A Brief History of Hackerdom

Eric Steven Raymond

Thyrsus Enterprises [http://www.tuxedo.org/~esr/]

<esr@thyrsus.com>

This is version 1.24

$Date: 2002/08/02 08:21:12 $

Revision History

Revision 1.24	25 August 2000	esr
First DocBook version.		
Revision 1.23	29 Dec 1999	esr
This version went into the first printed edition.		
Revision 1.20	17 August 1999	esr
First SGML version with bibliography.		
Revision 1.1	15 Feb 1997	esr
This document dates from around 1992, but was not version-controlled until 1997.		

I explore the origins of the hacker culture, including prehistory among the Real Programmers, the glory days of the MIT hackers, and how the early ARPAnet nurtured the first network nation. I describe the early rise and eventual stagnation of Unix, the new hope from Finland, and how 'the last true hacker' became the next generation's patriarch. I sketch the way Linux and the mainstreaming of the Internet brought the hacker culture from the fringes of public consciousness to its current prominence.

Table of Contents

Prologue: The Real Programmers

In the beginning, there were Real Programmers.

That's not what they called themselves. They didn't call themselves 'hackers', either, or anything in particular; the sobriquet 'Real Programmer' wasn't coined until after 1980, retrospectively by one of their own. But from 1945 onward, the technology of computing attracted many of the world's brightest and most creative minds. From Eckert and Mauchly's first ENIAC computer onward there was a more or less continuous and self-conscious technical culture of enthusiast programmers, people who built and played with software for fun.

The Real Programmers typically came out of engineering or physics backgrounds. They were often amateur-radio hobbyists. They wore white socks and polyester shirts and ties and thick glasses and coded in machine language and assembler and FORTRAN and half a dozen ancient languages now forgotten.

From the end of World War II to the early 1970s, in the great days of batch processing and the "big iron" mainframes, the Real Programmers were the dominant technical culture in computing. A few pieces of revered hacker folklore date from this era, including various lists of Murphy's Laws and the mock-German "Blinkenlights" poster that still graces many computer rooms.

Some people who grew up in the 'Real Programmer' culture remained active into the 1990s and even past the turn of the 21st century. Seymour Cray, designer of the Cray line of supercomputers, was among the greatest. He is said once to have toggled an entire operating system of his own design into a computer of his own design through its front-panel switches. In octal. Without an error. And it worked. Real Programmer macho supremo.

The 'Real Programmer' culture, though, was heavily associated with batch (and especially batch scientific) computing. It was eventually eclipsed by the rise of interactive computing, the universities, and the networks. These gave birth to another engineering tradition that, eventually, would evolve into today's open-source hacker culture.

The Early Hackers

The beginnings of the hacker culture as we know it today can be conveniently dated to 1961, the year MIT acquired the first PDP-1. The Signals and Power Committee of MIT's Tech Model Railroad Club adopted the machine as their favorite tech-toy and invented programming tools, slang, and an entire surrounding culture that is still recognizably with us today. These early years have been examined in the first part of Steven Levy's book *Hackers* [Levy].

MIT's computer culture seems to have been the first to adopt the term 'hacker'. The Tech Model Railroad Club's hackers became the nucleus of MIT's Artificial Intelligence Laboratory, the world's leading center of AI research into the early 1980s. Their influence was spread far wider after 1969, the first year of the ARPAnet.

The ARPAnet was the first transcontinental, high-speed computer network. It was built by the Defense Department as an experiment in digital communications, but grew to link together hundreds of universities and defense contractors and research laboratories. It enabled researchers everywhere to exchange information with unprecedented speed and flexibility, giving a huge boost to collaborative work and tremendously increasing both the pace and intensity of technological advance.

But the ARPAnet did something else as well. Its electronic highways brought together hackers all over the U.S. in a critical mass; instead of remaining in isolated small groups each developing their own ephemeral local cultures, they discovered (or re-invented) themselves as a networked tribe.

