Twenty Years of Berkeley Unix : From AT&T-Owned to Freely Redistributable

Marshall Kirk McKusick

Early History

Ken Thompson and Dennis Ritchie presented the first Unix paper at the Symposium on Operating Systems Principles at Purdue University in November 1973. Professor Bob Fabry, of the University of California at Berkeley, was in attendance and immediately became interested in obtaining a copy of the system to experiment with at Berkeley.

At the time, Berkeley had only large mainframe computer systems doing batch processing, so the first order of business was to get a PDP-11/45 suitable for running with the then-current Version 4 of Unix. The Computer Science Department at Berkeley, together with the Mathematics Department and the Statistics Department, were able to jointly purchase a PDP-11/45. In January 1974, a Version 4 tape was delivered and Unix was installed by graduate student Keith Standiford.

Although Ken Thompson at Purdue was not involved in the installation at Berkeley as he had been for most systems up to that time, his expertise was soon needed to determine the cause of several strange system crashes. Because Berkeley had only a 300-baud acoustic-coupled modem without auto answer capability, Thompson would call Standiford in the machine room and have him insert the phone into the modem; in this way Thompson was able to remotely debug crash dumps from New Jersey.

Many of the crashes were caused by the disk controller's inability to reliably do overlapped seeks, contrary to the documentation. Berkeley's 11/45 was among the first systems that Thompson had encountered that had two disks on the same controller! Thompson's remote debugging was the first example of the cooperation that sprang up between Berkeley and Bell Labs. The willingness of the researchers at the Labs to share their work with Berkeley was instrumental in the rapid improvement of the software available at Berkeley.

Though Unix was soon reliably up and running, the coalition of Computer Science, Mathematics, and Statistics began to run into problems; Math and Statistics wanted to run DEC's RSTS system. After much debate, a compromise was reached in which each department would get an eight-hour shift; Unix would run for eight hours followed by sixteen hours of RSTS. To promote fairness, the time slices were rotated each day. Thus, Unix ran 8 a.m. to 4 p.m. one day, 4 p.m. to midnight the next day, and midnight to 8 a.m. the third day. Despite the bizarre schedule, students taking the Operating Systems course preferred to do their projects on Unix rather than on the batch machine.

Professors Eugene Wong and Michael Stonebraker were both stymied by the confinements of the batch environment, so their INGRES database project was among the first groups to move from the batch machines to the interactive environment provided by Unix. They quickly found the shortage of machine time and the odd hours on the 11/45 intolerable, so in the spring of 1974, they purchased an 11/40 running the newly available Version 5. With their first distribution of INGRES in the fall of 1974, the INGRES project became the first group in the Computer Science department to distribute their software. Several hundred INGRES tapes were shipped over the next six years, helping to establish Berkeley's reputation for designing and building real systems.

Even with the departure of the INGRES project from the 11/45, there was still insufficient time available for the remaining students. To alleviate the shortage, Professors Michael Stonebraker and Bob Fabry set out in June 1974, to get two instructional 11/45's for the Computer Science department's own use. Early in 1975, the money was obtained. At nearly the same time, DEC announced the 11/70, a machine that appeared to be much superior to the 11/45. Money for the two 11/45s was pooled to buy a single 11/70 that arrived in the fall of 1975. Coincident with the arrival of the 11/70, Ken Thompson decided to take a one-year sabbatical as a visiting professor at the University of California at Berkeley, his alma mater. Thompson, together with Jeff Schriebman and Bob Kridle, brought up the latest Unix, Version 6, on the newly installed 11/70.

Also arriving in the fall of 1975 were two unnoticed graduate students, Bill Joy and Chuck Haley; they both took an immediate interest in the new system. Initially they began working on a Pascal system that Thompson had hacked together while hanging around the 11/70 machine room. They expanded and improved the Pascal interpreter to the point that it became the programming system of choice for students because of its excellent error recovery scheme and fast compile and execute time.

With the replacement of Model 33 teletypes by ADM-3 screen terminals, Joy and Haley began to feel stymied by the constraints of the *ed* editor. Working from an editor named *em* that they had obtained from Professor George Coulouris at Queen Mary's College in London, they worked to produce the line-at-a-time editor *ex*.

With Ken Thompson's departure at the end of the summer of 1976, Joy and Haley begin to take an interest in exploring the internals of the Unix kernel. Under Schriebman's watchful eye, they first installed the fixes and improvements provided on the "fifty changes" tape from Bell Labs. Having learned to maneuver through the source code, they suggested several small enhancements to streamline certain kernel bottlenecks.

