



© 2002 The Charles Babbage Institute for the History of Information Technology
211 Andersen Library, 222 – 21st Avenue South, Minneapolis, MN 55455 USA

A War on Two Fronts: The U.S. Justice Department, Open Source, and Microsoft, 1995-2002

Paul E. Ceruzzi
National Air & Space Museum
Smithsonian Institution

Date published: 13 September 2002

(Note: this material has been prepared for a Second Edition of *A History of Modern Computing*, by the author, to be published by MIT Press in 2002. MIT Press will assume the copyright for the printed version.)

Abstract: Beginning in the mid 1990s, the dominance of Microsoft in the arena of personal computer software was attacked on two fronts. One was an antitrust suit brought against Microsoft by the U.S. Justice Department. The other was the enthusiastic embrace of a rival operating system, Linux, that was available for “free” (in a carefully-defined way, as the paper explains). Both assaults were based on technical, social, political, and even emotional concerns and issues. This paper examines each attack, and attempts to place them in historical context. While it is too soon to know how they will turn out, enough has happened to provide at least a beginning of a history.

Keywords: Microsoft Corporation, Linux, Open Source, UNIX

The U.S. Justice Department vs. Microsoft

The commercial, aired during the third quarter of the game, was the most memorable part of the broadcast of the January 1984 Super Bowl. The Macintosh, Apple assured us, would usher in a new era of personal computing, and therefore the year 1984 would not be one of dreary conformity and oppression as prophesied by George Orwell’s novel *1984*. A revolution in personal computing was indeed in the works, and the Macintosh was leading the way. Unfortunately for Apple, it was Microsoft, not Apple, who brought the revolution to a mass market. That happened not in 1984, the year the Mac

appeared, but in 1992, when Microsoft began shipping version 3.1 of its Windows program. In 1984, Apple hoped that the Mac would bring the innovative ideas from the Xerox Palo Alto Research Center, ideas already present in a few personal computer systems, to the consumer. A dozen years later, Microsoft, not Apple, would dominate personal computing. And that domination, in turn, would lead to its entanglement in a bitter antitrust trial.

Just as IBM spent a significant fraction of its resources during the 1970s facing a challenge by the U.S. Justice Department, so too is Microsoft in the same situation, following a similar filing against it in 1997. In November 2001 the Federal government announced a settlement, but several states, and the European Union, refused to go along and are continuing to press charges. Almost daily, the business press reports whenever a judge or lawyer from either side makes a statement or renders a ruling. Until the case is settled, one can only make provisional comments about its significance. One lesson of the IBM trial, however, applies to the present case against Microsoft: the Justice Department once again does not recognize how advancing technology may render the lawsuit irrelevant. What is the Microsoft-equivalent of the personal computer, whose appearance in the midst of the IBM trial was ignored as the litigants fought over mainframe dominance? It is too early to tell, although I will discuss some candidates later. What is certain is that advances in computing already threaten, and will continue to threaten, Microsoft's ability to dominate personal computing, based on its Windows and Office software.

The licensing policies of Microsoft and Intel gave rise to clone manufacturers, like Dell, Compaq, and Gateway, who provided choices unavailable to Apple customers. That yielded a greater variety of products and, above all, lower prices. Windows version 3.1, the introduction of the Pentium processor, and the combining of applications software into a suite called Microsoft Office, together gave consumers approximately eighty percent of what the Macintosh was offering, at a lower price. To Apple's surprise (and to the chagrin of Mac fans), that percentage was good enough to tip the balance, perhaps forever, away from Apple. By 1995 the advantage of Apple's more elegant design no longer mattered, as the Microsoft/Intel combination became a standard, like COBOL in the 1960s. As with COBOL, what mattered was the very existence of a standard, not the intrinsic value or lack thereof of the software.

The Macintosh Connection

One could begin this story of Microsoft's triumph and troubles at any number of places, but the introduction of the Macintosh conveniently allows us to introduce several critical factors. The first was that when the Mac appeared in 1984, it had a magnificent user interface but almost no applications software—programs that people actually bought personal computers for. The most interesting application that it had was MacPaint, a drawing program descended from the pioneering work at Xerox, and something that no software for IBM compatibles could approach.

But for word processing, an application that any serious new computer had to have, Apple offered only MacWrite, which took advantage of its graphical interface, but which otherwise was extremely limited in capability.¹ Both MacPaint and MacWrite were developed in-house.

Early Mac customers could also get a spreadsheet: Multiplan, developed by Microsoft for other computers but ported to the Mac. Although some popular accounts enjoy setting up Bill Gates and Steve Jobs as mortal enemies, for much of this period the two men had a cordial and mutually beneficial business relationship. At the onset of the Mac's development, in June 1981, Jobs and Jef Raskin (who had the initial idea for the Macintosh) met with Gates, and in January of the following year Microsoft agreed to develop software for the new machine.²

Gates needed little convincing of where personal computing was going. Even as Microsoft was negotiating to supply DOS for the IBM PC in 1980, Gates hired a programmer who would take the company in the opposite direction. That was Charles Simonyi, a native of Hungary who learned how to program first on a Soviet-built vacuum tube computer called the Ural-II, then on a Danish transistorized computer that had an advanced Algol compiler installed on it. In the 1970s Simonyi worked at Xerox PARC, where he developed a word processor called Bravo for the Alto workstation. Bravo is often credited with having the first true WYSIWYG (“What-You-See-Is-What-You-Get”) display, a concept that other Xerox alumni brought to Apple.³

In 1985 Microsoft produced another spreadsheet, Excel, for the Macintosh, which took advantage of all that the Macintosh interface had to offer. Excel was a success and helped Apple get through a difficult period when Mac sales were in danger of completely drying up. Mac users finally had a spreadsheet that was comparable to Lotus 1-2-3 on the IBM PCs. For its efforts, Microsoft gained something too: besides producing a successful product, Microsoft programmers learned how to develop software for a windows-based interface—something that Lotus and Word Perfect would have a hard time learning.

The ultimate impact of hiring Simonyi, and of these interactions between Microsoft and Apple, was that Bill Gates decided to recreate the Macintosh experience on the Intel 80x86 platform. Consider the context of that decision. In the mid-1980s, “Windows” was but one of many graphical systems (e.g., VisiOn, GEM, et al.) proposed for IBM compatibles. And Microsoft's applications programs, like Multiplan, were not as well regarded by industry critics as programs like Lotus 1-2-3 or Word Perfect. The Windows model was also being challenged by a competing idea, mainly from Lotus: that of a *single* program, running under DOS, that combined spreadsheets, databases, and word processing. Lotus offered such a program called Symphony for the IBM PC and was working on one for the Mac called Jazz. At Ashton-Tate, the leading supplier of database

software for the PC, a Xerox-PARC alumnus named Robert Carr was developing a similar program called Framework.⁴

It turned out that keeping the applications separate, while insisting that each of them adhere to a common graphical interface, would prevail.⁵ That was what Jobs insisted for all Macintosh developers, and Gates made it the focus (slightly less sharp) at Microsoft. Simonyi developed a system of programming that allowed Microsoft to manage increasingly larger and more complex programming jobs as the company grew. The style involved a way of naming variables, and was called “Hungarian,” an inside joke referring to its incomprehensibility to anyone not familiar with Microsoft’s programming, like Simonyi’s native Hungarian language supposedly is to speakers of other European languages.⁶

If Hungarian captured the spirit if not the letter of formal computer science, so too did Microsoft’s management of software development. Although the technique had been used elsewhere, Microsoft embraced it and applied it on a large scale not seen before, and in doing so it broke radically from the way large projects were managed at mainframe software houses. At Microsoft, programmers working on a section of a new product were required to submit their work to a central file at the end of each day, where overnight it would be compiled, along with everyone else’s, into a daily “build.”⁷ If your contribution caused the central file to crash, you were responsible for fixing it. That build then became the basis for the next day’s work.⁸ As soon as the build became marginally functional, members of the programming team were required to use it, regardless of how inefficient that might be. This requirement made life difficult, especially when the software was in an early stage and little of it worked well, but it kept the programmers focused on shipping a finished product of high quality. This process, too, had an evocative name: “eating your own dog food.”⁹ The public has since become aware of the large fortunes amassed by Microsoft programmers who worked there long enough to have their stock options vest. Less well-known is the dog’s life of no sleep, eating out of vending machines, endless staring into a computer screen, no social or family life, and other amenities for a programmer caught in the “death march” of fixing a “broken” build while getting a program finished to ship on time.¹⁰