The first intentional artifacts of the hacker culture—the first slang lists, the first satires, the first self-conscious discussions of the hacker ethic—all propagated on the ARPAnet in its early years. In particular, the first version of the Jargon File [http://www.tuxedo.org/jargon] developed as a cross-net collaboration during 1973–1975. This slang dictionary became one of the culture's defining documents. It was eventually published as "The Hacker's Dictionary" in 1983; that first version is out of print, but a revised and expanded version is *New Hacker's Dictionary* [Raymond].

Hackerdom flowered at the universities connected to the net, especially (though not exclusively) in their computer science departments. MIT's AI and LCS labs made it first among equals from the late 1960s. But Stanford University's Artificial Intelligence Laboratory (SAIL) and Carnegie-Mellon University (CMU) became nearly as important. All were thriving centers of computer science and AI research. All attracted bright people who contributed great things to the hacker culture, on both the technical and folkloric levels.

To understand what came later, though, we need to take another look at the computers themselves; because the AI Lab's rise and its eventual fall were both driven by waves of change in computing technology.

Since the days of the PDP-1, hackerdom's fortunes had been woven together with Digital Equipment Corporation's PDP series of minicomputers. DEC pioneered commercial interactive computing and time-sharing operating systems. Because their machines were flexible, powerful, and relatively cheap for the era, lots of universities bought them.

Cheap time-sharing was the medium the hacker culture grew in, and for most of its lifespan the ARPAnet was primarily a network of DEC machines. The most important of these was the PDP-10, first released in 1967. The 10 remained hackerdom's favorite machine for almost fifteen years; TOPS-10 (DEC's operating system for the machine) and MACRO-10 (its assembler) are still remembered with nostalgic fondness in a great deal of slang and folklore.

MIT, though it used the same PDP-10s as everyone else, took a slightly different path; they rejected DEC's software for the PDP-10 entirely and built their own operating system, the fabled ITS.

ITS stood for 'Incompatible Time-sharing System' which gives one a pretty good fix on the MIT hackers' attitude (technically, the name was a play on its predecessor, the Compatible Time-Sharing System CTSS). They wanted it *theirtheir* way. Fortunately for all, MIT's people had the intelligence to match their arrogance. ITS, quirky and eccentric and occasionally buggy though it always was, hosted a brilliant series of technical innovations and still arguably holds the record as the single time-sharing system in longest continuous use.

ITS itself was written in assembler, but many ITS projects were written in the AI language LISP. LISP was far more powerful and flexible than any other language of its day; in fact, it is still a better design than most languages of *todaytoday*, twenty-five years later. LISP freed ITS's hackers to think in unusual and creative ways. It was a major factor in their successes, and remains one of hackerdom's favorite languages.

Many of the ITS culture's technical creations are still alive today; the EMACS program editor is perhaps the best-known. And much of ITS's folklore is still 'live' to hackers, as one can see in the Jargon File [http://-www.tuxedo.org/jargon].

SAIL and CMU weren't asleep, either. Many of the cadre of hackers that grew up around SAIL's PDP-10 later became key figures in the development of the personal computer and today's window/icon/mouse software interfaces. Meanwhile hackers at CMU were doing the work that would lead to the first practical large-scale applications of expert systems and industrial robotics.

Another important node of the culture was XEROX PARC, the famed Palo Alto Research Center. For more than a decade, from the early 1970s into the mid-1980s, PARC yielded an astonishing volume of groundbreaking hardware and software innovations. The modern mice, windows, and icons style of software interface was invented there. So was the laser printer, and the local-area network; and PARC's series of D machines anticipated the powerful personal computers of the 1980s by a decade. Sadly, these prophets were without honor in their own company; so much so that it became a standard joke to describe PARC as a place characterized by developing brilliant ideas for everyone else. Their influence on hackerdom was pervasive.

The ARPAnet and the PDP-10 cultures grew in strength and variety throughout the 1970s. The facilities for electronic mailing lists that had been used to foster cooperation among continent-wide special-interest groups were increasingly also used for more social and recreational purposes. DARPA deliberately turned a blind eye to all the technically 'unauthorized' activity; it understood that the extra overhead was a small price to pay for attracting an entire generation of bright young people into the computing field.