Early Distributions

Meanwhile, interest in the error recovery work in the Pascal compiler brought in requests for copies of the system. Early in 1977, Joy put together the "Berkeley Software Distribution." This first distribution included the Pascal system, and, in an obscure subdirectory of the Pascal source, the editor *ex*. Over the next year, Joy, acting in the capacity of distribution secretary, sent out about thirty free copies of the system.

With the arrival of some ADM-3a terminals offering screen-addressable cursors, Joy was finally able to write *vi*, bringing screen-based editing to Berkeley. He soon found himself in a quandary. As is frequently the case in universities strapped for money, old equipment is never replaced all at once. Rather than support code for optimizing the updating of several different terminals, he decided to consolidate the screen management by using a small interpreter to redraw the screen. This interpreter was driven by a description of the terminal characteristics, an effort that eventually became *termcap*.

By mid-1978, the software distribution clearly needed to be updated. The Pascal system had been made markedly more robust through feedback from its expanding user community, and had been split into two passes so that it could be run on PDP-11/34s. The result of the update was the "Second Berkeley Software Distribution," a name that was quickly shortened to 2BSD. Along with the enhanced Pascal system, *vi* and *termcap* for several terminals was included. Once again Bill Joy single-handedly put together distributions, answered the phone, and incorporated user feedback into the system. Over the next year nearly seventy-five tapes were shipped. Though Joy moved on to other projects the following year, the 2BSD distribution continued to expand. The final version of this distribution, 2.11BSD, is a complete system used on hundreds of PDP-11's still running in various corners of the world.

VAX Unix

Early in 1978, Professor Richard Fateman began looking for a machine with a larger address space on which he could continue his work on Macsyma (originally started on a PDP-10). The newly announced VAX-11/780 fulfilled the requirements and was available within budget. Fateman and thirteen other faculty members put together an NSF proposal that they combined with some departmental funds to purchase a VAX.

Initially the VAX ran DEC's operating system VMS, but the department had gotten used to the Unix environment and wanted to continue using it. So, shortly after the arrival of the VAX, Fateman obtained a copy of the 32/V port of Unix to the VAX by John Reiser and Tom London of Bell Labs.

Although 32/V provided a Version 7 Unix environment on the VAX, it did not take advantage of the virtual memory capability of the VAX hardware. Like its predecessors on the PDP-11, it was entirely a swap-based system. For the Macsyma group at Berkeley, the lack of virtual memory meant that the process address space was limited by the size of the physical memory, initially 1 megabyte on the new VAX.

To alleviate this problem, Fateman approached Professor Domenico Ferrari, a member of the systems faculty at Berkeley, to investigate the possibility of having his group write a virtual memory system for Unix. Ozalp Babaoglu, one of Ferrari's students, set about to find some way of implementing a working set paging system on the VAX; his task was complicated because the VAX lacked reference bits.

As Babaoglu neared the completion of his first cut at an implementation, he approached Bill Joy for some help in understanding the intricacies of the Unix kernel. Intrigued by Babaoglu's approach, Joy joined in helping to integrate the code into 32/V and then with the ensuing debugging.

Unfortunately, Berkeley had only a single VAX for both system development and general production use. Thus, for several weeks over the Christmas break, the tolerant user community alternately found themselves logging into 32/V and "Virtual VAX/Unix." Often their work on the latter system would come to an abrupt halt, followed several minutes later by a 32/V login prompt. By January, 1979, most of the bugs had been worked out, and 32/V had been relegated to history.

Joy saw that the 32-bit VAX would soon make the 16-bit PDP-11 obsolete, and began to port the 2BSD software to the VAX. While Peter Kessler and I ported the Pascal system, Joy ported the editors *ex* and *vi*, the C shell, and the myriad other smaller programs from the 2BSD distribution. By the end of 1979, a complete distribution had been put together. This distribution included the virtual memory kernel, the standard 32/V utilities, and the additions from 2BSD. In December, 1979, Joy shipped the first of nearly a hundred copies of 3BSD, the first VAX distribution from Berkeley.

The final release from Bell Laboratories was 32/V; thereafter all Unix releases from AT&T, initially System III and later System V, were managed by a different group that emphasized stable commercial releases. With the commercialization of Unix, the researchers at Bell Laboratories were no longer able to act as a clearing-house for the ongoing Unix research. As the research community continued to modify the Unix system, it found that it needed an organization that could produce research releases. Because of its early involvement in Unix and its history of releasing Unix-based tools, Berkeley quickly stepped into the role previously provided by the Labs.