The cumulative effect of these efforts was a steady stream of ever-improved versions of Windows and an ever-expanding suite of applications. Word and Excel were the two pillars of applications software, soon joined by the database Access, the presentation program PowerPoint, the management planning program Project, and a host of others (*Table 10.1*). Microsoft purchased many of these from smaller, independent companies and then reworked them to conform to the Windows interface.¹¹ Major applications were combined into an applications suite called Microsoft Office.¹²

In its earliest days Microsoft was a supplier of language compilers; then it became known as the supplier of operating systems for the PC. By 1991 over fifty percent

of Microsoft’s revenues came from applications, especially Office. The resulting juggernaut of Windows and Office rolled over IBM and Digital Research among the operating system suppliers, and Lotus, Ashton-Tate, and Word Perfect among the applications providers. By the mid-1990s, many independent software companies supplied applications for the Windows platform, but only a few were of significant size, and fewer still offered word processing, database, or spreadsheet applications.

Selected Chronology of Microsoft Software, 1983-2001¹³

YEAR	WINDOWS VERSIONS	APPLICATIONS	INTEL PROCESSORS
1983	“Interface Manager” announced, not shipped	Word for PC (DOS)	
1984		Project; Chart; Word for Macintosh	80286
1985	1.0	Word 2.0 for PC (DOS); Excel for Macintosh	
1986		Works	80386
1987	2.0	Forefront (later PowerPoint); Excel for PC (Windows)	
1988			
1989			80486
1990	3.0	Word for Windows 1.0; Office	
1991			
1992	3.1	Access	
1993	3.11; NT	Office 4.0	Pentium
1994	NT 3.5		
1995	95 (“Chicago”)	Office 95; Network (MSN); Internet Explorer 1.0	Pentium Pro
1996		Internet Explorer 3.0; Exchange	
1997		Office 97; Internet Explorer 4.0	MMX; Pentium 2
1998	98	Hotmail	Celeron
1999			Pentium 3
2000	2000; Me	Office 2000	
2001	XP		Pentium 4

Internet Explorer

When that juggernaut finally caught the attention of antitrust lawyers at the U.S. Justice Department, few outside of Microsoft were surprised, as pressure had been building up to do something to stop the company. The specific action that triggered the lawsuit was Microsoft's bundling a Web browser into Windows. In December 1994, Microsoft paid Spyglass for a license to use its work as the basis for a Web browser, which Microsoft renamed Internet Explorer. (Spyglass, like Netscape, descended from Mosaic at the University of Illinois.) In the summer of 1995, just after Netscape's public offering, Microsoft offered a version of Spyglass's browser as part of Windows. From this point Microsoft followed a familiar road: it issued successive versions of the browser, each one having more features and greater integration with the base operating system's functions. Note that by bundling Internet Explorer into Windows and selling it at a single price, Microsoft effectively prevented Spyglass from selling a Windows version of its browser.

Internet Explorer 4.0, introduced in the fall of 1997, was just another new version to Microsoft. It was something else entirely to Netscape and to the Justice Department. In their view, Microsoft's tight integration of IE 4.0 went too far. There had been plenty of early warnings of trouble. With the fast pace of events, few remembered these warnings by 1997. What follows is a brief overview of only the most visible. Some of the colorful phrases that arose in connection with those actions are also introduced.

The first indication of a legal issue did not involve Microsoft but did introduce a phrase that would figure in later trials. This was a suit filed in 1987 by Lotus against a company called Paperback Software, established by Adam Osborne of portable computer fame. Paperback was selling a spreadsheet that functioned identically to 1-2-3, but at a fraction of the price.¹⁴ Lotus charged that Paperback, even if it did not copy or steal Lotus's code, nonetheless copied the "look and feel" of 1-2-3, and that was illegal. While that lawsuit was in progress, in 1988 Apple sued Microsoft (and Hewlett-Packard) for copying the "look and feel" of the Macintosh in version 2.0 of Windows. Apple and Microsoft had signed a licensing agreement, but Apple charged that it had only licensed the Macintosh interface for Windows 1.0. It is worth noting that by this time the head of Apple was John Sculley, not Steve Jobs. Jobs not only admitted but even boasted of having stolen the graphical interface from Xerox PARC.¹⁵ Throughout 1989 the case dragged on, eventually to be overtaken by events. Both parties also realized that they had a fundamental need to do business with one another, a need that went back to the founding of both companies in the 1970s.¹⁶

In 1990 the Federal Trade Commission investigated Microsoft in connection with its agreements with IBM over the development of a joint IBM/Microsoft operating system, which IBM marketed as OS/2. The FTC had also investigated a charge that in 1990, Microsoft gained access to details of a prototype pen-based

computer developed by a start-up called GO, and then announced at a trade show that it would soon integrate pen-based input into Windows (something Microsoft never really did). The effect was to immediately dry up all financial support for GO, which eventually folded.¹⁷ This technique was known by the phrase “Fear, Uncertainty, and Doubt,” or “FUD.” It was a charge that Control Data leveled against IBM in the 1960s for the same reason, when IBM announced its System/360, Model 91 to compete with Control Data’s supercomputer. Who would buy a small company’s product when the dominant vendor promised that the same technology would soon be part of its mainstream line?¹⁸ Another phrase, which emerged at this time, was that Microsoft’s actions against GO amounted to “cutting off its air supply”: making it impossible for GO to sell its product or to raise money. Some accused Microsoft of doing that to Spyglass as well, when it bundled Internet Explorer. Beginning in 1999 and into 2001, litigants, journalists, and judges alike would parse this phrase at great length, as it applied—or not—to Netscape.

The Justice Department threatened to sue Microsoft in 1994 over bundling of products into Windows, but it dropped the suit after Microsoft entered into a Consent Decree. Microsoft promised not to engage in the practice of “tying” sales of one product to another: that is, insisting that customers who bought Windows also buy another Microsoft product. This concept of the tie-in was well understood and had historical roots in early twentieth-century antitrust legislation. Although Microsoft carefully guarded the source code for Windows, it agreed to make available the parts of Windows code that interacted with applications programs, the so-called “Applications Program Interface,” or “API.” Thus, for example, developers of a database program were assured that their product would, in theory, work as smoothly with Windows as any database developed by Microsoft.

However, the company developed a Windows pricing policy that was less magnanimous. By 1995 consumers rarely bought Windows in a “shrink-wrapped” package and installed it themselves; instead they bought a computer on which Windows was already installed at the factory. That brought distribution costs, already low, even lower; the computer companies could negotiate a low price for Windows, passing on part of the savings to consumers and taking care of the cumbersome installation process. By the Consent Decree, Microsoft could not insist that anyone who bought a computer from, say, Compaq *had* to buy Windows, too. However, Microsoft billed Compaq on a “per processor” basis, not on the actual numbers of Windows programs installed. Therefore Compaq had to pay for a copy of Windows even if it sold a computer that had another operating system—or *no* operating system—installed on it. Legal, but not a policy to calm the growing army of Microsoft critics.¹⁹

In 1995, Microsoft announced that it would buy Intuit, the maker of the financial program Quicken and one of the few independent suppliers of an application that had a commanding market share. After initiating the purchase (and, critics

charged, learning the techniques of Intuit’s best programmers), the acquisition was dropped when the Department of Justice objected. Clearly sentiment was building up against Microsoft.

That brings us to the fall of 1997 and Internet Explorer, version 4.0. For Netscape the case was complicated. Bundling Internet Explorer implied that Microsoft was guilty of a tie-in, making it impossible for Netscape to sell its browser. But when in December 1994 Netscape posted a preliminary version of its Navigator on a Web server, users could download it for free.²⁰ In fact, the trade press called that a brilliant marketing innovation by Netscape. By giving away the browser, Netscape would “lock in” customers who, from that moment onward, would be captive to a Netscape standard. The browser was free to individuals. Businesses were charged a modest licensing fee. The company assumed that once they established their browser as a standard, everyone would pay for other Netscape products that adhered to it.

