

The Impact of Free and Open Source Licensing on Operating System Software Markets

Mikko Välimäki
Helsinki University of Technology
Ville Oksanen
Helsinki Institute for Information Technology

P.O. BOX 9800, 02015 HUT, Finland
mikko.valimaki@hut.fi, ville.oksanen@hiit.fi

Abstract. This paper describes the impact of different open source copyright licensing arrangements to the competition in microcomputer operating system markets. We compare the historical evolution of the licensing policies of Microsoft Windows, Apple OS X and GNU/Linux operating systems. We argue that open source and free software licensing has been one of the most important changing factors in the microcomputer operating system markets in the recent years. We have seen both new entrants in the relatively closed markets and renewed business models by incumbents. However, there has been no single open source strategy but merely all market players have adopted open source into their operating system strategy in one form or other.

1. Introduction

Operating system software separated as a product from hardware in the 1970s. Following IBM's unbundling decision in 1969, software products markets started to emerge and there were soon different optional operating systems available. First popular interoperable operating systems were Unix systems started at AT&T in 1971. However, in the 1970s and 1980s Unix systems were not offered for microcomputers. First microcomputers in the late 1970s carried their own proprietary systems. (Campbell-Kelly, 2003)

In the early 1980s, there were basically two types on operating system markets on microcomputers: those controlled by hardware manufacturer such as Apple, Commodore and Atari, and new uncontrolled markets based on open PC hardware standard. Open hardware on PC meant that there was in principle no single manufacturer control on operating system software. Soon, however, de facto operating system standard on PC hardware became Microsoft, first with Microsoft Disk Operating System (MS-DOS) and later with Microsoft Windows.

With computing evolving into new directions, Microsoft's operating system monopoly on PC hardware became challenged from time to time with the introduc-

tion of e.g. graphical user interfaces and networking features. But Microsoft resisted.

In the late 1990s, Unix compatible systems were finally offered for PCs. Linux and different Berkeley Software Distribution (BSD) variants combined with GNU system software started to gain popularity on cheap PC hardware and Internet servers. According to West and Dedrick, this new major trend towards open source was influenced by a need for cheap and free Unix implementations, the rise of different philosophy towards software ownership, and the rise of Internet as a new development and marketing platform (West and Dedrick, 2001). Also proprietary mainframe and workstation Unixes were scaled down to run on cheaper hardware.

In the early 2000s, the operating system competition on the marketplace is again different. Microsoft Windows is still on the lead but competing especially on desktop markets with the new Apple OS X based on open source Unix compatible kernel. On server markets, new competition comes from a variation of highly developed free Unix implementations (GNU/Linux in the lead and different variants of BSD also available). Market leader Microsoft has been challenged on both fronts by either fully or partly open source software products.

This paper proceeds as follows. First we shortly overview the operating system markets on microcomputers as they stand today. Then, we discuss the historical development of different operating system licensing models and their implications to business model possibilities of Microsoft Windows, Apple OS X and GNU/Linux. We show that there has been no single open source strategy but merely all market players have adopted open source into their operating system strategy in one form or other. Finally, we conclude the paper arguing that free software and open source components have had a major impact on the micro-computer operating system markets in the recent years.

2. Market Overview

The table below illustrates main operating system options available in the early 2000s.

	Windows	OS X	GNU/Linux
Provider	Microsoft	Apple	Many vendors
License(s)	Proprietary	Proprietary / Free	GPL
Market share	95-98 %	1-3 %	0.2 - 0.4 %
Main income	Licenses	Hardware, licenses	Support, services
Development	Inhouse	Inhouse / open source	Open source / community
Standards	Proprietary / open	Proprietary / open	Open
Processor	One option (x86)	One option (PowerPC)	Various
Environment	Desktop, server	Desktop, server	Server, desktop
Other hardware	Open	Proprietary	Open

Table 1. Microcomputer operating system competition in the early 2000s.

