Proprietary vs. Open Standards in the Network Era: An Examination of the Linux Phenomenon

Joel West and Jason Dedrick
UC Irvine, Center for Research on Information Technologies and Organizations (CRITO)
joelwest@ieee.org and jdedrick@uci.edu

Abstract: For networked I.T. industries, standards adoption is a key prerequisite for attracting complementary assets. Producer firms that hope to profit from their standards success must trade off control of the standard against the imperative for adoption.

Moschella outlines three eras of modern computing: systems, personal computers and network, each with its own form of standards competition. During the systems era of computing, mainframe producers maximized their control by offering vertically integrated standards architectures. In the PC era, IBM unintentionally surrendered control to two key suppliers in its haste to launch the IBM PC and maximize its adoption. Microsoft and Intel in turn sought pervasive adoption of their technologies by appropriating only a single layer of the standards architecture and publishing a subset of the interfaces to other layers.

In reaction to these proprietary strategies, the open source movement developed software that relinquishes control in favor of adoption. Such free software has played an important role in Internet infrastructure, and its adherents argue that it will supplant such proprietary standards in the network era.

This study examines the rise of the Linux operating system, with particular focus on its role as a PC server operating system in competition with the established Microsoft Windows OS. While Linux has its origins in the 1984 GNU Project, and was widely available beginning in 1993, we focus on the adoption motivations of organizational buyers and suppliers of complementary assets during the period 1995-1999.

1 Introduction

Vertically integrated proprietary standards architectures were the norm for the first three decades of the postwar computer industry. Each computer maker developed most if not all of its technology internally, and sold that technology only as part of an integrated computer system. This systems era was ascendant from IBM’s 1964 introduction of its System 360 until the 1981 release of IBM’s first personal computer [20].

This strategy was challenged by two different approaches. One was the fragmentation of proprietary standards in the PC industry between different suppliers, which led firms like Microsoft and Intel to seek industrywide dominance for their proprietary component of the overall system architecture, marking what Moschella [20] terms the “PC era” (1964-1981). The second was a movement by users and second-tier producers to create industrywide “open” systems, in which the standard was not owned by a single firm.

The explosive adoption of the Linux operating system in the late 1990s was a response to these earlier approaches. Linux was the most commercially successful example of a new wave of “open source” software, in which the software and even source code are freely distributed to use and modify. At the same time, its adherents emphasized its advantages in contrast to the proprietary PC standards, particularly software standards controlled by Microsoft.

In the next section, we examine the evolution of standards competition strategies in the computer industry from the proprietary integrated systems to both horizontal specialization and open systems. We then trace the birth and growth of Linux and related open source systems, and then review rival explanations for their recent successes. Finally, we conclude with an analysis of the durability of the Linux phenomenon in the network era.

2. Era of proprietary standards competition

Product compatibility standards have typically been considered using a simple unidimensional typology, bifurcated between “compatible” and “incompatible.” However, to illuminate differences between proprietary and open standards strategies, we use Gabel’s [6] multi-dimensional classification attribute, with each dimension assuming one of several (discrete) levels:

- “multivintage” compatibility between successive generations of a product:
- “product-line” compatibility, providing interoperability across the breadth of the company’s product line — as Microsoft does (to at least some extent) with its “Windows everywhere” initiative.
- “multivendor” compatibility, i.e. compatibility of products between competing producers.

2.1 Vertically integrated proprietary systems

During the era of vertically integrated computer systems, the most successful architecture was IBM’s System 360 (later 370 and 390) line of mainframe computers. The 360 marked IBM’s switch from being a major buyer of electronic components to one of the largest

0-7695-0981-9/01 $10.00 (c) 2001 IEEE

manufacturers, all for internal use [28]. IBM produced most of the hardware — including electronic components, CPU board and peripherals — as well as the operating system, tools and much of the application software. Outside IBM, its mainframe and minicomputer competitors from 1950-1980 also sought vertical integration, although most lacked IBM's scope to achieve the same level of integration.

IBM and its System 360 were fantastically successful in this strategy, at one point garnering nearing half of the computer industry's profits [20]. A key reason was that the System 360 marked the first widespread implementation of what Gabel refers to as "product line compatibility" [6]. Unlike previous generations, IBM shared the same standards architecture across its product line — and thus enabling use of the same operating system, tools and user applications

But IBM was forced to limit its vertical integration in the face of a 1969 federal anti-trust lawsuit. To respond to one of the charges, that of illegal "tying" of its mainframe hardware and software, five months later IBM decided to end its decades-long practice of bundling its offerings, allowing companies to buy the hardware, software and services separately. As hoped by the government, this unbundling encouraged the third-party supply of peripherals, software and support

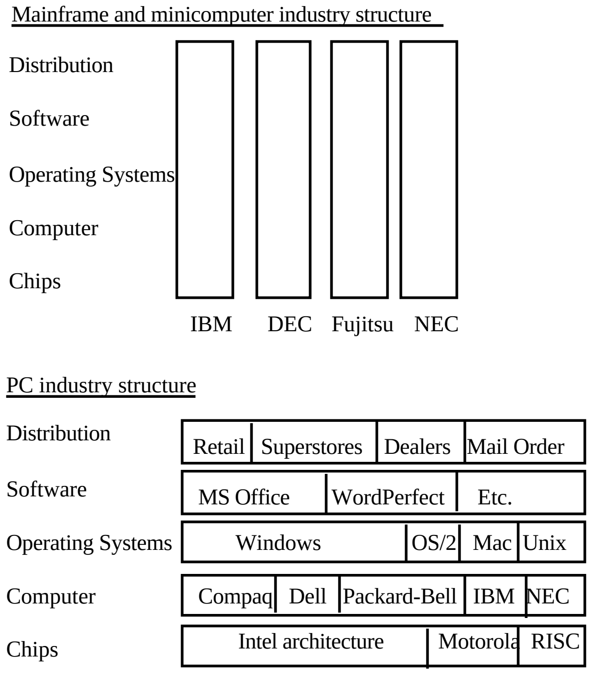
Still, many IBM customers bought a complete IBM solution. This helped IBM keep customers, because those customers who had custom-developed software would find it more expensive to adapt their software to a new processor, operating system or compiler.[1] As Greenstein's study of government procurement showed, such strategies helped discourage switching between vendors [9].

