

"The three ex-patchkit coordinators were all in agreement that the patchkit had to die."

—Jordan K. Hubbard, FreeBSD Web Page

"How can you people sleep at night?"
—Bill Jolitz, to the Berkeley CSRG

"Good luck!"
—Original UNIX Version 6 installation instructions, and still a valid sentiment

Mr. Protocol Goes to the Toy Store

Q: What in the world is that? It looks like a PC, but it has Sun stuff on the screen! Looks like an X Window session running an NFS server!

A: Too right. This is the legacy of the people who gave Sun its start. How did that come about? Good question. Excellent question, in fact. Let's take a look.

Mr. Protocol has held forth often ("Too often!" you cry. Tsk, tsk.) on the early days of the ARPANET. The net was originally started so that the recipients of ARPA's largesse could share computing resources and information, in the days when "computing resources" meant multimillion-dollar mainframes, of which you had only one...or two, if you were very big and/or very rich.

However, in those evolutionarily healthy days, there was a great diversity of species. This diversity had its good and bad points. It allowed for the exploration of many more hardware and operating system innovations than are commonly explored today. It allowed for the invention of a great number of the staples of today's Internet, such as mailing lists and anonymous ftp repositories, in a non-homogeneous environment. It allowed

very little actual research software to be exchanged, except among sites that happened to have chosen the same mainframe platform. All too often, the choice of mainframe was made on the basis of which mainframe manufacturer was most desperate for the educational market that year.

Finally, a number of researchers and

ers, it was decided to make the UNIX operating system, running on the DEC VAX, the "standard" research platform, in the sense that ARPA would fund an organization to remedy deficiencies in the software.

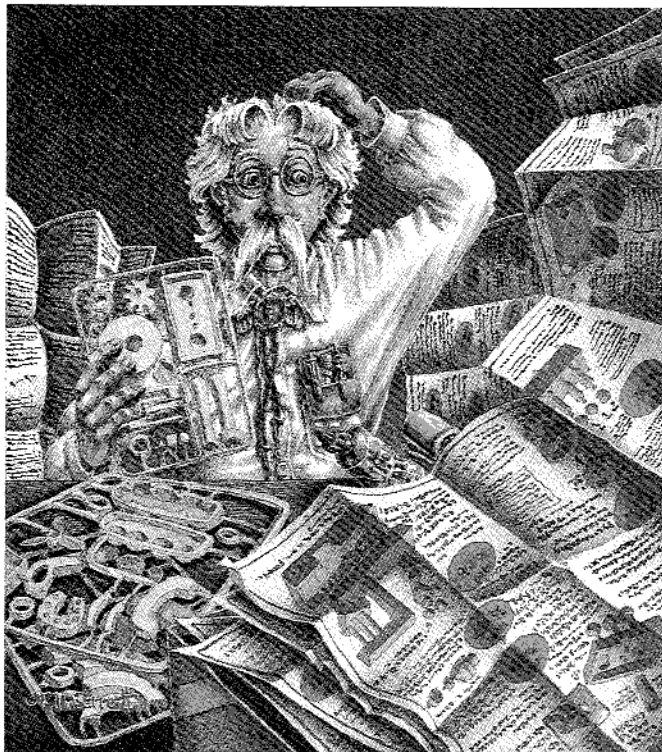
This did not mean that ARPA was going to pay for a tape librarian or a batch system. These are researchers

we're talking about, remember. What ARPA paid for was a general souping-up of the operating system, together with an implementation of the (then relatively new) TCP/IP Internet protocol stack.

Many people believe that the Berkeley implementation of TCP/IP is the only one ever written, aside from the AT&T STREAMS implementation, mainly because most people have never seen another one. But at about this time, BBN wrote its own TCP/IP implementation, under another ARPA contract, and there were more than a few folks who felt that, on the whole, this implementa-

tion was the better one. No one will ever know, since no real bake-off was ever held.

ARPA felt that paying for two competing implementations was a bit too evolutionary and told everyone to get



at least one ARPA program director became convinced that the best way to further the research interests of the community as a whole was to create a common operating environment. At a committee meeting of ARPA research-

together and pick one. Deadlock was the only result, and ARPA picked Berkeley. Its Berkeley Software Distributions, known as BSD UNIX, predated this contract and had put Berkeley on the map as a significant player in the UNIX market. Of course, this was partly because Berkeley had found it necessary to support 70 simultaneous users (though not well!) on a single PDP-11/70 minicomputer, which in turn resulted in significant work on the streamlining and stabilization of UNIX.