A few notes of the table are in place. First, market share estimates are based on desktop use. GNU/Linux systems have not been that successful in personal computer desktop software so far. If one uses searches made on Google as an indicator, during June 2001 and October 2003, a steady 1% of all searches came from computers using Linux as the operating system. The market share of Mac OS has been around 3-4% while other non-Windows operating systems gaining another 4%. The rest of Google queries, that is over 90%, were made from computers running Microsoft Windows.¹

While these figures indicate operating system popularity on desktop computers, they are a poor indicator of the overall operating system market size consisting of both desktops and servers. For example, GNU/Linux is relatively more popular on server markets where system consulting and support generate large parts of the business. According to an IDC study, Microsoft's share of all revenue generating server shipments in 2002 was 55% while GNU/Linux had 23%.² These two were the only server operating systems with a growing market share. IDC also estimated that the total market revenue from servers was around \$18,6 billion. (IDC 2003)

Second, there is no single source of revenues. Microsoft can be perhaps best described as a traditional software company in the sense their main income comes from license sales. Apple, on their part, still builds on the hardware controlled operating system model: Os X runs only on Apple hardware and in order to run Os X, one needs to first buy an Apple computer. Finally, GNU/Linux systems are typically not sold separately at all but rather installed as part of larger computer investment. Revenues are generated indirectly from e.g. services and support. (Daffara et al 2000, Raymond 2001, Fink 2003)

3. Study Framework

3.1. Theoretical Approach

¹ Information collected from Google Zeitgeist, available at <http://www.google.com/press/zeitgeist/archive.html>. Google is currently the most popular search engine on the Internet. It handled in February 2003 around 250 million queries per day and according to web analytics company OneStat.com it has over 55% market share in search engines. See http://www.onestat.com/html/aboutus_pressbox21.html. Google Zeitgeist's information on operating system market shares is only indicative. OneStat reported on September 24 2003 that among web users Linux would have 0.5% share and Apple only 1.5% with over 97% going to Microsoft. OneStat's press release is available at http://www.onestat.com/html/aboutus_pressbox24.html

² According to IDC, "Only paid software shipments, whether included with hardware or shipped exclusive of hardware, are incorporated in the revenue shipment statistics."

In order to understand more profoundly the impact of open source and free software licensing on operating systems competition, the licensing models of the three major operating systems were analyzed in more detail. The study was made by first collecting operating system market information from trade literature, vendor websites and expert interviews available on the Internet. Then, this information was analyzed through relevant economic theory. The presentation of each operating system is divided in three parts: historical background, licensing model, and impact on competition.

In this paper, we follow the now popular industrial economists' approach analyzing software markets in the context of network industries. (Economides 1996) We assume that operating system software is used in computer systems consisting of different hardware and software components. The system may consist of separate hardware and software components and for the system to work the components must in some sense work together. Further, in operating system software industry, only few products typically dominate the market at a time. It may not be rational for users to change from one product path to another because of lost system compatibility or benefits of existing user network. High enough switching costs and the benefits of positive network effects may in practice lead to sub-optimal lock-in situations. (e.g. Shy 2001, Shapiro and Varian 1999).

Relevant theoretical research can be traced back to Katz and Shapiro (1985) discussing the nature of network effects and market power especially from the producer's perspective and Teece (1986) considering how also complementary producers may generate profits. A summary of early papers from the perspective of component compatibility and standardization can be found in David and Greenstein (1990). A modern overview of the network economics and standards approach as applied to the software industry is presented in e.g. Grindley (1995), Shapiro and Varian (1999), Shy (2001), and Messerschmitt and Szyperski (2003).

The selected approach has its limitations. First, as Westrap notes, there is currently oversupply of fine-grained economic characterizations and classifications of software markets. (Westrap 2003, p. 5) For example, Shy's textbook identifies four main attributes as compatibility and standards; network externalities; lock-in and switching costs; and economies of scale. (Shy 2001) Gottinger lists sixteen "strategic characteristics". (Gottinger 2003, pp. xv-xvi) Our problem is to judge which of those are the most relevant.

Second, there are sound and empirically backed arguments suggesting the general role and explanation power of the network economics approach to be limited in the context of software markets. These arguments stem mainly from the critics of the Microsoft's antitrust trial at the end of the 1990s. For example, Liebowitz and Margolis point out that while the theory may be solid, the implications and usefulness of the results are not valid in all market situations. (Liebow-

itz and Margolis 2001, Liebowitz 2002). Also, while many economists used network economics to explain why Microsoft’s dominant position increased social (monopoly) costs, some used the same theory to explain why that power was both technically evident and economically efficient (e.g. Davis et al, 2001).