At the same time, IBM's vertical integration strategy had its limits as an exemplar for the rest of the industry. Without IBM's economies of scale and scope, rivals were eventually forced to choose between inferior performance on key dimensions, obtaining key components from outside suppliers, or exiting the computer business.

2.2 Horizontal specialization and quasi-monopolies

The first serious attack on the fully-integrated proprietary systems was against the integration rather than the proprietary nature. Ironically, IBM itself enabled this switch when, its haste to ship a personal computer, it purchased the processor and OS components from outside vendors. Its respective contracts with Intel and Microsoft enabled these firms to sell their components to rival systems vendors, enabling the IBM PC-compatible computer industry [4]; [20].

The net result was the personal computer industry developed with a standards environment that emphasized horizontal specialization in a given layer of the system, rather than vertical integration of internally developed components. In particular, each supplier had an incentive to obtain an industry-wide quasi-monopoly by selling its hardware or software to as many systems vendors as possible (Figure 1). By the mid-1990s it was clear that the horizontal proprietary model had overwhelmed the vertical one, at least as measured by overall industry spending.



Source: Adapted from [11]

Figure 1: Proprietary standards strategies in the computer industry

2.3 Competing through proprietary standards

Both the vertically integrated and horizontally segmented strategies used in the computer industry assumed proprietary (single-vendor) control of one or more standards in a systems architecture. The differences between these proprietary approaches (and with subsequent "open" standards) highlight the conflict between a firm's competing objectives in standards competition: adoption and appropriability.

Widespread adoption of the standard is important because the CPU, operating system and key application tools such as compilers and databases are more widely valued by users if they have a wide range of compatible software. Thus, to get the widest variety of software built upon their respective layer of the architecture, sponsors of a standard use pricing and other incentives to gain the largest number of early users and encourage the development of co-specialized software [30]; [19]; [27].

[1] Some CPUs were compatible with IBM's S/360 line, notably Amdahl (later Fujitsu) and NAS (later Hitachi). However, these apparently constituted a small fraction of IBM's sales, at least through the late 1970s [9].

At the same time, success of the standard provides no guarantee as to the sponsor's ability to profit from that standard. That ability will depend on intellectual property protection for the standard, and the relative importance of standard and complementary assets [30]. Control over the architecture enables the technological evolution of the standard and provides the incentive for the necessary investments [19]. However, in the increasingly common case of divided control of a standards architecture, control of the APIs determines access to the software and thus the right to profit from a standard [34].

So the sponsors of successful standards face competing incentives, as illustrated in the 1980s PC industry. They can gain wide adoption of their standard, at the risk of not profiting from its success (as IBM did). Or they can control the profit from the standard, at the risk of limited adoption and potential abandonment (as Apple did). The horizontal model adopted by Microsoft and Intel is an exemplar of what Morris and Ferguson termed "open but proprietary" standards — surrendering the appropriability of vertical integration in hopes of adoption as a universal layer in an industry-wide standards architecture [19].

3. Unix enables open systems

3.1 Unix and its bicoastal childhood

The first successful multi-vendor operating system was Unix, developed starting in 1969 by research group at Bell Telephone Laboratories in New Jersey. Forbidden by its 1956 consent decree from being in the computer business, AT&T did not sell the OS commercially. But after publishing influential research papers, Bell Labs licensed Unix to universities upon request, in source code form and without support [24].

The most important development was the licensing of Unix by the UC Berkeley Computer Science Department in 1974. The Berkeley group issued its own releases from 1977 to 1994, with much of its funding provided by the Defense Advanced Research Projects Agency (DARPA). The result of the Berkeley development included [7]; [24]:

- the first Unix version to support TCP/IP, later the standard protocols of the Internet;
- academic adoption of BSD Unix as the preferred OS by many computer science departments throughout the world;
- commercial spread of BSD-derived Unix through Sun Microsystems, cofounded by former BSD programmer Bill Joy;
- fragmentation of Unix versions in rival "BSD" and "AT&T" camps (Figure 2).

3.2 The push for formal standardization: the "open systems" movement

AT&T's Unix provided a multivendor standard which, when coupled with the BSD enhancements, helped spur the adoption of networked computing. Led by Sun, whose slogan became "the network is the computer," Unix rapidly gained acceptance during the 1980s as the preferred operating system for networked engineering workstations [7]. At the same time, it became a true multivendor standard as minicomputer producers with a small customer base, weak R&D and immature operating systems licensed Unix from AT&T. The main exceptions to the Unix push were the early leaders in workstations (Apollo) and minicomputers (DEC), who used their proprietary operating systems as a source of competitive advantage, and thus were the last to switch to Unix in their respective segments.

The adoption of Unix was also aided by user demand in key areas. Universities continued to favor the OS as a research and teaching tool. Meanwhile, to avoid repeating earlier mainframe switching costs, U.S. Defense Department procurement decisions began to favor Unix over proprietary systems.