Perhaps the best-known of the 'social' ARPAnet mailing lists was the SF-LOVERS list for science-fiction fans; it is still very much alive today, in fact, on the larger 'Internet' that ARPAnet evolved into. But there were many others, pioneering a style of communication that would later be commercialized by for-profit time-sharing services like CompuServe, GEnie and Prodigy (and later still dominated by AOL).

Your historian first became involved with the hacker culture in 1977 through the early ARPAnet and science-fiction fandom. From then onward, I personally witnessed and participated in many of the changes described here.

The Rise of Unix

Far from the bright lights of the ARPAnet, off in the wilds of New Jersey, something else had been going on since 1969 that would eventually overshadow the PDP-10 tradition. The year of ARPAnet's birth was also the year that a Bell Labs hacker named Ken Thompson invented Unix.

Thompson had been involved with the development work on a time-sharing OS called Multics, which shared common ancestry with ITS. Multics was a test-bed for some important ideas about how the complexity of an operating system could be hidden inside it, invisible to the user, and even to most programmers. The idea was to make using Multics from the outside (and programming for it!) much simpler, so that more real work could get done.

Bell Labs pulled out of the project when Multics displayed signs of bloating into an unusable white elephant (the system was later marketed commercially by Honeywell but never became a success). Ken Thompson missed the Multics environment, and began to play at implementing a mixture of its ideas and some of his own on a scavenged DEC PDP-7.

Another hacker named Dennis Ritchie invented a new language called 'C' for use under Thompson's embryonic Unix. Like Unix, C was designed to be pleasant, unconstraining, and flexible. Interest in these tools spread at

Bell Labs, and they got a boost in 1971 when Thompson and Ritchie won a bid to produce what we'd now call an office automation system for internal use there. But Thompson & Ritchie had their eye on a bigger prize.

Traditionally, operating systems had been written in tight assembler to extract the absolute highest efficiency possible out of their host machines. Thompson and Ritchie were among the first to realize that hardware and compiler technology had become good enough that an entire operating system could be written in C, and by 1978 the whole environment had been successfully ported to several machines of different types.

This had never been done before, and the implications were enormous. If Unix could present the same face, the same capabilities, on machines of many different types, it could serve as a common software environment for all of them. No longer would users have to pay for complete new designs of software every time a machine went obsolete. Hackers could carry around software toolkits between different machines, rather than having to re-invent the equivalents of fire and the wheel every time.

Besides portability, Unix and C had some other important strengths. Both were constructed from a "Keep It Simple, Stupid" philosophy. A programmer could easily hold the entire logical structure of C in his head (unlike most other languages before or since) rather than needing to refer constantly to manuals; and Unix was structured as a flexible toolkit of simple programs designed to combine with each other in useful ways.

The combination proved to be adaptable to a very wide range of computing tasks, including many completely unanticipated by the designers. It spread very rapidly within AT&T, in spite of the lack of any formal support program for it. By 1980 it had spread to a large number of university and research computing sites, and thousands of hackers considered it home.

The workhorse machines of the early Unix culture were the PDP-11 and its descendant, the VAX. But because of Unix's portability, it ran essentially unaltered on a wider range of machines than one could find on the entire ARPAnet. And nobody used assembler; C programs were readily portable among all these machines.

Unix even had its own networking, of sorts—UUCP: low-speed and unreliable, but cheap. Any two Unix machines could exchange point-to-point electronic mail over ordinary phone lines; this capability was built into the system, not an optional extra. In 1980 the first Usenet sites began exchanging broadcast news, forming a gigantic distributed bulletin board that would quickly grow bigger than ARPAnet. Unix sites began to form a network nation of their own around Usenet.

A few Unix sites were on the ARPAnet themselves. The PDP-10 and Unix/Usenet cultures began to meet and mingle at the edges, but they didn't mix very well at first. The PDP-10 hackers tended to consider the Unix crowd a bunch of upstarts, using tools that looked ridiculously primitive when set against the baroque, lovely complexities of LISP and ITS. "Stone knives and bearskins!" they muttered.

And there was yet a third current flowing. The first personal computer had been marketed in 1975; Apple was founded in 1977, and advances came with almost unbelievable rapidity in the years that followed. The potential of microcomputers was clear, and attracted yet another generation of bright young hackers. *TheirTheir* language was BASIC, so primitive that PDP-10 partisans and Unix aficionados both considered it beneath contempt.