For a while the strategy worked brilliantly. So great was the interest in Web browsers in general, and Netscape in particular, that it was able to offer shares to the public in August 1995 before the company was profitable. Economists wrote of the way Netscape exploited the phenomenon of “lock-in,” and of the “tipping point” where one standard overtakes over another—and these mattered more than whether or not Netscape was charging for its products.²¹ The soaring price for the stock made multimillionaires of its employees (on paper at least). The Internet madness began.

Microsoft was focused on the introduction of Windows 95, but it was aware of what was happening to the Internet. Microsoft’s critics were gloating over how, in his book *The Road Ahead* published in 1995, Gates missed the biggest thing on that road, namely the World Wide Web.²² The critics were wrong: the book frequently describes a future based on networked computers. It just got the details about the Web wrong. Most critics failed to note the passages in *The Road Ahead* where Gates wrote of how IBM and Digital Equipment Corporation failed to sense a sea change in computing and stumbled badly.²³ Gates implied that Microsoft faced the same risk.

As the release of Internet Explorer 4.0 approached, the company’s public relations apparatus kicked into high gear and touted the story of how Gates, unlike his “hero” Ken Olsen at DEC, listened to the message of his troops in the field.²⁴ The stories told of how Microsoft recruiters visited college campuses and found students and professors conducting their coursework by email and file transfers. Internal memos and transcripts of speeches, released to the press, revealed a fear among Microsoft employees that a properly-designed Web browser could replace the Windows desktop that users saw when they turned their computers on. On December 7th and 8th, 1995, Gates spoke with analysts, to whom he declared “the sleeping giant has awakened”—a reference to the American response to the Japanese attack on Pearl Harbor 54 years before.²⁵ Microsoft won a key battle in

March 1996, when America Online agreed to provide its customers with Web browsing through Internet Explorer instead of Netscape's browser.

From this point on, the sequence of events gets murky, and it is impossible to summarize in a few pages what has taken the courts several years, unsuccessfully, to straighten out. A shelf of books has already appeared on the trial, and once it is settled there will be more. The simplest description of the charge against Microsoft is that Internet Explorer (IE) 4.0 violated the Consent Decree by being tied too closely to Windows. That is, personal computer users, who by 1997 were overwhelmingly using Windows, could not easily access the Web using Netscape's or any other browser, but rather were steered too strongly to IE. A corollary to the charge was that by bundling IE into Windows, the browser was essentially free, for individuals as well as for business customers—an action that Microsoft took primarily to cut off Netscape's major source of revenue.

In its defense, Microsoft made blunders that led prosecutors to resurrect charges that otherwise might have remained buried—charges of holding back the details of APIs from third-party developers, for example. Microsoft's stormy relations with IBM during the development of OS/2 also resurfaced. In the fall of 1997 Microsoft executive Steve Ballmer blurted out, in a speech to employees, "to heck with [Attorney General] Janet Reno!" In the summer of 1998, Gates gave a deposition on video, in which he appeared nervous, evasive, and inarticulate—the polar opposite of the confident public image he so carefully cultivated. His evasiveness drew parallels with a video deposition by President Clinton, who was then sparring with lawyers over his involvement with a White House intern named Monica Lewinsky. A good portion of the trial was devoted to the question of whether one could remove the IE icon from the Windows desktop, and whether such a removal, if it could be done, implied that Microsoft was following the Consent Decree against a tie-in.²⁶ During the discovery process, internal email messages residing on Microsoft's company servers were introduced into the record, causing further embarrassment.

The prosecution made blunders, too. The worst was an interview by Judge Thomas Penfield Jackson, in which he flatly stated his prejudice against Microsoft. That was enough to get most of his judgment overturned in June 2001, and to have Jackson removed from the case.

Historians cannot afford to wait for things to settle down before writing a chronicle of these events, but historians do have an obligation to assert independence from the findings of the courts. Historians did this, with some difficulty, when they wrote about the invention of the electronic digital computer itself, an invention that the courts have credited to John V. Atanasoff. Atanasoff's invention happened 60 years ago. Regardless of how the Microsoft trial is settled, some historical themes still apply. One clue may be found in a statement by Judge Jackson, made in November 1999, that "There exists no commercially viable alternative [to Windows] to which [customers] could switch in response to

a substantial and sustained price increase or its equivalent by Microsoft.”²⁷ That sounds too close to the statement made by a government economist after the government dropped its suit against IBM in the 1980s.

From the perspective of the past fifty years of computing, one could conclude that Microsoft’s brave attempt to avoid the pitfalls that caught DEC and IBM may give it some breathing space but not for long. It has no choice but to accommodate itself to the Internet and its accessibility to the consumer via the World Wide Web. I will refrain from further speculation, but I do wish to mention one example that illustrates what is happening.

Hotmail, UNIX

Recognizing the threat of networking, Microsoft introduced a proprietary network, MSN, in 1995, and a “groupware” communications system, Exchange, in 1996. These were aimed at America Online and Lotus Notes, respectively. It also introduced a full-featured email program called Outlook. But as the Web exploded, Microsoft had to face a new threat: the advent of free services like Yahoo! that offered mail, news, chat, and a friendly on-ramp (called a “portal”) to the Information Highway. In 1997 Microsoft purchased (for \$400 million) a mail program called Hotmail to counter this threat. Hotmail was already growing rapidly and soon became Microsoft’s biggest presence on the Web.²⁸ MSN was reconfigured to be an Internet portal rather than a proprietary network, and Microsoft’s market share for these services began to grow rapidly. Not only was Hotmail free, it was only loosely coupled to Windows. And it ran on UNIX machines, not Windows NT, and continued to after Microsoft bought it. Thus even Microsoft violated the sacred dictum of “eating your own dog food.”²⁹

The parallels with IBM’s introduction of the Personal Computer, with its Intel processor, Microsoft software, ASCII codes, and an open architecture should be obvious. Microsoft derives most of its revenues from successive versions of Windows and a few applications tightly coupled to it; thus it will have to adapt or lose its place as the leading personal computer software company. Even if Microsoft does adapt successfully, it will be a different company. Again, consider the example of IBM, which under the leadership of Louis Gerstner successfully adapted to the changing computing field after 1991. IBM today is a successful and profitable company, but it is not the mainframe company it was in 1980, and it no longer dominates and controls the computer industry as its critics charged it would after the lawsuit against it was dropped.³⁰

Linux

In 1988 the late Seymour Cray gave one of his rare public appearances, in which he described plans to build a new supercomputer with chips made from gallium arsenide. A member of the audience asked him what it was like working with this unusual material, whose properties were far less understood than those of silicon, to which Seymour is said to have replied, “Well, if you can pronounce it, I guess that’s the hardest thing about working with it.”³¹

That also applies to Linux, which since 1995 has been at the center of a highly-visible movement to provide an alternative operating system to Microsoft’s Windows, and to break out of the Babel of dialects that was threatening Unix after 1990. Linux is the creation of Linus Torvalds, born in 1969 in Finland, where his name is pronounced “Lee-noose.” The name is not rare in Finland, but his parents named him after the American biologist Linus Pauling, whose name (like Snoopy’s friend) is pronounced with a long *i*. As for the operating system itself, Linux adherents say it is pronounced “Lih-nooks,” but many also pronounce it “Lie-nux,” or with a short *i*: “Lin-ux.” Hard-core Linux fans scoff at anyone who pronounces it “wrong,” but since one of Linux’s appeals is that it is an “open source” program (more on that later), they can only exclude newcomers so far. Pronouncing gallium arsenide is easy by comparison.³²

Leaving that issue unresolved for the moment, what is Linux and why all the interest in it? A little backtracking into the history of UNIX will be helpful here. UNIX first took form at Bell Labs in 1969 as a set of basic file management tools, initially written for the Digital Equipment Corporation PDP-7 and later PDP-11 minicomputers. The commands that actually moved data were machine-specific, but after Ken Thompson and Dennis Ritchie rewrote it in the programming language “C,” UNIX could be moved over (“ported”) to other computers from different manufacturers, as long as someone had written a C compiler for those machines. And because of AT&T’s position as a regulated monopoly, UNIX quickly spread to locations outside Bell Labs, especially in universities.

