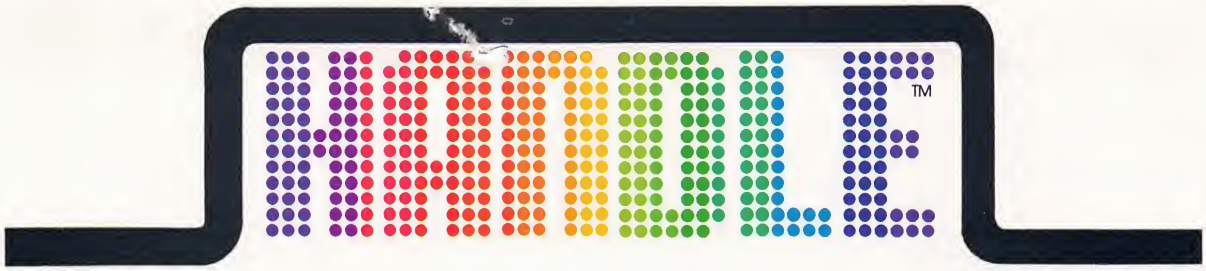
UNIX™ REVIEW

THE PUBLICATION FOR THE UNIX™ COMMUNITY

DECEMBER 1985 $3.95

INTERNATIONAL UNIX



Modular.
Integrated.
Now.

Handle Writer/Spell™
Word processing with integrated spelling correction and verification.

Handle Calc™
Spreadsheet with up to 32,000 rows and columns. Conditional and iterative recalculation.

The **Handle Office-Automation Series** is a powerful set of modular, integrated software tools developed for today's multiuser office environment. **Handle** application modules can be used stand-alone or combined into a fully integrated system.

The **Handle Office-Automation Series** modules offer:

- Ease of Use and Learning
- Insulation from **UNIX**
- Data Sharing Between Multiple Users
- Data Integration Between Modules
- Data Sharing with Other Software Products
- Sophisticated Document Security System

Handle Technologies, Inc.

Corporate Office
6300 Richmond
3rd Floor
Houston, TX 77057
(713) 266-1415

Sales and Product Information
850 North Lake Tahoe Blvd.
P.O. Box 1913
Tahoe City, CA 95730
(916) 583-7283

TM—HANDLE, HANDLE HOST, HANDLE WRITER, HANDLE SPELL, HANDLE WRITER/SPELL, and HANDLE CALC ARE TRADEMARKS OF HANDLE TECHNOLOGIES, INC.
TM—UNIX IS A TRADEMARK OF AT&T BELL LABORATORIES.

How to go from UNIX to DOS without compromising your standards.

It's easy. Just get an industry standard file access method that works on both.

C-ISAM™ from RDS.

It's been the UNIX™ standard for years (used in more UNIX languages and programs than any other access method), and it's fast becoming the standard for DOS.

Why?

Because of the way it works. Its B+ Tree indexing structure offers unlimited indexes. There's also automatic or manual record locking and optional transaction audit trails. Plus index compression to save disk space and cut access times.

How can we be so sure C-ISAM works so well? We use it ourselves. It's a part of INFORMIX®, INFORMIX-SQL and File-it!™, our best selling database management programs.

For an information packet, call (415) 322-4100. Or write RDS, 4100 Bohannon Drive, Menlo Park, CA 94025.

You'll see why anything less than C-ISAM is just a compromise.

RDS™

RELATIONAL DATABASE SYSTEMS, INC.

© 1985, Relational Database Systems, Inc. UNIX is a trademark of AT&T
INFORMIX is a registered trademark and RDS, C-ISAM and File-It! are trademarks of Relational Database Systems, Inc.

How we improved Structured Query Language.

Actually, we didn't change a thing.

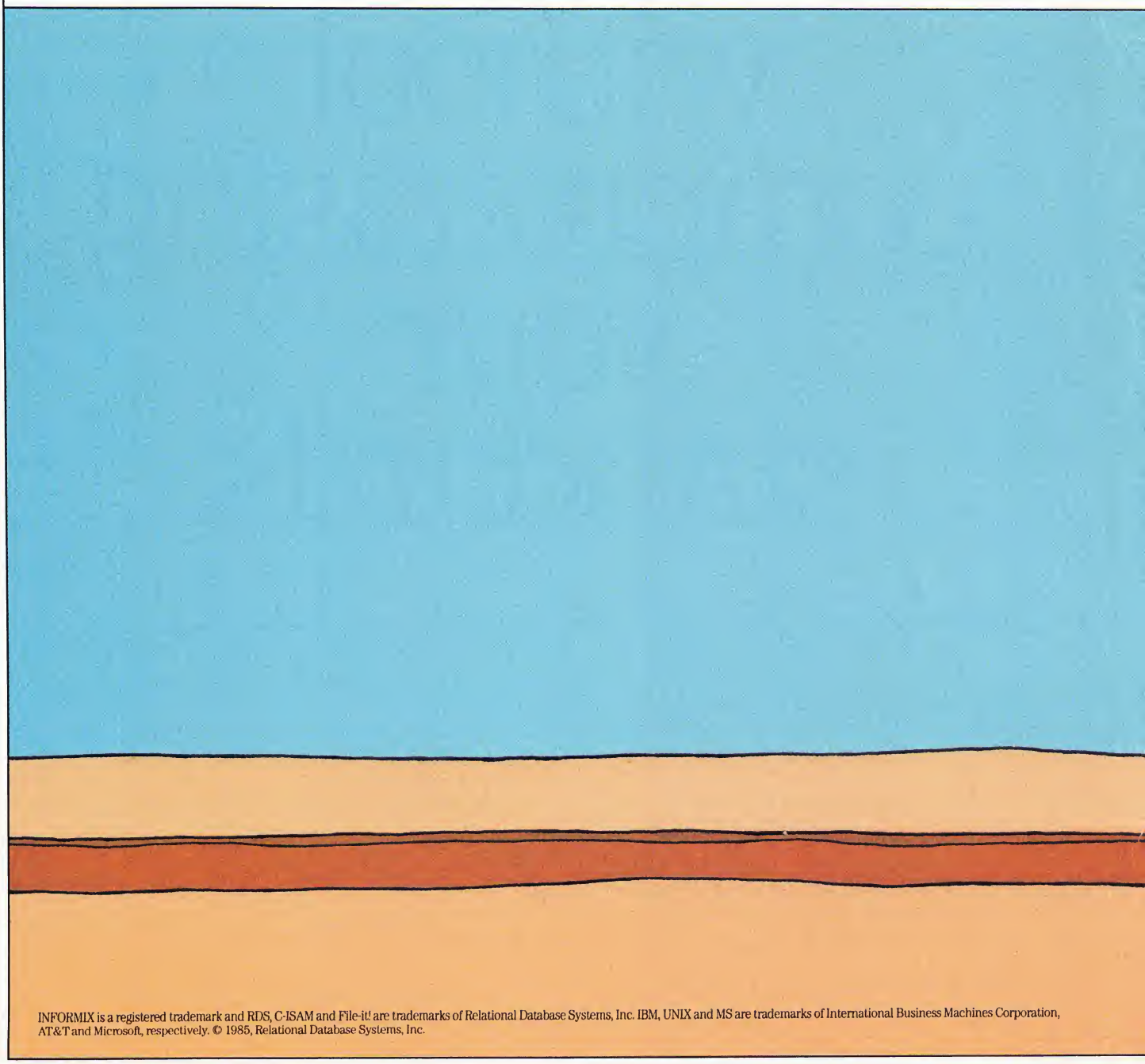
We just combined it with the best relational database management system.

Introducing INFORMIX®-SQL.

It runs on either MS™-DOS or UNIX™ operating systems. And now with IBM's SQL as part of the program, you can ask more of your database. Using the emerging industry-standard query language.

To make your job easier, INFORMIX-SQL comes with the most complete set of application building tools. Including a full report writer and screen generator. Plus a family of companion products that all work together.

Like our embedded SQLs for C and COBOL. So you can easily link your programs with ours. File-it!,™ our easy-to-use file manager. And C-ISAM,™ the de facto



INFORMIX is a registered trademark and RDS, C-ISAM and File-it! are trademarks of Relational Database Systems, Inc. IBM, UNIX and MS are trademarks of International Business Machines Corporation, AT&T and Microsoft, respectively. © 1985, Relational Database Systems, Inc.

standard ISAM for the UNIX operating system. It's built into all our products, but you can buy it separately.

And when you choose RDS, you'll be in the company of some other good companies. Computer manufacturers including AT&T, Northern Telecom, Altos and over 60 others. And major corporations like Anheuser Busch and The First National Bank of Chicago.

Which makes sense. After all, only RDS offers a family of products that work so well together. As well as with so many industry standards.

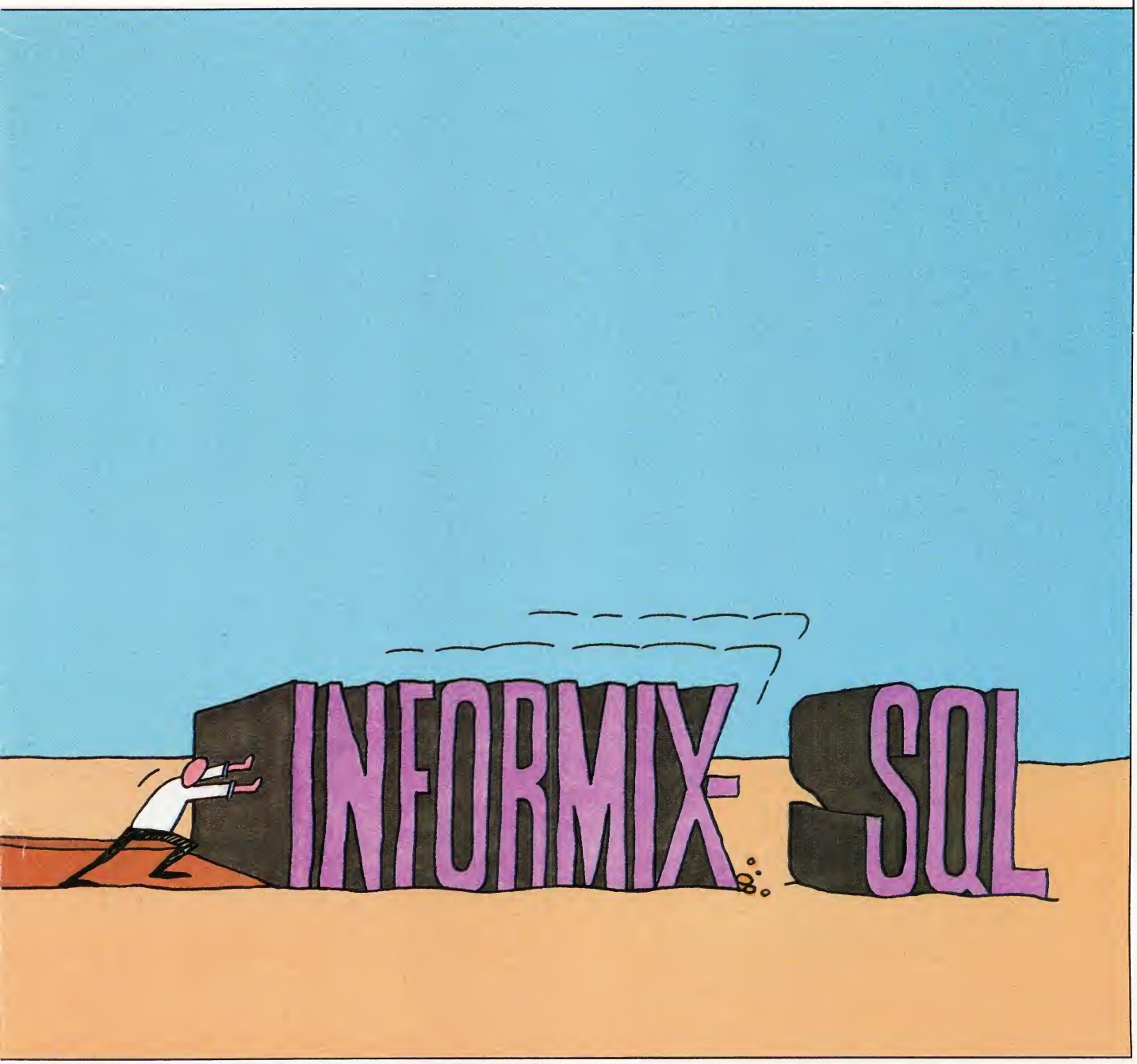
So call us for a demo, a manual and a copy of our Independent Software Vendor Catalog. Software vendors be sure to ask about our new "Hooks" software integration program. Our number: 415/322-4100.

Or write RDS, 4100 Bohannon Drive, Menlo Park, CA 94025.

And we'll show you how we took a good idea and made it better.

RDS™

RELATIONAL DATABASE SYSTEMS, INC.



Circle No. 41 on Inquiry Card

UNIX™ REVIEW
THE PUBLICATION FOR THE UNIX COMMUNITY

Volume 3,
Number 12
December 1985

DEPARTMENTS:

6 **Viewpoint**

8 **The Monthly Report**
By David Chandler

18 **The Human Factor**
By Richard Morin

66 **C Advisor**
By Bill Tuthill

74 **Rules of the Game**
By Glenn Groenewold

78 **Devil's Advocate**
By Stan Kelly-Bootle

80 **The UNIX Glossary**
By Steve Rosenthal

84 **Fit to Print**
By August Mohr

90 **Recent Releases**

102 **Calendar**

104 **The Last Word**

108 **Advertisers' Index**

Cover illustration by Hyon Kim

FEATURES:

22 **MADE IN THE USA**

By Mike Banahan

When UNIX travels, people immediately spot it as an American. Could this be why its reception often has been cool?

31 **TAKING A GLOBAL VIEW**

By Brian Boyle

Considerations of how UNIX lends itself to internationalization and what Americans stand to gain.

INTERNATIONAL UNIX

36 CHANGING CHARACTER

By Karen Barnes and Dan Epstein

Two engineers already at work on internationalization discuss some of the technical problems with the multilingual support of UNIX.

مؤسّسة الراجحي المَالية

44 BUREAUCRATIC BORDER SKIRMISHES

By Teus Hagen

Not the least of the hurdles facing UNIX users in Europe is the issue of international communications. The Director of the European UNIX systems User Group tells of the obstructions raised by bureaucracy.

50 FACING UP TO INTERNATIONALIZATION

By G.L. Lindgren

Nobody said it was going to be easy, but AT&T has some general guidelines for the efforts ahead.

58 INTERVIEW WITH JIM BELL

By Jeff Schriebman

The man in charge of Hewlett Packard's internationalization efforts provides an overview of the challenges ahead.

UNIX REVIEW (ISSN-0742-3136) is published monthly by REVIEW Publications Co. It is a publication dedicated exclusively to the needs of the UNIX community. Second class postage paid at Renton, WA 98055 and at additional mailing offices. POSTMASTER: Please send Form 3579 to UNIX REVIEW, 500 Howard Street, San Francisco, CA 94105.

Entire contents copyright 1985. All rights reserved and nothing may be reproduced in whole or in part without prior written permission from UNIX REVIEW.

Subscriptions to UNIX REVIEW are available at the following annual rates (12 issues): US$28 in the US; US$35 in Canada; US$48 in all other countries/surface mail; US$85 in all other countries/air mail. Correspondence regarding editorial (press releases, product announcements) and circulation (subscriptions, fulfillment, change of address) should be sent to 500 Howard Street, San Francisco, CA 94105. Telephone 415/397-1881. Correspondence regarding dealer sales should be sent to 901 South 3rd Street, Renton, WA 98055. Telephone 206/271-9605.

Letters to UNIX REVIEW or its editors become the property of the magazine and are assumed intended for publication and may so be used. They should include the writer's full name, address and home telephone number. Letters may be edited for the purpose of clarity or space. Opinions expressed by the authors are not necessarily those of UNIX REVIEW.

UNIX is a trademark of AT&T Bell Laboratories, Inc. UNIX REVIEW is not affiliated with AT&T Bell Laboratories.

PUBLISHER:
Pamela J. McKee
ASSOCIATE PUBLISHERS:
Ken Roberts, Scott Robin
EDITORIAL DIRECTOR:
Stephen J. Schneiderman
EDITOR:
Mark Compton
ASSOCIATE EDITOR:
David Chandler
EDITORIAL ADVISOR:
Dr. Stephen R. Bourne, Consulting Software Engineer, Digital Equipment Corporation
EDITORIAL REVIEW BOARD:
Dr. Greg Chesson, Chief Scientist, Silicon Graphics, Inc.
Larry Crume, President/Managing Director, AT&T UNIX Pacific Co., Ltd.
Ted Dolotta, Senior Vice President of Technology, Interactive Systems Corporation
Ian Johnstone, Project Manager, Operating Software, Sequent Computer Systems
Bob Marsh, Chairman, Plexus Computers
John Mashey, Manager, Operating Systems, MIPS Computer Systems
Robert Mitze, Department Head, UNIX Computing System Development, AT&T Bell Labs
Deborah Scherrer, Computer Scientist, Mt. Xinu
Jeff Schriebman, President, UniSoft Systems
Rob Warnock, Consultant
Otis Wilson, Manager, Software Sales and Marketing, AT&T Information Systems
HARDWARE REVIEW BOARD:
Gene Dronek, Director of Software, Aim Technology
Doug Merritt, Consultant
Richard Morin, Consultant, Canta Forda Computer Laboratory
Mark G. Sobell, Consultant
SOFTWARE REVIEW BOARD:
Eric Allman, Principal Systems Engineer, Britton Lee, Inc.
Ken Arnold, Consultant, UC Berkeley
Jordan Mattson, Programmer, UC Santa Cruz
Dr. Kirk McKusick, Research Computer Scientist, UC Berkeley
Doug Merritt, Consultant
Mark G. Sobell, Consultant
CONTRIBUTING EDITOR:
Ned Peirce, Systems Analyst, AT&T Information Systems
PRODUCTION DIRECTOR:
Nancy Jorgensen
PRODUCTION STAFF:
Cynthia Grant, Tamara V. Heimarck, Florence O'Brien, Denise Wertzler
BUSINESS MANAGER:
Ron King
CIRCULATION DIRECTOR:
Wini D. Ragus
CIRCULATION MANAGER:
Michael R. Newton
MARKETING MANAGER:
Donald A. Pazour
OFFICE MANAGER:
Tracey J. McKee
TRAFFIC:
Tom Burrill, Dan McKee, Corey Nelson
NATIONAL SALES OFFICES:
500 Howard St.
San Francisco, CA 94105
(415) 397-1881
Regional Sales Manager:
Colleen M. Y. Rodgers
Sales/Marketing Assistant:
Anmarie Achacoso

370 Lexington Ave.
New York, NY 10017
(212) 683-9294
Regional Sales Manager:
Katie A. McGoldrick

BPA

VIEWPOINT

Sometimes a great notion

Although it may not always be apparent, the themes UNIX REVIEW adopts for coverage are selected on the basis of news judgment. To qualify, a topic must be timely and have some bearing on a large number of readers. Certainly, the internationalization of UNIX meets this test. But, more than mere news, internationalization also is a cause.