DARPA Support

Meanwhile, in the offices of the planners for the Defense Advanced Research Projects Agency (DARPA), discussions were being held that would have a major influence on the work at Berkeley. One of DARPA's early successes had been to set up a nationwide computer network to link together all their major research centers. At that time, they were finding that many of the computers at these centers were reaching the end of their useful lifetime and had to be replaced. The heaviest cost of replacement was the porting of the research software to the new machines. In addition, many sites were unable to share their software because of the diversity of hardware and operating systems.

Choosing a single hardware vendor was impractical because of the widely varying computing needs of the research groups and the undesirability of depending on a single manufacturer. Thus, the planners at DARPA decided that the best solution was to unify at the operating systems level. After much discussion, Unix was chosen as a standard because of its proven portability.

In the fall of 1979, Bob Fabry responded to DARPA's interest in moving towards Unix by writing a proposal suggesting that Berkeley develop an enhanced version of 3BSD for the use of the DARPA community. Fabry took a copy of his proposal to a meeting of DARPA image processing and VLSI contractors, plus representatives from Bolt, Beranek, and Newman, the developers of the ARPAnet. There was some reservation whether Berkeley could produce a working system; however, the release of 3BSD in December 1979 assuaged most of the doubts.

With the increasingly good reputation of the 3BSD release to validate his claims, Bob Fabry was able to get an 18-month contract with DARPA beginning in April 1980. This contract was to add features needed by the

DARPA contractors. Under the auspices of this contract, Bob Fabry sets up an organization which was christened the Computer Systems Research Group, or CSRG for short. He immediately hired Laura Tong to handle the project administration. Fabry turned his attention to finding a project leader to manage the software development. Fabry assumed that since Joy had just passed his Ph.D. qualifying examination, he would rather concentrate on completing his degree than take the software development position. But Joy had other plans. One night in early March he phoned Fabry at home to express interest in taking charge of the further development of Unix. Though surprised by the offer, Fabry took little time to agree.

The project started promptly. Tong set up a distribution system that could handle a higher volume of orders than Joy's previous distributions. Fabry managed to coordinate with Bob Guffy at AT&T and lawyers at the University of California to formally release Unix under terms agreeable to all. Joy incorporated Jim Kulp's job control, and added auto reboot, a 1K block file system, and support for the latest VAX machine, the VAX-11/750. By October 1980, a polished distribution that also included the Pascal compiler, the Franz Lisp system, and an enhanced mail handling system was released as 4BSD. During its nine-month lifetime, nearly 150 copies were shipped. The license arrangement was on a per-institution basis rather than a per machine basis; thus the distribution ran on about 500 machines.

With the increasingly wide distribution and visibility of Berkeley Unix, several critics began to emerge. David Kashtan at Stanford Research Institute wrote a paper describing the results of benchmarks he had run on both VMS and Berkeley Unix. These benchmarks showed severe performance problems with the Unix system for the VAX. Setting his future plans aside for several months, Joy systematically began tuning up the kernel. Within weeks he had a rebuttal paper written showing that Kashtan's benchmarks could be made to run as well on Unix as they could on VMS.

Rather than continue shipping 4BSD, the tuned-up system, with the addition of Robert Elz's auto configuration code, was released as 4.1BSD in June, 1981. Over its two- year lifetime about 400 distributions were shipped. The original intent had been to call it the 5BSD release; however, there were objections from AT&T that there would be customer confusion between their commercial Unix release, System V, and a Berkeley release named 5BSD. So, to resolve the issue, Berkeley agreed to change the naming scheme for future releases to stay at 4BSD and just increment the minor number.

4.2BSD

With the release of 4.1BSD, much of the furor over performance died down. DARPA was sufficiently satisfied with the results of the first contract that a new two-year contract was granted to Berkeley with funding almost five times that of the original. Half of the money went to the Unix project, the rest to other researchers in the Computer Science department. The contract called for major work to be done on the system so the DARPA research community could better do their work.

Based on the needs of the DARPA community, goals were set and work begun to define the modifications to the system. In particular, the new system was expected to include a faster file system that would raise throughput to the speed of available disk technology, support processes with multi-gigabyte address space requirements, provide flexible interprocess communication facilities that allow researchers to do work in distributed systems, and would integrate networking support so that machines running the new system could easily participate in the ARPAnet.