Advocates among both producers and users formed a number of trade associations to promote Unix and related operating systems. By fueling the adoption and standardization of Unix, they hoped to increase the supply of application software to compete with sponsored, proprietary architectures [6]; [10]. These two groups promoted these under the rubric "open systems"; the editors of a book series on such systems summarized their goals as follows:

> Open systems allow users to move their applications between systems easily; thus, purchasing decisions can be made on the basis of cost-performance ratio and vendor support, rather than on which systems will run a user's application suite [24: v].

Despite such noble goals, the Unix community spent the 1980s and early 1990s fragmented into warring factions, each of which sought control of the OS APIs to maximize the software available for their respective versions. The initial split between the "AT&T" and "BSD" factions effectively ended after an AT&T/Sun agreement in 1987 merged the two streams to a common technical standard. Fearing AT&T control and increasing licensing fees, in 1988 larger computer makers established a rival group, the Open Software Foundation. But in 1989 both OSF and the AT&T-led Unix International effectively ended their rivalry by joining X/Open, a group of weaker European computer makers established in 1984. The APIs published after negotiation among X/Open members then became the common Unix standard for developers and end users [6]; [7]; [10].

The final division among the Unix community came between those versions of the operating system based on code licensed from AT&T, and those which were independent of the AT&T sources. The licensed OS versions included all the early Berkeley releases, as well as most of the commercial versions of the software, including the Xenix operating system from Microsoft.

At the same time, many university and hobbyist implementations of Unix-like operating systems were begun with the express intent of being free of the AT&T licensing royalties and non-disclosure terms. This would allow free (or nearly free) software to be distributed in source code form to the widest possible audience. Meanwhile, the 4.4 BSD developers fought and eventually won the right to release (in source code form) a version of BSD which did not include AT&T restricted code, launching the NetBSD, FreeBSD and OpenBSD operating systems. The volunteer groups developing these Unix clones were the forerunners of what later became the "Open Source" movement.

4. Gnu, Linux and open source

The ARPANET (and the subsequent Internet) provided a reliable way to organize dispersed groups of software engineers, through e-mail, discussion groups and primitive file servers. These groups worked to develop and expand the network, as well as the tools necessary to run it, such as e-mail and later web servers. Others worked on free Unix-like operating systems, eventually including Linux.

4.1 Software and growth of the Internet

The ARPANET had begun as a research network sponsored by the U.S. Department of Defense. So in 1986, government-sponsored researchers began quarterly meetings of what became the Internet Engineering Task Force. The group quickly broadened to include other U.S. and later international researchers, and in 1998 sponsorship shifted from the U.S. government to the non-profit Internet Society. While it lacks the statutory enforcement mechanisms of a formal *de jure* standards body, with implicit U.S. government sponsorships and a formal standardization process, the IETF standards are closer to *de jure* than *de facto* standards.

Its process is based on "rough consensus and running code." The adoption of any standard in final form requires at least two independent and mutually compatible implementations of that standard; when possible, the standard is defined in terms of a "reference implementation" with public source code. Unlike some standards organizations, the IETF discussions, draft specifications and all source code are public documents posted on Internet file servers [2]; [17]. As Bradner noted, "The IETF supported the concept of open sources long before the Open Source movement was formed" [2: 52].

These prototype implementations were increasingly developed for use on Unix-compatible computer systems. As the growth of the Internet shifted the user base from workstations to primarily personal computers, the client-based Unix implementations declined in importance, but they continued to enable the use of Unix (and the Unix clones) to run Internet servers using well-tested free software such as Sendmail, BIND, and Apache.

4.2 GNU becomes Linux

The origin of the open source movement came in 1984, when MIT programmer Richard Stallman quit his job and began the GNU project. His was driven by opposition to the norms of commercial software: proprietary technologies, secret source code and software copyright. In his view, all software should be "free software," with source code that can be read, modified and redistributed [29].

His GNU system began with the Emacs text editor and other development tools, and was widely distributed via tape and via the ARPANET research network and later the Internet. The C and later C++ compilers were especially popular with other developers seeking to implement a quick programming language. However, GNU lacked the essential core of a modern operating system, a kernel. Efforts to develop a kernel had first been delayed by work on other components, and then floundered for several years without strong leadership.

The eventual kernel came from Linus Torvalds, a Finnish undergraduate student, who in August 1991 announced in a worldwide Minix discussion group his plans for a free hobbyist operating system for Intel-based PCs. He soon attracted hundreds of co-authors and thousands of users for Linux, a complete Unix-like system that made extensive use of GNU components.

4.3 "Open Source" movement

By the mid-1990s, examples of "free software" included Project GNU, Linux, and the various BSD-derived OS and tools. But different groups argued over what freedoms were necessary and desirable. All agreed that users should have the freedom to read, use and modify their source code, but differed on two key issues: organization and appropriability (Table 1).

GNU and the free versions of BSD use a volunteer analog to the organizational approach used for proprietary commercial software, with development done by a small and carefully coordinated group. The Linux development is largely without structure and accepts contributions from anyone, with marketplace feedback from thousands of adopters to determine which changes were kept and which ones would be rejected. Raymond respectively labeled these two models "the cathedral and the bazaar" [23].

More significantly, some developers vehemently object to the idea that free source software be merged with proprietary software or modified to become proprietary; therefore, source distributed under the GNU Public License forbids such modifications. A rival group worried that the anti-capitalist rhetoric (particularly by Stallman) would limit the adoption of free software by both developers and users. They drafted a rival license, the "Open Source Definition," which allowed for commercialization. This business model was quickly endorsed by Torvalds, Netscape, and other key actors [22].

