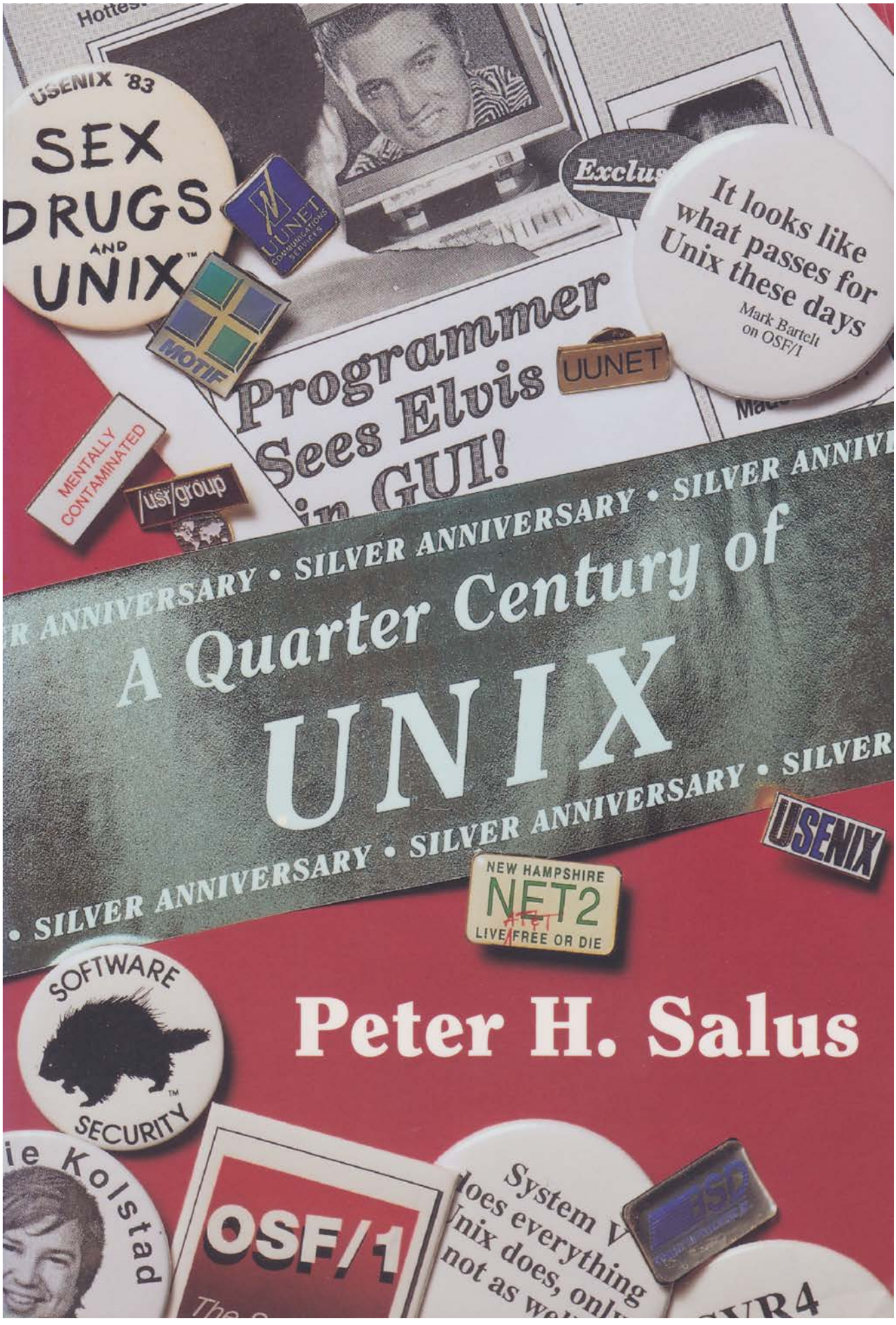

USENIX '83
SEX DRUGS AND UNIX™
UUNET COMMUNICATIONS SERVICES
MOTIF
It looks like what passes for Unix these days
Mark Bartelt on OSF/1
UUNET
Programmer Sees Elvis in GUI!
MENTALLY CONTAMINATED
/usr/group
SILVER ANNIVERSARY • SILVER ANNIVERSARY • SILVER ANNIVERSARY
A Quarter Century of
UNIX
SILVER ANNIVERSARY • SILVER ANNIVERSARY • SILVER
USENIX
NEW HAMPSHIRE
NET2
LIVE FREE OR DIE
SOFTWARE SECURITY
Peter H. Salus
OSF/1
System V does everything Unix does, only not as well
ASP

CHAPTER

12

Style and Tools

Just about the time that Ritchie and Thompson were working on their paper for the ACM symposium, Brian Kernighan and P.J. Plauger were collaborating on *Elements of Programming Style*. Kernighan has said, "We were writing it in 1973, and we finished it early in '74." Plauger remarked:

> Brian Kernighan and I ended up with adjacent offices at Bell Labs in Murray Hill almost by accident....
>
> We began by commiserating over the sad state of computer programming and ended up in the authoring business, both for the first time, by writing this little book over a period of about four months. As far as I know, it marks the first time that "programming style" was identified in print as a legitimate topic of discussion by adults.
>
> The book is terribly dated now, of course....