Some of you may remember what a cause is, but most of us have lost the concept somewhere along our weary trek through the arid '80s. Be that as it may, the symptoms are unmistakable. Unlike "objectives", causes take hold of the emotions and demand that you *believe*. The intellectual and emotional stimulation of belief itself is at least part of the payoff.

But why believe in UNIX internationalization? Let's start with the good of humankind. Overblown rhetoric? Perhaps, but bear with me briefly as I step through the logic.

UNIX, as you know, has its charms. Of these, one of the greatest is a design that engenders sharing. By definition, of course, UNIX is a time-sharing, multiuser system. But within a larger context, "sharing" also refers to the portability and communications capabilities that have attracted—and helped to shape—a number of communities.

Among these, the academic community of UNIX users is a prime example. As John Stoneback noted in the October issue of UNIX REVIEW, "The UNIX community has made inexpensive electronic communications available to all of its members via Usenet. A community that already had so much in common was strengthened and enhanced by the ability to move software easily among locations." *Moving* the software, of course, is different from actually being able to *use* it at a new site. The fact that UNIX has made this possible owes to the portability of the system's applications. Note that even though much of the software sent via Usenet has its roots in non-UNIX environments, academic users have been able to comb it into UNIX painlessly because of the flexibility of the system. This effectively has expedited research by obviating the need to "reinvent the wheel". What's more, a large number of small "teaching" institutions incapable of funding their own research have been able to benefit from work done at larger universities.

It doesn't take a great leap of imagination to see how this exchange between universities might be paralleled by an exchange between nations. Some connections to Europe, Australia, and Japan already have been established, but these links could be greatly expanded. More significantly, UNIX could become the means by which lifelines of information are provided for the many underdeveloped areas of the world. Without modifications to the system, though, this will be impossible. The charter, then, for efforts to internationalize the system must be the elimination of language and cultural barriers. Some of the problems that these efforts will encounter—and a number of thoughts on the strategies to be employed in addressing them—are the substance of this issue.

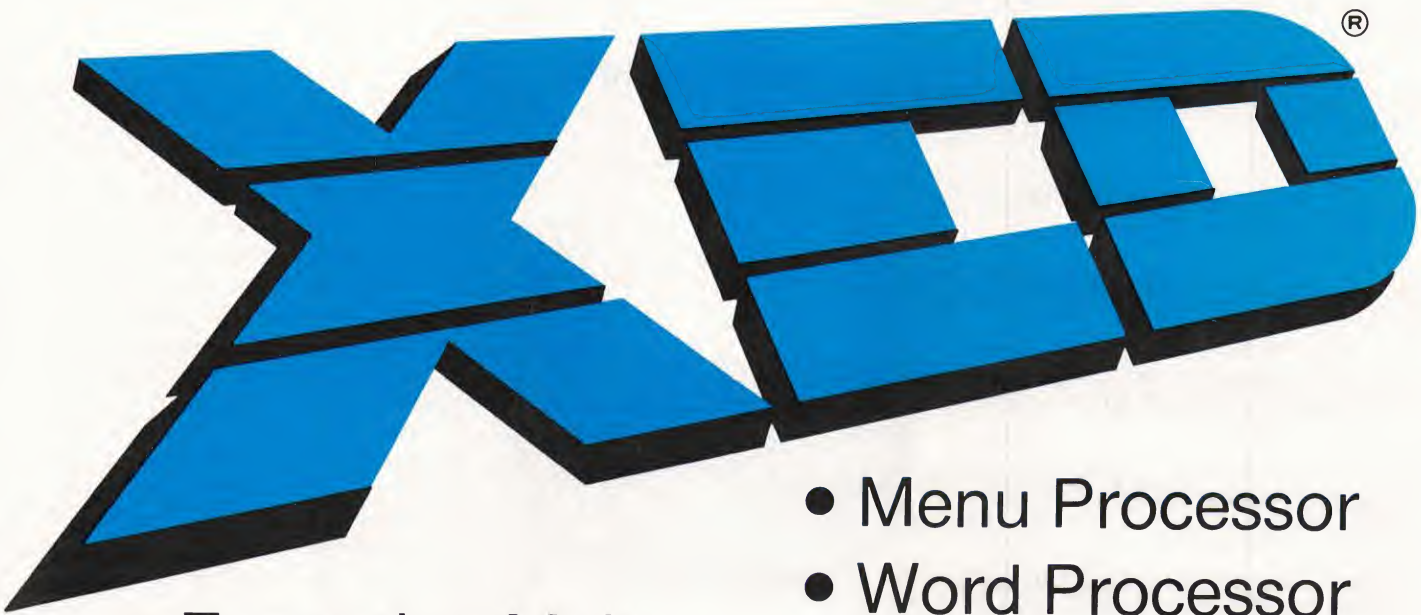
"Well and good," you say, "but I'm in business to make money." Fine. So forget all the bleeding-heart arguments and think strictly in terms of dollars and cents. UNIX is already a major presence in Europe and Japan; rumblings also have been heard in much of Asia and the Middle East. Many of the people in these areas have money; all of them have brains. Why not tap in?

Surely it can't be more difficult to cultivate essentially virgin markets than to carve out new niches in long-standing ones. As an added benefit, branching out would provide for enhanced intellectual dialog. The university community already has shown how vigorous electronic exchange can propel the state of the art.

The challenges of internationalizing the system, though, promise to be formidable. The close cooperation of many organizations and industries will be necessary if acceptable standards are to be developed. This, no doubt, will require much money and time, but the stakes are too high not to try.

Mark Compton

The First Name In Integrated Office Automation Software



- Executive Mail
- Telephone Directory
- Menu Processor
- Word Processor
- Forms/Data Base
- Spreadsheet

Certified and Deliverable Since 1981

XED was the first independent software company to introduce a Unix WP package and achieved early success by selling to the government and international market (XED is the only Unix WP package to meet government specifications). Worldwide sales of XED rank Computer Methods first in both sales and units installed in 1984.

XED® INTEGRATED OFFICE SOFTWARE

Box 3938 • Chatsworth, CA 91313 U.S.A. • (818) 884-2000
FAX (818) 884-3870 • Intl. TLX 292 662 XED UR

XED is a registered trademark of CCL Datentechnik AG
UNIX is a trademark of AT & T Bell Laboratories, Inc.

Circle No. 52 on Inquiry Card

THE MONTHLY REPORT

Pause to reflect

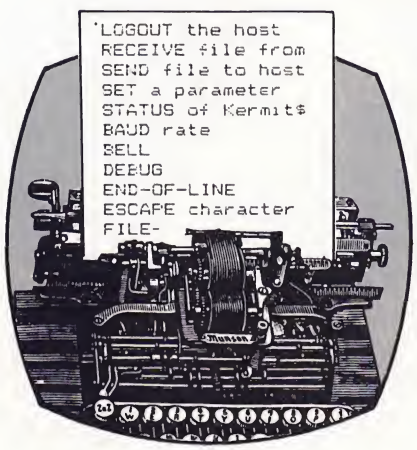
by David Chandler

In all labor there is profit, but mere talk leads only to poverty.
Proverbs 14:23 (NASV)

Plexus Computers Chairman Robert Marsh, in his closing remarks at last October's UNIX Market Forum in Los Angeles, reflected on what the commercial development of UNIX has brought us. As the man responsible for negotiating the first UNIX distribution license, founding the first UNIX users group, and developing the first commercial UNIX-based supermicro, Marsh spoke with knowledge—and good humor—on the topic.

Among other observations, Marsh noted that, from 1975 to '79, UNIX was in commercial "gestation". AT&T, he claimed, was "asleep at the wheel" during this period. But that changed during the first half of the current decade. These past five years have seen UNIX go through infancy, teen age, and young adulthood; commercially speaking, it would seem that the system already has reached middle age. Somewhere between the system's infancy and middle age, AT&T did, in fact, wake up, find the helm, and start steering the UNIX ship (or leading the fleet).

In approaching year's end, it is now appropriate to reflect on where this has brought us during



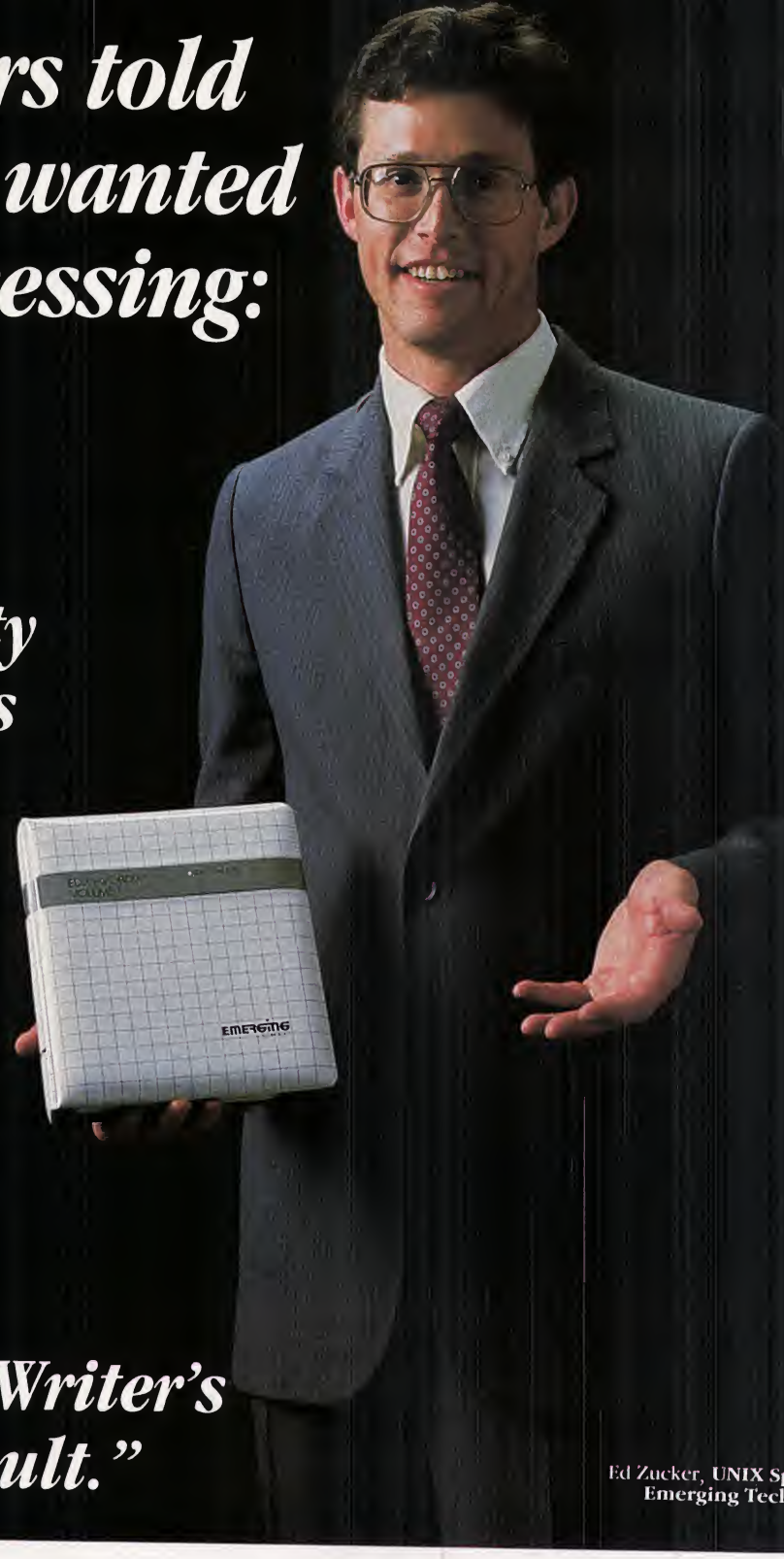
1985. Much as Marsh exhorted his audience to use history in shaping a perspective on the present, it can be helpful to consider the course charted this year by AT&T in determining what progress has been made in the UNIX industry as a whole.

To say that AT&T is the guiding force in the UNIX market is not to say that this pleases everyone. The Tel and Tel giant historically has had its share of critics and competitors. Many of these competitors, of course, feel they could do a better job of guiding the industry—and would love to have the chance. But the bearer of the UNIX trademark can no longer be criticized for lack of effort: AT&T is working—and working hard—to chart a definite course.

Consider the major moves of the past year: the publication of the *System V Interface Definition* (SVID), and all the events related to its announcement, including the signing of a contract with Microsoft to make Xenix (the most widely installed of the micro operating systems derived from UNIX) compatible with System V, and the announcement of contracts with Intel, Motorola, and National Semiconductor to port System V to those companies' microprocessors; the hardware and software announcements of last March, including the unveiling of the UNIX PC 7300, enhancements to the PC 6300, and the introduction of the Starlan network; X/OPEN's May announcement that it had adopted System V as its standard; the June announcements of some 70 hardware and software products, including the 3B2/400 and the 3B15, communications processors for connecting 3Bs to mainframes, and System V-VM; the disclosure of plans with UniSoft Systems to develop a Japanese language capability for System V; the elimination of 24,000 jobs from AT&T Information Systems, a major effort to reduce costs; September's unveiling of several proprietary and third-party software applications programs, adding to the more than 500 System V-compatible packages already created or adopted by AT&T (the company expects this number to nearly triple by mid-1986);

"Our customers told us what they wanted in word processing:

Compatibility
Flexibility
Ease of Learning
Vendor Reliability
Cost Effectiveness



Ed Zucker, UNIX Specialist
Emerging Technology

We listened. The Professional Writer's Package is the result."

Compatibility. A word processor for all machines, UNIX or MS-DOS™.

Flexibility. A word processor for every use, whether it's writing complicated manuals or preparing simple letters.

Ease of Learning. Time and money shouldn't be wasted in training. On-line help, on-line tutorials with step-by-step instructions.

Vendor Reliability. Extensive word processing experience, a large installed base, accessible technical support, and a liberal update policy.

Cost Effectiveness. With these capabilities you can't afford not to have the Professional Writer's Package.™

EMERGING TECHNOLOGY

Call or write for information:
4760 Walnut Street, Boulder, Colorado 80301 800/782-4896 303/447-9495

Circle No. 39 on Inquiry Card

UNIX is a trademark of Bell Laboratories. MS-DOS is a trademark of Microsoft Corporation. Professional Writer's Package is a trademark of Emerging Technology Consultants, Inc.

UNIX ™ is a trademark of AT&T Bell Laboratories
IBM® is a registered trademark of International Business Machines Corporation

UNIFY. JUDGED NO. 1 BY COMPUTER OEMS.

One UNIX computer manufacturer after another has come to the same decision: UNIFY is the fastest, most powerful, most flexible data base management system for users of all skill levels.

By their own investigation and by system integrator requests, computer manufacturers representing some 90% of the market choose to offer UNIFY with their UNIX computers.

They include IBM. AT&T. DEC. Data General. Honeywell. Four Phase. Sun. Perkin-Elmer. NCR. Tandy. Pixel. Cadmus. Plexus. Altos. Gould Computer. NBI. CCI. Sequent. And many more.

The evidence is overwhelming. In independent benchmarks, UNIFY consistently ranks as the top performer.

Completely menu-driven design and industry standard IBM SQL query language make it easy for non-programmers to develop data base applications.

The most powerful "back end" design in the industry, including 90 subroutines at the host language interface level, promises that UNIFY can keep adding features, keep adding users, without eroding performance.

Judge for yourself. Send for our demo kit—disks, tutorial and reference manuals, all for only $150—that shows you how to build virtually any application. Contact UNIFY, 4000 Kruse Way Place, Lake Oswego, OR 97034, 503/635-6265. TELEX 469220.

UNIFY

THE PREFERRED DBMS.

Circle No. 62 on Inquiry Card

and the September deal struck between AT&T and Sun Microsystems whereby Sun OS will be made compatible with System V and AT&T will blend certain 4.2BSD facilities into the SVID.

The month of October brought further announcements, including that of the PC 6300 Plus (see the following section for details) and the disclosure of an agreement with VMark Computer to convert applications software developed for the Pick operating system to UNIX.

VMark has an online DBMS called UniVerse that contains about 2500 Pick-based business applications programs. Roughly 20 of VMark's 300 distributors already have begun converting these packages. The rate of conversion varies with each program, of course, but it can be expected that, over time, the Pick packages will contribute significantly to AT&T's *Computer Software Guide*.

Though these many announcements are evidence of activity, they don't necessarily add up to everything that the UNIX community, the computer industry at large, or the general business community had hoped for. Neither can it be said that all of these groups are necessarily pleased with the speed at which developments have surfaced. The ship, though, is clearly out of port and heading on some specific course. We'll all continue to monitor the ship's bearing and status.

THE ICING, NOT THE CAKE

The October announcements by AT&T included 26 new software packages for the UNIX PC 7300; the introduction of the UNIX PC Model 3B1; the unveiling of the 3B2/310; and—spotlight, please—the announcement of the PC 6300 Plus.

Powered by an Intel 80286 microprocessor running at 6 MHz—a 16-bit data bus/16-bit CPU architecture—with 512KB of main memory (expandable to 7 MB), AT&T's latest PC comes in several configurations. On the high end can be included a 20 MB hard disk and a 12-inch color screen with 640 × 400 pixels.

For the time being, the machine runs only under MS-DOS 3.1. Sometime in the coming four months, however, 6300 Plus customers will be able to spend $395 to purchase the *OS Merge* package, a hardware/software combination that will allow concurrent use of UNIX and MS-DOS.