To assist in defining the new system, Duane Adams, Berkeley's contract monitor at DARPA, formed a group known as the "steering committee" to help guide the design work and ensure that the research community's needs were addressed. This committee met twice a year between April 1981 and June 1983. It included Bob Fabry, Bill Joy, and Sam Leffler of the University of California at Berkeley; Alan Nemeth and Rob Gurwitz of Bolt, Beranek, and Newman; Dennis Ritchie of Bell Laboratories; Keith Lantz of Stanford University; Rick Rashid of Carnegie-Mellon University; Bert Halstead of the Massachusetts Institute of Technology; Dan Lynch of The Information Sciences Institute; Duane Adams and Bob Baker of DARPA; and Jerry Popek of the University of California at Los Angeles. Beginning in 1984, these meetings were supplanted by workshops that were expanded to include many more people.

An initial document proposing facilities to be included in the new system was circulated to the steering committee and other people outside Berkeley in July, 1981, sparking many lengthy debates. In the summer of 1981, I became involved with the CSRG and took on the implementation of the new file system. During the summer, Joy concentrated on implementing a prototype version of the interprocess communication facilities. In the fall of 1981, Sam Leffler joined the CSRG as a full-time staff member to work with Bill Joy.

When Rob Gurwitz released an early implementation of the TCP/IP protocols to Berkeley, Joy integrated it into the system and tuned its performance. During this work, it became clear to Joy and Leffler that the new system would need to provide support for more than just the DARPA standard network protocols. Thus, they redesigned the internal structuring of the software, refining the interfaces so that multiple network protocols could be used simultaneously.

With the internal restructuring completed and the TCP/IP protocols integrated with the prototype IPC facilities, several simple applications were created to provide local users access to remote resources. These programs, rcp, rsh, rlogin, and rwho were intended to be temporary tools that would eventually be replaced by more reasonable facilities (hence the use of the distinguishing "r" prefix). This system, called 4.1a, was first distributed in April 1982 for local use; it was never intended that it would have wide circulation, though bootleg copies of the system proliferated as sites grew impatient waiting for the 4.2 release.

The 4.1a system was obsolete long before it was complete. However, feedback from its users provided valuable information that was used to create a revised proposal for the new system called the "4.2BSD System Manual." This document was circulated in February 1982 and contained a concise description of the proposed user interfaces to the system facilities that were to be implemented in 4.2BSD.

Concurrent with the 4.1a development, I completed the implementation of the new file system, and by June 1982, had fully integrated it into the 4.1a kernel. The resulting system was called 4.1b and ran on only a few select development machines at Berkeley. Joy felt that with significant impending changes to the system, it was best to avoid even a local distribution, particularly since it required every machine's file systems to be dumped and restored to convert from 4.1a to 4.1b. Once the file system proved to be stable, Leffler proceeded to add the new file system related system calls, while Joy worked on revising the interprocess communication facilities.

In the late spring of 1982, Joy announced he was joining Sun Microsystems. Over the summer, he split his time between Sun and Berkeley, spending most of his time polishing his revisions to the interprocess communication facilities and reorganizing the Unix kernel sources to isolate machine dependencies. With Joy's departure, Leffler took over responsibility for completing the project. Certain deadlines had already been established and the release had been promised to the DARPA community for the spring of 1983. Given the time constraints, the work remaining to complete the release was evaluated and priorities were set. In particular, the virtual memory enhancements and the most sophisticated parts of the interprocess communication design were relegated to low priority (and later shelved completely). Also, with the implementation more than a year old and the Unix community's expectations heightened, it was decided an intermediate release should be put together to hold people until the final system could be completed. This system, called 4.1c, was distributed in April 1983; many vendors used this release to prepare for ports of 4.2 to their hardware. Pauline Schwartz was hired to take over the distribution duties starting with the 4.1c release.

In June 1983, Bob Fabry turned over administrative control of the CSRG to Professors Domenico Ferrari and Susan Graham to begin a sabbatical free from the frantic pace of the previous four years. Leffler continued the completion of the system, implementing the new signal facilities, adding to the networking support, redoing the standalone I/O system to simplify the installation process, integrating the disc quota facilities from Robert Elz, updating all the documentation, and tracking the bugs from the 4.1c release. In August 1983, the system was released as 4.2BSD.

When Leffler left Berkeley for Lucasfilm following the completion of 4.2, he was replaced by Mike Karels. Karels' previous experience with the 2.9BSD PDP-11 software distribution provided an ideal background for his new job. After completing my Ph.D. in December 1984, I joined Mike Karels full-time at the CSRG.