Kernighan told Peter Collinson:

> Plauger and I wrote the style book and that worked out pretty well and was fun. The idea behind the style book was to take a large number of programs and criticize them: that isn't right, that could have been better. We weren't saying *how* to do things, rather how *not* to do them. It was FORTRAN and PL/I, at the time they were the dominant languages in the community we were aiming at.

Programming Style made quite a splash. But, as both Plauger and Kernighan remark, it concerned PL/I and FORTRAN. Their next collaboration was far more general, and far more influential. Collinson asked Kernighan how he became involved with the *Software Tools* book, he responded, "It's hard to say."

> A couple of years later, we decided that the time had come to tell people a little about how they ought to do things. By then, we had a clearer picture of some of the benefits of the Unix environment: the advantage that you can get by piping programs together and building things that were going to be filters. It wasn't clear what to use as a language for the book. C, of course, didn't exist in very many environments. I had already done Ratfor [Rational FORTRAN]. Ratfor was around, it had simply stolen the good appearance of C, but didn't add much beyond it. It was fine, it converted FORTRAN into a programming language.
>
> We decided to use Ratfor as the programming language for the book which was really unconventional, [because] there were very few users. There was no-one in this group here, because nobody wrote in FORTRAN at that time except the numerical analysts who were worried about portability and they did not want to produce unreadable FORTRAN which is what Ratfor did.
>
> The original version was written in C with a small amount of yacc grammar. [**yacc** is "yet another compiler-compiler," written by Steve Johnson.] Given that as a bootstrap, it took a very short time to write it in Ratfor. It can then be bootstrapped on a machine only running FORTRAN. Part of the program distribution for the Software Tools package was Ratfor in Ratfor, and [part] Ratfor in FORTRAN. The first thing on the tape was Ratfor in FORTRAN so you could just peel it off and start running.

Collinson then asked, "So you have Ratfor, how do you get from there to Software Tools?"

> Then you sit down and ask what are the things that are interesting in a Unix system. It's the fact that there is this large number of very small tools that you can glue together in interesting ways. That takes you through the first three chapters of the book: the ideas of character input and character output as the most common denominator, text is everything, data is just streams of lines of text, and maybe up as far as archiving.
>
> Then we do regular expressions. These are a very important fundamental notion in Unix. Using this, the next couple of things are the pattern searching, the **grep** family, and then the editor. Actually, sorting is one of those tools that you put in as well. There are interesting algorithmic things that you can say about sorting.
>
> The only other thing that we do is text formatting, so we do a little simple formatter. Finally, we do Ratfor, saying this is the program you have been using all along. This is Ratfor in Ratfor.

Software Tools came out with a Ratfor tape, which had been written for the book. But the book came first. Kernighan remarked,

> Ratfor was independent of the book, but it became the vehicle for the book. The version that was described in the book was simpler than the version that was on the tape because you didn't learn anything new by having the full implementation described to you. There were groups like the Software Tools group that sprang up; which was primarily done at the Lawrence Berkeley Lab in California. These were people like Debbie Scherrer, Joe Sventek, and Dennis Hall. They set up the Software Tools group and they did some really nice stuff with it to create the "Virtual Operating System" and all kinds of stuff. Neat tricks across multiple machines. It was all done in a very clean way.

Recall the Unix Philosophy: **Write programs that do one thing and do it well.** That was the idea behind the tools. And it was the idea that fired up those folks at LBL.

Debbie Scherrer told me:

> You know, I started at UOP [University of the Pacific], but they had only one computer course. So I transferred to Berkeley. And when I was working on my master's, around 1968, there was a job at LBL. It was great. There was Dennis Hall. And Joe Sventek was working downstairs, in an altogether different department. His degree was in physics, and he was doing something secret that had to do with weapons testing.... We were maintaining systems for all these researchers who didn't care at all about computing, just that it worked. But one day Andy Tanenbaum [of the Vrije Universiteit in Amsterdam] left a copy of this book, *Software Tools*, on the table in the Lab. He said, "You might be interested in this." So I read it. And I thought it was wonderful. At the same time that Andy Tanenbaum mentioned the tools book, Dennis Hall at the Lab discovered it. It was Dennis who dragged me into it (I had been working on another project at the time), and it was Dennis who was the PI, got the funding, directed us, to the extent that either Joe or I could be directed. So over the weekend I sat down and I started to implement all the tools. It was great! And Joe and Dennis and I thought, well, if we're supposed to be supporting research, then these tools support research, and we implemented all the rest. Oh, and some other stuff, too. Jim Poole at DOE was always extremely supportive and gave us funding. Nobody else ever got the point, and went on along with their ghastly FORTRAN programs, extended DCL scripts, and other sillinesses.