3.2. Evaluation Criteria

The first question is the licensing and implied ownership structure of the operating system software. Three main options used are proprietary (one owns), copyleft (no one owns) and permissive licensing (everyone owns). GNU GPL is a typical copyleft license while BSD and Apple Public Source License are more permissive. An overview of licensing options used at the different level of the system is illustrated in the table 2 below. These licensing options and their implications are explained later in detail with each operating system

	Proprietary	Permissive	Copyleft
Kernel	Microsoft	Apple	Linux
User interface	Microsoft, Apple	Linux	
Applications	All	All	All

Table 2. Main licensing options in different logical levels of the operating system.

Second, we discuss different business model alternatives. Possibilities for operating system vendors to benefit from components strategy include bundled proprietary software such as user interface, hardware and operating system integration. It is also possible to get involved in the complimentary products and services markets including participation in open source communities. We assume business model possibilities depend on e.g. the licensing choice and whether the operating system is competing on desktop or server and enterprise markets.

Third, we discuss the competitive environment of each operating system. How system vendors have differentiated their product from others? What are their competitive advantages? We are especially interested how open source licensing has affected the competitive environment and has it been used as a competitive tool.

4. Evaluation of Different Business Models

4.1 Microsoft Windows

Historical background. The first version of Microsoft Disk Operating System (MS-DOS) was released with IBM PC back in 1981. The first version of Microsoft Windows – at first basically a graphical user interface extension to MS-DOS – was announced in 1983 and published in 1985. From the beginning, Microsoft’s operating system has been sold as licensed software. It has been priced as a con-

sumer product being roughly 5-10% of the total price of a home PC system. (Computer Hope, 2003).

As graphical user interfaces became standardized Microsoft had a MS-DOS based edge on PC systems. Its many competitors including IBM's OS/2, Digital Research's Graphical Environment Manager (GEM), Berkeley Softworks' Ge-oWorks (GEOS) disappeared by the 1990s. While the competitors were perhaps technologically more advanced at the time compared to Windows they were more or less incompatible with MS-DOS programs. IBM's OS/2 offered the toughest competition. It was in the beginning supported by Microsoft but it never gained enough popularity to attract application developers, which were already developing for the growing Windows markets.

Licensing model and open source. Microsoft's licensing model has developed during time. Focus has been largely on pricing strategies and price discrimination. The basic idea has been to tie price with usage restrictions based on e.g. user group (students, home users) or computer (OEM sales). (Clemens and Szypenski, 2003) With the increasing commoditization of software Microsoft has bundled other office, Internet and multimedia application to its operating system. From economic perspective, Microsoft has been a text-book example of how to use network effects to create strong vendor lock-in and monopoly power. (cf. Liebowitz and Margolis, 2001). No wonder, antitrust officials have followed closely on Microsoft's licensing and bundling practices and the company has been the target of several unsuccessful suits. (US vs. Microsoft, 2003)

Recently, the company has changed its enterprise licensing model effectively from non-perpetual licenses to annual renewal fees. This "Microsoft Software Assurance" faced considerable resistance from their clientele. The licensing introduced three-year licensing cycles, which was perceived to force the users to upgrade their systems more often than their need really is. (Microsoft, 2003c) The licensing was also seen as a way to strengthen the lock-in to Microsoft products.³

In addition, Microsoft has to some degree opened Windows source code to some of its biggest customers in its "Shared Source Initiative". They first opened parts of Windows source code for inspections under strict non-disclosure agreements. As of today, the terms have been further relaxed. For example, the Shared Source License for Microsoft ASP.NET Starter Kit would almost pass the open source definition – the users are allowed to create derivate works and distribute them as long as they keep the original license agreement intact. A significant restriction, however, remains that the users are not allowed the mix the source code with copyleft licensed code: (Microsoft, 2003b)

³ The original deadline for signing up for the program had to be extended, which was well documented for example in The Register:

<http://www.theregister.co.uk/content/archive/26477.html> and
<http://www.theregister.co.uk/content/4/26456.html>

“That you are not allowed to combine or distribute the Software with other software that is licensed under terms that seek to require that the Software (or any intellectual property in it) be provided in source code form, licensed to others to allow the creation or distribution of derivative works, or distributed without charge.”