From its origins, UNIX acquired characteristics that have set it apart from all other programming systems, and these characteristics are crucial to an understanding of Linux. One was social, not technical: the wide use of UNIX in universities and other non-commercial locations outside Bell Labs. It grew by a cooperative effort of researchers scattered widely, who worked with an existing version of UNIX as they built improvements: thus an early instance of “eating your own dog food.” Thompson and Ritchie said,

If the designers of a system are forced to use that system, they quickly become aware of its functional and superficial deficiencies and are strongly motivated to correct them before it is too late. Because all source programs were always available and easily modified online, we were willing to revise and rewrite the system and its software when new ideas were invented, discovered, or suggested by others.³³

The technical characteristics include a “kernel” that consists of only the most essential commands to move data and control the processor. UNIX programmers are able to build up complex and sophisticated operations from combinations of simpler commands, primarily by allowing the output data of one program to serve as the input data for another (using so-called “pipes”). Files are stored in UNIX with a minimum of structure or formatting: “No particular structuring is expected by the system...the structure of files is controlled by the programs that use them, not by the system.”³⁴ A final characteristic is the system’s reliance on the C programming language and the use of C compilers. Those who work with UNIX speak of the elegance of its initial design. Ritchie and Thompson modestly point to the physical constraints of the PDP-7 that they had to use, a computer whose power was about equal to a Commodore VIC-20, one of which Linus’s grandfather let him play with as a child. They clearly put a lot of careful thought into the system’s design.³⁵

UNIX was created and nurtured on minicomputers and became the preferred operating system for workstations. On IBM compatible personal computers, which after 1985 were being sold in greater numbers, the situation was different. In 1980, before Microsoft acquired DOS for the IBM PC, it developed a version of UNIX called XENIX. Microsoft hoped it would become a standard for 16-bit processors.³⁶ XENIX sold well, but the industry evolved in a different direction. The portability of UNIX was less important once the IBM PC drove competitors from the personal computer market. And XENIX required more memory than MS-DOS. Thus personal computers standardized around MS-DOS, written by Tim Paterson of Seattle Computer Products. Later versions of DOS incorporated some of UNIX’s features, such as the tree-like structure of ordering files, and even something like pipes. Other features, especially the all-important multitasking, either never got into DOS or else had to wait for Windows, where they were grafted on in an inelegant fashion. The way that DOS stored files was also different. In other respects the two systems had much in common, especially in the way users typed in brief commands at a “command line” to move, copy, delete, or port files. Some of the commands, such as “mkdir” to create a directory, were identical in both systems. (In the fall of 2001 Microsoft officially retired DOS when it introduced a version of Windows that, finally, did not have DOS code at its core.)

In 1991, Linus Torvalds set out to write a version of UNIX for his IBM-compatible personal computer.³⁷ The details of how he came to do that are recorded in his memoir of the era, and are summarized briefly here. The previous year he had taken a course at the University of Helsinki on UNIX. The instructor, he said, was as new to the subject as the students, but that did not matter, as the enrollment in the course gave him access to a version of Digital Equipment Corporation’s UNIX running on a time-shared VAX. Linus was already familiar with UNIX from reading a book on operating systems by Andrew Tanenbaum, of Vrije University in Amsterdam. Tanenbaum had also developed a teaching

version of UNIX, which he called “Minix.”³⁸ Reading that book, combined with a childhood spent hacking on personal computers like the Sinclair QL and the Commodore VIC-20, convinced Torvalds that he wanted not only to learn all he could about UNIX, but also that he wanted Minix on his home computer. The Sinclair used an advanced Motorola processor, but Torvalds recognized that the IBM-PC standard was becoming so well-established that he had to switch to the Intel-based architecture. Just at that time, PCs began to appear on the Finnish market based on the Intel 80386 chip—a chip that enabled personal computers to offer performance that was competitive with minicomputers and workstations.³⁹

Torvalds bought (on the installment plan) a 386-based PC, and a version of Minix for it about a month later. According to his memoir, almost as soon as he had it installed and running, in the winter of 1991, he found Minix wanting. Specifically, he needed a terminal-emulation program, which would allow him to use the PC as a terminal to access the University’s computer, with its software resources and access to online discussion groups.⁴⁰ Rather than write it as a process under Minix, he wrote an emulation program of his own, programming “to the bare metal” as he described it. From that beginning eventually came a version of UNIX that was separate from Minix, and which was not crippled or restricted in any way, as he felt Minix was. A brief note posted to a newsgroup in July 1991 gave a hint to the world that he was thinking not just of an expanded terminal emulator but a 386 implementation of UNIX that would conform to a standard set out by a subcommittee of the International Standards Organization.⁴¹ One person responded to his query not with information on the standard but with an offer of space on a computer at the Helsinki University of Technology, where Torvalds could post versions of his work and where others could download it via the File-Transfer Protocol (FTP).

Another brief note to the discussion group in August 1991 went further: “...I’m doing a (free) operating system (just a hobby, won’t be big and professional like gnu) for 386 (486) AT clones... I’d like any feedback on things people like/dislike in minix, as my OS resembles it somewhat ...”⁴² Note that he was still thinking in terms of addressing the deficiencies that he saw in Minix. The next month he posted his work on the FTP site, and at the suggestion of Ari Lemke, who made the site available, the program was called “Linux.” As Linux got legs of its own, Torvalds thought less and less of it as a derivative of Minix and more of it as a new version of UNIX. That led to a break with Tanenbaum, who registered his disapproval of Torvalds’s approach in a posting to the newsgroup in early 1992. Torvalds responded with a vigorous defense of his work, accompanied by “flaming” rhetoric that was typical of online discussions in those days.⁴³ (The discussions had been going on in the “comp.os.minix” newsgroup; after a heated exchange with Tanenbaum, Linux discussions moved to a newsgroup of its own.)

Meanwhile Torvalds kept working on the program, with encouragement from people on the discussion list. Like Marc Andreessen at Netscape, Torvalds

exploited the Internet’s ability to distribute his work cheaply around the world, and receive almost instant feedback from enthusiastic users. No classical theories of marketing could have foreseen the close relationship between creator and user that Torvalds developed.

In spite of the various flavors of UNIX then available, Linux was filling a need. The fragmentation of UNIX after the breakup of AT&T in the early 1980s turned to Torvalds’s advantage. His version allowed users to get the operating system with none of the baggage associated with other versions then available. AT&T had hoped to profit from its creation once the company was free to market it; but its marketing was ineffective. AT&T did manage to annoy people with its insistence on owning the name “UNIX,” leading to those other names ending in “IX,” spelling it “UN*X,” or other silliness. Worse than that, the company sued vendors who were selling variants of UNIX that had portions, however small, of AT&T’s code in them. The Berkeley distributions of UNIX were less restricted, owing to the U.S. government’s support through ARPA, but even SUN moved to a closed version (Solaris) for its later generations of workstations. In 1993 AT&T sold its version of UNIX to Novell, who resold it shortly thereafter. Berkeley distributions of UNIX have evolved into several versions, some of them freely available.

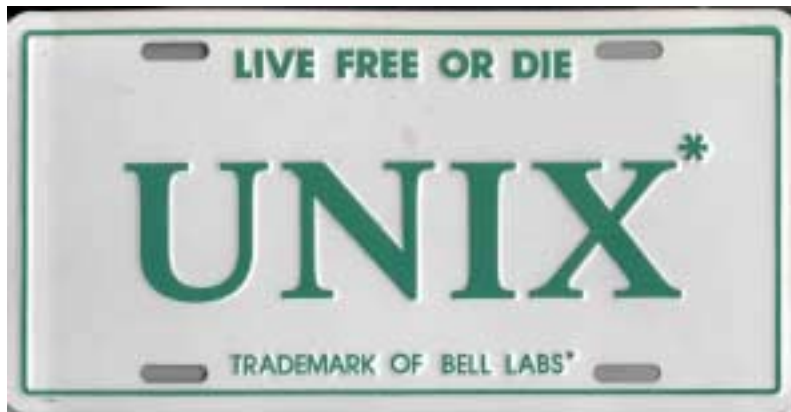


Figure 1: UNIX License Plate, replica. The plates were produced in 1983 by Armando Stettner, a DEC employee and resident of New Hampshire, whose state motto appears on its license plates. Stettner was the leading evangelist for UNIX at DEC, where he faced resistance from Dave Cutler, the architect of the VMS operating system and later of Microsoft’s Windows NT. Note the obligatory credit to Bell Labs, the result of a letter from an AT&T lawyer to Stettner.