It was that fast.

Mike O'Dell told me, "Remember, Debbie knew Brian, and he knew what they were doing, so he pointed people who asked about the tools to LBL. And that started the user group." Recall, *Software Tools* wasn't about Unix, it was about philosophy and style. Ritchie said:

> The tool-using approach is powerful and intellectually economical, but it takes imagination to use. It may also be more

> costly to combine simpler, more general tools than to build a more specialized one.

Interestingly, Scherrer and her colleagues realized just how powerful the tool concept was. It was clear to them that the software tools could be used on just about any available architecture. So Hall, Scherrer and Sventek wrote a Virtual Operating System (VOS) that would serve as a pseudo-interface between the software tools (in Ratfor) and whatever OS was running. Scherrer said, modestly, "VOS was Dennis' idea."

I asked Paula Hawthorn, who had the distinction of being Mike O'Dell's manager at LBL, about the tools. She told me:

> When I joined the Computer Science and Math Department at LBL, Debbie and Joe and Dennis Hall already had completed several releases of the software tools, and were in the middle of the "but is it really research?" issues.
>
> The problem was that LBL was supposed to be doing research, and how could we say that making another release of the Software Tools really was research? "But is it really research?" is the bane of the engineer who wants to make the results of the research into something that people actually use, because many of the things you do to deliver a working system are not necessarily research, but the research is totally invalid if it has had no field trial. So my memories of that time are colored with the "but is it really research?" fight, and the "if you need to use Unix, why not just use Unix?" battle. To a first approximation, the Software Tools appeared to be just papering over non-Unix systems so that they were more Unix-like. This also caused heated discussions...

Communicating with a few others about the tools, a Software Tools User Group was founded in 1978. The article on VOS appeared in *CACM* in September 1980, but a STUG meeting had been held on June 16, preceding the USENIX meeting at the University of Delaware. Wally Wedel reported on the meeting in *Software Tools Communications* (#4, October 1980)—an aperiodic newsletter initially produced through LBL. The software tools were already available for several DEC

operating systems, for the Control Data Cyber, for IBM systems in a TSO environment, and on an SEL 32/77 under MPX 1.3 (MPX was a primitive operating system; NASA's Ames Research Center was unable to implement a shell under it, though they got every tool to work, etc.). By the November 1981 issue, Phil Scherrer (formerly of Unicorn Systems, now at Stanford University) was able to report that:

> The software tools have now been completely ported to a microcomputer environment. The CP/M (trademark of Digital Research) operating system which runs on 8080, Z80 (trademark of Zilog), and 8085 processors, was chosen because of its wide availability on systems with (barely) sufficient hardware. ...
>
> All the tools from the STUG distribution tape, as well as well as many of the extensions specified in the CACM article, have been brought up and run quite well.

The same issue of *Software Tools Communications* listed nearly three dozen architectures on which the tools had been implemented, together with the names and addresses of the implementors. The manufacturers ranged (alphabetically) from Burroughs to Zilog; the machines (in size) from the Z80/8080 (64k bytes of program memory) to the IBM 370 and the DEC 20. Geographically, the implementors spanned the globe: from Kawasaki, Japan, across the US and Canada, to the UK and Eindhoven in the Netherlands.

One of the sites where the Tools became important was Georgia Tech. Gene Spafford, who was there for much of the time (though not for the very beginning), told me:

> In the mid-to-late 1970s, the folks at Georgia Tech had several PDP-11 machines running Unix. These were used both for research and as part of a medical database research program.
>
> Several of the students and faculty got hooked on Unix and wanted to bring it up on the main research computers in the department, which were Prime machines. The Primes, at the time, were really nice machines. They had virtual memory, good multi-user capacity, and many other nice features. Unfortunately, the Prime architecture was not an easy machine to write an OS for. Unix was not going to map onto the machines.

As an added inducement, the last remaining PDP-11 got "melted down" by a DEC field engineer who jammed the power supply in upside down and sent 110 VAC into the backplane. DEC never made the situation right, so we were without Unix....

[The students then wrote a system, but] Georgia Tech wouldn't allow this system to be given away, and so it was licensed to universities and companies who wanted it enough to pay for it. At one time, many score places were running it. Prime even marketed it. After several years, three things came along to kill it:

- Some folks at Prime weren't pleased that customers liked the Software Tools interface better than Primos, so they stopped providing special assistance to the Software Tools team. Many of the "champions" inside Prime engineering left to form Apollo. The ones who remained built some Software Tools ideas into Primos and turned their backs on the development team at Tech.
- The Software Tools group wrote a very good C compiler and library for the system. They wanted to switch everything over to C and form an independent company to provide support. Unfortunately, the administration at Georgia Tech got petty over ownership rights to the compiler and associated code. The result was that the team, as a whole, quit and Software Tools was effectively left without further support.
- VAXes began to be shipped in quantity with BSD Unix available. People who had previously used Software Tools could now get real Unix and virtual memory for the same (or less) cost than their Primes with Software Tools. Demand rapidly fell off.