Long, long ago, in a start-up far, far away, there were a bunch of filmmakers and special effects wonks who decided that, if money were no object, they could by golly build the best special effects and postproduction facility that man had ever seen. They quickly decided that Xerox PARC had had the right idea years ago, with its invention of the personal computer, not to be confused with the travesty of the same name perpetrated by IBM and Microsoft. These folks wanted virtual memory and a real multitasking operating

system, because that's what they'd all grown up with in the universities. MS-DOS was to laugh, if you weren't so busy crying, or doing less mentionable things.

Three Rivers Computing—in Pittsburgh, of all places—looked like the only place building real workstations for money at that point, Xerox having refused to cease contemplation of the wonders of selenium long enough to actually sell any workstations. This was a spin-off of technology developed at Carnegie Mellon University, which had evolved a rather lenient, if odd, policy on patents as a result of a graduate student having developed a best-selling text-processing system and refusing to give it to the university. The student got rich, and the university avoided a defeat in court.

Three Rivers developed a rather nifty-looking workstation called the Perq. The filmmakers, working provisionally from a company called Sprocket Systems, bought a bunch of them. The lead programmer figured out how to change the screen bitmap so that the

workstations labeled themselves "Jerq." Pascal was not a rampaging success as a system programming language.

This left Sprocket Systems in a sad state, with a bunch of money down the tubes. They decided to roll their own. But not alone. Being probably the only real customers Three Rivers ever had, they had had enough of "alone."

Sprocket formed a partnership with the fledgling Sun Microsystems Inc., which had in turn glommed on to a board designed at Stanford for its Stanford University Network, or SUN, project. The board was nice for its time but had no software. Sun and Sprocket joined forces, with Valid Logic Systems adding programming muscle, being themselves desperate for a platform on which to run their CAD/CAM design system.

Software was rocky in those days, in large part because the I/O controllers available were equally flaky. Originally, the test versions of the OS were downloaded over serial lines and used those same serial lines as a pseudo-disk. This was, to put it charitably, slow. But it



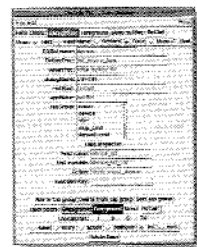
Searching For UNIX Tools?

Text & X Windows - Runs On All UNIX

PinQuik allows the user to build menus, forms and panels, quickly and easily. From simple menus to comprehensive interfaces - all through a simple menu driven set of tools. PinQuik enables the user to completely customize their system to meet their needs.

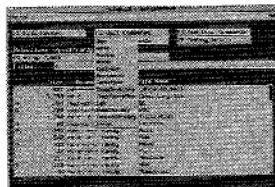
Tools For The Programmer

PinQuik incorporates Tcl - a powerful scripting language that easily allows users to build applications. With Tcl, no development system is needed and the user has the ability to perform rapid interface prototyping. The GUI and Text Form builder allows you to visually build screens. All screens run on both X and Text. Build interfaces on one UNIX box and it works on all of your other UNIX boxes.



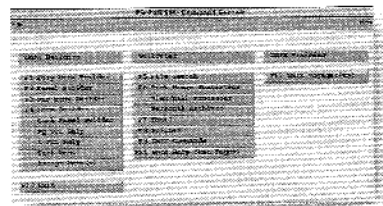
Tools for the SysAdm

- Easy Menu Builder allows a new SysAdm with limited knowledge of UNIX to quickly build menus for their users. As UNIX knowledge is gained they can move into more technical interfaces, forms, bar menus, panels, pull downs, etc.
- Screen and applications are portable to all UNIX without recompiling programs.
- PinQuik comes with a complete set of pre-built menus and utilities that are very beneficial to the SysAdm - navigational aids, safety net, personal archiver, are just a few examples.



PinQuik-Plus

PinQuik-Plus has all the power of PinQuik and 4 to 5 hours of UNIX Computer Based Training, the UNIX Voyager™ from UniWorx, built in. Now the user can learn UNIX while they work.



For information call 800-489-3592



PROSIM Corp.
14201 Memorial Drive, Suite Four
Houston, Texas 77079
(713) 558-6866 • FAX (713) 558-8857
EMAIL INFO@prosim.com

was enough to get running.

The most famous of the initial principals of Sun Microsystems was Bill Joy, who up until that time had been a graduate student at Berkeley, where he was the most prolific coder on the ARPA/Berkeley UNIX project. The software and most of the thought processes went with him to Sun.