Smithsonian Institution collections, gift of Eugene Miya.

For the next decade, Linux thus grew in size and quality. Around it grew a cadre of enthusiasts, for whom Linux was much more than just an operating system. To them, the successful development of Linux by dedicated volunteers working around the globe represented both a vindication and a refutation of the tenets of software engineering set out by Fred Brooks in his classic book, *The Mythical*

Man-Month (Addison-Wesley, 1975). Recall that Brooks wrote that book to explain the difficulties he had, as project manager, in developing an operating system for IBM's System/360 line of mainframes. He postulated "Brooks's Law": adding people to a software development team in the middle of a project actually hinders progress, as the difficulties of coordination among its members overwhelm any contributions the new members can make. To the growing numbers of Linux enthusiasts, that seemed to be precisely what Linus Torvalds was doing: he enlisted the help of users around the world, and gradually turned over portions of the project to those among that group whom he trusted.⁴⁴ Brooks suggested that one should organize a large software project around a leader with strong technical skills who is given proper authority. That certainly applies to Linus Torvalds, even if not as Brooks had in mind. Torvalds has no authority in any formal sense, yet he is able to act and speak with authority on the evolution of the system that bears his name. Brooks argued that large projects need both a "producer" and "technical director": roles which in Linux development are filled by Alan Cox and Torvalds, respectively.⁴⁵ In any event, it worked, and it continues to work as Torvalds moved from Finland to Silicon Valley, where he works at a company called Transmeta. Cox, a year older than Torvalds, lives in Wales but is employed by the Linux company Red Hat of North Carolina.

Whether the development of Linux is a vindication or refutation of Brooks's Law is of central importance to the Linux community. Fred Brooks's writings are respected, but *The Mythical Man-Month* is fundamentally a book about a project that failed. Linux fans come to his writings with an agenda: to demonstrate that what they are doing will not fail as IBM's OS/360 did. But Linux enthusiasts are less concerned with IBM than they are with the products of another company: Microsoft. Microsoft's software development techniques evolved quite differently from IBM's, and they owe something to the early development of UNIX, as mentioned above. During the joint IBM-Microsoft development of OS/2, Gates often expressed his disdain for the cumbersome ways IBM worked. Now, Linux advocates believe that their work will produce better quality software than what Microsoft can produce. Brooks himself added several chapters to the 1995 re-issue of his book, and in it he discussed at length the evidence both for and against his theories of how teams develop large-scale software projects. Although he discussed the impact of commercial PC software, at that time still sold in "shrink-wrapped" packages, the edition was published before Linux became well-known. (Brooks did mention UNIX as an example of an "exciting" project whose qualities stem from having one or a few dedicated individuals involved in its creation.)⁴⁶

Eric Raymond was among those programmers who saw the merits of this model of software development, and in an influential essay called "The Cathedral and the Bazaar," he analyzed it with regard to writings by Brooks and others about large software projects.⁴⁷ Raymond and others argued that by letting people look at, and modify, the source code, bugs are found and fixed. As they do that, the general level of quality rises, much faster than it possibly could in a closed

system. As Raymond's said (paraphrasing Torvalds), "Given enough eyeballs, all bugs are shallow."⁴⁸ Raymond's philosophy recalled the spirit of the Homebrew Computer Club of the 1970s, where sharing was the order of the day. It recalls an even earlier time, in the 1950s, when IBM 704 programmers banded together to form SHARE, even if the individual members worked for competing companies.

The Homebrew spirit of sharing software was opposed by Bill Gates in his famous "Open Letter to Hobbyists," written in 1976.⁴⁹ Twenty-five years later those arguments have resurfaced, and Microsoft again plays the role of the "enemy." In a few public speeches, Microsoft executives railed against the philosophy of making source code available, arguing as Gates did in his Open Letter that "free" software would not properly reward talented programmers for their hard work. Linux is free to the consumer, but who pays for its development is not a simple matter. Most members of the Linux development community have "day jobs," where their employers allow them to work on Linux as a perk, in hopes of keeping their talents available for developing commercial products. With the recent downturn in the economy, some of those employers may be less generous. Other free software is supported by the publisher O'Reilly & Associates, which makes money-selling books about UNIX. The commercial company VA Linux also supports this effort, but recently VA Linux announced that it was shifting its business model to proprietary products (and changing its name to VA Systems).

Microsoft executives also made the reasonable argument that nothing would prevent the system from careening off in a dozen directions—a likely scenario that is only prevented by the hard work and talents of Linus Torvalds and his inner circle of collaborators. Whether Linux can succeed without the dedication of Torvalds and Cox is an open question. An internal memo, written by someone in Microsoft in 1998, was leaked in October to Eric Raymond, who promptly posted it on the Internet where it became known as the notorious "Halloween Document." In it, Microsoft allegedly laid out plans to make it difficult for Windows users to install or use Linux on their machines.⁵⁰ Torvalds's claims that he was never motivated by political considerations, nor did he feel any animosity toward Microsoft, other than believing their code was technically not as good as his. In any event, Linux grew on the shoulders of a worldwide, voluntary group of enthusiasts.

GNU

Any discussion of UNIX must include a discussion of the C programming language, which was developed in tandem with it, and in which UNIX was rewritten. Thus one of the first things Linus Torvalds did, in the fall of 1991, was adapt a C compiler for use on his machine. It was not an easy task but once it was done he would gain access to a library of C programs, which allowed him to bootstrap his way to a more developed UNIX system. The compiler he chose was

GCC, the “GNU C Compiler,” which had been written by Richard Stallman (b. 1953) of Cambridge, Massachusetts.⁵¹

Linus Torvalds’s relationship to Richard Stallman, like his relationship to Andrew Tanenbaum, is a complex one. Torvalds relied on work done by the older programmers, but he deliberately set off in a direction different from what either Tanenbaum or Stallman would have preferred. Nevertheless, when it came to adopting a C compiler, Stallman’s was one of the best available, if not the best. Its widespread adoption established a *de facto* standard for the C programming language; i.e., a “standard C” was defined as a language that the GNU C Compiler understood. And it was free. *Very* free.

Richard Stallman’s campaign to develop and spread software that is “free” in a carefully-defined way is by now well-known.⁵² Stallman’s odyssey began at the MIT Artificial Intelligence Laboratory, the home of a PDP-10 and the legendary “Incompatible Timesharing System.” In the early 1980s he felt that the ethos of the Lab, in which programmers freely shared their work, was being drained away by the founding of a company that sought to commercialize some of the AI research being done there. Stallman resolved to produce similar software, with the distinction that it would be freely distributed. The business model of commercializing AI research was flawed anyway (especially after ARPA withdrew financial support around 1985), but Stallman’s resolution remained. He decided to create a free version of UNIX, an operating system he knew little about, and give that away.⁵³ He would have preferred copying ITS, the system used on the PDP-10, but UNIX had the advantage of being used on more than just one or two machines from a single manufacturer. He called his UNIX copy “GNU” (pronounced as a two-syllable word): a recursive acronym for “Gnu’s Not UNIX.”

In a Usenet post in September 1983 he wrote: “Starting this Thanksgiving I am going to write a complete Unix-compatible software system called GNU (for Gnu’s Not Unix), and give it away free to everyone who can use it. Contributions of time, money, programs and equipment are greatly needed.” Later in the same posting he stated his philosophy of free software: “I consider that the golden rule requires that if I like a program I must share it with other people who like it. I cannot in good conscience sign a nondisclosure agreement or a software license agreement. So that I can continue to use computers without violating my principles, I have decided to put together a sufficient body of free software so that I will be able to get along without any software that is not free.”⁵⁴ His definition of “free” would change but would always retain the notion that free software would never have a restrictive license attached to it.