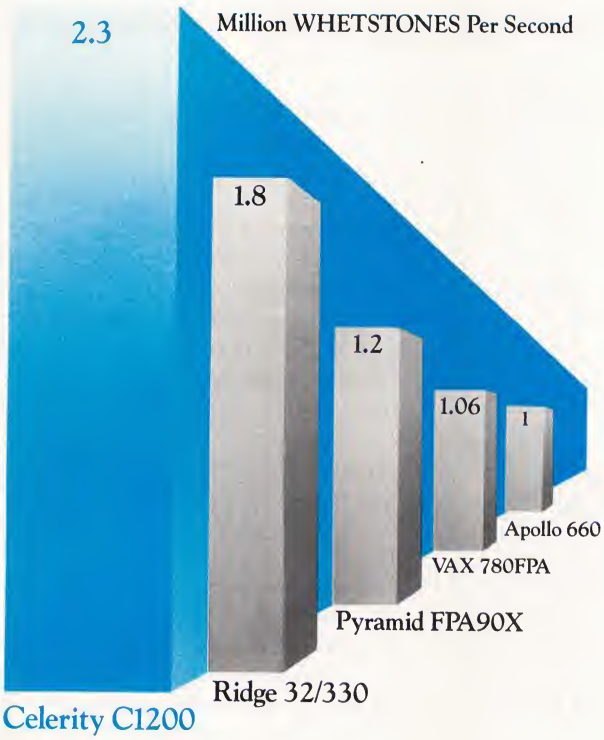
To say that AT&T is the guiding force in the UNIX market is not to say that this pleases everyone.

OS Merge—an AT&T designation—was developed by Locus Computing Corp., and is being made available to other hardware manufacturers and OEMs under the name *Multisystem Merge*. Working specifically on an 80286 computer, the package formats the hard disk and performs other system functions under UNIX. When a user enters a command at the system prompt, Multisystem Merge checks the command at a very low system level to determine if the command is in a UNIX or DOS format. If the command is a valid UNIX directive, it is executed just as it would be on any other UNIX system; if a DOS command is given, though, the package causes UNIX to allocate low-order memory for the execution of the DOS program. The processor then converts to 8086 compatibility mode. To allow for the concurrent operation of UNIX and DOS programs, the CPU is time-sliced.

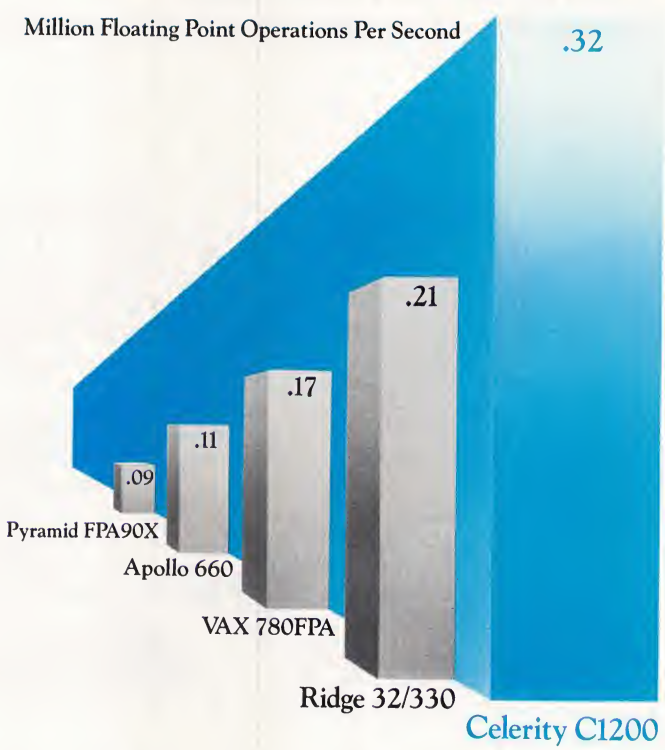
Since DOS is inherently single-user, only one DOS application can be run at a time (the DOS program runs as a process under UNIX). UNIX maintains its multiuser and multitasking capabilities. There is only one file structure—controlled by UNIX—for both operating systems, hence no file transfer utilities are required. When the user is in MS-DOS mode, the DOS interface gives the appearance of a DOS file structure; the user then can press a key to enter UNIX mode, and the file structure appears in UNIX format. Windows on the 6300 Plus allow the user to see and interact with both operating systems separately, concurrently, or cooperatively.

For all the convenience this offers the user, AT&T is marketing the 6300 Plus with the idea that OS Merge is a bonus—and not the focus—of the machine. Lawrence Dooling, vice president for product management and market development in the Small Business Systems arm of AT&T, called the 6300 Plus "fundamentally a DOS machine for use in a DOS environment". Unlike the developments listed at the beginning of this column, the purpose of the 6300 Plus is not to promote UNIX per se; it can be considered, nonetheless, as another step toward making more business users aware of UNIX. AT&T is offering the 6300 Plus as head-to-head competition for the IBM PC-AT, at a comparable cost ($6320 for the hard disk unit). For this price, the 6300 Plus offers what AT&T considers to be better graphics and stronger performance that—with no memory wait states—AT&T claims is 25 percent faster than that of the AT.

Too Good To Be True?



WHETSTONE Benchmark
Single Precision



LINPACK Benchmark
Double Precision

Call Us On It.

The C1200 computational system continues to set new performance standards. Results from industry-accepted benchmarks highlight the C1200's performance when executing workloads characteristic of compute-intensive engineering and scientific applications. Similar performance results are achieved in a broad range of application environments: modeling, simulation, analysis, image processing.

The C1200 combines mainframe performance and features with unmatched local control to provide you the best price/performance value available today.

- Optimized native UNIX 4.2BSD
- 32-bit RISC-like architecture
- Up to 24 MBytes physical memory
- 4 Gigabytes virtual memory
- Multiple high speed buses
- Industry-standard graphics, compilers, networking and communications options
- Up to 32 users

The proof of performance is in the execution of your application. We promise performance. We deliver performance.

So Call Us On It.

CELERITY COMPUTING

Corporate Headquarters: 9692 Via Excelencia, San Diego, CA 92126 (619) 271-9940

Circle No. 51 on Inquiry Card

UNIX is a registered Trademark of AT&T Bell Laboratories. VAX is a registered Trademark of Digital Equipment Corporation.

The AT&T PC 6300 Plus, capable of running both MS-DOS and UNIX.

"And, as icing on top of the cake", said AT&T spokesperson Craig Lowder, "with the $395 option to run System V applications, you can do lots of things at once. . .and switch back and forth [between operating systems] at the touch of a key."

An option though it may be, OS Merge can be significant for personal computing. The idea of switching back and forth between operating systems is not new; Xerox has a product that switches between DOS and its proprietary system, and various other products have permitted some interaction between DOS and UNIX—including the Connector from Uniform Software Systems (the first product to run DOS as a task under UNIX). OS Merge, however, is the first product to offer the integrated file system feature, and it is also the first to allow the change of environments at the touch of a key.

"The biggest problem facing the UNIX market", claimed Dr. Gerald Popek, president of Locus, "is that it has one-tenth the amount of off-the-shelf software applications that DOS has. The biggest problem facing the DOS system is it has limited power despite lots of software." Blending the strengths of the two systems, then, seems like a great gain.

Using both operating systems, of course, implies *knowledge* of how to use both. This then raises the question of whether there is a market of users educated in the use of both systems, or one that is willing to become so educated. There may well be a sizable number of users fluent in both DOS and UNIX. But even failing that, AT&T points out that the business user already familiar with MS-DOS can use the 6300 Plus for all MS-DOS needs, and then, with only minimal training in the more business-oriented features of UNIX (electronic mail, word processing, and the like) learn to be a multitasking user—a *UNIX* multitasking user at that.

Potential customers for the OS Merge-enhanced 6300 Plus have another point to consider, one raised by Peter Wensberg, president of Uniform. "There are a number of situations, notably in government agencies, where the multiuser feature is quite important, and where UNIX makes sense for a number of other reasons. . . . There are many government agencies implementing across-the-board moves to UNIX systems. One strong reason for this change from DOS to UNIX is that the strategic decision has come down from the top to implement UNIX. This often has raised the question, 'What do we do with all the DOS software we've accumulated over the years?' Though MS-DOS users can be trained to use UNIX, very often there are not good [UNIX] substitutes for DOS applications.

"So this is where a product like the Connector [or 6300 Plus] becomes valuable. It allows for the implementation of a UNIX system broadly across the organization, but it still allows people to access software with which they're familiar and where an investment already has been made."

Yet another group that could make use of a UNIX-with-DOS environment includes system integrators and VARs. According to Locus, these users "can mix and match standard off-the-shelf DOS programs with customized UNIX and DOS programs. . . . Applications can be developed in which a DOS program is used to read data while a UNIX program runs concurrently to process that data."

It seems then that there currently *is* a place in the market for the 6300 Plus, but, of course, it remains to be seen whether AT&T can define and command that niche. Note, however, the use of the word "currently", inserted on the suggestion of Scott McGregor, a consulting engineer with DEC. McGregor designed Microsoft Windows, and participated in the design of many later DOS products for Microsoft; he currently is busy at DEC with

UNIX workstation products. Betraying a long-range view, McGregor stated, "My belief is that a lot of the solutions around now are interim solutions. In the past, it's been difficult to go back and forth between different operating systems. One of the things that gives DOS and UNIX compatibility is the 80386 that Intel has just announced [see the last section below]. It has a lot of hardware features that make it trivial to implement that kind of thing. . . ."

In addition to hardware innovations, a second reason that current operating system switching schemes may only be interim measures is that the UNIX applications software base itself is growing. It someday may become unnecessary to switch. As McGregor noted, "The business community is mostly interested in spreadsheets and project management and such things. I think one of the things that's hampered UNIX on that issue is that it's not as easy to do a lot of the interactive graphics things [under UNIX]. If you use a very standard UNIX approach, you pay a lot for the system call overhead for that sort of thing. But I think that's changing now as people are learning that interactive graphics are important. So the thing that's going to motivate people to use UNIX in the business environment is going to have to be software."

For readers who have heard this point before, the song remains the same.

THE CHIPPER CLIPPER CHIP

Not only is it true that the end of the year is a time for reflection; it is also a time to look ahead. In this light, consider the realm of the 32-bit microprocessor. At the beginning of 1985, we were starting to see several implementations of the MC68010; and now—at year's end—we find that several implementations of the MC68020 are impressing observers. The success of these processors notwithstanding, 1986 could prove to be the year of the *high-end* 32-bit superchip; and though it will be a matter of months before implementations of these

UNIX SYSTEMS UTILITY SOFTWARE —SO YOU CAN GET ON WITH YOUR JOB.

For more information, call or write.
(703) 734-9844

UBACKUP
BACKUP, RESTORE, AND MEDIA MANAGEMENT

USECURE
SYSTEM SECURITY MANAGEMENT

SPR
PRINT SPOOLING AND BATCH JOB SCHEDULING

SSL
FULL-SCREEN APPLICATION DEVELOPMENT

S-TELEX
TELEX COMMUNICATIONS MANAGEMENT

SSE
FULL-SCREEN TEXT EDITOR

These products are available for most UNIX or UNIX-derivative operating systems, including System V, 4.2 BSD, 4.1 BSD, Xenix, Version 7, System III, Uniplus, and others.

UNIX is a trademark of AT&T Bell Laboratories.

UNITECH SOFTWARE INC.

Visit us at UniForum Booth No. 1151

8330 OLD COURTHOUSE RD; SUITE 800 VIENNA, VIRGINIA 22180

Circle No. 1 on Inquiry Card

chips come to market, word has it that things are aglow on the horizon.

The low end—or current generation—of the 32-bit microprocessor market is already quite competitive, and with such entries as the 68000 family, the competition will be tough to beat. There is activity on the high end, however, where researchers are borrowing a number of architectural concepts from supercomputer design. With rumors of RISC (Reduced Instruction Set Computer) technology and 5 MIPS chips floating about, curiosity has been stirred in the marketplace. MIPS Computer Systems, for one, is said to be working on a processor with such high-speed traits, but development is not yet along far enough for detailed public comment. There also are two other microprocessors worthy of note, and for these there *is* news available: the Fairchild Clipper and the Intel 80386.

Fairchild Camera and Instrument Corp. has entered the market with a CMOS (Complementary Metal-Oxide Semiconductor) processor that is said to execute instructions in 30 nanoseconds—or at an average rate of 5 MIPS. [It is left to the reader to remember that MIPS is a general rating, and that RISC MIPS do not equal non-RISC MIPS.] Dubbed the Clipper, the processor is a general-purpose, three-chip module designed for use in scientific and professional computing applications in the UNIX environment. The three chips comprise a CPU with an on-chip floating-point execution unit, and two combination cache/memory-management chips (one each for instructions and data). The two cache chips are linked to the CPU via a dual-bus architecture, with one 32-bit bus dedicated to instructions and the other devoted to data. A third 32-bit address/data bus—the Clipper's system bus—allows the chip set to interface with main memory and to work with a wide variety of industry-standard peripheral chips. The Clipper has a streamlined instruction set architecture that runs at 33 MHz, and offers the basic elements of RISC architecture coupled with a macroinstruction unit that provides high-level instructions and functions.

The Clipper employs a "scoreboard" mechanism that simultaneously keeps track of events taking place in all resources.

The processor also employs a "scoreboard" mechanism that simultaneously keeps track of events taking place in all resources. According to Tom Miller, Director of Systems Engineering and Marketing at Fairchild, "a small table keeps track of the registers and all of the data paths and logic elements inside the CPU chip. . . . Before an instruction is issued for execution, all resources required by that instruction are scanned, and if they are not in use or pending use, execution is issued. If a resource is in use or pending use, the instruction is stalled in the pipeline until the other instructions that are ahead of it have completed execution, thus freeing the resources. In other RISC architectures, these computations are done at compile time."

This is the first time such a concept has been applied to a microprocessor. Prior to this, only supercomputer suppliers like Cray and Control Data have used similar schemes. A key reason such techniques now are being used at the supermicro level is that scientists formerly at Cray Research have gone forth and multiplied. One such disciple is Howard Sachs, general manager of Fairchild's Advanced Processor Division.

Some in the field have expressed concern about the Clipper's high clock rate—33 MHz. The concern is that such a speed pushes the limits of current silicon technology, making it difficult—from the perspective of quality control—to assure that a processor will be able to function successfully and produce good yields.

Fairchild Systems Engineering Director Miller addressed this concern: "We have a new design technique that speeds up all circuit elements, and we also have tuned our CMOS process so that Fairchild CMOS is faster than other CMOS processes in the industry. These two effects combined allow us to manufacture 33 MHz devices with a minimum amount of process control, and with quality assurance."

The Clipper module is a 3.0 × 4.5-inch printed-circuit card that plugs into the user's system via a 96-pin connector. Priced at $2451.80, Clipper will be available in sample quantities come June of '86, with production volumes available in the fourth quarter. Initial software offerings include a port of System V.2; optimized Fortran, C, and Pascal compilers; as well as an assembler.

OUT OF THE GATE WITH A LEG UP

Though not as high on the high end of the 32-bit spectrum as the Clipper, Intel's new 80386 processor already is available in

sample quantities ($299 unit price), and you can believe it hit the ground running, blessed as it is with several advantages that not every chip can boast.

Intel claims the 80386 operates at a sustained speed of 3 to 4 MIPS—not quite the 5 MIPS of the Clipper, but as Scott McGregor stated (or rather, understated), "The 386 is capable of doing UNIX in an interesting way; that's enough horsepower to do UNIX."

A key advantage already at work for the 386 is that it has the largest base of existing software for any 32-bit processor. This is because the chip is fully compatible with all software generated for Intel's iAPX 86 family, including the 8086, 8088, 80186, 80188, and 80286 processors.

The 386 also contains on-chip support for large virtual memory addressability—more than 64 trillion bytes. Further, as alluded to earlier, the chip provides a "multiple-execution" environment that allows it simultaneously to run programs written for different operating systems such as UNIX and MS-DOS.

A series of products—the 386 family—was announced along with the microprocessor. Included among these were the iSBC (single-board computer) 386/20 Multibus I board, the iSBC 386/100 Multibus II board, 386 operating systems, and 386 development software. Two of the operating systems, due out in the fourth quarter of '86, are iRMX 286/386 for real-time multitasking systems, and System V/386, the result of a contract between Intel and AT&T that continues a partnership established earlier this year.

As columnist Richard Morin said last month, "These are exciting times for hardware junkies", and UNIX continues to play a major role in computer science advances. The past year certainly has proved this to be true, and 1986 promises to continue carrying the torch. Here's to a good year.

David Chandler is the Associate Editor of UNIX REVIEW. ■

UNIX*-C COURSES

THE UNIX SYSTEM FOR END USERS
Jan 6-10 • Feb 3-7 • Mar 10-14

THE UNIX SYSTEM FOR THE DP PROFESSIONAL
Jan 20-24 • Feb 24-28
Mar 31-Apr 4

C LANGUAGE PROGRAMMING
Jan 27-31 • Mar 3-7 • Apr 7-11

HANDS-ON SESSIONS
OTHER COURSES AVAILABLE
DISCOUNTS FOR EARLY REGISTRATION

COURSE LOCATION
WASHINGTON, D.C.
METROPOLITAN AREA

On-Site and customized courses also available. Call or write for course descriptions, complete schedule and registration information.

(301) 498-0722
WEBCO INDUSTRIES, INC.
14918 LAUREL OAKS LANE
LAUREL, MARYLAND 20707

*UNIX Is a trademark of AT&T Bell Laboratories

CLIP AND SAVE

Circle No. 2 on Inquiry Card

BLAST®

PC-MINI-MAINFRAME COMMUNICATIONS SOFTWARE

ANY COMPUTER WITH BLAST CAN TALK TO ANY OTHER COMPUTER WITH BLAST, the universal file transfer utility linking many different computers, operating systems, and networks, via RS 232 serial ports.

NO ADD-ON BOARDS TO BUY! BLAST software uses any asynchronous modems or direct connect for fast, error-free data transfer through noisy lines and PBXs, across LANs, and over satellites or packet switched networks.

THE PERFECT LOW-COST LINK FOR PC's, MINIS, MAINFRAMES
Transfer binary or text files, or executable commands. Use BLAST standalone, or build it into your application.

$250/Micros **$495-895**/Minis **$2495**/up Mainframes

COMMUNICATIONS RESEARCH GROUP (800)-24-BLAST
8939 Jefferson Hwy. Baton Rouge, LA 70809 (504)-923-0888

Circle No. 57 on Inquiry Card

THE HUMAN FACTOR

(Un)reasonable assumptions

By Richard Morin

If anything can go wrong, it will.