Sun kept on rolling, and so did Berkeley, but ARPA's work had essentially been done in the finest style—the sales of research machines slipped away from DEC and its VAX architecture and moved to Sun. Sun never enjoyed quite the dominance in the educational market that DEC did in its day, but to be fair, DEC peddled three different architectures—the PDP-11, the VAX and the DECSystem-20 (known in its earlier incarnation as the PDP-10). Sun marketed several different architectures but only one OS and made the OS as interoperable as possible. In trying to make one system be all things to all people, they did succeed in narrowing their base somewhat, so that companies such as HP could capture significant minority market share—along with such folks as IBM and DEC, once they woke up and sniffed the Folger's.

Berkeley kept on rolling too, but it seemed less and less relevant. Mind, there were some retro folks out there. One gent named Bill Jolitz insisted on folding all of the major BSD developments for the VAX back into a PDP-11 platform, for example, which kept at least some PDP-11 systems running long past the point when rational people would have phased them out.

Mr. Protocol has never known a single instance of an educational institution making anything like a rational decision on capital equipment, however. They seem to be constitutionally incapable of it, which is doubtless why state legislators delight in slashing the budgets of such institutions. So, far from developing strong constituencies, they in general seem to make ever-more dramatic and capricious decisions. As the bottom layers still have to get the work done, this has resulted in some of the most amazing resuscitation efforts in the history of computer science.

AN OBVIOUS PROBLEM:

**Your business applications
aren't running at peak efficiency,
so you're losing time,
customers, and money.**

1-800-488-1111

Circle No. 21 on Inquiry Card

Eventually, ARPA decided that the Berkeley UNIX project had served its purpose, at least as far as ARPA was concerned, by providing technology sufficiently mature for it to be picked up by the industrial community and continue under its own head of steam in the commercial arena. This is, in fact, a significant success for ARPA, and one that should not be underrated. This is how technology transfer is supposed to—and seldom does—work. The Berkeley Computer Systems Research Group, unable to obtain funding as a consortium for UNIX-related research, disbanded after delivering 4.4BSD, its final product.

Somewhere along the way, though, Bill Jolitz had had some fun. It had already been noted that Berkeley had hammered on so much code that very little of it bore any resemblance to the code originally produced and licensed by AT&T. In fact, only about six of the hundred or so modules making up the kernel of the operating system still contained code that could be considered to be “encumbered” by the AT&T license. Mr. Jolitz wrote new versions of those six modules, and proceeded to wow a Usenix Association meeting by showing off Berkeley UNIX running on his Toshiba laptop. This display of an airplane-capable UNIX was a great crowd-pleaser. It finally gave some people an idea of what they could do with all those cheap boxes they had sitting around that could run only one program at a time, a program whose kernel had to fit into 640 KB of memory, at that. They could run UNIX on it, instead.

Several things happened at once. First, a whole bunch of people no one had ever heard of suddenly started writing a whole lot of code. What kind of code? Mr. Protocol is glad you asked.

He's glad because there was an immediate dichotomy. On the one hand, we had a whole bunch of loosely associated people who believed that Richard Stallman's GNU Project was too right-wing. GNU software is bound by the GNU “copyleft,” which is actually a perfectly standard and perfectly legal copyright that insists that anyone who passes along software

derived from GNU software must also pass along the source code to that software. This has resulted in some very peculiar attitudes on the part of a large number of otherwise normal people.

It is a commonly acknowledged fact that organizations in late 20th-century America are not, in fact, run by the officers supposedly elected or appointed to this duty. They are run by their lawyers. It is almost always the case that any corporation, university or other public body will invariably do, or not do, whatever its lawyers tell it to do, or not do. Sun Microsystems, for example, originally expressed the intent to submit its reworked memory management code back to Berkeley for inclusion in the Berkeley release, as a token of appreciation for all the software they took, that got their company off the ground. Ha, ha. Only a short time later, when BBN looked like it was willing to pay any amount of money, under ARPA contract, for the source code to Sun's version of UNIX, Sun turned them down flat, on the advice of their lawyers. It wasn't until ARPA quietly informed Sun that such an attitude would result in No ARPA Dollars Going to Sun Ever Again that the lawyers were overruled and instructed to forge a source license. (Berkeley eventually adopted the memory management policy from CMU's Mach operating system.)

Lawyers are people too. If you ever doubt this, please note that no lower life form has ever exhibited such a wide range of demonstrably irrational or inconsistent behavior. The Law is whatever the courts say The Law is, and lawyers make most of their money by guessing what the courts will say in the future. Like all prognosticators, they are out of their depth when presented with anything new and original, such as the GNU copyleft.