Overall, Microsoft’s position on open source has changed considerably. Currently the company’s position on open source is still somewhat critical and dismissive but not totally hostile (Microsoft, 2003b):

“...The main benefit of the OSS model is that it allows any programmer to advance the ideas of the original developer, and global “communities” of programmers do emerge to contribute to major OSS projects. Another obvious benefit is that there is little or no cost in obtaining OSS software, although training, service, and support costs may be higher over the life of the software. The principal drawback of OSS is that no single entity can be held responsible for individual contributions of a far-flung army of unrelated programmers. There also is the possibility that one version of an OSS program will not work properly, or at all, with other versions. In addition, it is not clear that the OSS model can sustain software companies over the long term.”

However, Microsoft still makes a strict difference between GNU GPL and permissive licenses. The company has used considerable resources to lobby against the use of GNU GPL in governments and publicly funded research arguing that GPL threatens the “software ecosystem”. (Microsoft, 2003a) As long as open source supports and complements Microsoft’s product business and does not threaten the licensing business of its core intellectual property, this seems to be a rational policy in accordance with economic theory.

Impact on competition. Because of its current dominant market position (especially on desktop), it is reasonable to assume that Microsoft can only lose market share. Thus its competitive moves on operating system markets are more or less defensive. For example, the adoption of limited shared source programs is clearly aimed to undermine the threat presented by open source. One problem with these initiatives is that Microsoft can hardly exclude anyone from using their technology by placing membership in shared source as a condition to enter Windows markets.

Microsoft has also used other strategies. The infamous “Halloween document” – a leaked Microsoft’s internal memo on open source – draws an interesting picture of their options. First, Microsoft can use its market power to divert standards to include proprietary components. This behavior is sometimes called as “extend and embrace”. One well-documented example of this strategy is Kerberos-protocol, to which Microsoft made subtle changes to prevent Unix-servers to

interoperate with Windows-clients. (Livingston, 2000) Of course, this strategy assumes that Microsoft currently controls some essential operating system component that can be changed without notifying competitors.

Second possible strategy is aggressive use of intellectual property rights and especially software patents. However, it is worth noting that companies with large patent portfolios including IBM, which has the biggest patent portfolio of all, have now major business interest in GNU/Linux. They could and most likely would counter any move from Microsoft against GNU/Linux with similar or bigger charge of patent violations.

Microsoft's has also invested in trusted computing (TC).⁴ TC protects files with hardware authentication thus making it practically impossible to create interoperable programs without Microsoft's permission. Consequently, if the users adapt proprietary operating system specific architecture such as file formats, which are based on TC, they also lock themselves to the vendor stronger than ever before.

Ironically these actions against open source may end up being counter-beneficial to Microsoft. The users do not want to lock themselves up to one vendor if there is an alternative, because they realize the risks and there are also other operating system networks available. So by positioning itself better to block open source competitors from the market, Microsoft makes at the same its products less appealing to its customers, which may lead to reduced market share.

4.2 Apple OS X

Historical background. Apple Macintosh had a strong position in microcomputer markets in the 1980s. However, in the 1990s company run into troubles and much was due to its ancient operating system technology. After reviewing the possibilities, Apple acquired NeXT, a company headed by Apple's co-founder Steve Jobs. A completely new operating system called Mac OS X was developed with Jobs' lead in the late 1990s. (West 2003)

In short, Apple OS X can be described as a further developed version of NeXT's NeXTStep operating system with new graphical user interface and compatibility with old Mac software. Since OS X's kernel and other important parts – following NeXTStep's design – are largely based on BSD Unix and are open source, a brief note on BSD's history is in place here.

⁴ The company uses currently term Trustworthy Computing. At the other extreme of the naming game, Free software Foundation prefers term Treacherous Computing. In this article the same idea will be referred to "neutrally" as TC

In academic environment software had been for a long time developed with the principles of open source code and free distribution. Many universities chose to use AT&T's Unix operating system in the 1970s. It was licensed from the beginning to educational institutions with full source code under a trade secret agreement. Users were then encouraged to develop the system further.