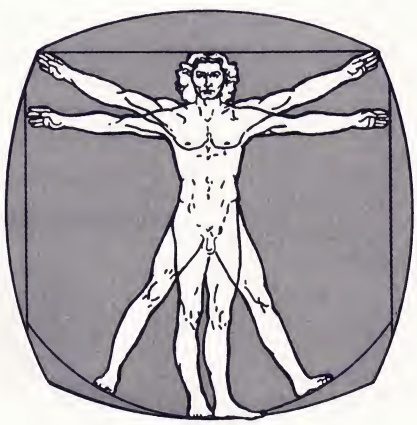
Murphy's Law

We mortals tend to expect a certain amount of consistency from the universe. One looks for the light switch in the same place it was found last time. Only when the switch has been moved or disabled does confusion set in. Despite Murphy's Law, this model actually works fairly well. The universe *is* pretty consistent, by and large, and we're reasonably good at handling the exceptions that occur. Besides, to expect utter chaos would be pretty disabling. A small amount of paranoia, on the other hand, can be very useful.

EXPECT THE UNEXPECTED

Programmers are all too acquainted with "little" changes to project specifications. Projects are seldom completely understood at the outset, and extraneous details often arise to complicate matters. Although these problems generally percolate up from within a company, one must also deal with occasional bolts from the blue. This can happen during the design, development, or even maintenance phases, and the impact can be substantial.

Commercial programmers use up quite a bit of time messing about with ZIP Code data. Untold



millions of lines of COBOL have been written to handle these codes in one way or another. You may have noticed that the Post Office has instituted an optional four-digit extended form, known as ZIP+4. First class mailers can ignore the extra digits, but commercial mailers have been given strong incentives to use all nine. Consequently, a certain amount of COBOL currently is being hacked to provide for the extra four digits. That's just the way it goes. The Post Office has its reasons, as well as the needed clout. There's really no way for programmers to anticipate such changes, so we just have to accept them gracefully.

EXPECT THE EXPECTED

Some changes can and should be expected, however. For instance, a very large and quite predictable recoding job will be needed over the next 15 years because a great deal of commercial code follows the quaint habit of storing only the last two digits of the year. Whether through economy ("every byte counts"), modesty ("this program won't last that long"), or simple sloth ("let the next guy worry about it"), a great deal of code has been written that will go up in flames on or around 1/1/00.

Fortunately, the creators of UNIX showed foresight in this regard. By coding time as the number of seconds from 1/1/70, they put the hassle off for quite a while. The **gettimeofday(2)** documentation says the value is an *unsigned long*, which gives us about a century and a half. Unfortunately, the documentation for the **ctime(3)** family of routines simply specifies the value as a *long*. Still, this will work until January 19, 2038, on most systems. But the **asctime(3)** routine on my workstation blows up when it hits the year 2000. Sigh. Well, *ars (unsigned) longa*, as they say. . . .

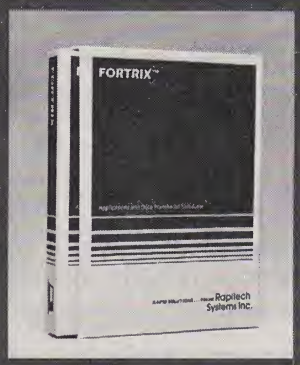
ROOM FOR GROWTH

A few other UNIX glitches also lay in wait to trap the unwary. Probably the most hideous example can be found in the *BUGS* section of the 4.2BSD **sort(3)** manual page: "Very long lines are

Without FORTRIX,™ moving up to 'C' can cost you a bundle!

The bundle we're referring to consists of your existing FORTRAN programs and files. Costly items you'll have to discard when you move up to C, **unless** you save them with FORTRIX™!

Here at last is a program that automatically and rapidly converts FORTRAN code to C code, allowing you to salvage your FORTRAN material at approximately 600 lines per minute. This incredible speed allows a single programmer to convert, debug and put into operation a typical 50,000 line package in only one to two weeks. Plus, the resulting "C" program will run 15% to 30% faster than the original FORTRAN program, while occupying 35% less disk space! And the system even helps you learn coding in C language as you compare your own familiar FORTRAN programs with the corresponding C language programs generated by FORTRIX™.



There's a complete selection of FORTRIX™ versions to suit the full range of user requirements: Original FORTRIX™-C, which translates FORTRAN code to C code, allowing input data files to remain fully compatible with your new C program; FORTRIX™-C+, with the added ability to handle COMMON and EQUIVALENCE statements, character handling and direct I/O; FORTRIX™-C′, the complete FORTRIX™-C+ package configured for non-UNIX* systems including VAX/VMS; and FORTRIX™-C/micro, standard FORTRIX configured for use on the IBM PC and compatibles.

FORTRIX™ has already been installed on 26 different brands of hardware, so whichever FORTRIX™ version meets your needs, you can be sure it will exceed your expectations in terms of speed and cost savings realized. Why not act now to save **your** bundle? Get full technical details, plus references from among over 100 satisfied licensees, from Jim Flynn at (212) 687-6255, Extension 44, or write to him at Rapitech Systems Inc., Dept. A2, 565 Fifth Avenue, New York, NY 10017.

Call toll free 1-800-FORTRIX
(in NY State 212-687-6255)

FORTRIX,™ Fortran-to-C Conversionware℠ from

Rapitech Systems Inc.™

Telephone (212) 687-6255/Telex 509210

NEW RELEASE!
NOW, FORTRIX FOR
VAX/VMS AND MS-DOS!

Circle No. 46 on Inquiry Card

*UNIX is a trademark of AT&T Bell Laboratories

silently truncated." Even ignoring the cuteness of the documentation (just how long *is* "very long"?), the bug simply is unacceptable. This is *the* system sort utility we're talking about here, and any number of UNIX applications use records that exceed the (approximately 500-byte) limit. I understand that AT&T has worked on the problem, and that **sort** now sets an error flag when it truncates a line. The documentation still fails to give the actual limit, but maybe that's just UNIX.

In any case, a number of implementation mistakes exist. They fail to:

1) Allow for growth in the input data.

2) Report on input limit overflows.

3) Document program limitations adequately.

Unfortunately, these kinds of errors are all too common. Many programs fail when subjected to gigantic input lines or the equivalent. Sometimes this is unavoidable; at other times it simply reflects a lack of effort or foresight on the part of the designer. In any case, few programs handle such conditions elegantly. Many UNIX utilities simply dump core on the slightest provocation. Program limitations are seldom admitted, let alone documented well. It should be noted, however, that UNIX is better than most operating systems in this regard. The UNIX manual's *BUGS* section is an unusual—and refreshing—attempt at honesty in documentation.

Other UNIX limitations are more forgivable, stemming from the days when the system was run only on small minicomputers. For one, the major and minor device number fields are too small for systems with large numbers of devices. For another, files cannot span physical devices, crippling some very large applications. A set of limitations having to do with character sets serves as a third example, and causes especially difficult problems on the international front.

CHARACTER-ISTICS

The de facto standard for UNIX character sets is seven-bit ASCII. This means that a number of nifty filters break on binary data, but that's merely unfortunate and annoying. The real problem is that seven-bit ASCII hasn't the room for additional characters found in foreign languages. Since ASCII is the *American* Standard Code for Information Interchange, this shouldn't be too surprising. Nor should the designers of UNIX be faulted for their choice. Imagine approaching Thompson or Ritchie 15 years ago to ask about international character set compatibility issues.

We find UNIX being used, as is, by programmers all over the world. They may not be totally happy with it, but it beats walking (and TSO).

However, some ingenious terminal manufacturers have found a solution of sorts by simply replacing little-used special characters with the needed European alphabetics. The last such terminal I tried sacrificed the back slash and the vertical bar to the cause, but who needs weird characters like that anyway?

The problem goes deeper, however. UNIX can and will be modified to use eight-bit characters, but some code still will break. This is due to some naive assumptions about the nature of alphabetic characters. For instance, every character has both upper and lower case forms, right? Not in German. Each character has a unique place in the sort sequence, doesn't it? Not in Danish and Finnish. A character is a character is a character, isn't it? Well, the German umlaut (¨) is simply a diacritical mark, and characters so marked sort as the same thing. In Finnish, on the other hand, the "ö" is actually a totally different character, sorting at the end of the alphabet.

Middle Eastern and Asian languages present even worse problems. The Japanese are busy designing character sets that can handle Kanji, which has literally thousands of characters. Arabic and Hebrew, reading *tfel-ot-thgir*, introduce their own dilemmas, particularly when mixed with other alphabets. Will UNIX adapt? Yes, but not overnight, and you should expect to hear some screaming in the process. Character manipulation is at the heart of many data processing operations. The authors of UNIX recognized this when they made the single character the basic unit of UNIX data. Character operations thus pervade UNIX, and the needed conversion will be a *major* operation. In the meantime, we find UNIX being used, as is, by programmers all over the world. They may not be totally happy with it, but it beats walking (and TSO).

WISHFUL THINKING

One of the really comfortable

things about the idea of top-down design is the belief that there won't be any really rotten surprises during implementation. This is largely wishful thinking, of course. Users change their minds with the passing breezes, special constraints arise from hardware and software peculiar to specific vendors, and problems are seldom understood completely at design time. Still, the belief in top-down design *is* comforting, particularly to managers.

On the standardization front, the corresponding belief is that, given a committee large and diverse enough, all major needs will be addressed. Maybe so, but again, the odds are against it. The same kinds of factors, in slightly disguised forms, still will be present. Things will be left out, conditions will change, and even the most beautiful standard will end up being modified *ad nauseum* as a result.

There's simply no way to plan for all possible gotchas. Even if Murphy wasn't completely right, he had very good reasons for his law. Specifications change, unexpected data shows up, and code gets used in *totally* weird ways. About the best one can do is to take appropriate precautions. Some coding practice maxims may be illustrative.

It is all too common for data structures to overflow. Therefore, use (ridiculously) large limits on data structures when the needed size isn't *absolutely* known. Virtual memory makes this cheap and easy, and dynamic storage allocation also can be used if it seems appropriate. A healthy bit of skepticism is suggested even so; it's best to check and report on storage overflows. While you're at it, avoid stuffing integers into *chars* and *shorts*. They overflow easily, and often are slower to use than *longs*. In general, assume that your code will be misused, and try to allow for it.

Develop programs iteratively, applying prototyping techniques. You'll still have gotchas, but they'll hurt less. Design for easy modification and debugging, because somebody (probably you) will end up doing it. Such design includes the use of nicely formatted **trace** and **dump** code, the defining of magic numbers as constants, and an attempt to produce clean, readable code. Program modification and debugging tasks are a bit like auto repair: if the engine has been steam-cleaned first, the problems probably will be much easier to find and correct. Finally, avoid cuteness, mysterious side effects, and the other sins detailed so well by Kernighan and Plauger in their classic *The Elements of Programming Style*. Though none of this will safeguard you completely from the forces of chaos, it should help quite a bit.

Mail for Mr. Morin can be addressed to Canta Forda Computer Lab, PO Box 1488, Pacifica, CA 94044.

Richard Morin is an independent computer consultant specializing in the design, development, and documentation of software for engineering, scientific, and operating systems applications. He operates Canta Forda Computer Lab in Pacifica, CA. ■

HALF-EMACSED?

Don't Be Half-EMACSed

HALF-EMACSED? It's when your current editor isn't CCA EMACS... when it doesn't run on all your systems or terminals, when it doesn't have windows, or multi-job processing. CCA EMACS is the most powerful editor ever developed. Over 400 corporations are saving programmer time today by using CCA EMACS on their VAX/VMS, Ultrix and Unix 4.2 systems. CCA EMACS is also available on many other Unix systems.

- **Horizontal and Vertical Windows**
- **Multiple Jobs Simultaneously**
- **Edit up to 200 Files at Once**
- **Program Mode for C, LISP, ADA, TEXT, more**

800-222-0214
in MA 617-235-2600

Uniworks, Inc.
20 William Street
Wellesley, MA 02181

UNIX is a trademark of Bell Laboratories, VAX, VMS and Ultrix are trademarks of Digital Equipment Corp., CCA EMACS is a trademark of Uniworks, Inc.

Circle No. 3 on Inquiry Card

MADE IN THE USA

UNIX discovers a big world out there

by Mike Banahan

Internationalization. We should have known UNIX would come to this someday. Had we anticipated better, we might have thought of a better term.

There's no getting around the fact that "internationalization" is a polysyllabic monster. Little wonder that not everyone who hears the term is blessed immediately with a clear understanding of what it means. To the degree that I can, then, allow me to summarize some of the goals that the term suggests:

- Support for character sets containing "funny" letters. Although it's very inconvenient of them, there are some people outside of the US who have a peculiar desire to use their own alphabets when they correspond with one another. Of course, this inconvenience is tempered by the fact that these people are willing to spend lots of money to obtain systems that provide what they want.

- As if this weren't enough, some of these same people claim that they don't want to see prompts, error messages, and the like in English. A message like "Jeg forstår ikke" apparently generates a warmer feeling in some places than "Command not recognized"—even though neither actually contains any useful information. Ideally, an "internationalized" program would contain no built-in strings of any sort. Rather, it might come complete with a database capable of providing for the program's "localization" for each language.

- Strings used in the system's interface are one thing, but the issue of localization actually takes on a much broader scope of concerns. Different cultures require different date formats, different currency symbols, and different collating sequences. In the instance of collating, variations depend both on the language *and* the reason for sorting. String comparison cannot always be done by simple lexical equivalence.

- Solutions to problems such as these often generate technical problems of their own. Regular expressions offer a famous example. Consider alphabets with characters that are not encoded sequentially, for instance. Suddenly, [a-z] does not necessarily mean "all lower case alphabetic characters".

The last point raises an interesting issue. It is often forgotten just who the exercise of internationalization is intended to please. For a long time, UNIX has been used chiefly by software developers—people who are quite unrepresentative of the general community of computer users. Although it's true that most of these people would be quite happy to continue working only in the restricted alphabet offered by ASCII, there are many others in the world who respectfully disagree. None of the UNIX strings I've ever seen has made much sense even in English. But we have all learned, with more or less success, to attach appropriate meanings to the strings; the job is made only somewhat easier if you understand English. Perhaps the obtuseness that has come to be recognized as a UNIX signature was intended as subtle preparation for the coming internationalization effort. If so, the crowning

Illustration by Heda Majlessi

UNIX

achievement must be **ed**, whose famous "?" message is understood equally badly in almost all existing cultures.

Although none of this seems be bother software developers, they are not the users who matter. It's true that they are the ones who *care* about character encodings, regular expressions, and what have you, but they aren't the ones with the money. *Those* are the end users, and they don't give a monkey's *** about regular expressions. They want word processing packages that will allow someone in Greece to draft a contract in Finnish before sending a copy over a network to their head office in Moscow, where it can be examined for ideological correctness and filed. If the contract lawyer in Greece happens to be *une Française*, then she would probably like to be able to work in French on the machine. Anyone who can bring out a system that can do all of that will be laughing so hard by the time they get to the bank that they'll need oxygen (let's hope their account is at the Chemical Bank).

"But what has that got to do with UNIX?" comes the cry. The answer, simply, is: nothing at all. But the desires of end users do affect *vendors* in the UNIX community in two ways. First, like the rest of the world, suppliers of most UNIX systems have no real objection to getting rich, and multilingual systems might help them achieve that end. Second, because of the portability of the system and the applications packages it hosts, it is likely that the sheer volume of work involved in internationalizing UNIX will be substantially smaller than it would be for a proprietary system.

This, though, is not to suggest that the internationalization of UNIX will be easy in any absolute sense. The jury is still out, but there is a gathering consensus about some of the classes of problems that exist. This account attempts to draw from that consensus, though it admittedly is instilled with a strong European bias.

CHARACTER SETS

Character sets have been the source of much confusion. Those already at work on internationalization understand perfectly well that it is only within the context of ASCII that "\101" and "A" have any relationship whatsoever. To emphasize the distinction, these people use a special terminology.

Within this lexicon, the *repertoire* of a character set is the collection of graphical symbols the user wants to see. The appearance of a character, of course, depends on the font, point size, orientation, and color in which it's printed. The curious thing is that despite all these variants, most of the western world would recognize an "A" as an "A". There is an obvious Aness about it that shines through virtually any impediment.

Computers, though, lack this flexibility. In any given *codeset*, a character set's repertoire is represented strictly in terms of *encoded* bit-patterns. Thus, in the ASCII codeset, "A" forms part of the repertoire and its encoding is "\101".

The first task confronting anyone who tries to support multilingual functionality is to find an appropriate repertoire. A number of international standards exist, but that's just the problem—there are so *many* that it is hard to know which one to choose. Some of the standards worth considering include graphical characters—the teletex standard, for example. Machines like the Apple Macintosh have proven conclusively that users want not only a large repertoire of characters to choose from, but also the ability to manipulate at least the font and size of those characters. Systems that aren't equivalent to or better than the Mac in this regard may never get out of the gate.

The repertoire issue is worthy of serious debate, but the organizations currently at work on international standards seem mesmerized by discussions of how much material to encode in a given number of bytes. Draft International Standard 8859 offers a standard for eight-bit encoding that includes ASCII in the bottom half (with the eighth bit off), and allows for a number of variants in the top half. Variant 1 (DIS 8859-1) will support most of the western European characters in the top half of the encoding. It still isn't possible to mix Greek and French, though, without switching around. DIS 2022 provides for a number of shift-in/shift-out mechanisms that allow the meanings of encodings to be changed, with the effect of making character streams somewhat context-dependent. None of the standards currently in widespread use, though, allow attributes to be attached to characters.

My own point of view is that the standards debates are being driven from the wrong end. At the moment, there is considerable concentration on how to encode characters and what to do with a given number of bits. But until some market research is done to investigate the needs of customers, design teams will be left to work with a meaningless charter. It is pointless to brag about capabilities that nobody is willing to buy. If a study of the market shows that 19½ bits-per-character would be optimal, so be it. *Then* is when cost should be thrown into the equation so that cost/performance tradeoffs can be assessed.

One thing is for sure. Implementations *must* allow for upward expansion; the initial releases of

DOUBLE THE PERFORMANCE OF YOUR VAX

For 1/3 of the price of a MicroVAX II an AP/10A Attached Processor converts a VAX 11/750 into a multiprocessor with more power than a VAX 11/780.

Additional AP/10A systems give greater performance.

The AP/10A can be used with any VAX 11/730, VAX 11/750, VAX 11/780, VAX 11/785, or VAX 8600.

AP/10A programs can access all VAX peripherals and all VAX operating system capabilities.