The lawyers in most organizations never hear about the copyleft. Most people are too smart ever to give them a look at it. In some of the remaining organizations, the lawyers think the copyleft is a copyright, and say that it means what it says it means. The remainder of the organizations are told by their lawyers that the copyleft might as well be written in Urdu, and that the

courts are going to decide what it means by casting chicken bones, and that they should stay the hell away from anything remotely resembling copylefted software, so there.

The interesting result is that the people working on this new, funky, PC-based free UNIX stuff decided early on that they were sick and tired of encumbrances. Any encumbrances. And so was created 386BSD, a freely available UNIX lookalike system for IBM PC clones.

Of course, there was a little more to it than that. Quite a bit, in fact. In the absence of a copyright *or* a copyleft, except for the original copyright of the Regents of the University of California, a for-profit (for immense profit, its founders sincerely hope) company was founded to support and sell it, with the cooperation of the remaining CSRG members at Berkeley (some of whom, not being entirely dim, perceived a way to cash in on their expertise). This exercised Mr. Jolitz no end, resulting in one of the more hair-raising exchanges ever seen at a Usenix. Mr. Protocol, in fact, had never seen anything like it since the day when Jeff Schriebman got up in front of the entire UNIX Users' Group, all 200 or so of them, and announced that he was actually in business to support UNIX for money. This blatant commercialism and self-promotion appalled everyone present and quite naturally made the president of the users' group livid with rage. UNIX was a research tool, and commercialism in the academic arena was anathema and would obviously result in the ruin of the group.

At the Berkeley BOF session, Bill Jolitz stood up and stunned the audience and CSRG members alike with his opening question: “How can you people sleep at night?” To which Keith Bostic replied, “What?” Bill repeated the question, and went on to castigate the entire CSRG for taking what Bill regarded as the best free software effort anywhere and subverting it to their own personal profit, by permitting BSD Inc. to make off with the software and attempt to earn a skazillion dollars from it. The CSRG members pointed out that 4.4BSD was still in process and that they were earnestly going to

try to come out with an unencumbered version of it. In the end, this did not quite happen. The final, unencumbered version of 4.4BSD, called "4.4BSD Lite," is not a bootable operating system.

At a loss, as usual, on how to cash in on anything as far removed from telephones as UNIX, AT&T eventually spun off a subsidiary, UNIX Systems Laboratories, to deal with it. USL was acquired in fairly short order by Novell, which had apparently decided if you can't lick 'em, buy 'em out. Novell quickly noticed that this bunch calling themselves BSD Inc. was selling something that sure as heck looked like what they'd just paid a boodle to get the licensing rights to. Negotiations and court appearances alternated in a slow and stately dance, until Berkeley finally solved the problem by issuing its final product, 4.4BSD, which all sides agreed had no encumbered software in it anywhere. It looked very dicey there for a while, though.

For whatever reason, after releasing Version 0.9 of 386BSD, Bill Jolitz effectively disappeared from the scene. He spent considerable time rewriting large portions of the operating system, but no newer release ever appeared from him. An ad hoc group started to pull together via the Net to handle matters in the interim. This ad hoc group shortly split into two ad hoc groups, each with a different focus. Mr. Protocol has mentioned both of these groups in a previous column. One effort, called NetBSD, is devoted to porting 4.4BSD to every platform in sight, including Suns as well as PCs. The other effort, called FreeBSD, concentrates on platforms based on the Intel x86 chipset, that is, on PCs and PC clones.

Both operating systems have been substantially rewritten since Jolitz's original efforts, to incorporate 4.4BSD, the final Berkeley software release. The impetus to port the system to the Intel platform was Jolitz's, though, and Mr. P. sends him a crate of Big Stuf Ding-Dongs.

This story would not be complete without mention of a significant parallel development. Linus Torvalds, not content with basking in the balmy

THE OBVIOUS SOLUTION:

LANDMARK[®]



Landmark Systems Corporation helps you monitor every component that affects your mission-critical applications.

To find out more, call us today for your free white paper "Extending Performance Management Across the Enterprise."

LANDMARK SYSTEMS CORPORATION

8000 Towers Crescent Drive
Vienna, Virginia 22182-2700

1-800-488-1111

Circle No. 22 on Inquiry Card

atmosphere of his native Finland, apparently spent many dark winter nights writing a lookalike version of AT&T's UNIX System V Release 4 (SVR4). This package, called Linux, is the latest entry in the juicy operating system wars that keep the Berkeley and AI&'I' camps amused for hours and hours by lobbing overripe fruit at each other.