Stallman found that some software components of a UNIX system were already available free—the X-Windows system and Donald Knuth’s TeX typesetting program, among others. To those he added the Emacs text editing system he had already written while at MIT, which he rewrote for UNIX, and of course, the C

compiler.⁵⁵ To avoid legal problems with AT&T, he refused to look at any source code for AT&T UNIX, just as those who recreated the BIOS for the IBM Personal Computer worked in a “clean room.” Thus the Gnu C Compiler (gcc) was offered as a replacement for “pcc,” a compiler written by AT&T’s Steve Johnson and offered for a price. Stallman’s “Bison” parser generator (a component of a compiler) was offered as a free replacement for “yacc” (“yet another compiler compiler”) also written by Steve Johnson. Note the pun on the word “yak”; Stallman follows an old hacker tradition of punning whenever possible.⁵⁶ He did much of this himself, in an office generously loaned to him by MIT. For a while that was also his sleeping quarters. (As with Linux, GNU benefited from an informal but nevertheless real generosity from those who had access to computing resources.) He suffered from carpal-tunnel syndrome from keyboarding. Some observers skeptically compared his effort to building a commercial jet airliner in your basement workshop. With the help of a few colleagues, Stallman began building up a body of software that was comparable to, or even better than, what it took large teams of people to create in the commercial or academic world.

Just as important was the legal agreement he crafted to guarantee the rights to his work. With the help of an attorney he developed a “GNU General Public License” (GPL) that not only put his work into the public domain, it also required that those who used and modified it put *their* modifications in the public domain as well. In his words, “It is a legal instrument that requires those who pass on a program to include the rights to use, modify, and redistribute the code; the code and the freedoms become legally inseparable.”⁵⁷ It did not prevent someone from selling the code in a packaged product, as Red Hat does. The GPL does, however, prevent Red Hat from owning the code portion of what they sell.⁵⁸

It is this last provision that was so radical, and it defines the character of Linux (which Torvalds released under the GPL) and other so-called “open source” software. “Free” software, defined by its specifications, was nothing new: Microsoft began with the BASIC programming language, developed at Dartmouth but modified by Gates, Allen, and Davidoff. The creators of BASIC wanted their program to become widespread and publicly described its specification with that in mind. Microsoft’s version of BASIC for the Altair was very different from Dartmouth’s BASIC, something that Kemeny and Kurtz did not approve of, but such modifications of existing languages were common. Other personal computer software, including dBase II and some word processors, were modifications of software developed at universities or with government support and therefore considered “free.” But once modified, their creators could and did sell them for a profit. One cannot do that with Linux, with GNU tools, or with other software under the GPL. What is more, the GPL requires that if one uses such software in a product, the *entire* product must be covered by the GPL, even if other parts of it had previously been proprietary. The GPL thus has the potential to “convert” proprietary software into free software, the opposite of what had been happening in the industry. It is this provision which is so threatening to companies like

Microsoft, and which led senior executives at Microsoft to denounce the movement as being contrary to the “American Way” of free enterprise.⁵⁹

Many popular accounts of the history of Linux emphasize the rift between Linus Torvalds and Richard Stallman, who are only a few years apart in age but represent different generations in many ways. Such accounts neglect the fact that Torvalds completed what Stallman set out to do, and Linux is protected by the GPL. Stallman ran into difficulties in developing a UNIX kernel, which was the first thing that Torvalds wrote. Stallman came from an environment of DEC mainframes like the PDP-10 and minicomputers like the PDP-11; he may not have recognized how the Intel 80386 and successor chips were taking over. Or there may have been other reasons, related to the underlying philosophy of how to write a good kernel. In any event, Stallman reminds people, sometimes to the point of annoyance, that the full set of UNIX tools should be called “GNU/Linux,” not just “Linux.” On that he is correct. Some advocates of the GPL philosophy coined the term “open source,” mainly to distance themselves from him personally. Their commitment to the GNU Public License however is as strong as anyone’s. Software that is in the public domain but not covered by the GPL is, strictly speaking, not a part of this social phenomenon.

This activity might have gone unnoticed had it not been for the explosion of Internet activity after the creation of the World Wide Web and the Netscape browser. Suddenly there was a demand for larger computers to run Web sites and route Internet traffic. UNIX fit that need perfectly. Linux, and a free version of Berkeley UNIX that had all the AT&T “cleansed,” became the systems of choice. Each offered Web administrators an ability to modify and extend their systems without legal troubles or charges. As of this writing, the most popular Web server software is Apache (“a patchy server”), coordinated by Brian Behlendorf; the most popular mail-routing program is “Sendmail,” written by Eric Allman; and the most-used scripting language for Web pages is Perl, written by Larry Wall. All are based around UNIX, and all are free. Web giants like Google rely on Linux, and as mentioned, Microsoft’s Hotmail service runs on the free Berkeley Distribution.

IBM

Apple’s 1984 Super Bowl commercial announced that the future would be nothing like Orwell’s vision, but in one respect Apple was wrong. Orwell predicted a world where today’s enemy becomes tomorrow’s ally, and as that happened, all memory of the previous alliance or warfare was erased. That was Winston Smith’s job: to put all records of previous alliances down the “memory hole.” In 2001 IBM, the implied enemy in the Super Bowl commercial, became an ally; Microsoft is the enemy. The trade press dutifully does Winston Smith’s job of erasing everyone’s memory of the relative roles of those two entities 20 years ago.

IBM has embraced Linux as an alternative to the closed, proprietary operating systems it so jealously guarded in its days of “Big Iron.” Throughout its reign as the dominant computer company, IBM was known as a company that set its own standards—think of EBCDIC instead of ASCII—and tried to impose them on the rest of the computing community. But IBM suffered huge losses in the early 1990s and might have vanished or shriveled into insignificance, along with the rest of the BUNCH (Burroughs, Univac, NCR, Control Data, and Honeywell), had it not changed. The company’s embrace of Linux is too recent for a coherent story to emerge, but it appears that it began as a “Skunk Works” project at IBM’s Böblingen, Germany, lab outside Stuttgart, where in late 1999 a team of young programmers succeeded in porting Linux to an IBM 390 mainframe. That was not IBM’s first exposure to open source software. For the 1996 Summer Olympics in Atlanta, IBM was the prime contractor in providing computer services, including a Web presence. Instead of “eating your own dog food” and using a server offered by the IBM subsidiary Lotus, it chose Apache instead.⁶⁰ IBM was in trouble financially, it knew that the world would be watching how well it handled the Olympics, and it had more faith in Apache than in the Lotus product.

Nor was the notion of a Skunk Works all that foreign to IBM either. It had always supported such activities, going back at least to the famous “wild duck” memo from Thomas Watson, Jr. Perhaps senior executives remembered how software developed that way at IBM’s Cambridge, Massachusetts, laboratory stepped into the breach when official versions of System/360 software came in late or did not work. What was startling is how this effort made its way through the layers of managers and programmers committed to the classic IBM way, until it gained the attention and approval of the Chairman, Lou Gerstner. According to one account, the Linux effort was stuck in a vortex of memos, meetings, and committees when John Markoff, a reporter for the *New York Times*, reported in December 1998 on IBM’s plan to release a mail program developed by one of its staff as Open Source.⁶¹ Gerstner allegedly read the story and demanded that IBM develop a coherent policy on Open Source from that day onward. In any event, IBM made a substantial commitment to Linux and announced that for the year 2002 it will devote 20 percent of its R&D budget toward getting Linux to run on its product line.⁶² In a series of ads in the trade press, IBM announced: “The facts are clear: Linux is here and Linux is ready. Ready for business, Ready for e-business. Ready for enterprise.” One ad showed a fuzzy black-and-white image of a penguin, the Linux mascot, walking through the towers of a mainframe installation, with the caption, “Fact: Linux in the enterprise.” Is it coincidental that Sasquatch, the elusive man-beast of lore, is periodically sighted in the woods of the Pacific Northwest, where Microsoft employees live?