The AP/10A can execute programs so transparently that users cannot tell whether a program is executing on an Attached Processor or on the VAX processor.

The AP/10A gives your overloaded VAX unlimited expandability with the most cost effective computing power available while retaining all of your substantial investment in VAX peripherals, software, and system organization.

PRICE 6990 QTY 1, 4990 QTY 10
SPECIFY UNIX OR VMS
C, FORTRAN-77 AND PASCAL AVAILABLE

VAX, MicroVAX II, and VMS are trademarks of Digital Equipment Corporation.
UNIX is a trademark of Bell Laboratories.

Circle No. 58 on Inquiry Card

AVALON

Avalon Computer Systems
425 E. Colorado Blvd. #710
Glendale, California 91205

Telephone: 818 247-2216
Telex II: 910 588-3274
Cable: AVALONCOM

Implementors who find an assignment difficult to accomplish will do well not to complain.

an internationalized version of UNIX might be excused for not including everything, but developers who need added functionality shouldn't be frozen out until hardware and storage costs have fallen enough to prompt a complete redesign. Any company prepared to shoulder the additional cost before then should be allowed to.

It should be emphasized that cost is the real issue in any of these considerations. Technical difficulty is only an issue if it makes a solution too expensive, or if it prevents a solution from being regular and expandable. Implementors who find an assignment difficult to accomplish will do well not to complain. They were *hired* to solve hard problems, after all—if they don't like the job, they should consider a career in politics.

COMPARING, COLLATING, AND CONVERTING

Comparisons, collations, and conversions of characters are accomplished completely independent of encoding. That's because these are semantic—rather than lexical—issues. The distinction may make it difficult to base sorting algorithms closely on encoding, but that's the implementor's problem. Users will need table-driven routines to perform tasks like sorting, character comparison, and other similar functions if their systems are to be capable of adapting to different languages and intended uses.

English itself needs help with these tasks. In British telephone directories, for instance, it would be helpful if the names "McDonald" and "MacDonald" could be sorted to the same position. Such approaches can be extended into absurdity, of course, with the suggestion that "O'Donnell" also be sorted to the same position—but it is not for the implementors to say where the line should lie between a genuine language-dependent problem and absurdity. The customer decides that question. And, in keeping with that, users should be able to generate their own collating and sorting tables so that they can introduce new schemes when the standard ones aren't adequate.

Conversions. Capabilities like **toupper** and **tolower** run into some entertaining internationalization problems. As far as I can ascertain, the German lower case character β (the "sharp s") has no upper case equivalent, but converts into "SS" instead. What does **toupper** return in such circumstances? A string? This is a piece of woodwork that promises to reveal many worms once the paint is peeled off.

REGULAR EXPRESSIONS

The ramifications of internationalized regular expressions are actually unlikely to concern the great majority of system users. However, implementors may want to use them to express concepts with which most of us are familiar. Some common ones include:

```
[a-z]                 all lower case alphabetics
[A-Za-z][A-Za-z0-9]*  identifiers in C (almost)
[^A-Za-z]             any non-alphabetic character
```

We could go on, but let's assume that these have been used in **sed** and **awk** scripts for data validation. How then are they to deal with Norwegian—a language that is nearly as hard to pronounce as it is to write regular expressions for? All of the Scandinavian languages have the interesting feature of "extended" alphabets, with more letters than English. (Of course, the Scandinavians think that English suffers from a "restricted" alphabet.) All of the extra letters in Norwegian are vowels: æ, ø, and å. They follow immediately after "z" in the language's collating sequence, but this isn't where they fit in any proposed encoding I am aware of. Consider the following:

```
"å vaere eller ikke å vaere, det er spøren?"
```

How are regular ASCII expressions to deal with something like this? There doesn't seem to be any consensus at the moment.

MESSAGES

The suggestion that all prompts, strings, and so forth be extracted from programs and listed in databases is fine in theory, but difficult in practice. Let's take a simple example. A program is written to prompt for the user's name and then use it in responses. Some of the responses look like:

```
printf("OK,%s,now I am going to %s your %s\n",name,action,object);
```

If "joe" has signed on, it might say:

```
"OK, joe, now I am going to test your typing skills"
```

This works fine in English, but few other languages

TANDY... Clearly Superior™

The Tandy 6000 lets your office balance the books, track sales and write memos...simultaneously.

For many companies, it's hard to justify the cost of a separate computer for each employee. That's why we designed the efficient Tandy 6000 multi-user computer.

The Tandy 6000 system allows three people to simultaneously access programs and data, and you can expand with up to six users at any time.

With a single Tandy 6000 and printer, you can save time and effort. Your accounting can be processed in one office, word processing in another, and data base management in a third office.

The Tandy 6000 can also help with other departmental functions, like financial planning, inventory, job costing, and sales analysis.

The Tandy 6000 comes with 512K of memory, XENIX 3.0 operating system and a 15-megabyte hard disk drive (26-6022, $5499).

Discover how your business can benefit from a Tandy 6000 multi-user office system. Drop by your local Radio Shack Computer Center for a free demonstration. Ask about our leasing plan, too.

Available at over 1200 Radio Shack Computer Centers and at participating Radio Shack stores and dealers.

Radio Shack®
COMPUTER CENTERS
A DIVISION OF TANDY CORPORATION

Prices apply at Radio Shack Computer Centers and participating stores and dealers. Display terminals sold separately.
XENIX/TM Microsoft Corp. UNIX/TM Bell Laboratories.

Circle No. 43 on Inquiry Card

Send me a Tandy 6000 brochure.
Mail To: Radio Shack, Dept. 86-A-963, 300 One Tandy Center, Fort Worth, TX 76102

NAME _____
COMPANY _____
ADDRESS _____
CITY _____ STATE _____ ZIP _____
PHONE _____

Tools must not make arbitrary judgments about the meaning of the byte streams they process.

have the word order of English. It is almost certain that a German translation would have to transpose the object and the verb(s). Worse yet would be a language where a change to a user of the opposite gender would change word endings throughout the sentence.

There are ways of obtaining partial relief. For instance, under English, one can use the following:

```
printf("%d %s deleted\n", nfiles, nfiles = 1 ? "file" : "files");
```

It's a remarkable database that can persuade **printf** to accept variable numbers of different typed arguments in an already compiled C program, especially if some of the arguments depend on the values of other arguments. To accommodate different languages, you might simply compile a different version of the program for each—but that isn't really an acceptable solution over the long term.

This obviously is a fruitful area for more work. It looks as if one approach might be to ditch **printf** for a runtime interpreter that can be fed a program, as well as any appropriate strings, from a localized database.

CULTURAL ISSUES

What discussion of internationalization would be complete without a reference to cultural issues? Oddly enough, they aren't particularly hard to deal with. Some, though, do have nuisance value. One must account, for instance, for those countries that use "." to separate every three digits in large numbers and use "," as a decimal point. Remember also that though some countries place a currency symbol before an amount, there are others that transpose this order. Also bear in mind that not all currency symbols are single-character, and that some are special graphic symbols.

Date and time formats vary depending on where you are, so **ctime** and its associated routines must be adjusted accordingly. The Gregorian calendar is not universal, and there are countries that don't synchronize their clocks to Greenwich Mean Time, although I suspect most businesses in those countries do. Then again, that might be a bit much to assume since it would seem that the European Economic Community works out its daylight savings time schedules by consulting the Oracle at Delphi.

OTHER MATTERS

A fair amount of effort has already been expended in trying to get the UNIX system "eight-bit clean". If these efforts succeed, tools should be able to pass all eight bits of a byte without corruption, unless they have been designed to transform data. This obviously is a necessary condition. Tools must not make arbitrary judgments about the meaning of the byte streams they process. There are many tools that depend on the meaning of these byte streams: **sed**, **vi**, **sh**, and **sort** are just a few examples. It is most important that these tools be modified so that they place as little dependence as possible on the meaning of data. That way, they will be relatively decoupled from the way that characters are eventually encoded. Of the tools requiring attention, editors will be the hardest to fix.

It seems quite likely that multilingual systems will contain files written in a mixture of languages and possibly a variety of codesets. One obvious question is how a user is supposed to know what language or codeset a file is using. Some have suggested that this information be encoded into the inode of the file, or that an announcement detailing these particulars be placed at the beginning of the file. In truth, though, this is a non-issue. UNIX has always had files that needed to be processed by special tools. The C compiler doesn't understand shell scripts, **troff** input can be processed only superficially by the other tools, **troff** output is generated in a new codeset altogether—and, of course, object files can be understood only by a very restricted set of tools.

The user has always borne the responsibility for this, so there is no point in trying to fix the problem now with *ad hoc* warts. What is called for—and always has been—is a decent file-management system (FMS) that can keep track of file types and dependencies. Naive users then can have their hands held by the FMS, while *experts* continue their battles against archaic *bodges* like **make**.

Mike Banahan is one of the founders and directors of The Instruction Set. A user of UNIX since 1977, he is of the camp that believes the decision to move away from ***cat -s*** *was a retrogressive step. He has lectured in the Department of Computer Science at the University of Bradford and is a popular speaker at European UNIX conferences.* ■

NOW WE CAN TALK

cLINE/cENGLISH™

Features & Benefits

- **cEnglish** is a fourth generation procedural language using English-like syntax.
  Easy to learn and use.
- **cEnglish** has all the advantages of a full programming language including the use of functions and arrays.
  Ability to develop any type of application.
- **cEnglish** programs are compiled into "C" source code.
  "C" programs without "C" programming.
- **cEnglish** permits embedded "C" code.
  "C" language power for systems integration.
- **cEnglish** programs are 100% portable.
  No duplication of development effort.

Sample Program

```
IDENTIFICATIONS
  MODULE: Mininame
  AUTHOR: bcs
  DATE:   8/29/84
/* program that adds first name to a file
GLOBALS
  FIXED LENGTH 1 ans
  FIXED LENGTH 15 fname
END GLOBALS

MAIN PROGRAM

  BEGIN
    CLEAR SCREEN

    USE "NAMES"
    VIEW BY "ID_FNAME" ASCENDING

    STORE "?" TO ans
    AT 23,01 SAY "Add a record ?
    Enter Y or N "
    WAIT TO ans

    WHILE UPPERCASE(ans) EQ "Y"
      CLEAR GETS
      AT 06,01 SAY "Please enter first
      name"
      AT 06,20 GET fname
      READ 1

      STORE fname TO record_name
      APPEND RECORD

      AT 12,10 SAY "Welcome to
      cENGLISH " & fname
      AT 14,10 SAY "Press any key to
      continue. "
      WAIT
      STORE " " TO fname
      STORE "?" TO ans
      AT 23,01 SAY "Add another record
      Enter Y or N "
      WAIT TO ans
      CLEAR ROW 1 THRU 24

    END WHILE

    AT 12,10 SAY "Thats all for now !"
    UNUSE "NAMES"
END PROGRAM
```

Availability

- Computers

  IBM, INTEL, DEC, NCR, AT & T, COMPAQ, SUN, PLEXUS...

- Operating Systems

  UNIX system V, UNIX version 7, Xenix, BSD 4.2, MS / DOS, PC / DOS, ULTRIX...

- Interfaces with data base managers offering High Level C interface such as ORACLE, INFORMIX, C-ISAM (requires no additional C programming)...

- Also available dBASEII to **cEnglish** converter...

cLINE™, Inc.

3550 Camino del Rio North, San Diego, CA 92108 • (619) 281-5593
Distributors Worldwide

IBM is a trademark of International Business Machines. MS/DOS and Xenix are trademarks of Microsoft, Inc. UNIX is a trademark of Bell Laboratories. C-ISAM and INFORMIX are trademarks of Relational Database Systems, Inc. NCR Tower is a trademark of NCR Corp. **cLINE/cEnglish** is a trademark of cLINE, Inc. All other computer names are trademarks of their respective manufacturers.

Circle No. 59 on Inquiry Card

H.KIM

TAKING A GLOBAL VIEW

An appeal for reason

by Brian Boyle

"The UNIX Philosophy" might be as good a name as any for the one thing that all alleged UNIX implementations seem to share. It is this underlying point of view that makes the UNIX system a logical contender to become a reasonably global operating environment.

How might this philosophy be described? Think back to when UNIX was first developed. Even though the commercial operating systems of the day were strictly hardware-centered, UNIX was designed as a *function*-oriented system. The fact that it nevertheless has been accepted by the commercial world—unlike some of its logical predecessors (Multics, for example)—means that an intuitively appealing transition has occurred in the industry. But although this new approach has been accepted, it remains as disquieting today to the established powers of the field as it was when first introduced. It is hardly surprising, then, that UNIX is still the only significant operating system on the market that attempts to be blind to hardware differences.

Still, given that the challenges faced by those who wish UNIX to be a global presence differ only minimally from those faced by people who have similar ambitions for other systems, a valid question to pursue is: "Why UNIX?" Why not let the vendor(s) of some proprietary operating system(s) scout these uncharted territories? (After all, you can always tell the pioneers by the arrows in their chests.) Since several such vendors already offer economically successful proprietary approaches in each of several regions, why not pick the best—or most applicable—elements from each and consolidate them into a single standard?

Probably the most straightforward response is that such an approach simply will not work. The communications and computer industries—now apparently on a convergent (or collision) course—have long been distinguished by the fact that while communications standards are *de jure* (made by formal agreement), computer standards are *de facto* ("might makes right").

It always has been clear to those in the communications industry that the ability to interface consistently, reliably, and bidirectionally with as many other units as possible—*regardless of vendor*—is fundamental. Vendors in the industry know it is imperative that they differentiate their offerings by how well they perform a function—power, features, cost, and the like—rather than by function itself. That's because functions—like 60 Hz, 110 volt AC "house current"—need to be predefined. (Would you buy an "enhanced" house that offered 80 Hz, 160 volt, four-prong wall plugs?)

Unlike those in the communications industry, however, established vendors in the computer industry have resisted standardization. This has been reflected in the commonplace corruptions of formally defined standard programming languages—proprietary "enhancements" that permit portability *from* but not *to* the legitimate standard. The object of these corruptions, of course, has been to develop a captive base of customers.

Even in a climate of genuine altruism, it would be difficult to escape the historically autocentric view sometimes labeled as the "NIH syndrome"—Not Invented Here. Well-intentioned concerns for efficiency of *imple-*

Illustration by Hyon Kim

mentation on a particular architecture (such as byte addressability) often dictate *function* at a higher level of abstraction.

Such parochial decisions often are harmless until 1) a system is moved to a different architecture, 2) machines from different vendors need to communicate, or 3) a system has to be used in a "foreign" environment. This third concern is something of a combination of the first two in that it involves handling new (data) structures *internally* and communicating via new (user) interfaces *externally*.

One fundamental reason that UNIX is the best candidate to become a global operating system standard is that it already exists as a functional definition independent of its implementation. Given the historical development of the UNIX system itself, it is unlikely that any operating system written in assembly language or evolved across a less diverse set of architectures, instruction sets, word lengths, memory systems, peripheral types, and environments could be as readily *ported* to the world's many languages and cultures.

VIEWED IN THE ABSTRACT

The barriers to international portability and communications parallel the problems that proprietary software raises for people who wish to modify system architecture or establish communications between machines from different vendors. This parallel actually may help in classifying the problems and perhaps will even point in the direction of a general solution.

> Even in a climate of genuine altruism, it would be difficult to escape the historically autocentric view sometimes labeled as the "NIH syndrome".

The essential strength of the UNIX focus on function is that it allows a complex system to be divided into minimally coupled "levels of abstraction". As this notion is fundamental to modern computer science, mathematics, and—at some level—all abstract thought, it is not surprising that it is expressed in numerous hardware and software standards and taxonomies.

To see the benefits of abstraction more clearly, let's look at how it already has been employed. Figure 1 depicts the International Standards Organization's *Open Systems Interconnect* (OSI) communications model, the /usr/ group Technical Advisory Committee on Internationalization's model for internationalization (the "INTECH" model), and a model describing the levels of compatibility among the various versions of UNIX.

The latter two models are based on the template provided by the OSI scheme for distributed networks. (See the April, 1985, issue of UNIX REVIEW, pp. 25-26, for Mark Hall's "The Potpourri of Networks".)

By charting out abstract pictures such as these, it becomes possible to explore what otherwise might be obscure. As an example, one of the sorest weaknesses in the UNIX market is revealed by the *Compatibility Levels* component of Figure 1. At the top of the model we find considerations of the portability of UNIX *Applications*. This is because applications are the component of the system that show the greatest variation from version to version—at the user interface level at least. Our abstract picture reflects this since, as with

COMMUNICATION ISO-OSI Model	INTERNATIONALIZATION INTECH Model	PORTABILITY Compatibility Levels
Application	Application/Language	Application
Presentation	Syntactic/Format	Source Code
Session	Lexical/Group	System Interface
Transport	Classification	Data Format
Network	Logical Character	Data Media
Data Link	Physical Character	Instruction Set
Physical	Byte/Bit-String	Architectural

Figure 1 — *Three standards models that exhibit the benefits of abstraction.*

all of these models, the top is reserved for the system layer with the least degree of commonality.

The *Source Code* layer listed immediately below *Applications* is a level of portability that long has been addressed by standardized programming languages—generally unsuccessfully, since the standard source language program invariably includes *system calls* to a proprietary operating system. The typical language for UNIX programs has been the C language, which is now being defined by the IEEE X3J11 Committee. It's here that the format for calling sequences is defined.

Standardization of the *System Interface* level (one rung lower in the model) represents a major step toward achieving true UNIX portability. This is the level addressed by the AT&T *System V Interface Definition* (SVID)—or the /usr/group *Interface Standard*. Each defines a "conforming" UNIX system in terms of *what* functions are provided rather than *how* the underlying implementation is to be carried out by a vendor.

Jumping directly to the bottom of the model, we find the compatibility layers that least concern UNIX applications vendors. Portability between nearly identical systems is seldom a goal for these vendors, few of whom would bind themselves to either of these levels. For the community at large, *Instruction Set* portability is a job for compilers, while the *Architectural* differences of machines employing the same processor concern only the manufacturer itself or perhaps one of the established system software houses.

Nevertheless, this still leaves UNIX applications developers to consider the two "trivial" levels of *Data Media* and *Data Format*. Although both have been largely unaddressed, they have drastically affected UNIX acceptance and market penetration.