A major variant of AT&T's Unix called Berkeley Software Distribution (BSD) came from the University of California at Berkeley. In fact, much of the Unix development happened in BSD.⁵ First fully copyright-independent BSD releases were distributed under the liberal BSD license terms. Berkeley computer scientist Marshall Kirk McKusick later explained (McKusick 1999):

"The BSD originated networking code and supporting utilities were released in June 1989 as Networking Release 1, the first freely-redistributable code from Berkeley. The licensing terms were liberal. A licensee could release the code modified or unmodified in source or binary form with no accounting or royalties to Berkeley."

BSD itself has never been that popular operating system but its legacy lives on. In 1993, BSD was ported to cheap Intel hardware by William Jolitz and later three separate projects – FreeBSD, NetBSD and OpenBSD – were founded to continue BSD development. However, Apple's Mac OS X is by all accounts the first popular BSD implementation for microcomputers. (Howard, 2001)

Licensing model and open source. The user interface and many other system tools related to OS X technology are proprietary. Thus, the whole OS X operating system package is licensed much the same way as Microsoft Windows. There are no licensing requirements for applications running on Mac OS X. A traditional Mac software culture has perhaps favored proprietary licensing such as shareware for hobbyist programs. However, OS X has provided access to the system for many GNU/Linux open source programmers.

Because of the BSD license Apple has been able to also re-license everything in Mac OS X kernel with its own terms. The kernel, called Darwin, uses Apple Public Source License, which is accepted as free software license by the Free Software Foundation. Their approach has been to keep the source code of the kernel open and have good contacts with the open source community. For example, Apple has hired key developers including Jordan Hubbard, who is one FreeBSD founders and was a long time a core member of the development group. At the moment, Apple says it uses FreeBSD as a reference operating system. Apple also

⁵ According to Kirk McKusick, over 50% of Unix System V Release 4 was derived from BSD including file system and networking code. This information is from email exchange with mr. McKusick 12.10.2003

makes its modifications publicly available for the community even if there is no such license requirement. According to Darwin FAQ, the reason is: (Apple, 2003)

“Although the BSD licenses don’t require companies to post their sources, divergent code bases are very hard to maintain. We believe that the open source model is the most effective form of development for certain types of software. By pooling our expertise with the open source development community, we expect to improve the quality, performance, and feature set of our software.”

Impact on competition. Theoretically Apple’s approach takes the best of all possible worlds. The open sourced Darwin and good community relations allowed Apple to leverage off the large community networks around different BSD variants. Good interoperability with Unix allowed also easy porting from rich collections of server applications. Open source code has helped hardware vendors to port their drivers to Apple. Also, Apple has worked in collaboration with Microsoft and most of the proprietary Microsoft-controlled standards are supported in OS X (file formats, Internet extensions). Finally, and perhaps also most importantly, Apple held control over the OS X user interface, which allowed it to differentiate the product from the rest of the Unix variants. (West, 2003).

Unfortunately, the market realities have somewhat spoiled Apple’s approach. Even if Darwin has been ported to Intel-based hardware, without user interface it is more or less useless. This initially limited the developer base to the core Macintosh users, which had already the hardware from Apple. The migration of open source developers to OS X has taken years. (Singer, 2004) This most likely also explains why Darwin has not gained similar momentum as Linux did. It is also important to note that Apple cannot benefit directly from the Linux-development because it is not possible to attach GPL’ed operating system source code to Darwin.

4.3 GNU/Linux Distributions

Historical background. The history of both GNU and Linux are by now well known and easily available on the Internet. Let’s summarize the history from licensing perspective shortly. First, there was the GNU project. Richard Stallman started it publishing GNU manifesto in 1983 and then founding the Free Software Foundation. While GNU manifesto had a political and ideological tone, the project was at first technologically focused. The aim was no more or less than to write a complete operating system. But it didn’t take long for licenses to get into the picture.

