

The Impact of Free and Open Source Components on the Competition in Microcomputer Operating System Software Markets

*Mikko Välimäki and Ville Oksanen
Helsinki Institute for Information Technology
P.O. BOX 9800, 02015 HUT, Finland
[mikko.valimaki, 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

In the early 1980s, there were basically two types on operating system markets on microcomputers: (1) those controlled by hardware manufacturer such as Apple, Commodore and Atari, and (2) uncontrolled markets based on 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.

In the 1990s 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 Unix operating

systems were scaled down to run on cheaper hardware. Most proprietary microcomputer operating systems died including Atari and Commodore.

In the early 2000s, 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 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). Interestingly, when basic hardware and software components have become commoditized, the competition has focused on new factors. Instead of market dominance and features, usability has become more important desktops. Reliability, costs and vendor independence through open source code are important factors on server and enterprise system markets. 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 describe operating system markets from historical perspective. 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 microcomputer operating system markets in the recent years.

2. Market Overview

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. (Grad 2002, Humprey 2002). 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.

Mass-market microcomputer industry started in the late 1970s. In the early 1980s IBM entered the new markets by introducing its PC with open hardware standards and free competition on both system and application software. Operating system was, however, a de facto standard component provided by

Microsoft. With computing evolving into new directions, Microsoft's operating system monopoly on PC hardware became challenged from time to time with e.g. graphical user interfaces in the 1980s and networking features in the 1990s. Especially Linux with GNU system tools running powerful Internet software was seen as a direct threat to Microsoft's operating system monopoly in the late 1990s. Also proprietary mainframe and workstation operating systems were scaled down to run on cheaper hardware. At the other extreme, computing was directed towards handheld devices where main competitors of Microsoft were Symbian OS and Palm OS.¹

In the early 2000s the main options in microcomputer operating systems were Microsoft Windows, Apple OS X and GNU/Linux on personal computers. In this study we ask how open source and free software licensing has shaped the structure of the microcomputer operating system markets as it is today. 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.²

¹ Both mainframes and handheld operating systems are not discussed further in this paper.

² 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

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, Microsoft's share of all revenue generating server shipments in 2002 was 55% while GNU/Linux had 23%. 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.

Second, there is no single open source development model. Linux development is open in the sense that principally anyone can download the source code and follow closely how it grows. It is also possible for anyone to submit new source code or patches to be included (after a specific process) in the source tree. We can therefore characterize Linux development as "community" open source. To compare, Apple's Mac OS X has an open source kernel (based on BSD source code) with a development model controlled by Apple. Also, only the kernel is open source and under a free license; other parts of the operating system including the user interface are proprietary. Hence, we call Apple's model here "inhouse" open source. 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.

Third, the approach to standardization is subject to further definitions. Here, we understand open standards as such where anyone can develop interoperable hardware or software without explicit permission. In other words, there can't be any license fees over an open standard. So-called reasonable and non-discriminatory (RAND) licensing terms are thus essentially proprietary while royalty-free terms are defined as open. Historically, Microsoft Windows and Apple OS X have used proprietary standards in many instances such as web and multimedia features while many GNU/Linux distributions are build only on open standards.

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

Finally, hardware environment is also subject to fine-grained definition. It is difficult to define a desktop microcomputer based on the physical appearance. One should include laptops in the desktop computer category but PDAs and mobile phones not. On the upper extreme, most web and workgroup servers can be counted as microcomputers but scientific and corporate mainframes not.

3. Study Framework

3.1. Approach

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 collecting information from relevant literature, vendor websites and referring to expert interviews available on the Internet. The presentation of each operating system is divided in three parts: historical background, licensing model, and impact on competition.

Main theoretical arguments are based on the economics of complementary products and services. The assumption is that operating system software is used in computer systems consisting of different hardware and software components. Software is never used in isolation but as part of a system. The system may consist of separate hardware and software components and for the system to work the components must in some sense work together. In other words, they must be compatible with each other. 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. (Shy 2001, Shapiro and Varian 1999).

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 question is whether the system is competing on desktop or server and enterprise markets. Instead of market dominance and features, usability has become more important desktops. Reliability, costs and vendor independence through open source code are important factors on server markets.

Third, we discuss different business model alternatives based on the components approach. Possibilities for operating system vendors to benefit from components strategy include bundled proprietary software such as user interface, hardware and operating system integration and involvement in the complimentary products and services markets including participation in open source communities.

4. Evaluation of Different Business Models

4.1 Microsoft Windows

Historical background. The first version of Microsoft Windows was announced in 1983 and published in 1985. It was at first stage mainly a graphical user interface extension to Microsoft Disk Operating System (MS-DOS), which run on PC hardware. - The first version of MS-DOS was released with IBM PC back in 1981.