Perhaps another reason for this embrace of Linux was IBM’s experience with Microsoft during the development of the personal computer operating system, OS/2. By most accounts, Microsoft got the best of that deal, and OS/2, the product of a great effort by a large team of IBM programmers, was forgotten. Or

perhaps the reason was simply that Linux offered the best set of capabilities. This story is still developing. It is not clear that Linux will prevail, but since those ads appeared IBM has gone even further and introduced a line of computers that run only Linux.⁶³ In the press release IBM claimed that Linux systems account for eleven percent of the computing capacity that it shipped in late 2001, based on millions of instructions per second. All of this happening ten years after Linus Torvalds began writing a terminal emulator for his PC.

Conclusion

Seymour Cray never succeeded in building gallium arsenide circuits that could compete with silicon. Linux, however it is pronounced, is going to have to deal with Microsoft one way or another. Simply being an alternative to Microsoft is not sufficient in itself to prevail. Among Linux evangelists are a strong and vocal group who tout Linux-based programs that offer a graphical interface like Windows (“KDE” and “GNOME”), word processors (“AbiWord”), and other products.⁶⁴ In keeping with the UNIX philosophy, and in contrast to Windows, the code that generates the graphical user interfaces is kept separate from the base Linux code. Linux is still accessed by typing at a command line, like DOS of old. As Microsoft moved away from DOS, Linux enthusiasts steadfastly prefer typing cryptic commands at the command line. Apple seems to be of two minds on this. When it introduced the Macintosh in 1984 it got rid of the command line, but with the latest version of the Mac operating system (“X,” based on UNIX), a savvy user can bypass the graphical interface that Apple made so famous.

The Web site “Slashdot.org” posts daily messages on the battle against Microsoft, the tone of which suggests that flame wars are not extinct after all. But not all messages are in favor of this approach: Rob Malda, one of the founders of Slashdot who goes by the screen name “CmdrTaco,” recently took a less aggressive stance. And Russ Mitchell, who worked for Red Hat, is even more skeptical. He argues that going against Microsoft head-to-head is a waste of time; Microsoft has won this battle. He hopes to see Linux establish a stronger position in the server market, as IBM has done. If well-executed, Microsoft might be unable to threaten that market.⁶⁵ For its part, Microsoft is not going to let this happen without a fight.

Linux evangelists might learn from the experience of Marc Andreessen, when he was touting the Netscape Navigator as a competitor for Windows. In an interview he described Windows as “a partially-debugged set of device drivers.”⁶⁶ Bill Gates and Steve Ballmer did not think that was funny. Today, Netscape is buried in a corner of America Online.⁶⁷ Neither Andreessen nor Jim Clark have been forthright about why Netscape ultimately lost the browser war to Microsoft, but the hubris of statements like that one did not help. Someone should have reminded Andreessen of the folk wisdom, “you don’t tug on Superman’s cape.” Unless you are IBM. In any event, what started out as a footnote to the Microsoft antitrust trial, something that Linus Torvalds claimed was “...just a hobby, [and]

won't be big and professional..." is turning out to be quite interesting. We shall see.

Paul E. Ceruzzi, "A War on Two Fronts: The U.S. Justice Department, Open Source, and Microsoft, 1995-2002," *Iterations: An Interdisciplinary Journal of Software History* 1 (September 13, 2002): 1-26.

¹ MacWrite had full WYSIWYG capabilities, which set it apart from the IBM PC and its clones, but MacWrite users could not write documents longer than ten pages. See Frank Rose, *West of Eden: The End of Innocence at Apple Computer*, (New York: Penguin Books, 1989), chapter 11.

² Stephen Manes & Paul Andrews, *Gates* (New York: Doubleday, 1993), Chapters 12, 13.

³ Susan Lammers, *Programmers at Work* (Redmond, Washington: Microsoft Press, 1986), pp. 6-22.

⁴ Lammers (1986), pp. 207-225. Figures for the relative sizes of Microsoft, Lotus, and Ashton-Tate may be found in Martin Campbell-Kelly, "Not Only Microsoft: the Maturing of the Personal Computer Software Industry, 1982-1995," *Business History Review* (Spring 2001), p. 109.

⁵ The so-called "WIMP" interface: Windows, Icons, Mouse, and Pull-down-menus. The reasons it prevailed over the integrated single program are complex and probably have as much to do with social forces as with technical superiority.

⁶ Lammers (1986), p. 13.

⁷ Frederick P. Brooks, Jr., *The Mythical Man-Month; Essays on Software Engineering. Anniversary Edition* (Reading, MA: Addison-Wesley 1995), pp. 270-271.

⁸ Michael A. Cusumano and Richard W. Selby, *Microsoft Secrets* (New York, Free Press, 1995), pp. 36, 269-270. The authors note that Version 3.0 of Excel was the first Microsoft product to employ this tactic.

⁹ G. Pascal Zachary, *Showstopper! The Breakneck Race to Create Windows NT and the Next Generation at Microsoft* (New York: Free Press 1994), Chapter 1.

¹⁰ Microsoft programmers who owned dogs probably fed them well, but the poor animals must have been starved for affection. At rival Netscape, programmers were allowed to bring their dogs to work, an amenity the company was very proud of. Terms like "Death March" and "broken" are also peculiar to Microsoft and its peers among software developers.

¹¹ Access had its roots in a database offered by Fox Software, which Microsoft purchased in 1992.

¹² For an amusing story of how Power Point was first developed and then acquired by Microsoft, see Ian Parker, "Absolute PowerPoint," *The New Yorker* (May 28, 2001), pp. 86-87. According to Parker, "Today...there are great tracts of corporate America where to appear at a meeting without PowerPoint would be unwelcome and vaguely pretentious, like wearing no shoes" (p. 78).

¹³ Data taken from a number of sources, including Cusumano, Campbell-Kelly, Manes & Andrews, (all op. cit.), and a clipping file in the possession of the author

of selected articles from *Infoworld* and *PC Week*, 1990-1994. In some cases the dates are approximate, reflecting the difference between the announcement of a product and its actual availability to consumers.

¹⁴ Adam Osborne and John Dvorak, *Hypergrowth* (Berkeley, CA: Idthekekthian Publishing, 1984), pp. 162-165; also Manes & Andrews, *Gates*, pp. 360-361. Martin Campbell-Kelly has pointed out that the pricing of programs like Lotus 1-2-3 in the range of \$350 - \$550 was not based on any classical models of economics. Few economic theories applied to PC software.

¹⁵ Jobs quoted Picasso, who allegedly said, “Great artists steal.” Some critics of Jobs claim that Picasso never said that. As if that were enough, Jef Raskin claimed that Jobs’s boast was idle; he did not steal from Xerox after all! Only Steve Jobs can have his reputation tarnished by charging that he is not a thief.

¹⁶ As that litigation proceeded, Apple and Microsoft entered an agreement to license a technology for displaying and printing type fonts, which later became a fundamental feature of Windows.

¹⁷ Jerry Kaplan, *Start Up: A Silicon Valley Adventure* (New York: Houghton Mifflin, 1994), chapters 8, 9.

¹⁸ After several false starts including GO’s Penpoint and Apple’s Newton, pen-based computing gained a market foothold with the Palm Pilot, introduced by a start-up called Palm Computing in 1996. Microsoft countered with its own pen-based operating system that was not compatible with the Palm, and at present the two competing systems have about equal market share.

¹⁹ Kaplan, *Start Up*, pp. 178-181.

²⁰ Joshua Quittner and Michelle Slatana, *Speeding the Net: the Inside Story of Netscape and how it Challenged Microsoft* (New York: Atlantic Monthly Press, 1998), pp. 172-174.

²¹ These concepts appear initially to have descended from a paper by the economist Paul David, on the reasons for the persistence of the QWERTY typewriter keyboard. Its extension to things like browsers and computer operating systems has been analyzed in Stan J. Liebowitz and Stephen E. Margolis, *Winners, Losers, and Microsoft: Competition and Antitrust in High Technology* (Oakland, CA; the Independent Institute, 1999, 2001).