The popularity of 4.2BSD was impressive; within eighteen months more than 1,000 site licenses had been issued. Thus, more copies of 4.2BSD had been shipped than of all the previous Berkeley software distributions combined. Most of the Unix vendors shipped a 4.2BSD system rather than the commercial System V from AT&T. The reason was that System V had neither networking nor the Berkley Fast filesystem. The BSD release of Unix only held its dominant commercial position for a few years before returning to its roots. As networking and other 4.2BSD improvements were integrated into the system V release, the vendors usually switched back to it. However, later BSD developments continued to be incorporated into System V.

4.3BSD

As with the 4.1BSD release, criticism was quick in coming. Most of the complaints were that the system ran too slowly. The problem, not surprisingly, was that the new facilities had not been tuned and that many of the kernel data structures were not well-suited to their new uses. Karels' and my first year on the project was spent tuning and polishing the system.

After two years of work spent tuning the system and refining the networking code, we made an announcement at the June 1985 Usenix conference that we anticipated releasing 4.3BSD later that summer. However, our release plans were brought to an abrupt halt by the folks at BBN. They correctly pointed out that we had never updated 4.2BSD with the final version of their networking code. Rather, we were still using the much-hacked initial prototype that they had given us many years earlier. They complained to DARPA that Berkeley was to implement the interface while BBN was supposed to implement the protocol, so Berkeley should replace the TCP/IP code in 4.3BSD with the BBN implementation.

Mike Karels got the BBN code and did an evaluation of the work that had been done since the prototype that was handed off to Berkeley. He decided that the best plan was to incorporate the good ideas from the BBN code into the Berkeley code base, but not to replace the Berkeley code base. The reason to retain the Berkeley code base was that it had gotten considerable testing and improvements from the widespread distribution in 4.2BSD. However, as a compromise, he offered to include both implementations on the 4.3BSD distribution and let users select which one to use in their kernel.

After reviewing Mike Karels' decision, DARPA decided that releasing two code bases would lead to unnecessary interoperability problems, and that just one implementation should be released. To decide which code base to use, they gave both to Mike Muuse of the Ballistics Research Laboratory, who was viewed by both Berkeley and BBN as an independent third party. After a month of evaluation, the report came back that the Berkeley code was more efficient but that the BBN code handled congestion better. The tie breaker was that the Berkeley code flawlessly ran all the tests while the BBN code panicked under some stress conditions. The final decision by DARPA was that 4.3BSD would stick with the Berkeley code base.

The polished 4.3BSD system was finally released in June 1986. As expected, it quelled many of the performance complaints, much as the 4.1BSD release quelled many of the complaints about 4BSD. Although most of the vendors had started the switch back to System V, large parts of 4.3BSD were carried over into their systems, particularly the networking subsystem.

In October 1986, Keith Bostic joined the CSRG. One condition of his employment was that he be allowed to finish up a project that he had been working on at his previous job, which was to port 4.3BSD to the PDP-11. While both Karels and I believed that it would be impossible to get a system that compiled to 250 Kbytes on the VAX to fit in the 64-Kbyte address space of the PDP-11, we agreed that Bostic could finish up his attempts to do so. Much to our amazement, the port was done successfully, using an intricate set of overlays and auxiliary processor states found on the PDP-11. The result was the 2.11BSD release, done by Casey Leedom and Bostic, which is still in use on some of the last remaining PDP-11's still in production in 1998.

Meanwhile, it was becoming increasingly obvious that the VAX architecture was reaching the end of its life and that it was time to begin considering other machines for running BSD. A promising new architecture of the time was made by Computer Consoles, Incorporated, and was called the Power 6/32. Unfortunately, the architecture died when the company decided to change its strategic direction. However, they did provide the CSRG with several machines that enabled us to finish the work, started by Bill Joy, of splitting the BSD kernel into machine-dependent and machine-independent parts. The result of this work was released as 4.3BSD-Tahoe in June 1988. The name Tahoe came from the development name used by Computer Consoles, Incorporated, for the machine that they eventually released as the Power 6/32. Although the useful lifetime of the Power 6/32 machine support was short, the work done to split the kernel into machine-independent and machine-dependent parts proved to be extremely valuable as BSD was ported to numerous other architectures.