The first software Stallman worked on was Emacs text editor. After copyright problems with source code he used from a third party, Stallman coined the idea of copyleft. (Stallman 1986) Copyleft was for the first time implemented in Emacs General Public License (GPL) in 1988. It held that GNU Emacs was not public

domain but under copyright. It was free to copy and distribute but it wasn't allowed to change the license terms in any derivative work. It is worth noting that before Emacs GPL Free Software Foundation didn't use any license for their software and Stallman appeared to be opposing copyrighting software.

With an innovative license Stallman was able to go against copyright with the help of copyright itself. In 1989 Emacs GPL license text was partly rewritten for clarity and the license was renamed to GNU General Public License. It became the default license for all GNU programs. The second version of GNU GPL was published in 1991 and the third version is currently in preparation.

Meanwhile, Linus Torvalds had started to develop his new operating system in 1991. Linux was the first Unix implementation targeted for microcomputers. In 1992, Torvalds decided to license Linux under GNU GPL and subsequently all source code written to the Linux kernel by numerous contributors was under that license. (Torvalds and Diamond, 2001, p. 96) Most system tools of Linux were taken from GNU project and other sources including BSD. Linux itself became the kernel of the new operating system.

Later, Free Software Foundation and Stallman wanted the full operating system based on main components from the GNU project to be called GNU/Linux. While perhaps technically applicable, trade press has so far neglected Stallman's desire. In this article, however, we have followed Stallman's recommendation in order to emphasize that an operating system consists of different components (kernel, user interface, system tools etc.).

Licensing model and open source. Most software in GNU/Linux distributions are under GNU GPL license. The license states that the software must be free to copy, distribute and modify. Also, the source code must be open and available practically free of charge. It is not allowed to use the GNU GPL copyrighted code in derivative works (as defined in law) under any other license but GNU GPL. The interpretation of the last mentioned term is unfortunately quite open.

GNU/Linux includes also essential system software licensed with other terms. Free Software Foundation has for example used other license terms for their programming language compilers. The idea is that any new program created in GNU/Linux – being perhaps technically speaking a derivative of the compiler – is not automatically under GNU GPL. The current interpretation of GNU GPL and derivative works means that proprietary applications and system tools can be developed, run and even sold with proprietary licenses for Linux. However, the rough rule is that it must be possible to run any such application separately (using only dynamic runtime-calls) from other program including the kernel. (Free Software Foundation, 2003)

A major part of GNU/Linux is X11 windowing system based user interfaces. X11 is licensed with liberal BSD-like terms but the most popular graphical user interfaces including KDE and Gnome are under GNU GPL. Major problems in their development have been usability issues as well as the lack of open source GUI code. Developing a major free software operating system component from scratch has proven to be a time consuming project.

Impact on competition. In a way, GNU/Linux is not a ready-to-run product. It is a set of commodity tools and components that can be used as such and tailored for different uses. (e.g. Fink 2003). There are numerous companies distributing or otherwise using GNU/Linux under different business models such as Red Hat and SuSe for desktops and servers, and MontaVista for embedded systems. Large software companies including IBM, Oracle and Sun support Linux development and use it to sell their proprietary enterprise application software running on Linux. (IBM, 2003 and Sun, 2003) For example, Sun states on their website:

“Seven Sun ONE products are available on Linux today - Application server, Directory server, Web server, Active Server Pages, Studio, Grid Engine and Message Queue - with plans to deliver more in the near future.”

To conclude, GNU/Linux can be characterized as an open network providing opportunities to companies that sell related and interoperable products or services to the network members. Therefore, it is perhaps more appropriate not to take the GNU/Linux as a direct competitor to Microsoft Windows or Apple Mac OS X. First, GNU/Linux distributions have a minimal market share on desktop users. In addition, Microsoft has been hesitant to support Linux and all kinds of interoperability problems with the dominant operating system have made consumer markets difficult. Second, on servers a large market share is captured by enterprise application providers, which use GNU/Linux as an independent and cheap platform for their applications. These applications can be also run on different operating system's including provider's own (e.g. IBM AIX and Sun Solaris).