In response to the increasing popularity of open source and Linux, a number of firms offered partial open source strategies to spur adoption and thus complementary assets:

- In 1998, Netscape released the source to its browser to the open source group Mozilla.org, although outsider contributions were dwarfed by those from Netscape employees.
- Later that year, Sun announced the distribution (under a "Sun Community Source License") of Java-related source code but not its Solaris operating system.
- In 1999, Apple announced its "Darwin" open source initiative, providing the BSD part of its new Mac OS X operating system but excluding Macintosh emulation and its new Aqua user interface.

Interestingly, even the most restrictive "free software" license does not require that the software be distributed without charge: "to understand the concept, you should think of 'free speech,' not 'free beer'" [5]. From a practical standpoint, the earliest GNU distributions charged a nominal handling fee for mailing magnetic tapes; more recent free software distributions have been sold on CD-ROM, even though all contemporary open source projects are freely available on the world wide web. This intentional loophole enabled creation of for-profit distribution firms such as SuSE (1992), TurboLinux (1992), VA Linux (1993), Red Hat (1994) and Caldera Systems (1998).

4.4 Commercialization of Linux

The notion of commercializing Linux seemed a contradiction to many. Commercial software development is based on the premise that the developer must be able to protect the intellectual property embedded in software code through trade secrets, copyright and even patents. Without intellectual property protection, there was no way for the developer to collect economic rents and thus no incentive to develop new technologies. Simply put, how can someone make money selling something that was given away for free on the Internet?

The answer turned out to be simple: the same way that companies sell bottled drinking water when it's available for virtually free out of the tap. Just as bottled water companies filter drinking water so it tastes better, new companies started to package Linux with supporting applications, documentation and services to add value and make it easier to use. Companies such as Red Hat offered "distributions" of Linux on CD-ROM with installation software, utilities, and user manuals. They also usually offered some level of technical support, while others such as Linuxcare provided support for all Linux distributions.

After years of being developed by a group of reputedly idealistic programmers who selflessly gave their time without financial gain, Linux suddenly got caught in the updraft of the IPO frenzy of the late 1990s. Red Hat went public at $14 on August 11, 1999, and topped $250 by December of that year.

The Linux boom was short-lived as stock prices collapsed in the spring of 2000. By May 2000, Red Hat lost 90% of its value, VA Linux lost 88%, and Caldera lost 75%. The bursting of the stock bubble raised the issue of whether the Linux companies had developed viable profit models. However, it did not appear to slow the adoption of Linux, which already had established itself with a significant number of users, and was supported by major hardware, software and IT services companies. This continued growth raises the question of why these users and providers of complementary assets were now making Linux a key part of their technology strategies.

5. Adoption motivations

5.1 End user adoption

Since its commercialization, Linux has become the fastest growing operating system on the market, with an estimated 25% of server shipments in 1999 according to IDC [15]. There are several possible explanations that Linux has been able to capture such a large market share against entrenched competitors.

Fragmented Server Market. The operating system market for client PCs has evolved along the lines predicted by theories of increasing returns and network externalities [27]. Most PC users have strong incentives to choose Windows, given the huge library of Windows applications and the ability to share files with other Windows users. The result is a 90% market share for Windows in desktop and notebook PCs.

On the other hand, the server market is much different. Microsoft's Windows NT was a late starter: IDC estimated that it accounted for only 38.3% of the server market in 1998 mostly at the low end [16]. Microsoft faces strong competition from Novell NetWare, with 22.8% of the market and the various versions of Unix from Sun, HP, IBM, Compaq and SGI, which had 18.8%. It also is quite common for large enterprises to support more than one server platform. For instance, Dell Computer runs many of its applications on its own PC servers running Windows NT, yet its order management system runs on high-end Tandem servers [13]. In such a mixed environment, there may be less resistance to adopting another operating system to handle new functions such as web hosting.

New Uses. IT departments tend to resist changing platforms for established applications, as there are rarely compelling enough benefits to justify the high switching costs (and the threat to personnel whose expertise is in the existing platform). As a result, new platforms usually gain acceptance when applied to new uses. For instance, the minicomputer initially was adopted as a single-purpose platform for engineering applications, rather than a replacement for the general purpose mainframe computer. Likewise, the PC was used for personal productivity applications that were difficult to perform in a time-sharing centralized computing environment.

The most common applications for Linux are infrastructure applications, mostly centered around the Internet, where Linux servers are commonly used for web/Internet applications, e-mail, firewall, and proxy/caching/security services. According to IDC, 40% of all customer spending for Linux servers in 1999 was for Internet-related applications [1]. The fact that these applications were Internet-centric made them ideal for Linux, which grew up on the Internet and was developed to a large extent with Internet applications in mind, just as Windows was developed to run desktop PC applications.

New Users. New standards are adopted not only for new applications, but also often by a new set of users who are not invested in existing technologies. The minicomputer was first adopted by scientists and engineers, rather than the data processing departments who were more locked into the mainframe, and PCs were adopted by individual users rather than companies or departments.

The early adopters of Linux were neither corporate IT departments, who were busy supporting various proprietary server platforms, nor individual PC users, who were using Windows. Instead, they included computer scientists, hackers, students, and others who need to modify the software to fit particular uses, and who in many cases were part of the Linux programming community. They also included Internet service providers, web site developers, and intranet developers who wanted a low-cost, robust operating system suited to such Internet applications.

According to IDC, Linux is used in a variety of industries, with the highest concentration in educational institutions and the IT industry [1]. This is not surprising, given the high numbers of computer scientists and engineers in those two settings.