Networking, Release 1

Up through the release of 4.3BSD-Tahoe, all recipients of BSD had to first get an AT&T source license. That was because the BSD systems were never released by Berkeley in a binary-only format; the distributions always contained the complete source to every part of the system. The history of the Unix system and the BSD system in particular had shown the power of making the source available to the users. Instead of passively using the system, they actively worked to fix bugs, improve performance and functionality, and even add completely new features.

With the increasing cost of the AT&T source licenses, vendors that wanted to build standalone TCP/IP-based networking products for the PC market using the BSD code found the per-binary costs prohibitive. So, they requested that Berkeley break out the networking code and utilities and provide them under licensing terms that did not require an AT&T source license. The TCP/IP networking code clearly did not exist in 32/V and thus had been developed entirely by Berkeley and its contributors. The BSD originated networking code and supporting utilities were released in June 1989 as Networking Release 1, the first freely-redistributable code from Berkeley.

The licensing terms were liberal. A licensee could release the code modified or unmodified in source or binary form with no accounting or royalties to Berkeley. The only requirements were that the copyright notices in the source file be left intact and that products that incorporated the code indicate in their documentation that the product contained code from the University of California and its contributors. Although Berkeley charged a $1,000 fee to get a tape, anyone was free to get a copy from anyone who already had received it. Indeed, several large sites put it up for anonymous ftp shortly after it was released. Given that it was so easily available, the CSRG was pleased that several hundred organizations purchased copies, since their fees helped fund further development.

4.3BSD-Reno

Meanwhile, development continued on the base system. The virtual memory system whose interface was first described in the 4.2BSD architecture document finally came to fruition. As was often the case with the CSRG, we always tried to find existing code to integrate rather than write something from scratch. So, rather than design a new virtual memory system, we looked around for existing alternatives. Our first choice was the virtual memory system that appeared in Sun Microsystem's SunOS. Although there was some discussion about Sun contributing the code to Berkeley, nothing came of those talks. So we went with our second choice, which was to incorporate the virtual memory system from the MACH operating system done at Carnegie-Mellon University. Mike Hibler at the University of Utah merged the core technology from MACH with the user interface described by the 4.2BSD architecture manual (which was also the interface used by SunOS).

The other major addition to the system at the time was a Sun-compatible version of the Network Filesystem (NFS). Again the CSRG was able to avoid writing the actual NFS code, instead getting an implementation done by Rick Macklem at the University of Geulph in Canada.

Although we did not yet have the complete feature set of 4.4BSD ready to ship, the CSRG decided to do an interim release to get additional feedback and experiences on the two major new additions to the system. This licensed interim release was called 4.3BSD-Reno and occurred in early 1990. The release was named after a big gambling city in Nevada as an oblique reminder to its recipients that running the interim release was a bit of a gamble.

Networking, Release 2

During one of our weekly group meetings at the CSRG, Keith Bostic brought up the subject of the popularity of the freely-redistributable networking release and inquired about the possibility of doing an expanded release that included more of the BSD code. Mike Karels and I pointed out to Bostic that releasing large parts of the system was a huge task, but we agreed that if he could sort out how to deal with reimplementing the hundreds of utilities and the massive C library then we would tackle the kernel. Privately, Karels and I felt that would be the end of the discussion.

Undeterred, Bostic pioneered the technique of doing a mass net-based development effort. He solicited folks to rewrite the Unix utilities from scratch based solely on their published descriptions. Their only compensation would be to have their name listed among the Berkeley contributors next to the name of the utility that they rewrote. The contributions started slowly and were mostly for the trivial utilities. But as the list of completed utilities grew and Bostic continued to hold forth for contributions at public events such as Usenix, the rate of contributions continued to grow. Soon the list crossed one hundred utilities and within 18 months nearly all the important utilities and libraries had been rewritten.

Proudly, Bostic marched into Mike Karels' and my office, list in hand, wanting to know how we were doing on the kernel. Resigned to our task, Karels, Bostic, and I spent the next several months going over the entire distribution, file by file, removing code that had originated in the 32/V release. When the dust settled, we discovered that there were only six remaining kernel files that were still contaminated and which could not be trivially rewritten. While we considered rewriting those six files so that we could release a complete system, we decided instead to release just what we had. We did, however, seek permission for our expanded release from folks higher up in the University administration. After much internal debate and verification of our method for determining proprietary code, we were given the go-ahead to do the release.