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' GeoWorks (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 has sold from the start proprietary licenses to binary versions of Windows. It has been priced as a consumer product being roughly 5-10% of the total price of a home PC system. (Computer Hope, 2003). The 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). With the increasing commoditization of software Microsoft has bundled other office, Internet and multimedia application to its operating system. 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 its customers. Bigger customers and especially governments are hesitant to use products, whose source code they cannot inspect for security bugs and possible backdoors. Therefore, Microsoft introduced "Shared Source Initiative", which opened the Windows source code for inspections under strict non-disclosure agreements. Unlike in normal open source, the users were not allowed to make any changes to the source code or use it in their own products. (Microsoft, 2003b)

Microsoft has subsequently gradually relaxed the licensing terms of at least part of products under the Shared Source Initiative. 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,

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>

however, is that the users are not allowed to mix the source code with copyleft licensed code: (Microsoft, 2003b)

“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 license 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)

Impact on competition. Because of its current dominant market position (especially on desktop), Microsoft can only lose market share. Thus its moves 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.

Microsoft has also used other strategies. The infamous “Halloween document” – a leaked Microsoft’s internal memo on open source – draws an interesting overall 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)

Second possible strategy is aggressive use of intellectual property rights and especially software patents. Traditional open source companies were very reluctant to apply for patents because of practical (too expensive) and philosophical (prevents sharing) objections. The market realities have forced the companies to adapt to try to counter this threat and thus companies such as Red Hat have started to build their own patent portfolios. (Red Hat 2003) It is also 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. So by positioning itself better to block the open source from the market Microsoft makes at the same its products less appealing to its customers, which may lead the 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 trouble 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)

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 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 Apple OS X's kernel and other important parts – following NeXTStep's design – are largely based on BSD and are open source, a brief note on BSD history is in place here.

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. Much of the Unix development in fact happened in BSD, which was funded generously by Defense Advanced Research Projects Agency (DARPA) in the United States.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 a 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

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

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 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 it to leverage off the large communities 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 has limited the developer base to the core Macintosh users, which had already the hardware from Apple. 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 software 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 GNU. Richard Stallman, a former staff member at MIT computer lab, started his GNU project 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. 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 follow Stallman's

recommendation in order to emphasize difference components of an operating system (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. 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.”

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 user. 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 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 components 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 Unices 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 new entrant GNU/Linux hold stronger position largely thanks to open source licensing methods and the rapid growth of Internet and networked computing.

However, how open source actually works as a competitive tool is more difficult to understand. Much remains for further study especially on the history of the industry structure. The little data we have presented in this article can be summarized as follows:

- Open source and free software code have proved to be powerful ways to standardize and stabilize operating system technology
- On desktop markets the impact of open source code and free software has been limited mainly because of compatibility and usability issues
- On server and enterprise application markets open source and free software have had more changing impact based on the benefits of standardized independent technology, stability and security

- 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

References

- Apple (2003): Darwin FAQ, available at <http://developer.apple.com/darwin/projects/darwin/faq.html>, visited November 28, 2003
- Computer Hope (2003): "History of Microsoft Windows", available at <http://www.computerhope.com/history/windows.htm>, visited November 28, 2003
- 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
- Grad, Burton (2002): "A Personal Recollection: IBM's Unbundling of Software and Services", *IEEE Annals of the History of Computing*, Volume 24, Issue 1, pp. 64-71.
- 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
- Humprey, Watts S. (2002): "Software unbundling: a personal perspective", *IEEE Annals of the History of Computing*, Volume 24, Issue 1, pp. 59-63.
- IBM (2003): Linux at IBM, available at <http://www.ibm.com/linux/>, visited November 28, 2003
- 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.
- Microsoft (2003a): Microsoft Shared Source Initiative Frequently Asked Questions, available at <http://www.microsoft.com/resources/sharedsource/Initiative/FAQ.mspix>, visited November 28, 2003
- Microsoft (2003b): Shared Source Licensing Programs, available at <http://www.microsoft.com/resources/sharedsource/Licensing/default.mspix>, visited November 28, 2003
- Microsoft (2003c): Software Assurance, available at <http://www.microsoft.com/licensing/programs/sa/>, visited November 28, 2003
- Red Hat (2003): Red Hat, Inc. Statement of Position and Our Promise on Software Patents, available at http://www.redhat.com/legal/patent_policy.html, visited November 28, 2003
- 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.
- 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.

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

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.