Also, as Lerner and Tirole point out, the greatest use of open source software such as Linux takes place "in settings where the end users are sophisticated...In these cases, users are apparently more willing to tolerate the lack of detailed documentation or easy-to-understand user interfaces in exchange for the cost saving and the possibility of modifying the source code themselves" [14].

Price/Performance. While Linux is a multi-platform OS like Unix, most Linux users stick with low cost Intel-based hardware, which enjoys huge economies of scale from the global PC industry. Adopters say that Linux has better reliability than other OSes available on the Intel platform, such as Windows NT and NetWare. For example Burlington Coat Factory made a major commitment to Linux in 1998, buying 1,250 desktop PCs running Linux to manage the day-to-day retail business of its 264 stores. In the words of Mike Prince, Burlington's chief technology officer, "Linux is rock-solid. It has a tremendous amount of mind share. It is unifying Unix in a way that it has never been before" [21].

Acer Computec Latino America (ACLA) is another Linux user, partly due to the fact that its CIO is a Linux developer himself. ACLA runs all of its operations in Mexico on Intel-based Acer hardware, using both NT and Linux. Linux is used for ACLA's public web site and runs on firewalls to protect servers running NT, which is said to be vulnerable to attack (author's interview with ACLA executives, August 1998).

5.2 Adoption by software and service vendors

A key issue in any standards competition is whether a standard can attract developers of complementary assets, which account for much of the value of a standards platform [30]. In order to succeed Linux needed a supply of applications, development tools, and support services, especially if it was to be accepted in more conservative business settings.

The first key complementary asset for Linux originally was the large pool of professionals who wrote most of the code, found and fixed bugs, and also provided technical support to users via the Internet. In many cases, a user could get a problem diagnosed and solved via e-mail faster than he or she would get similar support from a software vendor's technical support team. For businesses and other large institutions, however, relying on the kindness of strangers is simply too risky. MIS departments are used to having support contracts with vendors or third party service providers, and having legal recourse if such support fails them. This required that someone step in and offer service and support on a commercial basis if Linux were to penetrate deeper into user organizations.

Just as the Linux community provided valuable support for users, the open-source community was also responsible for the first wave of applications to run on Linux, including Apache web server, Sendmail e-mail routing, Perl and TCL programming languages, and the MySQL database. To advance beyond running Internet servers, however, Linux needed software companies to port leading business applications to the Linux platform.

As Linux evolved, it began to attract support from commercial suppliers of services and software. First, were the startups such as VA Linux, Red Hat, Caldera, SuSE, and TurboLinux. Some of these also offered consulting services, training and certification, and technical support on a contract basis. Others developed applications for Linux, including Linux distributors such as Red Hat and SuSE and independent software vendors such as Applixware [8].

Around 1998, a new wave of companies entered the picture. These were large established software and services vendors, such as IBM, Oracle, Novell, Hewlett-Packard, Computer Associates, and Corel. They began to port applications to Linux, from productivity suites such as Sun's Star Office and Corel's WordPerfect to enterprise applications such as Oracle databases and IBM's DB2 database application and MQ Series messaging software. The support of these established suppliers was vital to providing confidence to potential users that Linux was a viable standards choice for business applications.

Why did these providers of complementary assets choose to support Linux, when they had ignored other standards? The decision was not without costs or risks, especially to more established vendors. Some had their own operating systems that might be threatened by the success of Linux (e.g., Compaq, HP and IBM, which had their own versions of Unix as well as other proprietary OSes). Second, there were significant costs to developing and supporting Linux versions of major applications, and the Linux market was still small and growth uncertain. Third, Linux did not offer the opportunity to lock customers in to the vendors' own technologies. Finally, there was the threat of retaliation from Microsoft, which held significant leverage over many of these companies.

Under these conditions, such widespread support for Linux might be explained by the following factors:

- Responding to customer demand: As Linux gained traction in organizations, there was a need for more robust support, and users turned to their traditional IT vendors. According to IBM, its decision to offer Linux on the System 390 mainframe was made "in response to a groundswell of demand by S/390 users" [12].
- Cost: Vendors such as IBM, HP and Compaq may be hoping to reduce their own software development costs, and are looking at Linux as an alternative to their second-tier Unix systems. In this case, Linux is ideal, as its development costs are borne by the Linux community, and adopting it does not put the vendors at the mercy of a competitor.
- Strategic reasons: Companies such as Oracle, Sun, and IBM all compete aggressively with Microsoft, and see Linux as a tool for challenging Microsoft in the server market. IBM and HP also are anxious to catch up to Sun in fast-growing hardware markets for Internet and e-commerce servers, and are capitalizing on the popularity of Linux in the Internet world.

5.3 Adoption by hardware vendors

Table 2: Top Linux Server Vendors

	Market share
Compaq	25%
IBM	10%
HP	7%
Dell	7%
Fujitsu Siemens	3%
Others	48%

Source: IDC, 2000, Q4 1999 unit shipments

PC Vendors. Linux originally was developed on the Intel platform, and is still predominantly used on Intel servers. Early adopters used older PCs, self-built systems, or so-called "white box" clones. As of September 1999, self-built systems accounted for 39.5% of installed Linux servers and white boxes another 14.9%, according to IDC. However, the big PC makers have entered the market and captured a growing share, with Compaq, IBM, HP and Dell leading the way (Table 2).

Why have PC companies decided to offer Linux as an alternative to Windows NT?