5. Concluding Remarks

From historical perspective, it seems clear that open source and free software have changed fundamentally the operating system software markets in the recent years. The fact is that today the main operating system alternatives are either fully or partly open source UNIXes or Microsoft. Ten years ago, Microsoft had gained practical monopoly on microcomputer operating systems with only Apple Macintosh barely hanging along. As of today, both Apple and the new entrant GNU/Linux hold stronger position largely thanks to open source licensing policy, development methods and the rapid growth of Internet and networked computing.

However, there seems to be no “one-size-fits-all” open source operating system strategy. The open community model guaranteed fast growth for GNU/Linux. Anyone can download, modify, and distribute the source code of the whole operating system free of charge. Anyone can develop new complementing components to the system without restrictions. To compare, Apple’s Mac OS X has only an open source kernel with in-house development model controlled by Apple. Other parts of the operating system including the user interface are proprietary and the whole system is licensed for fee. Finally, Microsoft’s reply to open source has been their much debated shared source initiative. Significant institutional users have been granted under specific shared source agreements a limited access to view parts of Windows source code. While their means to exclude competitors from Windows technology is limited – especially under the on-going antitrust investigations – Microsoft basically still believes in the fully proprietary development and licensing model.

From economical point of view, an open platform is usually socially preferable to a closed one. However, individual companies may generate better profits from closed systems and may therefore have incentives to such system design and licensing choices we have seen with Apple and Microsoft. Since competition among closed systems is not a perfect substitute for competition among components on an open system, each company in the software industry must look at its own specific competencies when making operating system choices. (Farrell et al 1998)

To conclude, it is difficult to draw a comprehensive picture on how open source and free software actually work as competitive tools. Much remains for further study. The little data we have presented in this article can be perhaps summarized as follows:

- Open source code and free software have proved to be powerful ways to standardize and stabilize new operating system technology and compete against established market powers
- On desktop markets, their impact has been limited mainly because of compatibility and usability issues (strong lock-in to incumbent operating system vendors)
- On server and enterprise application markets, they have had more changing impact based on the benefits of standardized independent technology and other technical features (weaker lock-in to incumbents)
- Proprietary components are still a major competitive factor and there remains large areas in both desktops (e.g. user interface) and servers (e.g. enterprise applications) without major open source impact from where proprietary vendors can generate revenue

References

- Apple (2003): Darwin FAQ, available at <http://developer.apple.com/darwin/projects/darwin/faq.html>, visited November 28, 2003
- Campbell-Kelly, Martin (2003): *From Airline Reservations to Sonic the Hedgehog. A History of the Software Industry*. MIT Press.
- Computer Hope (2003): "History of Microsoft Windows", available at <http://www.computerhope.com/history/windows.htm>, visited November 28, 2003
- Daffara, Carlo and Carlo González-Barahona, Jesús M. (ed.) (2000): *Free Software / Open Source: Information Society Opportunities for Europe?*, Working Group on Libre Software, Version 1.2., available at <http://eu.conecta.it/paper/paper.html>, visited April 5, 2004
- David, Paul A. and Greenstein, Shane (1990): "The Economics of Compatibility Standards: An Introduction to Recent Research", *Economics of Innovation and New Technology*, Volume 1, Issue 1, pp. 3-41.
- Davis, Steven J. and MacCrisken, Jack and Murphy, Kevin M. (2001): "Economic Perspectives on Software Design: PC Operating Systems and Platforms", National Bureau of Economic Research Working Paper Series, paper #8411, available at <http://www.nber.org/papers/w8141>, visited April 5, 2004
- Economides, Nicholas (1996): "The Economics of Networks", *International Journal of Industrial Organization*, Volume 14, Number 6, pp. 673-700.
- Farrell, Joseph and Monroe, Hunter K. and Saloner, Garth (1998): "The Vertical Organization of Industry: Systems Competition versus Components Competition", *Journal of Economics & Management Strategy*, Volume 7, Number 2, pp. 143-182.
- Fink, Martin (2003): *The Business and Economics of Linux and Open Source*. Prentice Hall.
- Free Software Foundation (2003): Frequently Asked Questions about the GNU GPL, available at <http://www.gnu.org/licenses/gpl-faq.html>, visited November 28, 2003
- Gottinger, Hans-Werner (2003): *Economies of Network Industries*. Routledge.
- Grindley, Peter (1995): *Standards, Strategy, and Policy*. Oxford University Press.
- Howard, James (2001): "The BSD Family Tree", *Daemon News*, available at http://www.daemonnews.org/200104/bsd_family.html, visited November 28, 2003.
- Hubbard, Jordan (2003): "A Brief History of FreeBSD", available at http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/history.html, visited November 28, 2003
- IBM (2003): Linux at IBM, available at <http://www.ibm.com/linux/>, visited November 28, 2003
- IDC (2003): "IDC Says Microsoft Is Moving into Dominant Role in Server Operating Environments, Even as Linux Grows", IDC press release, 8th October 2003.
- Katz, Michael and Shapiro, Carl (1985), "Network Externalities, Competition and Compatibility," *American Economic Review*, Volume 75, Issue 3, pp. 424-440.
- Liebowitz, Stan J. and Margolis, Stephen E. (2001): *Winners, Losers & Microsoft. Competition and Antitrust in High Technology*. Revised edition. The Independent Institute.
- Liebowitz, Stan J. (2002): *Re-Thinking the Network Economy*. Amacom.
- Livingston, Brad (2000): "Is Microsoft's change in Kerberos security a form of 'embrace, extend, extinguish'?", *InfoWorld*, available at <http://archive.infoworld.com/articles/op/xml/00/05/15/000515oplivingston.xml>, visited November 28, 2003
- McKusick, Marshall Kirk (1999): "Twenty Years of Berkeley Unix. From AT&T-Owned to Freely Redistributable", in DiBona, Chris et al (eds.) (1999): *Open Sources: Voices from the Open Source Revolution*, O'Reilly, pp. 31-46.