For instance, mid-range supermicros—systems too large to be backed up on standard floppy disks, too small to warrant a standard nine-track tape drive, and too cost-sensitive to have multiple tertiary storage—have been plagued by the lack of an adequate standard software distribution mechanism. The simple

Freedom of Communication in the Xenix Environment

Nine Track Tape Subsystem

If you are an IBM AT user running IBM Xenix, you can now declare your independence. Overland Data has just introduced the TC50X, a high performance tape controller and software package specifically designed to provide you unlimited freedom to communicate with other Unix systems.

Tape oriented Unix™ interchange commands such as *tar*, *cpio* and *dd* are fully supported and are completely compatible with the same format tapes from mainframes and minis.

Standard file system *backup* and *restore* commands are supported as well. Frequent backup is now simple and fast. Disk allocation fragmentation is eliminated on a restore, preventing gradual performance degradation common on poorly maintained systems.

The software driver is interrupt driven, with multiple data buffers for high speed transfer using a low cost ½-inch streaming drive supplied with the system. Error handling and recovery is fully automatic. Standard features of character tape device drivers are supported including rewind and no rewind on close (*/dev/rmtl* and */dev/rnmtl*). This feature allows multiple *tar* volumes or file system dumps on one tape.

For more information contact Overland Data and ask for Xenix System Support.

ODI Overland Data, Inc.
5644 Kearny Mesa Road
San Diego, CA 92111
Tel. 619-571-5555
Telex 754923 OVERLAND

Xenix is a Registered Trademark of Microsoft Corp.
Unix is a Registered Trademark of AT&T Bell Laboratories.
IBM AT is a Registered Trademark of International Business Machines Corp.

Circle No. 4 on Inquiry Card

absence of a "software aftermarket" comparable to the one attached to MS-DOS has been a major factor in the lethargy of the UNIX industry—and all for want of one or two levels of standardization.

Similar dangers apply to an incomplete or insufficient international UNIX standard.

FROM THEORY TO EXECUTION

The specific difficulties of European and Asian implementations are well addressed in the other articles of this issue, but a few examples will illustrate the use and value of the internationalization model depicted in Figure 1. Attention to this model, it should be noted, will not only allow developers to serve the needs of UNIX users in the world at large, but also those who live in the US. Many domestic "exceptions" may be classified and treated as subsets of a more general approach, which should serve to point out that internationalization is something more than a genteel gesture toward "foreigners".

At the lowest, most physical level of the internationalization model, we find the basic elements of our data representation, the universally accepted bits and bytes that know no national borders. To many of us, bits of each eight-bit byte are synonymous with "character"—a property of which a number of UNIX utilities and programs "take advantage". One needn't have been hacking around with **vi** or **sed** for very long to have been bit by a bit in a returned byte that wasn't a bit like the bit in the byte we had sent earlier. Both the SVID and /usr/group definition are quite explicit in separating type *char* from type *byte*, *integer*, *long*, or any other explicit bit-strings. Depending on the specific Asian country and the richness of the set of ideographs permitted and the method of packing and framing allowed, characters might be seven, eight, 14, 15, 16, 21, 23, 24, 28, 30, 31, or 32 bits, framed in the appropriate number of bytes. (One Korean representation of components of the Hangul trigram uses radix reduction to pack three ostensibly six-bit subcharacters into a single 15-bit character framed in two bytes.)

One needn't have been hacking around with **vi** or **sed** for very long to have been bit by a bit in a returned byte that wasn't a bit like the bit in the byte we had sent earlier.

As noted elsewhere in this issue, we need a separate, independent, more explicit level to define the character and its frame as a *Physical Character*. In the UNIX asynchronous byte-stream orientation, what happens to reliability when we begin to handle multibyte characters with parity only in selected bytes? On the one hand, if we use an entire extra byte to handle extended ideographs or mixed Roman, Greek, and Cyrillic sets, will we sacrifice acceptance in countries where 50 or 100 percent increases in already outrageous line-cost simply will not be tolerated? On the other hand, if we restrict the set to, say, 14 or 21-bit characters—in an effort to gain added reliability—are we merely painting ourselves into a different corner? (Why are all fixed disks at least 85 percent full regardless of size or load?) If character semantics are governed by toggled multibyte shift-out and shift-in characters, how does one handle non-sequential starting points in files and database records?

The third and fourth levels of our internationalization model, the *Logical Character* and the (logical character) *Classification*, must be conceptually divorced from the physical representation of the character encoding scheme. The classification of logical characters includes both collating sequence and categorization (as **isalpha**, **isprint**, or whatever), while the mapping from *physical* to *logical* characters is more subtle. Just as we have synonyms and homonyms in alphabetic written languages, ideographic written languages include instances of many-to-one mappings known as "homographs". Collation is far from simple in these ideographic languages, since it must account for varying sequences of stroke order, stroke count, radical base, phonetic code, telegraphic code, and "dictionary" order. Complex though it may be, this problem must be addressed if limitations are to be lifted for much of the world. (Consider, for instance, that the Japanese have felt strongly enough about this issue to have imposed restrictions on the first names that children may legally be given so as to facilitate Katakana recordkeeping.)

Even the European languages contain some tricky idiosyncrasies that collating algorithms must take into account: one-to-two character mappings (German lower case sharp-s "β" shifts to upper case "SS"); two-for-one character mappings (Spanish *ch* to *c* or Scandinavian æ to *a*); and

one-to-zero mappings, in which an embedded hyphen is ignored for sorting or matching purposes.

Collating sequence itself is a major issue, considering that the Anglocentric ASCII omits all of the accents and diacritical marks that occasionally serve as the only distinction between a "minimal pair" of otherwise identical words. What is worse, when ASCII omits three entire Scandinavian characters, they are kludged into the 27th, 28th, and 29th character encodings—which not only do not sort correctly (rationally) but also manage to conflict with various UNIX "magic" characters.

The uppermost three levels of the INTECH internationalization model lie in the *semantic* category, beyond either the two *physical* levels or the intermediate *logical* levels. Aside from generally unacceptable word-for-word interlingual substitution in skeletal system messages, the *Lexical/Group* level of the model deals with such external structures as immediate-context sensitivity (position in word—in order to account for the four possible Arabic character types: initial, medial, final, and single); presentation sequence (not necessarily left-to-right within a top-to-bottom framework *and* not necessarily consistent within a single framework—Arabic numbers are written left to right within a right-to-left frame); and internal structures such as numeric group separators (comma versus decimal point, and vice versa).

The *Syntactic/Format* level of the model covers message restructuring at a simple syntactic level (number agreement, conjugation, declension, and gender in inflected languages) and format transformations such as date, time, and currency presentation. The top level, *Application/Language*, is probably beyond the scope of a UNIX standard at this time, given the past failures of most of the "mechanical translation" projects undertaken by DARPA, RAND, and numerous academic institutions. Still, this level of the model is valuable as a placeholder—as in the case of the *Application* level of the OSI and UNIX compatibility models.

WHY ME?

Following logically from the

Continued to page 100

140 MB disk/tape subsystem for XENIX

Finally, a complete XENIX subsystem for the AT.

Disk Features
- 30, 40, 55, 72, 118 Megabytes (formatted)
- Combine drives with each other or existing drive
- 25 milliseconds average access time
- Simplified installation
- Necessary file modifications done automatically

Tape Features
- 60 Megabyte 1/4 inch cartridge
- Standard XENIX commands (cpio, tar, dd, etc.)
- Fully integrated driver software

Subsystem Features
- Entire subsystem fits inside the AT
- External version with 6 expansion slots available (pictured)
- One year factory warranty

Emerald
Systems Corporation

Mainframe Storage for Micros

4757 Morena Boulevard
San Diego, CA 92117
(619) 270-1994
Telex: 323458 EMERSYS
EasyLink: 62853804

Emerald & Mainframe Storage for Micros™ Emerald Systems Corp.

CHANGING CHARACTER

Technical problems in the multilingual support of UNIX

by Karen Barnes and Dan Epstein

The UNIX system was designed around the assumption that a single character set would suffice. However, the need to represent data and to communicate with users in languages other than English has made apparent the requirement for supporting multiple character sets.

Character sets can be encoded in many ways, as some current national standards attest. For example, Japan's JIS 6226 uses 14 bits of two bytes to define space for 8864 characters (about 7000 of which already have been defined). ASCII, on the other hand, can be encoded in seven bits of one byte. To contend with this disparity, some coding schemes have been adopted which distinguish between languages that can be encoded in one byte and those that require two.

It is not the intention of this article to suggest the correct encoding method but rather to identify the problems that stem from supporting a chosen scheme. We therefore will only discuss the challenges of using all bits of a byte for European and Middle Eastern languages, and of representing the ideographic characters of Asian languages in two bytes. We restrict ourselves in this manner because all standards currently are based on one of these two approaches.

In reviewing the behavior of UNIX commands and routines, we find that new encoding schemes cause difficulties under the following conditions:

1) Use of the eighth bit of a data byte.

2) The editing and display of data.

3) Collation.

4) Regular expression parsing.

5) Time and date processing and display.

6) Character classification.

7) The printing of messages.

UNIX COMMANDS, UTILITIES, AND ROUTINES

The behavior of commands, utilities, and routines is determined by the language—and, by extension, the character set—in which the user wishes to work. The reason that we make this distinction between languages and character sets is that the characters from more than one language can be represented in a single character set. For instance, one of the standards proposed by the European Computer Manufacturing Association (ECMA) supports the development of a character set containing characters from over 40 countries. Let's probe into the areas where the support of the encoding for this or any other character set is likely to encounter problems.

Local Customs. Besides multiple languages, several conventions and local customs that vary from country to country concern system developers. Among them can be counted: 1) time formats, 2) date formats, 3) abbreviations, 4) decimal delimiters, 5) prompts, 6) alternate sets of digits, and 7) currency formats.

A significant problem of supporting local customs involves the calculation and reporting of time and date. This is not a trivial problem since some cultures use a different calendar than the one used in the Western world. The proper reporting of time and date depends on the algorithm used to calculate time divisions—days, months, and years. In addition, month and day names and their abbreviations are custom-dependent. For instance, some countries use more than three letters to abbreviate month and day names.

The **date(1)** command obtains the system date by accessing the *tz* environment variable. Cur-

Illustration by Stephen G. Luker

لات - توظيف أموال - تجارة - عمولة
مؤسَّسَة الراجحي المَالية

rently, UNIX works only in Greenwich Mean Time, calculating and reporting the date according to the Gregorian calendar. The Hebrew, Arabic, and Asian calendars, though, are measured from different dates of origin and use different algorithms for converting system clock cycles into local time.

Commands that accept date and time input from the user limit variables to specific field sizes. For example, in the **date(1)** command, "year" is defined as a number between 0 and 99. The field descriptor "d" defines the date format as *mm/dd/yy*, but in Europe it is commonly formatted as *dd/mm/yy*. There are also country-dependent field descriptors, such as "a" for abbreviated weekday and "h" for abbreviated month.

In addition to the **date(1)** command, other system utilities report date and time. For instance, various options to the **ls(1)** command report when a file was last modified, and **pr(1)** reports the time of day at which it produced paginated output. These commands call the routine **ctime()**, which must be language and character set-sensitive so as to report date and time in the language most appropriate to the user.

Character Classification. It is inappropriate to classify a character strictly with a seven-bit code value when multiple character sets are supported. Any change in the character codes will affect the performance of UNIX macros such as **isalpha()**, **isupper()**, and **isprint()**—among others that provide information to the system about the properties of character-coded integer values.

Character classification macros like these are part of the standard C library that operate by table-lookup. The ASCII-based tables that are referenced are hard-coded into the **ctype** macros. For indexing, UNIX table-lookup logic currently uses characters encoded in seven bits. With the introduction of character sets that require more bits for encoding, this logic produces unreliable results.

Character sets can be encoded in many ways, as some current national standards attest.

One problem is caused by sign-extension. When an eight-bit character (index) is loaded as a signed integer, it results in a negative index that subsequently causes unpredictable results. Compounding this problem is the fact that hard-coded tables are not character set-sensitive. A third problem relates to size: character sets for languages with as many characters as Chinese simply cannot be accommodated. The memory requirements alone can be staggering since encoded tables containing information on the properties of characters each can consume at least 50 KB (25,000 two-byte characters).

The problems hardly end here. Character classification macros that test for upper and lower case or do up-shifting and down-shifting, for instance, are totally useless when confronted by languages that do not differentiate between upper and lower case. This is an important issue since many commands use **ctype** and **conv** macros such as **islower()**, **isupper()**, **tolower()**, **toupper()**, and **isprint()**. These macros, for instance, are an integral part of the **passwd** logic used to encrypt and decrypt passwords.

Scanning and Filtering. A plethora of UNIX programs read some input, perform a transformation on it, and then write out the result. The behavior of commands that perform text scanning is wholly dependent on the correct interpretation of the input. Unfortunately, UNIX's rich set of scanning tools can have problems when confronted by languages other than English.

The UNIX "filters" that scan data assume that a character is encoded in seven bits of one byte. This rigid design makes it difficult to support non-English character sets. As an illustration, compilers scan source files for comment-delimiters and string-literals. Although the syntax of a programming language is no different for non-English-speaking programmers than for those who speak English, it is essential that people be able to write comments in their native language. But a compiler cannot scan source files for non-English comments as long as its scanning process is restricted to recognizing ASCII characters only. Because half of an Asian character (one byte) can have the same value as a comment delimiter, this is a particularly urgent problem.

The **awk(1)** pattern-scanning language is another utility that's locked into seven bits. In performing a specified action on lines that match a particular pattern, **awk(1)** uses fixed-size 128-byte tables to check for things such as FS (field separator) and OFS (output field separator). It also uses **lex(1)** and **yacc(1)** for building a lexical analyzer and for converting a context-free grammar into a set of tables.

The **lex(1)** analyzer reads an input stream by byte-processing rather than by character-processing. As part of this work,

lex(1) determines whether each byte is an ASCII character or not. There are many instances where **lex(1)** scans strings byte-by-byte by incrementing character pointers within loops. Constants based on the fixed size of the ASCII character set are sometimes used in *for* loops to step through the character set. If the character set in question contains 25,000 items, this may be unsatisfactory. The analyzer also uses its own hard-coded character-classification functions. In any event, **lex(1)** builds a transition matrix for use by the program it generates. This program recognizes patterns in a stream of text.

Unfortunately for **lex(1)**, the number of transitions between the current state and the next state grows exponentially each time the size of the character set is doubled. Although there is compacting logic in **lex(1)**, the internal tables used to generate the analyzer's matrix consume huge amounts of memory.

Collation. Correct behavior of any sorting algorithm depends on the language in which users want their data sorted. Several UNIX commands and routines perform sorting—the commands **ls(1)** and **comm(1)**, and the routines **strcmp(3c)** and **strncmp(3c)**, for example. There's also the command **sort(1)**, of course.

A language-independent sorting algorithm must be able to order data lexicographically. But UNIX sorting algorithms only make use of seven-bit ASCII keys. They also machine-collate according to a character's binary value. If it is eight-bit data that is to be sorted, **sort(1)** uses signed characters for character comparison, and subsequently sign-extension might take place. This causes characters encoded in eight bits to sort in reverse machine-collating order, with eight-bit keys appearing before seven-bit ones. In theory, **sort(1)** should provide dictionary-order collation but, in truth, it merely ignores punctuation and proceeds to machine-collate. If two words are equivalent in every way except their case, true dictionary sorting would place a precedence on upper case; but **sort(1)** retains the order of the input data when this occurs.

Some European languages require two adjacent characters to occupy a position in the collating sequence (for example, *ch* follows *c* in the Spanish alphabet). We refer to this as a two-to-one conversion. Other languages require a one-to-two conversion (for example, *sharp s* (*β*) is equivalent to *SS* in German).

Some languages also designate certain characters to be ignored in character comparison algorithms. For instance, if "-" is an ignored (or "don't-care") character, then the strings "REACT" and "RE-ACT" are equivalent. The two-to-one mapping, one-to-two mapping, and "don't-care" issues cause problems for UNIX because of the system's insistence on collating byte-by-byte.

Asian languages, of course, raise still other concerns. They are built on a foundation of characters that have meanings unto themselves. Lexicographical ordering of Asian data, by definition, implies sorting according to several collating sequences. (See Figure 1.) The UNIX **sort** utility, however, is locked into a hard-coded ASCII character-mapping table of 128 bytes. This is insufficient to support Asian collation tables—if only on the basis of size (some Asian languages contain as many as 25,000 characters). What's more, byte-by-byte comparisons and byte-oriented table-lookup simply do not produce the correct results.

When handling Asian characters, it is necessary to store

CEEGEN-GKS GRAPHICS SOFTWARE in C for UNIX

- ☐ Full implementation of Level 2B GKS.
- ☐ Outputs, Inputs, Segments, Metafile.
- ☐ Full Simulation for Linetypes, Linewidths, Fill Areas, Hatching.
- ☐ Circles and Arcs, Ellipses and Elliptic Arcs, Bezier Curves.
- ☐ Ports Available on all Versions of UNIX.
- ☐ **CEEGEN-GKS** is Ported to Gould, Masscomp, Plexus, Honeywell, Cadmus, Heurikon, Codata, NBI, NEC APCIII, IBM-AT, Silicon Graphics, Pyramid, Tadpole Technology, Apollo, AT&T 3B2, AT&T 6300, DEC VAX 11/750, 11/780 (4.2, 5.2), NCR Tower.
- ☐ **CEEGEN-GMS** GRAPHIC MODELING SYSTEM: An Interactive Object-Oriented Modeling Product for Developers of GKS Applications. **CEEGEN-GMS** and **GKS** Provide the Richest Development Environment Available on UNIX Systems.
- ☐ Extensive List of Peripheral Device Drivers Including Tektronix 4010, 4014, 4105, 4109, HPGL Plotters, Houston Instruments, Digitizers, Dot Matrix Printers and Graphics CRT Controllers.
- ☐ **END USER, OEM, DISTRIBUTOR DISCOUNTS AVAILABLE.**

CEEGEN

CEEGEN CORPORATION
20 S. Santa Cruz Avenue, Suite 102
Los Gatos, CA 95030
(408) 354-8841
TLX 287561 mlbx ur

EAST COAST:
John Redding & Associates
(617) 263-8206

UNITED KINGDOM:
Tadpole Technology PLC
044 (0223) 861112

UNIX is a trademark of Bell Labs.
CEEGEN-GKS is a trademark of Ceegen Corp.