The End of Elder Days

So matters stood in 1980; three cultures, overlapping at the edges but clustered around very different technologies. The ARPAnet/PDP-10 culture, wedded to LISP and MACRO and TOPS-10 and ITS and SAIL. The Unix and C crowd with their PDP-11s and VAXen and pokey telephone connections. And an anarchic horde of early microcomputer enthusiasts bent on taking computer power to the people.

Among these, the ITS culture could still claim pride of place. But stormclouds were gathering over the Lab. The PDP-10 technology ITS depended on was aging, and the Lab itself was split into factions by the first attempts to commercialize artificial intelligence. Some of the Lab's (and SAIL's and CMU's) best were lured away to high-paying jobs at startup companies.

The death blow came in 1983, when DEC cancelled its 'Jupiter' followon to the PDP-10 in order to concentrate on the PDP-11 and VAX lines. ITS no longer had a future. Because it wasn't portable, it was more effort than anyone could afford to move ITS to new hardware. The Berkeley variant of Unix running on a VAX became the hacking system *par excellence*, and anyone with an eye on the future could see that microcomputers were growing in power so rapidly that they were likely to sweep all before them.

It's around this time that Levy wrote *Hackers*. One of his prime informants was Richard M. Stallman (inventor of Emacs), a leading figure at the Lab and its most fanatical holdout against the commercialization of Lab technology.

Stallman (who is usually known by his initials and login name, RMS) went on to form the Free Software Foundation and dedicate himself to producing high-quality free software. Levy eulogized him as "the last true hacker", a description which happily proved incorrect.

Stallman's grandest scheme neatly epitomized the transition hackerdom underwent in the early eighties—in 1982 he began the construction of an entire clone of Unix, written in C and available for free. His project was known as the GNU (Gnu's Not Unix) operating system, in a kind of recursive acronym. GNU quickly became a major focus for hacker activity. Thus, the spirit and tradition of ITS was preserved as an important part of the newer, Unix and VAX-centered hacker culture. Stallman's design finished what Berkeley had started, fusing the remains of the PDP-10 hacker culture with the Unix crowd.

Indeed, for more than a decade after its founding RMS's Free Software Foundation would largely define the public ideology of the hacker culture, and Stallman himself would be the only credible claimant to leadership of the tribe.

It was also around 1982–83 that microchip and local-area network technology began to have a serious impact on hackerdom. Ethernet and the Motorola 68000 microchip made a potentially potent combination, and several different startups had been formed to build the first generation of what we now call workstations.

In 1982, a group of Unix hackers from Stanford and Berkeley founded Sun Microsystems on the belief that Unix running on relatively inexpensive 68000-based hardware would prove a winning combination for a wide variety of applications. They were right, and their vision set the pattern for an entire industry. While still priced out of reach of most individuals, workstations were cheap for corporations and universities; networks of them (one to a user) rapidly replaced the older VAXes and other time-sharing systems.

The Proprietary-Unix Era

By 1984, when Ma Bell divested and Unix became a supported AT&T product for the first time, the most important fault line in hackerdom was between a relatively cohesive "network nation" centered around the Internet and

Usenet (and mostly using minicomputer- or workstation-class machines running Unix), and a vast disconnected hinterland of microcomputer enthusiasts.

It was also around this time that serious cracking episodes were first covered in the mainstream press—and journalists began to misapply the term "hacker" to refer to computer vandals, an abuse which sadly continues to this day.

The workstation-class machines built by Sun and others opened up new worlds for hackers. They were built to do high-performance graphics and pass around shared data over a network. During the 1980s hackerdom was preoccupied by the software and tool-building challenges of getting the most use out of these features. Berkeley Unix developed built-in support for the ARPAnet protocols, which offered a solution to the networking problems associated with UUCP's slow point-to-point links and encouraged further growth of the Internet.