Messerschmitt, David G. and Szyperski, Clemens (2003): *Software Ecosystem. Understanding an Undispensable Technology and Industry*. MIT Press.

Microsoft (2003a): Microsoft Shared Source Initiative Frequently Asked Questions, available at <http://www.microsoft.com/resources/sharedsource/Initiative/FAQ.aspx>, visited November 28, 2003

Microsoft (2003b): Shared Source Licensing Programs, available at <http://www.microsoft.com/resources/sharedsource/Licensing/default.aspx>, visited November 28, 2003

Microsoft (2003c): Software Assurance, available at <http://www.microsoft.com/licensing/programs/sa/>, visited November 28, 2003

Raymond, Eric S. (2001): *The Cathedral and the Bazaar*, 2nd Edition, O'Reilly.

Shapiro, Carl and Varian, Hal R. (1998): *Information Rules. A Strategic Guide to the Network Economy*. Harvard Business School Press.

Shy, Oz (2001): *The Economics of Network Industries*. Cambridge University Press.

Singer, Michael (2004): "Apple Sees a Shift in Developer Profiles", *InternetNews*, available at <http://www.internetnews.com/ent-news/article.php/3335591>, visited April 5, 2004

Stallman, Richard (1986): Lecture at Kungliga Tekniska Högskolan (Royal Institute of Technology), Stockholm, Sweden, 30th October 1986, available at <http://www.gnu.org/philosophy/stallman-kth.html>, visited 28 November 2003.

Sun (2003): Linux from Sun, available at <http://www.sun.com/software/linux/>, visited November 28, 2003.

Teece, David J. (1986): "Profiting from technological innovation: Implications for integration, collaboration, licensing and public policy", *Research Policy*, Volume 15, Issue 6, pp. 285-305.

Torvalds, Linus, and Diamond, David (2001): *Just for Fun. The Story of an Accidental Revolutionary*. Harper Business.

US vs. Microsoft (2003): United States v. Microsoft – Current Case, available at http://www.usdoj.gov/atr/cases/ms_index.htm, visited November 28, 2003

von Westrap, Falk (2003): *Modeling Software Markets. Empirical Analysis, Network Simulations and Marketing Implications*. Physica-Verlag.

West, Joel and Dedrick, Jaseon (2001): "Open Source Standardization: The Rise of Linux in the Network Era," *Knowledge, Technology & Policy*, volume 14, issue 2, pp. 88-112.

West, Joel (2003): "How open is open enough? Melding proprietary and open source platform strategies", *Research Policy*, Volume 32, pp. 1259–1285.