The principals involved were Dan Forsyth, Paul Manno, Perry Flinn, Allen Akin, and Win Strickland.

Spafford's tip led me to Manno and Forsyth. Forsyth, after huddling with his former colleagues for the sake of accuracy, told me:

> The Software Tools project at the Georgia Tech School of Information and Computer Science began as the result of several independent events in 1976. First was the publication of *Software Tools*. Second was the Army's interest in portable COBOL. Finally, and closer to home, was the plight of a group of seniors who returned from summer break to find that their beloved Burroughs B5500 had met its end for lack of money for maintenance and that all the PDP-11s had been walled off to protect confidential medical research data. In the place of their formerly idyllic computing environment, funded largely by the Army's research project, was a shiny new Prime 400.
>
> A UNIX port was out of the question for many reasons, including the lack of access to a PDP-11 for porting, the initial lack of documentation on the internals of the Prime, and a lack of faculty interest. Meanwhile the local programming environment consisted of a machine architecture that looked like a cross between a GE-645 and what was to become an Intel 80286, a FORTRAN-66 compiler, and a timeshared operating system that, at its best, could accept commands consisting of two six-character file names and ten octal numbers!
>
> One of the senior design projects suggested by Dr. Philip H. Enslow (PI for the Army's research project and soon-to-become faculty advisor to many of the students) was the porting of Ratfor to the Prime. This was clearly before any of us knew what an excellent job Brian [Kernighan] and Bill [Plauger] had done. The tape was obtained from Addison-Wesley and Ratfor was running before we knew it.
>
> Allen Akin, Perry Flinn, and Jack Waugh began to recognize how the tools could be put to use. At some point in early 1977, the Primos command line interface was deemed entirely unsuitable for the graceful connection of tools, so a simple UNIX-like shell was cobbled together. Paul Manno and I were drawn into the effort shortly thereafter.

Although the desire to create an elegant computing environment was strong, each of us was primarily commited to other projects. Nonetheless, we often found that by following K&P's advice to extend or add new tools, we could accomplish our project work more effectively than starting from scratch each time. With this incentive, the Software Tools Subsystem began to grow rapidly. Since it was a much more effective environment, other student and faculty use began to grow as well. We received many contributions from other researchers and in return found it necessary to establish documentation standards and write bundles of documentation to avoid losing all of our time to questions. Of course, we had to extend the text formatter and other tools to accomplish it.

By the summer of 1978, we had an effective replacement for both the Primos command and programming environments. In addition to the original Software Tools, we had a large subroutine library, basic shell, extended Ratfor, electronic mail, bulletin board, full screen editor, and many other new tools.

Through the efforts of Phil Enslow and David Nelson, head of research at Prime, the research division of Prime Computer also became interested in our efforts. With the assistance of Georgia Tech and Prime, Allen, Perry, and I made a visit to Prime Research that summer and came away with the inspiration to produce an "advanced" command interpreter. We set about combining features from Multics, V6 UNIX, RDOS, and other systems into a single command language. The examples set for us by Software Tools, Unix, and Algol 68 convinced us to strive for the goals of elegance, orthogonality, and reusability.

By the beginning of 1979, we had produced the "new shell" that, in retrospect, looked quite a bit like the externals of the Bourne shell (although we had never seen it). It included several interesting concepts: multiple standard inputs and outputs along with syntax to connect directed graphs of tools, arbitrarily long command lines, control structures managed by external commands, and scoped variables that were

treated as objects whose execution yielded their value on standard output.

Prime Research acquired a copy of the Subsystem, as it was fondly called, and made it available internally, since it was superior for text processing and research programming. It was perhaps during this time that we established a less-than-desirable relationship with the engineering group at Prime. It seems we had a knack for implementing and releasing features, especially those dealing with improving the operating system interface, at nearly the same time that Prime engineering declared the task to be impossible or impractical. Of course, we had the luxury of a small installed base and no commitment to backward compatibility. For whatever reason, our work had only an indirect influence on later releases of Primos.

With the acquisition of more Prime systems and a newfangled way of connecting terminals called "ethernet," the Software Tools Subsystem became the administrative link for the School of ICS. It handled local electronic mail as well as text processing for the department, and it provided many graduate students, including me, with submission-quality theses without paying a typist. Rather than release our efforts to the public, the powers-that-be at Georgia Tech decided that licensing for a fee was more appropriate. Even at the then-astronomical fee of $3,000, many dozens of copies were licensed to Prime sites around the world, including England, Germany, and Australia.