More significantly, software is traded back and forth between the two camps. Sun's Solaris product also represents a blend of BSD UNIX and System V, but it must be noted that this was a shotgun marriage and resulted in a single product. Linux and NetBSD/FreeBSD remain very distinct products, and the requisite amount of fruit-lobbing is competently handled by both sides. It is perhaps surprising, then, how few people enter into the spirit of good old wholehearted inter-nicine warfare and how many people pragmatically borrow vital code back and forth between the two systems. Furthermore, the systems make an

effort to provide a compatibility mode in their kernels to run binaries created for a different flavor of UNIX.

The most interesting feature of these systems is their cost. Powerful Pentium workstations cost about one-fourth to one-fifth as much as an equivalent Sun product, but the Pentium running a free UNIX clone can communicate with the Sun not only via standard TCP/IP services such as Telnet and FTP, but also via NFS and NIS. In fact, these systems are already finding their way into cracks and crannies of the commercial world, often as Web servers and firewalls. And, since the source code is unencumbered and freely available, the halcyon days of self-maintenance and source code hackery are returning in force as a whole new generation finally gets its hands on UNIX source code. This is a frightening thought, actually. And, while NetBSD, FreeBSD and Linux are supported only by consultants, if at all, the BSDI product is fully supported commercially.

One's immediate reaction is to wonder how to make use of this phenomenon to either earn or to save a boodle of money. The obstacles to achieving either are formidable. Without support, a home-grown systems team of moderate-to-severe competence levels must be fielded to handle installation and support problems throughout the organization. System support takes time and money if you're maintaining the source yourself. Of course, the various consortia are actually quite snappy about fixing major bugs, because they all depend on the same system.

Another problem is hardware specification. Sometimes it's hard to determine a configuration in which all of the pieces are supported by the UNIX flavor du jour, be it NetBSD, FreeBSD or Linux. Even a configuration that is supported in theory may run up against the fact that standards in the PC industry are a bit thin. It is very possible, in fact it is all too easy, to buy a system in pieces and discover that the pieces don't quite work together when assembled into a complete computer.

The third major headache is software. Many free packages, and even a few commercial packages, are available

for these new operating systems. However, many research and commercial efforts depend upon third-party software packages from companies that not only have never heard of, say, FreeBSD, but don't want to hear of it, or about it.

In many shops, a middle ground may be acceptable. At the expense of retraining and beefing up the systems administration personnel to handle the self-support aspect, a large percentage of the end-user workstations on people's desks might be replaced by equivalent PC-based systems, at substantially lower cost.

To handle proprietary or commercial software, the organization would still maintain several centrally located "compute servers," which are machines manufactured by major vendors, upon which the necessary commercial applications could be run.

It should be noted that these operating systems are not toys, nor do they have gigantic step-by-step manuals from Microsoft (or Sun!). This is UNIX the Cantankerous, the original buster of chops, and if you do not know what you are doing when you start out, you will soon learn. You will learn fastest if you read code, but in the finest tradition of UNIX Darwinism, you will learn all kinds of arcane stuff or you will quit running UNIX.

But it sure is cheap. ➔

Mike O'Brien has been noodling around the UNIX world for far too long a time. He knows he started out with UNIX Research Version 5 (not System V, he hastens to point out), but forgets the year. He thinks it was around 1975 or so.

He founded and ran the first nationwide UNIX Users Group Software Distribution Center. He worked at Rand during the glory days of the Rand editor and the MH mail system, helped build CSNET (first at Rand and later at BBN Labs Inc.) and is now at an aerospace research corporation.

Mr. Protocol refuses to divulge his qualifications and may, in fact, have none whatsoever. His email address is amp@expert.com.

Where to Get It

Linux is available by consulting www.linux.org. FreeBSD's home page is at www.freebsd.org, and NetBSD, while lacking a Web page, is available for anonymous ftp at <ftp://netbsd.org>. Linux and FreeBSD are available on CD-ROM from:

Walnut Creek Inc.
1547 Pike Lane, Ste. 260
Walnut Creek, CA 94596
Phone: (510) 674-0783
FAX: (510) 674-0821
Email: info@cdrom.com
Web: <http://www.cdrom.com/>

NetBSD (as well as FreeBSD and Linux) is available on CD-ROM from:

Infomagic Inc.
P.O. Box 30370
Flagstaff, AZ 86003-0370
Orders: (800) 800-6613
Phone: (602) 526-9565
FAX: (602) 526-9573
Email: info@infomagic.com
Web: <http://www.infomagic.com/>