Library: How the Clash between Freedom and Control is Hacking the Real World and Crashing the System*. New York: Basic Books.

Villanueva Nuñez, E. D. (2002) 'Congressman Villanueva Nuñez's Response to Microsoft Peru.' Retrieved 21 May 2008 from www.proposicion.org.ar/doc/gob/VillanuevaNunez-080402.html

Virno, P. (2004) *A Grammar of the Multitude*. New York: Semiotext(e).

Waskul, D. (1996) 'Ethics of Online Research: Considerations for the Study of Computer Mediated Forms of Interaction.' *The Information Society* 12 (2): 129–40.

Weber, S. (2003) 'Open Source Software in Developing Economies.' Retrieved 21 May 2008 from www.ssrc.org/programs/itic/publications/ITST_materials/webernote2.pdf

—— (2004) *The Success of Open Source*. London: Harvard University Press.

Webster, F. (1995) *Theories of the Information Society*. London: Routledge.

Weinstein, M. (2003) 'Text Analysis Markup System (Version 2.15a7).' Retrieved 21 May 2008 from www.sourceforge.net/projects/tamsys

Wheeler, D. A. (2002) 'More Than a Gigabuck: Estimating Gnu/Linux's Size.' Retrieved 21 May 2008 from www.dwheeler.com/sloc/redhat71-v1/redhat71sloc.html

—— (2005) 'Why Free-Libre/Open Source Software (FLOSS)? Look at the Numbers!' Retrieved 21 May 2008 from www.dwheeler.com/numbers/oss_fs_why_presentation.pdf

Wightman, A., Callander, R. and Boyd, G. (2004) 'Common Land in Scotland: A Brief Overview.' Retrieved 21 May 2008 from www.scottishcommons.org/docs/commonweal_3.pdf

Williams, S. (2002) *Free as in Freedom*. London: O'Reilly.

Winner, L. (1986) *The Whale and the Reactor: A Search for Limits in an Age of High Technology*. London: University of Chicago Press.

—— (2001) *Autonomous Technology: Technics-out-of-control as a Theme in Political Thought*. London: MIT Press.

Witheford, N. (1994) 'Autonomist Marxism and the Information Society.' *Capital and Class* 52, Spring: 85–125.

Withers, K. (2006) *Intellectual Property and the Knowledge Economy.* London: IPPR. Retrieved 21 May 2008 from www.ippr.org.uk/ecomm/files/intellectual_property.pdf

INDEX