- Customer demand: As in the case of software and service providers, hardware companies are responding to demand for Linux. If end users are going to download Linux or install it from a CD-ROM, PC makers would rather offer it pre-loaded and give customers what they want on the vendor's own hardware.
- Independence from Microsoft: PC makers have been entirely dependent on Microsoft for operating systems for both client and server PCs. While there is little alternative to Windows on the client side, Linux offers an option on servers that does not depend on Microsoft's often shifting OS road maps. Even Microsoft's closest partner in Windows NT, Compaq, has not hesitated to push aggressively into the Linux market.

Intel. Intel has supported Linux in two ways. First, it has invested in a number of Linux startups, including Red Hat, SuSE, VA Linux, TurboLinux, eSoft, and Cobalt Networks.[2] It also supported Linux developers porting Linux to Intel's next generation IA-64 microprocessor, while providing information about the new chip to developers of Linux applications. It also has set up Itanium servers at VA Linux that are available over the Internet for Linux developers to work on [26].

Intel's support for Linux and its investment in Linux companies is important in establishing the legitimacy of the new OS. One likely reason is Intel's strong push for its processors into the fast-growing Internet server market. Linux gives Intel a robust operating system that runs on Intel hardware and can compete against Unix for some applications where Windows NT/2000 is still considered less reliable. Intel also may be trying to distance itself from Microsoft, as it has never felt quite comfortable with Microsoft's first-among-equals position in the Wintel partnership.

Beyond the PC. IBM has moved Linux beyond the PC platform by announcing it would support Linux on all of its server platforms, including RS/6000, AS/400 and S/390. It has designed its implementation of Linux on the S/390 so that users can run Linux alongside the standard OS/390. It also offers a Linux virtual machine so that a customer can offer a complete Linux server environment to each of its application developers and also host production systems all on the same S/390. This puts the S/390 in competition with both Sun and Microsoft in the e-business market, and also allows IBM's installed base of S/390 users to extend the functionality of their hardware into both Internet and business applications.

At the other end of the spectrum, the viability of a non-Microsoft OS is likely to be extended to newly emerging information appliances. Thus far, Microsoft's Windows CE has failed to dislodge the majority share held by the Palm OS in the PDA market. Linux has started to enter this segment too, as companies such as Gateway (in partnership with AOL) are designing new Internet appliances based on Linux and new chip developed by Transmeta, the Silicon Valley company that employs Linus Torvalds.

[2] Dell, Oracle, IBM and others have likewise invested in Linux startups.

Torvalds himself led an effort to modify Linux to run on consumer devices without a hard disk [18].

5.4 Assessment of adoption patterns

- Adoption of Linux has been driven by the Internet. A large share of Linux use is for Internet applications, and many users are Internet-oriented businesses such as ISPs who take advantage of Linux's suitability for such applications. Also, Linux developers use the Internet as a tool for collaborative development. Microsoft, in its so-called "Halloween" memo, acknowledges that the Internet provides a vital channel to coordinate development and create awareness of Linux. "OSS [Open Source Software] projects the size of Linux and Apache are only viable if a large enough community of highly skilled developers can be amassed to attack a problem. Consequently, there is direct correlation between the size of the project that OSS can tackle and the growth of the Internet" [31].
- Early Linux adopters had different characteristics from other IT users. They were younger, had stronger technical skills than most PC users, and had not built careers administering proprietary operating systems as most IT managers had. They brought Linux into the enterprise through the back door, just as an earlier generation had introduced PCs outside the reach of MIS departments.
- Companies offering complementary assets for Linux are motivated by a variety of economic and strategic incentives. The "anybody but Microsoft" sentiment is strong among companies that had lost earlier market battles with Microsoft and those who do not want to see Microsoft to extend its dominance into new markets. Others simply see a new market opportunity or a chance to leverage the development efforts of the Linux community at little cost to themselves.

6. Conclusions

Successful promotion of a standard must balance the demands of adoption and appropriability. Achieving the widest adoption of a standard attracts suppliers of complementary assets such as software and services, which in turn fuels further adoption [27]. This can be achieved by widespread technology licensing on favorable terms, but by doing so, the sponsor runs the risk of losing the ability to appropriate economic rents from the standard.

In the systems era, IBM maintained tight control over all key mainframe technologies and used its own sales and marketing efforts to achieve widespread adoption among large organizations that could afford such systems. In the PC era, Microsoft and Intel each controlled a key technology standard in an otherwise open architecture, and worked with system and software vendors to achieve the widest possible adoption in a much larger market. The network era has favored even more open standards due to the need for device compatibility with underlying open network standards.

Promoters of open systems have tried various strategies to balance adoption and appropriability. One strategy is for a corporate sponsor such as Sun to make its technology widely available while relying on its ability to earn temporary monopoly rents on new innovations [7]. Another is for a consortium of companies to establish an industry association to develop and promote a technology, as X/Open promoted a common Unix standard [6]. The risk here is that free-riding on the part of member companies can lead to underinvestment in the core technologies of the standard.

"Open source" software such as Linux takes another approach to the promotion of so-called "open systems." Linux has encouraged the widest possible adoption through a liberal licensing regime which encourages copying and distribution. On the other hand, it offers a novel solution to the problem of sustaining investment in a technology standard in which there is no appropriability of new innovations. A core group of developers led by a highly visible individual (Tovalds) plays the role of sponsor, but does not represent any company or group of companies with a commercial interest in the promotion of Linux. This group instead coordinates a network of thousands of Linux developers who contribute time and effort to upgrading Linux. As analyzed elswhere (e.g., [23]; [14]), these developers are believed to be motivated by some combination of gratification of working on challenging programming problems, career advancement and other factors. Another possible explanation has been user-driven innovation, or co-invention [3]; [32]. Linux programmers have improved Linux in part to solve their own problems as users, and are willing to contribute their innovations to the community in return for having access to the contributions of others.