In 1980, Debbie Scherrer from Lawrence Berkeley Labs invited us to attend a Software Tools User's Group meeting as part of the USENIX meeting in Toronto. Georgia Tech assisted us in flying to Toronto and we found ourselves surrounded by many new ideas. These were greatly interesting meetings that exposed us to new perspectives that were unimagined in our insular, departmental environment. We were allowed to contribute some of our software to the user's group, but because of the licensing considerations, we could never release the "new shell" and the more interesting tools.

> By the time Allen, Perry, and I left in 1981, the Subsystem had a completely rewritten Ratfor preprocessor, a parser generator, a language-independent code-generator and a C compiler (in Ratfor!), as well as thousands of pages of documentation. The project was taken on by Jeanette Myers, Terry Countryman, Peter Wan, Scott Lee, and Arnold Robbins who managed it well. During that period, AT&T began offering low-cost Unix binary licenses. Within a year, Unix boxes could be purchased for one-fifth the price of a Prime or PDP-11. Before long, Unix, vi, and troff were available to the masses, and the commercial need for the Subsystem began to wane.
>
> Around Georgia Tech, 4.1BSD Unix VAXen, AT&T 3Bs, and Sun 3s had been sprouting like weeds, much to the consternation of the central computing czars. CSNet, NSFnet, and USENET made Unix cycles a necessity for participation in the growing Internet community. A final release of the Subsystem was made in 1985 and the code was put into the public domain. Shortly thereafter, it and the Primes were retired from local use, fittingly replaced by the software whose lack had spurred its creation.

The Software Tools had given rise to another grassroots movement. Another set of avid users, only partially overlapping those using Unix. But the philosophy was portable: Write programs that do one thing and do it well.

Many of the names involved in STUG are familiar to Unix users, too: Neil Groundwater, Mike O'Dell, the Scherrers, Joe Sventek, Dave Stoffel, and Wally Wedel.

I last mentioned O'Dell when he was still an undergraduate at the University of Oklahoma. He had now completed his master's thesis ("I looked at it again a while ago," he told me. "It's not terribly embarrassing.") and was job-hunting. In his words:

> I was within two weeks of taking a job with 1127 [the Research Group] at Murray Hill. I went off and interviewed. And then the death march started, where the paperwork went from desk to desk to desk, arriving on a person's desk just as

> they were leaving on a two-week vacation. So what should have been a two-week approval process ... turned into two months. So I went —in the meantime—to the [USENIX] January meeting in Boulder and went to the Software Tools session. I had been interested in the Software Tools stuff as a way to get interesting stuff onto the IBM machine, while we were still fighting [trying to get Oklahoma to purchase a DEC machine]. So I went to the Software Tools thing and Debbie Scherrer spoke and she talked about all the wonderful stuff they were doing and at the end of her talk she said, "Oh, and by the way, we're looking for someone to run our Unix systems for us." And I all but climbed over people, I remember literally throwing chairs out of the way as the meeting was breaking up, to get up there. There were six people from the Lab there—Joe Sventek, and Peter Krebs, Debbie, Dennis Hall, Roland Johnson was probably there. The interviewing procedure was go to lunch with six people and they get to eat Chinese while you talk—tag-team interview.
>
> Anyway, I went home and they flew me out and I gave a talk on my thesis and they made me an offer. So I had this awful decision to make between LBL and BTL. And BTL kept delaying and delaying and delaying. And finally LBL said they had to have a decision. So I called up Dennis [Ritchie] ... and he said I should take the other job.

So O'Dell went to LBL, where he became the Unix guru, the ARPANET liaison (the 11/70 was an early ARPANET host), and several other things. Neil Groundwater was also at the Boulder meeting. He, too, got involved with STUG:

> I became familiar with the STUG at the Boulder USENIX Conference in 1980. At about that time it seemed that LBL (Sventek, Scherrer, et al.) had done 90% of the tools and work, but others were beginning to "use the source" (tools).
>
> We were involved in consulting to the US Navy on several projects that had differing programming environments.