Our initial thought was to come up with a whole new name for our second freely-redistributable release. However, we viewed getting a whole new license written and approved by the University lawyers as an unnecessary waste of resources and time delay. So, we decided to call the new release Networking Release 2 since we could just do a revision of the approved Networking Release 1 license agreement. Thus, our second greatly expanded freely-redistributable release began shipping in June 1991. The redistribution terms and cost were the same as the terms and cost of the first networking release. As before, several hundred individuals and organizations paid the $1,000 fee to get the distribution from Berkeley.

Closing the gap from the Networking Release 2 distribution to a fully functioning system did not take long. Within six months of the release, Bill Jolitz had written replacements for the six missing files. He promptly released a fully compiled and bootable system for the 386-based PC architecture which he called 386/BSD. Jolitz's 386/BSD distribution was done almost entirely on the Net. He simply put it up for anonymous FTP and let anyone who wanted it download it for free. Within weeks he had a huge following.

Unfortunately, the demands of keeping a full-time job meant that Jolitz could not devote the time needed to keep up with the flood of incoming bug fixes and enhancements to 386/BSD. So, within a few months of the release of 386/BSD, a group of avid 386/BSD users formed the NetBSD group to pool their collective resources to help maintain and later enhance the system. Their releases became known as the NetBSD distribution. The NetBSD group chose to emphasize the support of as many platforms as possible and continued the research style development done by the CSRG. Until 1998, their distribution was done solely over the Net; no distribution media was available. Their group continues to target primarily the hardcore technical users. More information about the NetBSD project can be found at *http://www.netbsd.org.*

The FreeBSD group was formed a few months after the NetBSD group with a charter to support just the PC architecture and to go after a larger and less technically advanced group of users, much as Linux had done. They built elaborate installation scripts and began shipping their system on a low cost CD-ROM. The combination of ease of installation and heavy promotion on the Net and at major trade shows such as Comdex led to a fast, large growth curve. Certainly FreeBSD currently has the largest installed base of all the Release 2-derived systems.

FreeBSD has also ridden the wave of Linux popularity by adding a Linux emulation mode that allows Linux binaries to run on the FreeBSD platform. This feature allows FreeBSD users to use the ever-growing set of applications available for Linux while getting the robustness, reliability, and performance of the FreeBSD system. The group recently opened a FreeBSD Mall (*http://www.freebsdmall.com*), which brings together many parts of the FreeBSD community, including consulting services, derived products, books, and a newsletter.

In the mid-1990s, OpenBSD spun off from the NetBSD group. Their technical focus was aimed at improving the security of the system. Their marketing focus was to make the system easier to use and more widely available. Thus, they began producing and selling CD-ROMs with many of the ease-of-installation ideas from the FreeBSD distribution. More information about the OpenBSD project can be found at *http://www.openbsd.org.*

The Lawsuit

In addition to the groups organized to freely redistribute systems built around the Networking Release 2 tape, a company, Berkeley Software Design, Incorporated (BSDI), was formed to develop and distribute a commercially supported version of the code. (More information about BSDI can be found at *http://www.bsdi.com*.) Like the other groups, they started by adding the six missing files that Bill Jolitz had written for his 386/BSD release. BSDI began selling their system including both source and binaries in January 1992 for $995. They began running advertisements touting their 99% discount over the price charged for System V source plus binary systems. Interested readers were told to call 1-800-ITS-Unix.

Shortly after BSDI began their sales campaign, they received a letter from Unix System Laboratories (USL) (a mostly-owned subsidiary of AT&T spun off to develop and sell Unix). The letter demanded that they stop promoting their product as Unix and in particular that they stop using the deceptive phone number. Although the phone number was promptly dropped and the advertisements changed to explain that the product was not Unix, USL was still unhappy and filed suit to enjoin BSDI from selling their product. The suit alleged that the BSDI product contained proprietary USL code and trade secrets. USL sought to get an injunction to halt BSDI's sales until the lawsuit was resolved, claiming that they would suffer irreparable harm from the loss of their trade secrets if the BSDI distributions continued.

At the preliminary hearing for the injunction, BSDI contended that they were simply using the sources being freely distributed by the University of California plus six additional files. They were willing to discuss the content of any of the six added files, but did not believe that they should be held responsible for the files being distributed by the University of California. The judge agreed with BSDI's argument and told USL that they would have to restate their complaint based solely on the six files or he would dismiss it. Recognizing that they would have a hard time making a case from just the six files, USL decided to refile the suit against both BSDI and the University of California. As before, USL requested an injunction on the shipping of Networking Release 2 from the University and on the BSDI products.