As for the development of complementary assets, the incentives of hardware and software companies are simpler. Those companies do not profit by appropriating rents from the core technology; rather they build profitable businesses that leverage the standard. Companies such as Dell, Compaq and Gateway have done so in the Windows market and can as easily do so in the Linux market if it gains sufficient size.

Linux's future prospects depend on the ability to maintain a single common standard while ensuring continued investment in that standard. One threat is "forking," which could happen if companies try to appropriate economic rents by creating incompatible proprietary implementations, or individual research groups focus on specialized needs without regards to the common standard. As shown earlier, this was the norm with Unix, and this risk is inherently greater with fully open source software.

At the other end of the spectrum is the risk of underinvestment due to the lack of incentives to invest in a non-appropriable standard. The continued volunteer efforts of the Linux community are essential, and could be threatened if volunteers become disillusioned by having others make large amounts of money off of their efforts. After the early success of Linux IPOs in 1999, Linux websites such as Slashdot.com hosted a heated debate over the contradictions between open source and commercialization.

There are also external risks from proprietary hardware vendors whose market share is threatened by Linux, including Sun and other Unix vendors. As Gabel noted, even vendors who've nominally endorsed a shared standard have difficulty making credible commitments to a standard that eliminates their options for proprietary advantage [6]. More ominous still is Microsoft, which has bet heavily on Windows 2000 and is well aware of the threat presented by Linux [31].

Factors favoring Linux include its high growth and significant market share, which is attracting a healthy supply of complementary assets and giving more confidence to users who consider adopting Linux. Companies making money from Linux can invest in the continued development of the core technology, supplementing or subsidizing the efforts of volunteers. As for the Microsoft challenge, Linux may be benefitting from Microsoft's legal problems, which appear to have emboldened Microsoft's customers to support alternative operating systems such as Linux.

It will not be known for a few years whether Linux or other open source projects have truly created a viable new model for standards competition. While the threats to the model should not be underestimated, we feel that the timing is favorable. The shift from personal computing to network-oriented computing is as significant a shift as the PC revolution of the 1980s, and there are opportunities for new standards and forms of industrial organization to succeed. Linux may not be the winner, but its early success has given it a first mover advantage in emerging Internet markets. If Linux does succeed, it will suggest that the open source model offers a viable solution to the dilemma of balancing adoption and appropriability, and may be more generally applicable to other network era standards.

7. References

[1] Bailey, Michelle, Vernon Turner, Jean Bozman and Janet Waxman, "Linux Servers: What's the Hype, and What's the Reality," IDC Bulletin #21610, March 2000.

[2] Bradner, Scott, "The Internet Engineering Task Force," in Chris DiBona, Sam Ockman and Mark Stone, eds., *Open Sources: Voices from the Open Source Revolution,* Sebastopol, Calif.: O'Reilly, 1999, pp. 47-52.

[3] Bresnahan, Timothy F. and Shane Greenstein, "Technological competition and the structure of the computer industry," *Journal of Industrial Economics* 47, 1 (March 1999):1-40.

[4] Chposky, James and Ted Leonsis, *Blue Magic: The People, Power and Politics Behind the IBM Personal Computer,* New York: Facts on File, 1988.

[5] Free Software Foundation, "What is Free Software?" May 2000, URL: http://www.gnu.org/philosophy/free-sw.html

[6] Gabel, H. Landis, "Open Standards in Computers: The Case of X/OPEN." In H. Landis Gabel, ed., *Product Standardization and Competitive Strategy,* Amsterdam: North-Holland, 1987.

[7] Garud, Raghu and Arun Kumaraswamy, "Changing competitive dynamics in network industries: An exploration of Sun Microsystems' open systems strategy," *Strategic Management Journal* 14, 5 (July 1993), 351-369.

[8] Gillen, Al and Dan Kusnetzky, "Linux Overview: Understanding the Linux Market Model," IDC Bulletin, International Data Corporation, 2000.

[9] Greenstein, Shane M., "Lock-in and the Costs of Switching Mainframe Computer Vendors: What Do Buyers See?" *Industrial and Corporate Change* 6, 2 (1997): 247-274.

[10] Grindley, Peter, *Standards, strategy, and policy: cases and stories,* Oxford: Oxford University Press, 1995.

[11] Grove, Andrew S., *Only the Paranoid Survive: How to Exploit the Crisis Points that Challenge Every Company and Career,* Doubleday, New York, 1996.

[12] IBM, "IBM Unveils Linux Software and Services for S/390 Server, Introducing Big Iron to Next Generation e-businesses," press release, Somers, NY, May 17, 2000.

[13] Kraemer, Kenneth L., Jason Dedrick and Sandra Yamashiro, "Refining and Extending the Business Model with Information Technology: Dell Computer Corporation," *The Information Society* 16, 1 (2000), 5-21.

[14] Lerner, Josh and Jean Tirole, "The Simple Economics of Open Source," Harvard Business School Working Paper #00-059, February 25, 2000.

[15] Lillington, Karlin, "Linux's Perilous Victory," *Guardian Unlimited,* May 18, 2000, URL: http://www.guardianunlimited.co.uk/online/story/0,3605,221905,00.html

[16] MacCormack, Alan and Kerry Herman, "Red Hat and the Linux Revolution," Harvard Business School case #9-600-009, Harvard Business School, 1999.