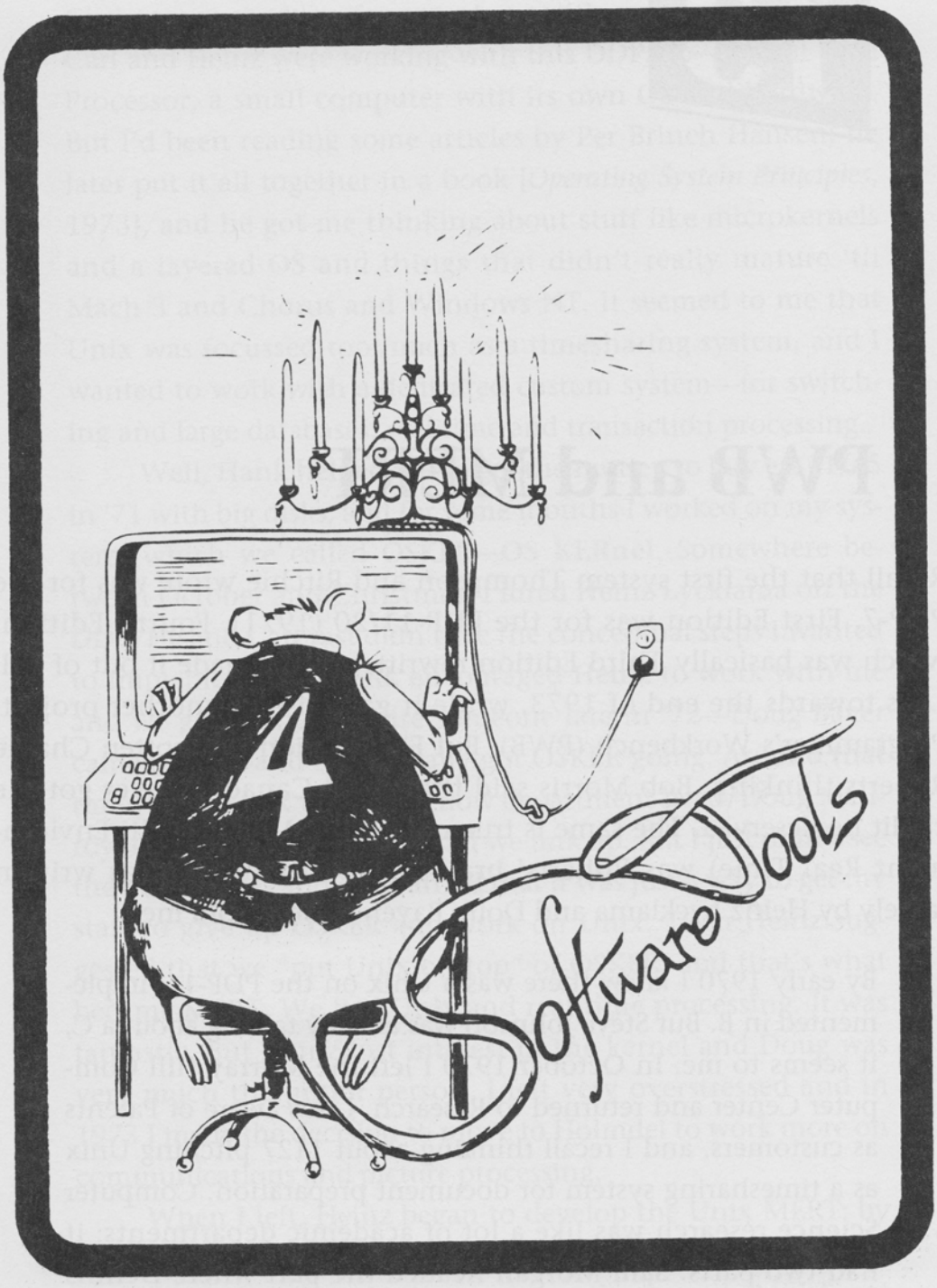
We suggested the Tools as a means for them to control their build environment on multiple platforms, but it turned out that they primarily used it on VAX/VMS.

My biggest contribution was a text control system (SCCS to Unix fans) that performed **create**, **delta**, **get**, and **edit** functions for version control of text files. It was based on the **diff** algorithm published as a Bell Labs Technical Memorandum by J. W. Hunt and Doug McIlroy.

STUG was formally set up at that Boulder meeting. Groundwater was elected to the first Board. And O'Dell moved to Berkeley.

Debbie and I shared an office for the three years I was at the Lab. She's the dearest person in the world, with more energy than any six people should be allowed to have. The Software Tools stuff sort of peaked, leveled off. And the sad part is that the Unix community has never really learned the lessons that the Software Tools people were teaching. The Software Tools folks, I would argue, understand portability to a degree that no one does. And the Unix community is much poorer for it. The problem with Unix is that it was so easy to write code, you saw no reason not to. This is not a version of salvation through suffering. A lot of things in the Software Tools stuff worked better because they really got it right, and 5,700 people didn't try to reinvent something.

The tools concept was flourishing beyond the realms of Unix.



The picture of Rat-4, drawn by George Kapus, is courtesy of Debbie Scherrer.

CHAPTER

25

Duelling Unixes

Up to 1978, Unix meant AT&T's operating system. It was only with the advent of PWB 2.0, V7, 32V, and 3BSD that confusion set in. But Unix was not an AT&T OS product. It was a "telecommunications support tool." AT&T, recall, felt that because of the consent decree it was constrained to stay out of the computer business. It wasn't permitted to compete in the computer marketplace. The future of Unix within AT&T was handed off from Research to USG—the UNIX Support Group, which controlled its future for most of the 1970s. Then, in 1979, Microsoft and the Santa Cruz Operation came out with XENIX2 and Berkeley with 4BSD, each a V7 derivative. XENIX was the first implementation of Unix for the Intel 8086 and many other architectures. It remains a very popular implementation, though it has become increasingly incompatible with others. Len Tower of the Free Software Foundation remarked to me, "Using SCO Unix is like travelling back in time."