With the impending injunction hearing just a few short weeks away, preparation began in earnest. All the members of the CSRG were deposed as were nearly everyone employed at BSDI. Briefs, counter-briefs, and counter-counter-briefs flew back and forth between the lawyers. Keith Bostic and I personally had to write several hundred pages of material that found its way into various briefs.

In December 1992, Dickinson R. Debevoise, a United States District Judge in New Jersey, heard the arguments for the injunction. Although judges usually rule on injunction requests immediately, he decided to take it under advisement. On a Friday about six weeks later, he issued a forty-page opinion in which he denied the injunction and threw out all but two of the complaints. The remaining two complaints were narrowed to recent copyrights and the possibility of the loss of trade secrets. He also suggested that the matter should be heard in a state court system before being heard in the federal court system.

The University of California took the hint and rushed into California state court the following Monday morning with a counter-suit against USL. By filing first in California, the University had established the locale of any further state court action. Constitutional law requires all state filing to be done in a single state to prevent a litigant with deep pockets from bleeding an opponent dry by filing fifty cases against them in every state. The result was that if USL wanted to take any action against the University in state courts, they would be forced to do so in California rather than in their home state of New Jersey.

The University's suit claimed that USL had failed in their obligation to provide due credit to the University for the use of BSD code in System V as required by the license that they had signed with the University. If the claim were found to be valid, the University asked that USL be forced to reprint all their documentation with the appropriate due credit added, to notify all their licensees of their oversight, and to run full-page advertisements in major publications such as *The Wall Street Journal* and *Fortune* magazine notifying the business world of their inadvertent oversight.

Soon after the filing in state court, USL was bought from AT&T by Novell. The CEO of Novell, Ray Noorda, stated publicly that he would rather compete in the marketplace than in court. By the summer of 1993, settlement talks had started. Unfortunately, the two sides had dug in so deep that the talks proceed slowly. With some further prodding by Ray Noorda on the USL side, many of the sticking points were removed and a settlement was finally reached in January 1994. The result was that three files were removed from the 18,000 that made up Networking Release 2, and a number of minor changes were made to other files. In addition, the University agreed to add USL copyrights to about 70 files, although those files continued to be freely redistributed.

4.4BSD

The newly blessed release was called 4.4BSD-Lite and was released in June 1994 under terms identical to those used for the Networking releases. Specifically, the terms allow free redistribution in source and binary form subject only to the constraint that the University copyrights remain intact and that the University receive credit when others use the code. Simultaneously, the complete system was released as 4.4BSD-Encumbered, which still required recipients to have a USL source license.

The lawsuit settlement also stipulated that USL would not sue any organization using 4.4BSD-Lite as the base for their system. So, all the BSD groups that were doing releases at that time, BSDI, NetBSD, and FreeBSD, had to restart their code base with the 4.4BSD-Lite sources into which they then merged their enhancements and improvements. While this reintegration caused a short-term delay in the development of the various BSD systems, it was a blessing in disguise since it forced all the divergent groups to resynchronize with the three years of development that had occurred at the CSRG since the release of Networking Release 2.

4.4BSD-Lite, Release 2

The money received from the 4.4BSD-Encumbered and 4.4BSD-Lite releases was used to fund a part-time effort to integrate bug fixes and enhancements. These changes continued for two years until the rate of bug reports and feature enhancements had died down to a trickle. The final set of changes was released as 4.4BSD-Lite, Release 2 in June 1995. Most of these changes eventually made it into the other systems source bases.

Following the release of 4.4BSD-Lite Release 2, the CSRG was disbanded. After nearly twenty years of piloting the BSD ship, we felt that it was time to let others with fresh ideas and boundless enthusiasm take over. While it might seem best to have a single centralized authority overseeing the system development, the idea of having several groups with different charters ensures that many different approaches will be tried. Because the system is released in source form, the best ideas can easily be picked up by other groups. If one group becomes particularly effective, they may eventually become the dominant system.

Today, the open source software movement is gaining increased attention and respect. Although the Linux system is perhaps the most well-known, about half of the utilities that it comes packaged with are drawn from the BSD distribution. The Linux distributions are also heavily dependent on the complier, debuggers, and other development tools written by the Free Software Foundation. Collectively, the CSRG, the Free Software Foundation, and the Linux kernel developers have created the platform from which the Open Source software movement has been launched. I am proud to have had the opportunity to help pioneer the Open Source software movement. I look forward to the day when it becomes the preferred way to develop and buy software for users and companies everywhere.

First Published on O'Reilly.com | This pdf version is created by LINUXjunkies.org