[17] Malkin, Gary, "The Tao of IETF: A Guide for New Attendees of the Internet Engineering Task Force," RFC 1539, Information Sciences Institute, Los Angeles, August 1993, URL: ftp://ftp.isi.edu/in-notes/rfc1539.txt

[18] Markoff, John, "Gateway and AOL Bypass Industry Stalwarts on Components," *New York Times,* May 30, 2000, p. C1.

[19] Morris, Charles R. and Charles H. Ferguson, "How Architecture Wins Technology Wars," *Harvard Business Review,* 71, 2 (March/April 1993), 86-96.

[20] Moschella, David C., *Waves of power: dynamics of global technology leadership, 1964-2010,* New York: AMACOM, 1997.

[21] *New York Times,* "Dell Bolsters Support of Windows Rival Linux," April 7, 1999, p. C2.

[22] OpenSource.org, "History of the Open Source Initiative," 1999, URL: http://www.opensource.org/history.html

[23] Raymond, Eric S., *The cathedral and the bazaar: musings on Linux and open source by an accidental revolutionary*, Cambridge, Mass.: O'Reilly, 1999.

[24] Salus, Peter H., *A quarter century of Unix,* Reading, Mass.: Addison-Wesley, 1994.

[25] Schneider, Wolfram, "The Unix system family tree / BSD history chart", v1.24, FreeBSD, Inc., April 30, 2000, URL: ftp://ftp.freebsd.org/pub/FreeBSD/FreeBSD-current/src/share/misc/bsd-family-tree

[26] Shankland, Stephen, "Linux programmers get a taste of the future," CNET News.com, May 25, 2000, URL: http://news.cnet.com/news/0-1003-200-1951178.html

[27] Shapiro, Carl and Hal R. Varian, *Information rules: a strategic guide to the network economy*, Boston, Mass.: Harvard Business School Press, 1999.

[28] Sobel, Robert, *I.B.M., colossus in transition.* New York: Times Books, 1981.

[29] Stallman, Richard, "The GNU Operating System and the Free Software Movement," in Chris DiBona, Sam Ockman and Mark Stone, eds., *Open Sources: Voices from the Open Source Revolution,* Sebastopol, Calif.: O'Reilly, 1999, pp. 53-70.

[30] Teece, David, "Profiting from technological innovation: Implications for integration, collaboration, licensing and public policy," *Research Policy* 15, 6 (Dec. 1986), 285-305.

[31] Valloppillil, Vinod, "Open Source Software: A (New?) Development Methodology," internal memo, Microsoft Corp., Aug. 11, 1998, URL: http://www.opensource.org/halloween/halloween1.html

[32] von Hippel, Eric, *The sources of innovation,* New York: Oxford University Press, 1988.

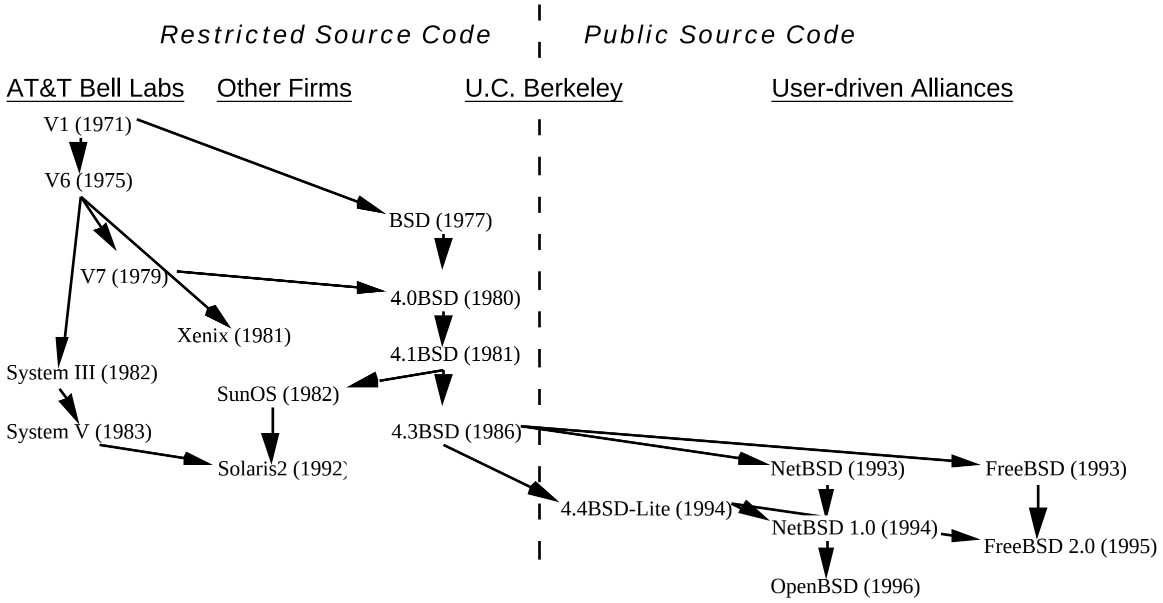
[34] West, Joel and Dedrick, Jason, "Innovation and Control in Standards Architectures: The Rise and Fall of Japan's PC-98," *Information Systems Research* 11, 2 (June 2000), 197-216.

Table 1: Comparison of commercial and "open source" software licenses

Type of Software	Zero Price	Source Code Available	Source Code Modifiable	Public changes to core code	All derivatives must be free
Commercial					
Shareware	†				
Open Source (BSD-Style)	X	X	X		
Open Source (Apache Style)	X	X	X	X	
Open Source (Linux/GNU style)	X	X	X	X	X

† Nominal price, but unenforced

Source: Adapted from [31]



Sources: [24], [25], company web sites

Figure 2: Simplified genealogy of Unix implementations