As Andy Tannenbaum pointed out, "Within BTL there was a UNIX SUPPORT GROUP, but you never heard about the UNIX DEVELOPMENT GROUP. This is because no Bell System organization had the charter to develop computer operating systems..." In 1981, UniSoft (founded by Jeff Schriebman) brought out a port called UniPlus+, which has remained compatible with System III and now System V. In 1983 a group of Berkeley-ites formed mt Xinu (*Unix tm* backwards) to commercialize and support BSD. Among these were Bob Kridle, Alan

Tobey, Ed Gould, and Vance Vaughan. Debbie Scherrer soon joined them. They marketed first more/4.2BSD and then more/4.3BSD. (mt Xinu excelled in customer service: Debbie installed my 4.3 tape herself in 1986.) Also in 1983, USG and the PWB group were merged into the Unix System Development Lab.

Within five years, Apollo, DEC, Eakins, Gould, Integrated Solutions, Masscomp, mt Xinu, NSC, and Wollongong were among the companies marketing Berkeley Unix. Among those marketing AT&T System III or System V derivatives were: AT&T, Altos, Apollo, Compaq, Convergent, HP, Honeywell, IBM, ITT, Intel, Interactive, Masscomp, Microport, Microsoft, Motorola, NCR, NUXI, Opus, SCO, Silicon Graphics, Sperry, Sun, Tandy, UniSoft, and Wollongong. Furthermore, Amdahl, Apollo, Apple, Cray, DEC, Data General, HP, IBM, Intel, Motorola, Unisys and a host of others offer proprietary versions of Unix, several of which are 4.2BSD-based.

All of these, whether AT&T or BSD-derived, require licenses from AT&T. Recently, several versions of Unix that do not require such licensing have become available. Though none is a truly robust, commercial product, BSDI, 386/BSD, and NetBSD run on any 386/486 machine. [These 'license-free' versions, all derived from the free CSRG releases, were still under litigation when I wrote; this changed on 4 February 1994; see Chapter 29.] Linux, written by Linus Torvalds, also runs on 386/486 machines and uses no CSRG code. All four systems employ many of the programs that the Free Software Foundation (FSF), founded by Richard M. Stallman, has written for their near-finished GNU ("GNU's Not Unix") system. GNU programs include original, freely redistributable recreations of most Unix software (they are distributed under a scheme called "copyleft" by those involved with the GNU Project. This means that both the original and any improved versions must remain distributable and modifiable by all and that source code must be distributed, if binaries are). gcc, GNU's C compiler, is probably the most important successor to Steve Johnson's pcc.

Finally, various of these derivatives and clones run on a catalog of computer chips: DEC, Intel, MIPS, Motorola, NSC, to name a few.

The result of this thick sludge of alphabet soup has been confusion on the part of the users and prospective purchasers: what will work with what? Certainly, despite all the to-do, there are no "open

Three too many? Microsoft, if one can believe the trade press, had 600 programmers working on Windows NT. But there were three working at Holmdel, creating 32V (London, Reiser, and Swanson); three who wrote HoneyDanBer; Hall, Sventek and Scherrer wrote VOS; Lorinda Cherry and Bob Morris wrote bc and dc; and the CSRG at Berkeley rarely had more than a handful of full-time workers: Haley and Joy; Haley, Joy and Kridle; Joy, Kridle and Leffler; McKusick part-time and in the summers until he earned his Ph.D.; McKusick, Jolitz and Karels; McKusick, Karels and Bostic. Larry Wall wrote patch, perl, rn by himself. And look at those who just "dropped by": Ken Thompson, George Coulouris, various Australians, Jan-Simon Pendry. And look at the other contributors: Mike Muuss, Doug Kingston, Jim Curry, Rick Adams, James Gosling, Rob Pike, Armando Stettner, Bill Shannon, etc.

Steve Johnson told me:

> If I had been at a university, I would have been considered either a software person or a theory person. There would have been a pot of money that paid my salary. And if I were a software person and talked to theoreticians, the software people who were paying my salary would have felt cheated. And conversely, if I had been a theory person, and tried to do something useful.... I've really always been a generalist. I've operated typically in the cracks between different disciplines, where I found a lot of very fertile ground. So I will write a program and this will suggest some problems that I can prove theorems out of and turn about and put the theorems back in the programs. That happened with yacc and pcc and some things I've done since then. It just doesn't fit well into a compartmentalized structure.

Here's another point: avoid compartmentalizing researchers and developers.