There were several attempts to tame workstation graphics. The one that prevailed was the X window system, developed at MIT with contributions from hundreds of individuals at dozens of companies. A critical factor in its success was that the X developers were willing to give the sources away for free in accordance with the hacker ethic, and able to distribute them over the Internet. X's victory over proprietary graphics systems (including one offered by Sun itself) was an important harbinger of changes which, a few years later, would profoundly affect Unix as a whole.

There was a bit of factional spleen still vented occasionally in the ITS/Unix rivalry (mostly from the ex-ITSers' side). But the last ITS machine shut down for good in 1990; the zealots no longer had a place to stand and mostly assimilated to the Unix culture with various degrees of grumbling.

Within networked hackerdom itself, the big rivalry of the 1980s was between fans of Berkeley Unix and the AT&T versions. Occasionally you can still find copies of a poster from that period, showing a cartoony X-wing fighter out of the "Star Wars" movies streaking away from an exploding Death Star patterned on the AT&T logo. Berkeley hackers liked to see themselves as rebels against soulless corporate empires. AT&T Unix never caught up with BSD/Sun in the marketplace, but it won the standards wars. By 1990, AT&T and BSD versions were becoming harder to tell apart, having adopted many of each other's innovations.

After 1987, the workstation technology of the previous decade began to look distinctly threatened by newer, low-cost and high-performance personal computers based on the Intel 386 chip and its descendants. For the first time, individual hackers could afford to have home machines comparable in power and storage capacity to the minicomputers of ten years earlier—Unix engines capable of supporting a full development environment and talking to the Internet.

The MS-DOS world remained blissfully ignorant of all this. Though those early microcomputer enthusiasts quickly expanded to a population of DOS and Mac hackers orders an magnitude greater than that of the "network nation" culture, they never become a self-aware culture themselves. The pace of change was so fast that fifty different technical cultures grew and died as rapidly as mayflies, never achieving quite the stability necessary to develop a common tradition of jargon, folklore, and mythic history. The absence of a really pervasive network comparable to UUCP or Internet prevented them from becoming a network nation themselves.

Widespread access to commercial online services like CompuServe and GEnie was beginning to take hold, but the fact that non-Unix operating systems don't come bundled with development tools meant that very little source was passed over them. Thus, no tradition of collaborative hacking developed.

The mainstream of hackerdom, (dis)organized around the Internet and by now largely identified with the Unix technical culture, didn't care about the commercial services. These hackers wanted better tools and more Internet, and cheap 32-bit PCs promised to put both in everyone's reach.

But where was the software? Commercial Unixes remained expensive, in the multiple-kilobuck range. In the early 1990s several companies made a go at selling AT&T or BSD Unix ports for PC-class machines. Success was elusive, prices didn't come down much, and (worst of all) you didn't get modifiable and redistributable sources with your operating system. The traditional software-business model wasn't giving hackers what they wanted.

Neither was the Free Software Foundation. The development of HURD, RMS's long-promised free Unix kernel for hackers, got stalled for years and failed to produce anything like a usable kernel until 1996 (though by 1990 FSF supplied almost all the other difficult parts of a Unix-like operating system).

Worse, by the early 1990s it was becoming clear that ten years of effort to commercialize proprietary Unix was ending in failure. Unix's promise of cross-platform portability got lost in bickering among half a dozen proprietary Unix versions. The proprietary-Unix players proved so ponderous, so blind, and so inept at marketing that Microsoft was able to grab away a large part of their market with the shockingly inferior technology of its Windows operating system.

In early 1993, a hostile observer might have had grounds for thinking that the Unix story was almost played out, and with it the fortunes of the hacker tribe. And there was no shortage of hostile observers in the computer trade press, many of whom had been ritually predicting the imminent death of Unix at six-month intervals ever since the late 1970s.

In those days it was conventional wisdom that the era of individual techno-heroism was over, that the software industry and the nascent Internet would increasingly be dominated by colossi like Microsoft. The first generation of Unix hackers seemed to be getting old and tired (Berkeley's Computer Science Research group ran out of steam and would lose its funding in 1994). It was a depressing time.

Fortunately, there had been things going on out of sight of the trade press, and out of sight even of most hackers, that would produce startlingly positive developments in later 1993 and 1994. Eventually, these would take the hacker culture in a whole new direction and to undreamed-of successes.