Circle No. 6 on Inquiry Card

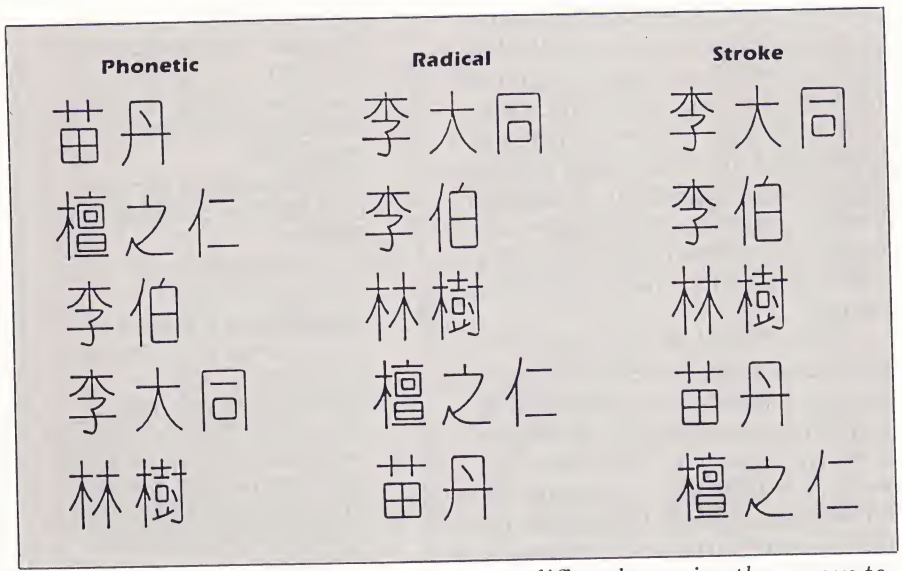


Figure 1 — *An example of how results can differ when using three ways to sort the same data.*

information about them so that each character can be sorted on multiple keys. Under this scenario, each key is weighted equally, with none designated as either primary or secondary. For instance, the user might want to sort Asian characters on a combination of the following keys: radical, number of strokes, phonetic, and country-code. In addition, user-defined collating sequences and special characters must be supported.

There still are problems this ordering does not address, however—such as those raised by languages which, unlike English, do not assume a left-to-right orientation of strings. UNIX sorting algorithms do not correctly collate Middle Eastern data that must be scanned from right to left. The problem is aggravated when Western data appears in the same file or field as Middle Eastern data. To complicate matters further, Middle Eastern data fields also must be delimited by right-to-left white space or field separators. The sorting algorithms of UNIX, though, expect data to be separated by ASCII white space. Asian space characters and field separators thus are not recognized as delimiters.

Regular Expressions. The capability to process regular expressions is an integral part of the UNIX programming environment. Among the many commands that support regular expression processing are **grep(1)**, **ed(1)**, **lex(1)**, **awk(1)**, **sed(1)**, and **sh(1)**.

Regular expressions use machine-collation when a character range is specified. But this might not produce correct results for eight-bit character sets because the range will not represent the language's correct collating sequence. This occurs if more than one language is represented by a single character set (for instance, the proposed ECMA standard). Depending on the hardware, sign-extension might take place when the eighth bit of a character is set to "1". What's more, the character-classification tables used by the regular-expression logic of UNIX are hard-coded for ASCII, and therefore are not large enough to support eight-bit character sets. As a result, problems can occur in code using the *include* file **regexp.h(7)**. This affects several system commands that process regular expressions.

Regular expressions are defined by single-byte characters under standard UNIX. Thus, when a two-byte Asian character is used in defining a pattern, the regular expression is compiled into something very different from what the user initially intended. For example, assume 李 is a Chinese character encoded in two bytes. If a regular expression for the **grep(1)** command is defined like so:

```
grep '^李+z$' Chinesefile
```

then the regular expression ought to match all strings starting with the two-byte Chinese character "李" followed by zero or more occurrences of this character and ending with "z". Unfortunately, the regular expression would compile into a pattern defined as starting with the first byte of a Chinese character, followed by zero or more occurrences of the value of the second byte, and then end in "z".

Edit, Word Process, and Display Utilities. Eight-bit data encounters obstacles throughout the UNIX system whenever it is formatted or edited for display purposes. There is no UNIX editing command that works correctly with non-ASCII one-byte—or two-byte—data. The reason is that in addition to editing, most of these commands collate, classify characters, and process regular expressions. We've already addressed some of the problems these actions suggest, but there are still others. One problem is caused by the escape sequences or shift characters used to switch from or to a "Math" or "Linedraw" character set. Many existing standards use mechanisms such as these to indicate a change

in character set, but the support of such "special characters" creates immense problems for the display and editing of data. Although escape sequences and shift characters are embedded transparently in text, they can be unintentionally corrupted. This easily could lead to display, processing, and printer problems.

One interesting problem arises when either Hebrew or Arabic is edited with a screen editor. When editing data that has a right-to-left orientation, the mapping between the cursor location on the screen and the cursor location in the editor buffer can be different. In addition, word wrapping does not work correctly when the editor's logic cannot accurately determine the location of the last word on a line.

The way in which UNIX preserves and records special information on how text should be formatted causes a major problem when the system is confronted with new character encoding schemes. Formatters can give special meaning to bytes by employing their own encoding schemes. For instance, **nroff(1)** and its macro packages encode each seven-bit character and its flags in 16 bits, where bits 1-7 encode the character and bits 8-16 are used as flags to contain information on things such as character font and size.

Another danger with preparing non-ASCII data for display is that the data might become corrupted and unprintable. For instance, the command **more(1)** can truncate a two-byte character at column 80 because of margin limitations in the logic; but the display logic never realizes that it is dealing with only half of a two-byte character when byte one is in column 80. Editing algorithms under UNIX can split an Asian character in two. There are many other commands that format data based on display widths, such as **pr(1)**, that experience similar problems.

USER INTERFACE and I/O

Hard-Coded Messages. The UNIX system and most software that runs under it use hard-coded messages in print statements that are compiled into object code. This poses a massive challenge for support and maintenance because, to localize such software, it is necessary to edit the source for message translation and subsequently support a different version of source and object for each language. Besides the additional effort, this increases the number of opportunities to introduce new bugs into code results. A copy of the source code is also necessary for message translation and modification. Traditionally, only binary licenses are provided for system software.

Context Analysis. Arabic is a cursive language in which a character can have up to four different shapes depending on where it falls within a word (first, medial, last, or in an isolated position). There is no current support under UNIX for the mapping of a character according to its shape. Neither is the process of selecting a proper shape for a character currently handled by the system's I/O drivers for display and printing.

Input/Output. UNIX terminal drivers function in both "raw" and "cooked" modes. In cooked mode, characters are saved in terminal driver buffers for input editing, and lines are made available when a carriage return or EOT signal is received. In raw mode, no input editing is done, and characters are made available to the program as they are typed. These distinctions can cause difficulties for UNIX in supporting languages and character sets.

INTERACTIVE
Offers Comprehensive Training for All UNIX Systems

No matter which UNIX system you are using, INTERACTIVE Systems has a training program to match your requirements. Our basic curriculum applies to all standard UNIX systems. Our supplementary courses describe the use of our products on IBM, DEC, WANG, AT&T, SUN, and SCI equipment. INTERACTIVE also offers training specifically for PC/IX, IX/370, and IN/ix users.

INTERACTIVE's curriculum is designed to meet the needs of all users — novice to expert. Our hands-on workshops and seminars cover the UNIX operating system and the C programming language, as well as INTERACTIVE's widely adopted proprietary packages such as INed, INmail, INnet, and TEN/PLUS.

Instruction is available at your site or at an INTERACTIVE Training Facility.

Our video-based training is an excellent tool to help meet your continuing education requirements.

INTERACTIVE's extensive line of UNIX-based products is currently available on:

- IBM PCs, XTs, and ATs
- IBM mainframes
- DEC systems
- AT&T 3B family
- SCI 1000 systems
- WANG systems

For more information, call the INTERACTIVE Training Department today.

INTERACTIVE
SYSTEMS CORPORATION
2401 Colorado Ave., 3rd Floor
Santa Monica, CA 90404
TWX 910-343-6255; Telex 18-2030
Telephone (213) 453-UNIX

IN/ix and TEN/PLUS are registered trademarks of INTERACTIVE Systems Corporation.
INed, INmail, and INnet are trademarks of INTERACTIVE Systems Corporation.
UNIX is a trademark of AT&T Bell Laboratories.
IBM is a registered trademark of International Business Machines Corporation.
DEC is a trademark of Digital Equipment Corporation.

Circle No. 7 on Inquiry Card

For one thing, a terminal driver can sometimes strip the eighth bit off every byte because it expects that bit to be used for parity checking. This is true of the 4.2BSD *tty* driver, for instance. The available encoding schemes are thus substantially limited.

There also is the complex problem of organizing large sets of Asian characters for input, output, and processing. A keyboard of 25,000 keys is, of course, unreasonable. There is no explicit way for a user to specify an Asian character through a keyboard, and, moreover, current terminal drivers do not provide the necessary interaction with the user.

Characters that are written and read from right to left have other problems to contend with as well. The general terminal interface expects a character at a time as it is typed. But this does not agree with the order of characters as they ultimately will be displayed. Some terminals deal with this by buffering strings a line at a time. When a linefeed is detected, the buffered characters are shipped out over the interface to the terminal driver in the order that they are displayed. But none of the drivers currently available under UNIX are able to support a general interface for the entry, storage, and display of Middle Eastern data.

IMPLEMENTATION

The magnitude of the problems involved in supporting a variety of character sets is so significant that solutions will not come easily. However, we have identified some mechanisms for achieving advances.

To begin with, an "announcement mechanism" is needed. This can be used to identify which character set, local customs, and language rules should govern the behavior of a user process. This will provide the user with the capability of dictating which language conventions to employ.

Editing algorithms under UNIX can split an Asian character in two.

One way to support processing requirements for a variety of languages is to develop mechanisms for extracting and utilizing information stored in a database. UNIX already uses this approach to support a large number of unique terminals through the terminal capabilities database, **termcap(5)**. Under this scheme, the software can be "extended" to provide a more general model of processing. The actual information used to support a particular language, though, is moved from the code itself into the file system where a new language can be configured, modified, or removed at will.

If the goal is to have a truly language-independent system, it also will be necessary to have a facility that enables local language strings to be "substituted" at runtime. To simplify this, a message catalog subsystem should provide tools for extracting hard-coded messages from source. Catalogs of translated messages should be easy to build and edit. In addition, a process should be able to identify the correct message catalog depending on the language of the user. Support for parameter substitution in messages at runtime is also needed.

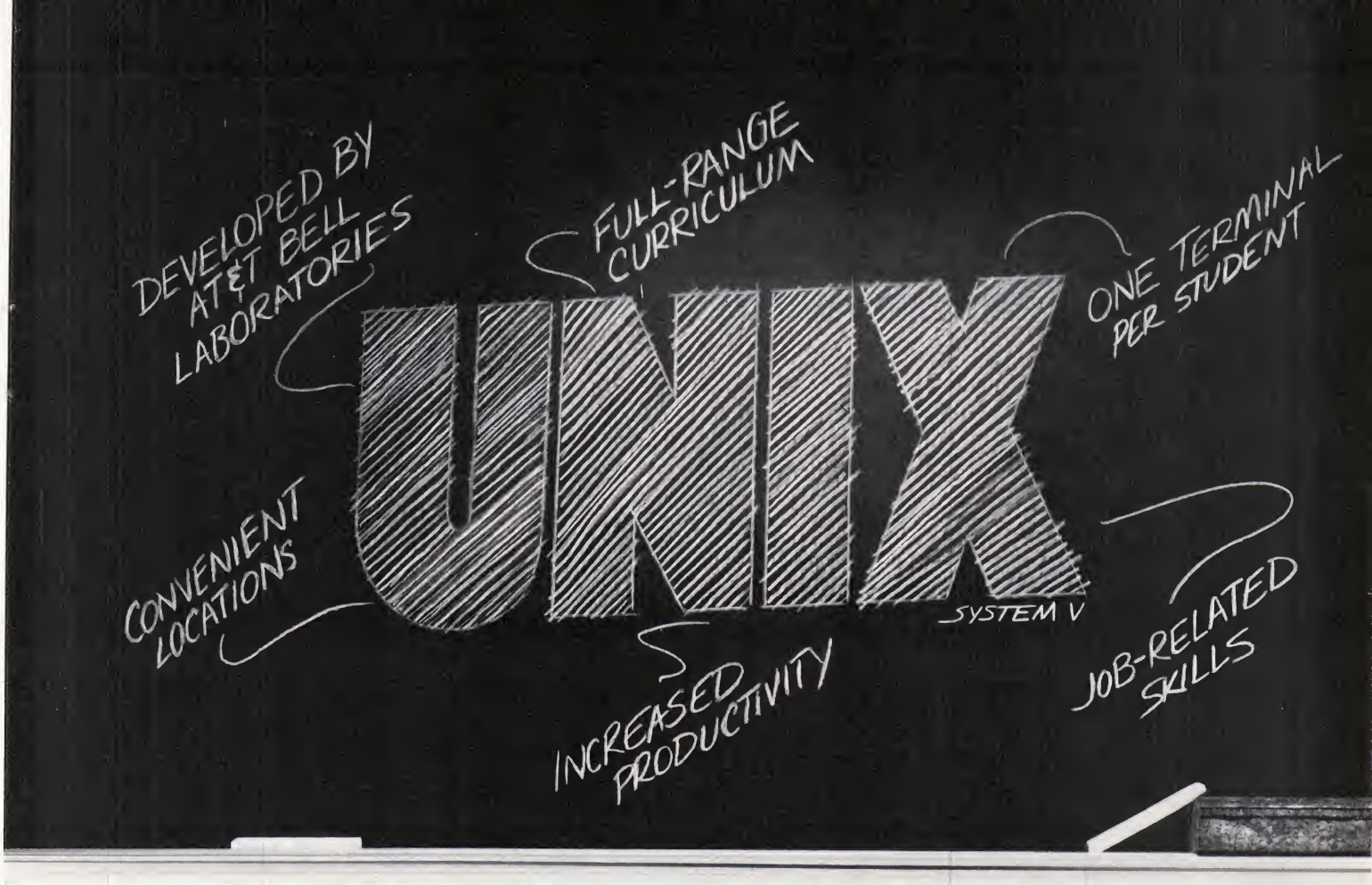
True language independence, moreover, will require that users have access to error or diagnostic messages written in their native language. The UNIX environment is built on top of system calls that involve direct entry into the kernel. Thus, if an error occurs during a system call, users should be informed about it in a comprehensible way. To rely on "errno" to flag an error is fine, but to report it through access to an external catalog of translated messages is risky. The implementation of a language-independent system largely involves the removal of all built-in language dependencies and the storage of this information in external files. Therefore, a method for supporting error-handling and error-reporting during the runtime access of these files is needed.

CONCLUSION

None of the problems mentioned in this article have easy solutions. Although many of the complications discussed here apply to all character encoding schemes, each method has its tradeoffs. In addition, any one encoding scheme can be implemented in many different ways. Therefore, if we are ever to have a portable solution to this challenge, companies and organizations within the computer industry will need to move together to work toward a standard we all can accept.

Karen Barnes is the Project Manager for HP-UX Native Language Support in Hewlett-Packard's Operating Systems Development Laboratory. Nearly half of her nine years with HP have been spent in the company's International Division.

Dan Epstein is a member of the technical staff in Hewlett-Packard's Operating Systems Development Laboratory. His work has helped shape HP's specifications for Middle Eastern and African computing environments. ■



HANDS-ON TRAINING THAT ISN'T SECONDHAND

When you learn the UNIX™ System directly from AT&T, you learn it from the people who develop it. So all the information you get is firsthand.

For over fifteen years, we've been teaching our people to use the UNIX System—which makes us the best trained to help you learn.

The best training starts at your own terminal. That's why, at AT&T each student gets the use of an individual terminal for real hands-on training.

Take your pick of courses from our extensive curriculum. Whatever your level of expertise, from first-time user to system developer, we have a course that will suit your individual needs. And all our courses are designed to teach you the specific skills that will soon have you using the UNIX System to organize and expand your computing system for maximum efficiency.

You also get experienced instructors, evening access to training facilities, and your choice of training centers. We can even bring our courses to your company and hold the training at your convenience.

And because we are continually expanding our courses to incorporate the developments of UNIX System V, you're assured of always getting the most up-to-date information.

So take your training from AT&T. And discover the power of UNIX System V—right from the source. **Call us today to reserve your seat or for a free catalog.**

1-800-247-1212, Ext. 418

Yes, I'd like some firsthand information on all UNIX System training courses.

Name

Title

Company

Address

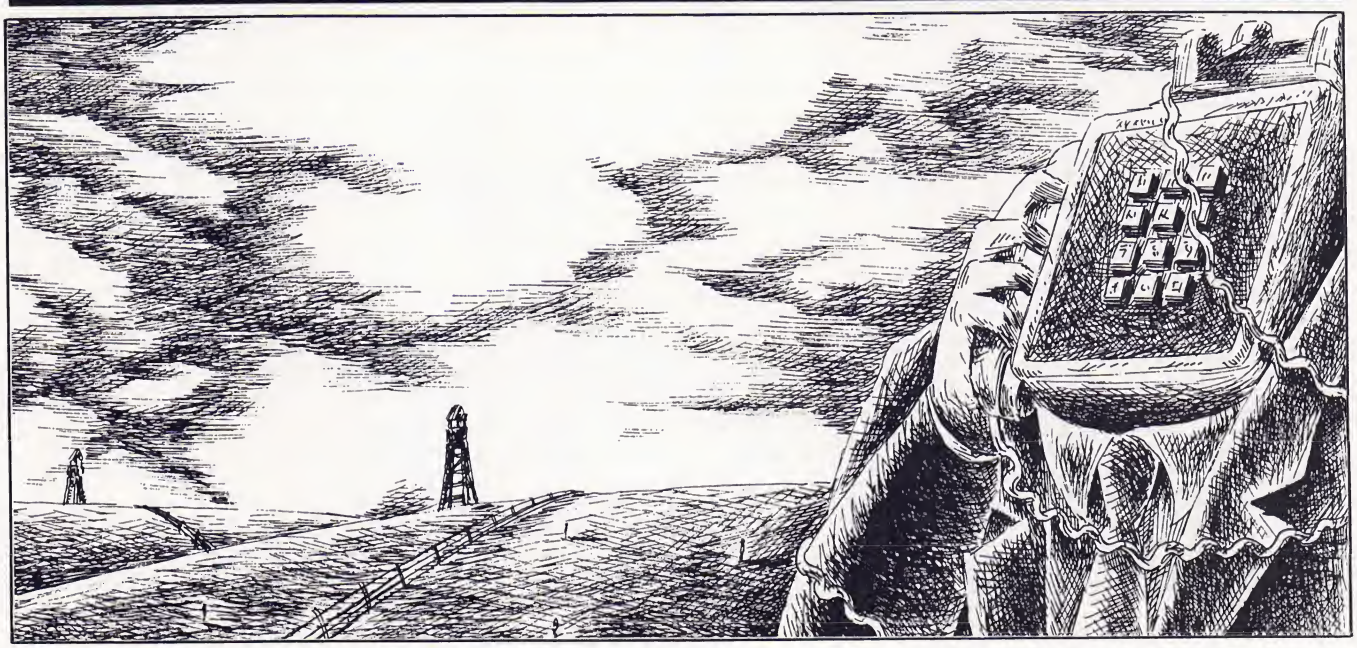
City State Zip

Call 1-800-247-1212, Ext. 418
or send coupon to:
AT&T Information Systems
P.O. Box 45038, Jacksonville, FL 32232-9974

AT&T
The right choice.