²² Bill Gates, with Nathan Myhrvold and Peter Rinearson, *The Road Ahead* (New York: Viking, 1995). For an example of some of the criticism, see Mike Wilson, *The Difference Between God and Larry Ellison: Inside Oracle Corporation* (New York: William Morrow, 1997), pp. 337-338.

²³ *Ibid.* At one point (p. 36), Gates says that Ken Olsen of DEC was “a hero of mine, a distant god.”

²⁴ Articles appeared in several popular magazines; see for example, “Microsoft’s Road to the Internet,” *Business Week* (July 15, 1996), cover, pp. 56-59.

²⁵ The fierce competition in computer software often led to metaphors of combat and war. After the attacks on the United States in September 2001, these metaphors no longer seem as harmless or appropriate.

²⁶ The computer on which this paper has been written, a computer that uses Windows NT and MS-Word, has no IE icon on its desktop.

²⁷ Jackson, quoted in Richard B. McKenzie, *Trust on Trial: How the Microsoft Case is Reframing the Rules of Competition* (Cambridge, MA: Perseus Books, 2000), p. 51.

²⁸ Research for this section consists mainly of observing the author's pre-teen daughter and her friends, all of whom seem to be addicted to Hotmail. See also Po Bronson, "Hot Male," *Wired* (December 1998), accessed electronically.

²⁹ McKenzie, *Trust on Trial*, p. 53.

³⁰ David Bank, *Breaking Windows: How Bill Gates Fumbled the Future of Microsoft* (New York: the Free Press, 2001).

³¹ I have heard this story from several sources, and it is in keeping with the legend surrounding Seymour Cray, but I have been unable to verify it. It does not appear on a videotape of a meeting he gave in Orlando, Florida, in November 1988, but he had given essentially the same briefing at other venues around that time.

³² Glyn Moody, *Rebel Code: Inside the Open Source Revolution*, (Cambridge, MA: Perseus Publishing, 2001), pp. 3, 7.

³³ D. M. Ritchie and K. Thompson, "The UNIX Time-Sharing System," *The Bell System Technical Journal*, 57/6, part 2 (July-August), 1978, pp. 1905-1929, especially p. 1927.

³⁴ *Ibid.*, pp. 1907-8.

³⁵ The PDP-7 on which UNIX was first written had a memory capacity of 18K, 18-bit words, or about 40K bytes. See Dennis M. Ritchie, "The Development of the C Programming Language," in Thomas J. Bergin and Richard G. Gibson, eds., *History of Programming Languages—II* (Reading, MA: Addison-Wesley, 1996), pp. 671-698.

³⁶ Stephen Manes and Paul Andrews, *Gates* (New York: Doubleday, 1993), p. 147. AT&T's licensing policies placed restrictions on the name "UNIX," hence Microsoft's (and others') adopting a different name.

³⁷ Linus Torvalds and David Diamond, *Just for Fun: The Story of an Accidental Revolutionary* (New York: Harper, 2001), Chapter 2, 3.

³⁸ Andrew S. Tanenbaum, *Operating Systems: Design and Implementation* (Engelwood Cliffs, NJ: Prentice Hall 1987).

³⁹ Torvalds, *Just for Fun*, Chapters 2, 3.

⁴⁰ *Ibid.*, pp. 61-62.

⁴¹ Robert H. Follett and Jean E. Sammett, "Programming Language Standards," in Anthony Ralston, Edwin Reilly, and David Hemmendinger, eds., *Encyclopedia of Computer Science, Fourth Edition* (London, 2000), pp. 1466-1470. The standard was called "POSIX," for "Portable Operating System Interface for Computer Environments."

⁴² Torvalds, p. 85. Punctuation and spelling are original. Some of these postings have been saved and archived on the web site Google.com.

⁴³ The notion of "flame wars," and whether they truly represented the feelings of the persons posting such messages, is a matter for future research and will not be further discussed here. Seen out of context, phrases calling another's work "brain damaged" or stating that it "sucks" can indeed appear shocking, especially given the traditional respect accorded to professors in European universities. Flame

wars seem to have died out recently, although they are alive in a restricted form on the Web site Slashdot.org (see text).

⁴⁴ Frederick P. Brooks, Jr., *The Mythical Man-Month: Essays on Software Engineering* (Reading, MA: Addison Wesley, 1975). Russ Mitchell lists the fifteen people in the “inner circle” of Linux developers as of 2001, in his essay “Open War,” (*Wired*, October 2001, p. 137). They come from nine countries and nearly all have “day jobs”: they do something else to earn money.

⁴⁵ Brooks, op cit., pp. 80-81.

⁴⁶ Frederick P. Brooks Jr., *The Mythical Man-Month: Essays on Software Engineering, Anniversary Edition* (Reading, MA: Addison-Wesley), p. 203. New material was added after Chapter 15. I have avoided relying on this edition, as I feel that with a few exceptions it does not add much to the classic qualities of the original.

⁴⁷ Raymond’s essay is available on the Internet, but I have relied on a published version, in *Knowledge, Technology, and Policy*, 12/3 (Fall 1999), pp. 23-49.

⁴⁸ *Ibid.*, p. 29.

⁴⁹ The letter was originally published in the newsletter of the MITS Corporation, makers of the Altair computer, in February 1976, p. 3. Smithsonian Institution, National Museum of American History, Archives.

⁵⁰ Bryan Pfaffenberger, “The Rhetoric of Dread: Fear, Uncertainty, and Doubt (FUD) in Information Technology Marketing,” *Knowledge, Technology & Policy* 13/3 (Fall 2000), pp. 78-92.

⁵¹ Torvalds, *Just for Fun*, pp. 87-89.

⁵² Stallman’s personal Web page is <http://www.stallman.org/>; the Free Software Foundation’s official page is <http://www.gnu.ai.mit.edu>. The online magazine *Salon.com* has been running an ongoing chronicle of the Free Software movement, by Andrew Leonard. These sites were accessed by the author in winter, 2001-2002, and they may change.

⁵³ Glyn Moody, *Rebel Code* (Cambridge, MA: Perseus, 2001), pp. 14-19.

⁵⁴ The Usenet posting, to net.unix-wizards, was recovered and archived in 2001 by the search engine Google.com, from which this passage was taken.

⁵⁵ Richard M. Stallman, “What is a GNU/Linux system?” in GNU’s Bulletin, vol. 1 n. 23, [1997], pp. 4-5.

⁵⁶ Steven Johnson, personal communication to the author, Jan. 31, 2002. The names of these programs are often, but not always nor consistently, written in lower-case letters. I have tried to follow the conventions of those who created them wherever possible.

⁵⁷ *Ibid.*, p. 3.

⁵⁸ Stallman’s relations with companies like Red Hat are fairly cordial, but he objects to O’Reilly & Associates’ making money by selling books that serve as manuals for free software. According to Stallman, those manuals are an integral part of the software and should be free as well.

⁵⁹ The phrase was used by Microsoft VP Jim Allchin, quoted by Andrew Leonard, “Life, Liberty, and the Pursuit of Free Software, *Salon.com* (online), Feb. 15, 2001.

⁶⁰ The Lotus server was called “Domino.” The basic outline of this story has been taken from Andrew Leonard’s online history of Open Source, chronicled in Salon.com.

⁶¹ Ibid.; John Markoff, “Sharing Software, I.B.M. to Release Mail Program Blueprint,” *New York Times* (December 14, 1998), p. C-5.

⁶² Spencer E. Ante, “Big Blue’s Big Bet on Free Software,” *Business Week*, December 10, 2001, pp. 78-79. That same magazine issue carried a two-page ad in which IBM affirmed its commitment to Linux.

⁶³ “IBM unveils first Linux-only Mainframes,” IBM press release, Friday, January 25, 2002.

⁶⁴ GNOME stands for “Gnu Network Object Model Environment”; KDE for “K Desktop Environment.”

⁶⁵ Russ Mitchell, “Open War,” *Wired* (October 2001), pp. 135-139.

⁶⁶ Andreessen is quoted in David Banks, *Breaking Windows*, p. 26, but the phrase has become part of common folklore.

⁶⁷ Open Source advocates are eagerly anticipating Netscape’s latest version of its browser, which it promises to be Open Source. As of this writing that version, 7.0, has not been released.