Eric Allman remarked to me that:

> I think one general rule of software design is that you should be writing a program that you want to use. Ken and Dennis wanted to use Unix. They did what they needed in order to make it work. We wanted to use **sendmail**, it wasn't something where we said "Oh, let's write a mailer and send it

Groundwater, Neil first user of Unix outside of New Jersey; first user of INGRES outside of Berkeley; early supporter of STUG

Gurwitz, Bob BBN member of DARPA's steering committee; author of TCP/IP code

Hagen, Teus established first trans-Atlantic uucp connection

Haight, Dick major contributor to PWB; wrote find, cpio, etc.

Haley, Chuck collaborator with Joy on ex and Pascal shell; wrote tar

Hall, Dennis co-implementor of VOS

Hawthorn, Paula database activist; manager of O'Dell and Allman at different times

HCR Human Computing Resources, the first Canadian Unix company

Henry, Robert creator of error

Holmgren, Steve coauthor of first ARPANET code for Unix

HoneyDanBer most common upgrade of UUCP

Honeyman, Peter the Honey of HoneyDanBer

Hume, Andrew one of the editors of 10th Edition

Idris Plauger's Unix-like system

IEC International Electrotechnical Commission

IIASA International Institute for Applied Systems and Analysis, Laxenburg, Austria

IIJ commercial Japanese IP network

INGRES Interactive Graphics and Retrieval System; first Unix-based relational database

IP Internet Protocol

IPC interprocess communication

Ishida, Haruhisa first user of Unix in Japan

JAWs Just Another Workstation

Johnson, Steve wrote lint, yacc, spell, pcc; worked with Ritchie on Interdata port; fifth president of USENIX

Joy, Bill Unix enthusiast; created much of BSD, 2BSD, 3BSD, 4BSD; co-founder of Sun Microsystems; designed NFS

JUNET Japan Universities' Network

jus Japan Unix Society

Pmax DEC workstation

POSIX set of computer standards committees

Presotto, Dave wrote vgrind with Bill Joy; involved with Plan 9

PTT Post, Telephone, Telegraph

PWB Programmer's Workbench

QMC Queen Mary College, University of London; now Queen Mary and Wakefield College

Quarterman, John S. author of *The Matrix* and editor of *Matrix News*

Rashid, Rick responsible for Mach

Redman, Brian the Ber of HoneyDanBer

Reiser, John coauthor of 32V

RFC Request for Comment

RISC Reduced Instruction Set Computer

Ritchie, Dennis M. one of the originators of Unix; principal author of C

RJE Remote Job Entry

RK a family of DEC drives

Roberts, Charlie creator of MERT; director of 32V project

Saito, Nobuo one of the founders of jus

SCCS Source Code Control System

Scherrer, Debbie one of the implementors of VOS; founder of STUG; president of mt Xinu; equestrian extraordinaire; third president of USENIX

Scherrer, Phil founder of Unicorn Systems; early STUG booster

Schriebman, Jeff founder of UniSoft

Schulman, Bob installer of Unix on Japan's first VAX

SCO Santa Cruz Operation

Seeley, Donn worked on f77 and pcc as well as Net-2

SGI Silicon Graphics, Inc.

Shienbrood, Eric wrote more

SIG Special Interest Group

SMTP Simple Mail Transfer Protocol

SOSP Symposium on Operating System Principles

Spafford, Gene involved with Georgia Tech tools effort

SSEC Selective Sequence Electronic Calculator

Stallman, Richard M. chief GNUsance; responsible for emacs, GNU and FSF

Standiford, Keith installer of Unix at Berkeley in January 1974

Stettner, Armando got DEC to acknowledge Unix; instigator of OSF

Stettner, Heidi owner of Biff

STUG Software Tools User Group

Sventek, Joe one of the implementors of VOS; co-founder of STUG

SVID System V Interface Definition

Tanenbaum, Andy creator of MINIX; originator of Amoeba

Tague, Berkely secretary to the Multics triumvirate; founder of USG

TCP Transmission Control Protocol

TECO early MIT editor

TENEX BBN OS for the DEC-10

Tevanian, Avadis co-originator of Mach

Thompson, Ken originator of Unix; implementor of many things; creator of Belle, sometime computer chess champion

Tilbrook, David originator of NEWSWHOLE, founder of HCR; program chair of first EUUG conference

Tilson, Michael president of HCR; now VP of SCO

Tobey, Alan co-founder of mt Xinu

Torvalds, Linus creator of Linux

Tower, Len Associate GNUsance; finder of vegetarian restaurants

Trickey, Howard part of Plan 9 team

Truscott, Tom co-originator of USENET

TWENEX BBN follow-up OS for the DEC-20

Ubell, Mike wrote history prototype

UEG DEC's Unix Engineering Group

UKUUG United Kingdom Unix Systems User Group

Ultrix DEC's version of 4.2BSD

UNICOM the 1983 joint STUG, USENIX and /usr/group conference

UNICS original name of Unix

UniForum current name of /usr/group

UniPlus UniSoft's port of Unix

USENET over 6000 examples of chaos theory