The Early Free Unixes

Into the gap left by the Free Software Foundation's uncompleted HURD had stepped a Helsinki University student named Linus Torvalds. In 1991 he began developing a free Unix kernel for 386 machines using the Free Software Foundation's toolkit. His initial, rapid success attracted many Internet hackers to help him develop Linux, a full-featured Unix with entirely free and re-distributable sources.

Linux was not without competitors. In 1991, contemporaneously with Linus Torvalds's early experiments, William and Lynne Jolitz were experimentally porting the BSD Unix sources to the 386. Most observers comparing BSD technology with Linus's crude early efforts expected that BSD ports would become the most important free Unixes on the PC.

The most important feature of Linux, however, was not technical but sociological. Until the Linux development, everyone believed that any software as complex as an operating system had to be developed in a carefully coordinated way by a relatively small, tightly-knit group of people. This model was and still is typical of both commercial software and the great free-software cathedrals built by the Free Software Foundation in the 1980s; also of the freeBSD/netBSD/OpenBSD projects that spun off from the Jolitzes' original 386BSD port.

Linux evolved in a completely different way. From nearly the beginning, it was rather casually hacked on by huge numbers of volunteers coordinating only through the Internet. Quality was maintained not by rigid standards or autocracy but by the naively simple strategy of releasing every week and getting feedback from hundreds of users within days, creating a sort of rapid Darwinian selection on the mutations introduced by developers. To the amazement of almost everyone, this worked quite well.

By late 1993 Linux could compete on stability and reliability with many commercial Unixes, and hosted vastly more software. It was even beginning to attract ports of commercial applications software. One indirect effect of this development was to kill off most of the smaller proprietary Unix vendors—without developers and hackers to sell to, they folded. One of the few survivors, BSDI (Berkeley Systems Design, Incorporated), flourished by offering full sources with its BSD-based Unix and cultivating close ties with the hacker community.

These developments were not much remarked on at the time even within the hacker culture, and not at all outside it. The hacker culture, defying repeated predictions of its demise, was just beginning to remake the commercial-software world in its own image. It would be five more years, however, before this trend became obvious.

The Great Web Explosion

The early growth of Linux synergized with another phenomenon: the public discovery of the Internet. The early 1990s also saw the beginnings of a flourishing Internet-provider industry, selling connectivity to the public for a few dollars a month. Following the invention of the World Wide Web, the Internet's already rapid growth accelerated to a breakneck pace.

By 1994, the year Berkeley's Unix development group formally shut down, several different free Unix versions (Linux and the descendants of 386BSD) served as the major focal points of hacking activity. Linux was being distributed commercially on CD-ROM and selling like hotcakes. By the end of 1995, major computer companies were beginning to take out glossy advertisements celebrating the Internet-friendliness of their software and hardware!

In the late 1990s the central activities of hackerdom became Linux development and the mainstreaming of the Internet. The World Wide Web has at last made the Internet into a mass medium, and many of the hackers of the 1980s and early 1990s launched Internet Service Providers selling or giving access to the masses.

The mainstreaming of the Internet even brought the hacker culture the beginnings of respectability and political clout. In 1994 and 1995 hacker activism scuppered the Clipper proposal which would have put strong encryption under government control. In 1996 hackers mobilized a broad coalition to defeat the misnamed "Communications Decency Act" and prevent censorship of the Internet.

With the CDA victory, we pass out of history into current events. We also pass into a period in which your historian (rather to his own surprise) became an actor rather than just an observer. This narrative will continue in *Revenge of the HackersRevenge of the Hackers*.

Bibliography

[Levy][Levy] Levy, Steven; *HackersHackers*, Anchor/Doubleday 1984, ISBN 0-385-19195-2.

[Raymond][Raymond] Raymond, Eric S.; *The New Hacker's DictionaryThe New Hacker's Dictionary*, MIT Press, 3rd edition 1996. ISBN ISBN 0-262-68092-0.

David E. Lundstrom gave us an anecdotal history of the "Real Programmer" era in *A Few Good Men From UNIVACA Few Good Men From UNIVAC*, 1987, ISBN-0-262-62075-8.