©1985 AT&T Information Systems.

BUREAUCRATIC BORDER SKIRMISHES



Hyon Kim

European barricades to international telecommunications

by Teus Hagen

Just as people from different European countries sometimes have difficulty communicating with each other, computers in Europe often have trouble talking across national boundaries. In the place of language differences, the machines encounter bureaucratic incompatibilities that are every bit as frustrating.

In large part, this owes to the fact that each country is served by a separate telecommunications company. Apart from sharing the common name "PTT" (for *Post, Telephone, and Telegraph*) and a similar bureaucratic style, these companies hold very little in common. While some use touch-tone dialing, others dial by way of pulses. Dial tones and busy signals also vary from country to country. And so it goes: on each and every technological front, there seem to be as many views as there are PTTs.

The permutations that surface can sometimes be amusing. An automated dialer in France, for instance, must by law be prepared to offer a proper metallic excuse like, "I'm sorry, but I'm the modem of a computer", if a person answers the call. Good manners also suggest that the dialer not wake up this person with daily 3 a.m. calls.

To complicate matters, dialers themselves vary from country to country. PTTs seem to agree on only one rule: no phone equipment can be connected to a public phone system until it first has gained official approval. A single violation of this dictum is all it takes to lose service for life—a threat that is taken most seriously since there are no competitors to turn to.

The testing of dialers checks for PTT conformance on both the hardware *and* software levels. Among other valuable tests, dialer hardware is examined for its ability to resist thousands of volts of input or output. Software, meanwhile, generally is required to meet horribly outmoded specifications. The PTTs nevertheless are quite serious about their tests. When they examine software, they test *all* of it. Thus, if a dialer happens to be contained in a UNIX system, all UNIX system software is subject to PTT review. In order to cope with such bureaucratic requirements, most sites have learned to provide their local PTTs with only the most modest of dialers operating only the sparest of software. In this way they not only avoid a grueling initial review, but also steer clear of the reviews that necessarily

would follow if the system ever were to be upgraded.

UNIX ENTERS THE FRAY

Frustrated by incompatibilities and hounded by regulations, European UNIX users took action following their first meeting under the banner of the European UNIX system User Group (EUUG). The linking of three sites in two countries immediately after that 1982 meeting in Paris was hailed as the beginning of the European UNIX systems Network (EUnet). Support from Digital Equipment Corporation and Philips Laboratories soon brought connections to the US. Within a year, the network had expanded to include 43 sites in nine countries. It now links over 500 sites in 16 West European countries and one East Bloc nation (Yugoslavia).

The EUnet has thrived because it offers access to strong, permanent links between backbone sites that serve each European country, allowing UNIX users to communicate via dependable channels with users served by other PTTs.

Although relying on X.25 links between the backbone sites, the basic carrier of all the network's data is UUCP. For the most part, this has been a satisfactory arrangement, but it hardly has been without its trials. In fact, much effort has been invested throughout the EUnet community in the repair of a number of the UUCP bugs that have raised serious cost concerns in Europe.

From the start of the network, most of the serious concerns have focused on transatlantic calls to the US. It didn't take long to discover that UUCP had a nasty tendency to keep phone lines open for hours while it requested the re-transmission of packages. In addition to killing UUCP performance, this threatened to be the death of more than a few employees.

Although the cost penalties produced by this problem admittedly were extreme, it by no means was the only circumstance where cost-per-byte concerns ran high. Communications in Europe—whether with other continents or between countries, cities, or neighbors—are expensive. It is largely for this reason that batching and compacting always have been popular (even electronic mail often is compacted). This is also why UNIX users have leaped at the opportunity to communicate internationally via UUCP. Even though it has been necessary to make modifications, UUCP, on the whole, is far less expensive than X.25 for low-volume communications over a general carrier.

PTTs seem to agree on only one rule: no phone equipment can be connected to a public phone system until it first has gained official approval.

The typical European price of an X.25 PAD (Packet Assembler/Disassembler), for example, is about $5000. Add to this an average monthly subscription fee of about $330 (which, of course, varies from country to country), and you can see why X.25 tends to surface only at sites that handle lots of international mail.

THE EUnet STORY

A look at how EUnet took shape and dealt with certain impediments should serve to illustrate just how steep the communications challenges faced by European users of UNIX really are. At the time the network was started, the only available carriers were public phone lines. Automatic dialers were illegal in many countries, so some national networks had to make secret connections using smuggled equipment. By law, only CCITT (V21 and V22) modems can be used over public phone lines, but some of these have proved to be incapable of working in environments where equipment from a variety of vendors are mixed.

Technical advances have aided in alleviating some of these compatibility problems, but at the time EUnet was formed, it was not uncommon to find modems from European manufacturers that could not connect with American modems. Even today, European modems that work fine are still priced much higher than their American counterparts because of additional costs related to CCITT regulations.

This is not to suggest that all the hurdles faced by EUnet have been hardware-related. There also have been serious financial and software concerns to address. In attacking these problems, the founders of EUnet have taken heed of the example of the Usenet network in the US. So as to avoid some of the load problems suffered by the Americans, it was decided that participation in the EUnet should *not* be free—one would have to be a member of the EUUG to qualify.

In order to gain better control over the flow of traffic, EUnet also adopted a more centralized approach than its American counterpart. The decision was made to route all calls from overseas through a central node called *mcvax*, located at the Centre for Mathematics and Computer Science in Amsterdam. Message routing from there—as well as central maintenance—is performed by a network manager.

At the next step in the network hierarchy, each backbone site coordinates and maintains communications among the various EUnet sites within its particular country. Backbone site managers look after the peculiar needs of their own countries, tending to software (screening, maintenance, and local development), hardware (PTT connections and maintenance), and administra-

tion (accounting, address contacts, routing information exchange, and various housekeeping tasks).

The current throughput of EUnet's international gateway is about 250 MB per month (about 60 percent mail and 40 percent news). Monthly transport costs of approximately $15,000 (exclusive of equipment and personnel) are shared by EUnet members according to use.

The "store and forward" nature of the network provides for convenient accounting. Backbone sites carry the cost for hardware, energy, and logistics, while all carrier costs external to EUnet are paid by those who actually send and receive data. Senders typically pay only for the transport of information to the nearest backbone site. From that point forward, the phone charges are billed to recipients. Likewise, calls received from the US are billed to those who receive the messages.

The basic charging strategy employed by EUnet is top-down: first, the international gateway bills the national backbone sites, which in turn bill all the subsites. This scheme promises to change somewhat over time as backbone sites become more closely connected to nodes in the US.

USE OF UUCP

Although UUCP has made EUnet possible, it has been necessary over time to modify the network somewhat. Among the changes that have been made are:

- Modifications to the software so that it can work with a variety of European dialers and phone numbering schemes.
- Mechanisms that keep cost/performance ratios dynamically so that data transmissions can be automatically terminated once they reach a specified limit.

Most American manufacturers, unaware of the subtleties of European problems, focus on the obvious language issues, to the detriment of the technology.

- Numerous UUCP bug fixes that improve the reliability of unmonitored calls. This is especially important because most data transmissions are made late at night to take advantage of lower rates.
- The integration of network accounting software that logs the origin and destination of calls automatically so that transmission costs can be charged back later.
- The inclusion of a special protocol for X.25 (the so-called F-protocol).
- The addition of a special package that allows X.25 PADs to be initialized with UUCP.
- The development of a package able to determine whether **pack** or **compact** has been used to compress a message. This program also determines the *version* of software used in the procedure.

Without these changes, it would be very difficult to run UUCP in Europe. Most European manufacturers understand this and thus have included EUUG's version of UUCP in their UNIX products. Most American manufacturers, however, do not appreciate the need for a modified UUCP. The reason should be familiar to those who know the European market: US manufacturers typically sell to Europe through third-party distributors. More often than not, these distributors are not abreast of UNIX technology—whether American or European. Whatever they do know usually has come by way of indoctrination from the US manufacturer. But most US manufacturers, unaware of the subtleties of the problem, focus on the obvious language issues, to the detriment of the technology.

What's more, the typical US manufacturer looking to expand into Europe will first look for a British sales point. The reason should be fairly obvious: American and English people think they speak the same language. Unfortunately, these companies then believe that they have succeeded in penetrating the European market. But, as Dutchman Piet Paaltjes once said, "All we have in common with England is some small water. All that we have in common with the States is about the same, except that the water is bigger."

USING X.25

When it comes to communications, the rule of thumb is that England is the European country most likely to take the initiative. It was there, for instance, that X.25 (or "PSS", as it is known in Britain) first was established. Being the first, it shouldn't be surprising that England's version of X.25 is somewhat different from what the rest of Europe ultimately adopted. In fact, there are variances between the X.25 implementations of almost all the countries. Happily, most of these differences are transparently re-

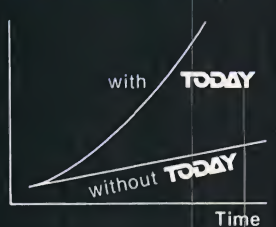
See Us At UniForum Booth 1042

TODAY

PRODUCTIVITY = PROFITABILITY

TODAY Application Generator is a true Fourth Generation Language. **TODAY** applications are immediately runable under MS DOS, Unix, Xenix, Ultrix, and VMS.

Let us explain how you too can obtain maximum productivity with **TODAY**.



Ask about our **TODAY** *Seminars*

San Francisco
Los Angeles
Dallas
Miami
Washington, D.C.
New York
Boston

bbj Computer Services, Inc. 2946 Scott Blvd. • Santa Clara, CA 95054 • (408) 727-4464

MS DOS, Xenix is a Trademark of Microsoft; Ultrix, VMS is a Trademark of DEC;
Unix is a Trademark of Bell Labs; TODAY is a Trademark of bbj Computer Services, Inc.

Circle No. 14 on Inquiry Card

solved by international PTT gateways.

International X.25 links were first made available in mid-1983. By the end of 1984, the quality of service afforded by most PTT gateways was actually pretty good. But, even yet, there are some countries—like Finland—that cannot be reached by way of X.25 connections.

To avoid the cost and delay that the checking and rechecking of UUCP can cause, most X.25 sites have resorted to PADs. But PADs require some basic CCITT protocol knowledge. For example, human interaction is required to prevent PADs from capturing some of the control characters used by UUCP. Other tricky questions for X.25 site managers revolve around the length of packages and the sending of messages. For instance, should a timeout be used to initiate the sending of a package, or should some other delimiter be employed?

Using X.25 also requires care in the use of UUCP protocols. The G-protocol will work, but it destroys performance on reliable connections. In order to facilitate those particular connections, the F-protocol was built. But complications such as these are reason enough to explore alternatives to UUCP.

Certainly, if anything is to be replaced, it will be UUCP and not X.25. The X.25 standard is an important international communications fixture. Cost savings alone ensure its survival. At present, the average transatlantic data transmission sent over traditional phone lines costs $.95 per kilobyte, while with X.25 this cost falls to $.15 per kilobyte.

Meanwhile, X.400 often has been proposed as a possible successor to UUCP. But attention must be paid to availability, which—at present—is not an X.400 strong suit. There are two X.400 implementations currently available. One that originated at the University of British Columbia is currently being evaluated for European use. The goal, of course, is to find a solution that is both dependable and inexpensive. Toward this end, an effort to implement X.400 under UNIX software should be considered. This certainly would be more economical than leaving each country to formulate solutions independently. At present, however, X.400 is more expensive than UUCP. If it remains so, UUCP and the various public domain UNIX packages that make use of it (**sendmail**, MMDF, and MH.5) can expect a long life in Europe.

> "All we have in common with England is some small water. All that we have in common with the States is about the same, except that the water is bigger."

On another front, the European Economic Community (EEC) is at work on UNIX-to-UNIX communications based on the International Standards Organization's Open Systems Interconnect (OSI) implementation. Most European manufacturers are partners in this effort. The package that ultimately is released will be one of the first implementations of OSI available under UNIX. Without question, this package will be significant, but it hardly will be a panacea. For one thing, numerous modifications in the UNIX kernel will be required to run implementations of the package. This, of course, will cause compatibility problems with systems not manufactured by European OSI partners. What's more, the software is unlikely to be cheap. This probably will put off veteran UNIX users who have come to appreciate the standard Berkeley fee.

ACADEMIC NETWORKS

No discussion of bureaucracy and European networking would be complete without a reference to "academic networks". These grew popular in every country in Europe a few years ago. Budgets were raised and plans were made to connect all the university computer centers with each other, but only on a nation-by-nation basis. Within this context, of course, each country considered only its own communications problems.

It is only now—by way of the EEC—that work finally is underway to combine the networks. The Réseau Académique de Recherche Européenne (RARE) is charged with accomplishing the merger. It actually wasn't until 1984, however, that any headway was made toward cooperation, and even then it was left to an American, Larry Landweber of CSnet, to get representatives of the various European network religions together at meetings in Paris (1984) and Stockholm (1985).

Meanwhile, on another front, a different network—European Academic Research Network (EARN)—has started to gain momentum. EARN connections are made via leased lines that crisscross Europe. Academics seem to find this approach attractive, doubtless (at least in part) because the costs for all the necessary lines and equipment are

carried by a Big Blue American-based computer company. Though it staggers the imagination, it is expected that users in academic computer centers will end up shipping data to neighboring computer science departments by way of the US. That's because computer centers will be able to ship their data free of charge to the US. A transatlantic call will then be made to EUnet, which will in turn relay the data on to the appropriate node. Guess who picks up the bill for this—the computer science department, of course.

Won't this change once the EARN network infrastructure is fully established and the EUnet has integrated some X.400 capabilities? The two, after all, will

American and English people think they speak the same language.

then share some European gateways. No matter—it appears as if data shipped from a computer center still will end up taking 80 trips around the world be fore making its way to a UNIX machine next door. One has to wonder, though, if change would not come quickly if EARN members suddenly were forced to shoulder the cost of their transmissions.

Teus Hagen is the Director of the European UNIX systems User Group and the Chairperson of the Netherlands UNIX systems User Group. He was the first person to install UNIX on a VAX 11/780 in Europe, and he also made Europe's first Ethernet connection. Until 1984, Mr. Hagen served as head of the Computer Laboratory in Amsterdam's Centre for Computer Science. He now works with ACE Associated Computer Experts in Amsterdam, and serves as a consultant to X/OPEN, ESPRIT, and various European companies producing UNIX products. ■

IBM XENIX DISK-TAPE-RAM

FOR THE IBM PC/AT—XENIX® OR DOS:

86 MEGABYTE HARD DISK - $2495
28ms average access, longest MTBF, 1 year warranty. Finest quality drives made.

60 MEGABYTE TAPE BACKUP - $1695
90 IPS, 5MB/minute cartridge tape. We ship the same unit IBM sells. Highest performance.

2 MEGABYTE RAM CARD - $745

120ns RAM, fully populated. Why pay more?

FIND OUT about the IBM Unix® Solution with 86MB Disk, 60MB tape, 2.5MB RAM and better than VAX 750® floating point for under $9500, quantity one price including all software. Why pay three times more for a slower machine?

Bell Technologies
415-794-5908 / PO Box 8323
Fremont, California 94537

Call today for quantity discounts.

Circle No. 12 on Inquiry Card

New from **Image Network!**

Documenter's Workbench®

for laserprinters and typesetters.

DWB is *troff, eqn, tbl,* and *pic* interfaced to raster printing devices.

Our existing **XROFF** product allows **DWB** to work with the following systems and printers:

- *System V*
- *Berkeley 4.2*
- *VAX/Ultrix*
- *IBM/PC MS/DOS*
- *Eunice*
- *UniPlus+*
- *System III*
- *V 7*
- *VAX/VMS*
- *Amdahl/UTS*
- *Xenix*
- *UNOS*
- *DEC LN01s, LN03*
- *APS–5 typesetter*
- *Compugraphic 8400*
- *Xerox 2700, 3700*
- *Xerox 8700, 9700*

Use **DWB** with a laser printer to make high quality documents or to make proof copies before typesetting.

Call or write to tell us *your* printing requirements!

Image Network, (415) 967-0542
448 Middlefield Road, Mountain View, CA 94043

*Documenter's Workbench is a trademark of AT&T Bell Laboratories.

Circle No. 13 on Inquiry Card

FACING UP TO INTERNATIONALIZATION

A layered approach to native language support

by G. L. Lindgren

Today, the UNIX system rides a wave of popularity in the US. This is also true in other parts of the world, where, despite the system's inadequate support for languages other than English, it quickly is becoming a standard. But before the UNIX system can be considered a *true* international standard, it will need to provide for the use of native languages and conventions.

Since the UNIX system was originally designed by software developers for their own use, it is not surprising that historically it has been popular in software engineering environments. Today, however, the UNIX system can be found in other walks of life as well. Much of the acceleration in its popularity can be attributed to a new class of users: *computer-literate knowledge workers* who are interested in computers strictly as a means to accomplish a task. This group is not the least bit interested in the sorts of programming languages that are supported, the types of shell procedures that are available, or, in fact, the number of software development tools that are included. Neither are they interested in learning a new syntax for each new task that they perform. No, these are users who want systems that conform to them because they don't have time to conform to systems.

This class of users, of course, has members who live outside the US. In a way, building an international version of the UNIX system is very similar to building a *friendly* user interface. Both efforts involve making the interaction between human and computer as simple, comfortable, and error-free as possible. The development of a system able to converse in the user's own language would certainly be one way to improve this interaction, both for new users and traditional users (software developers).

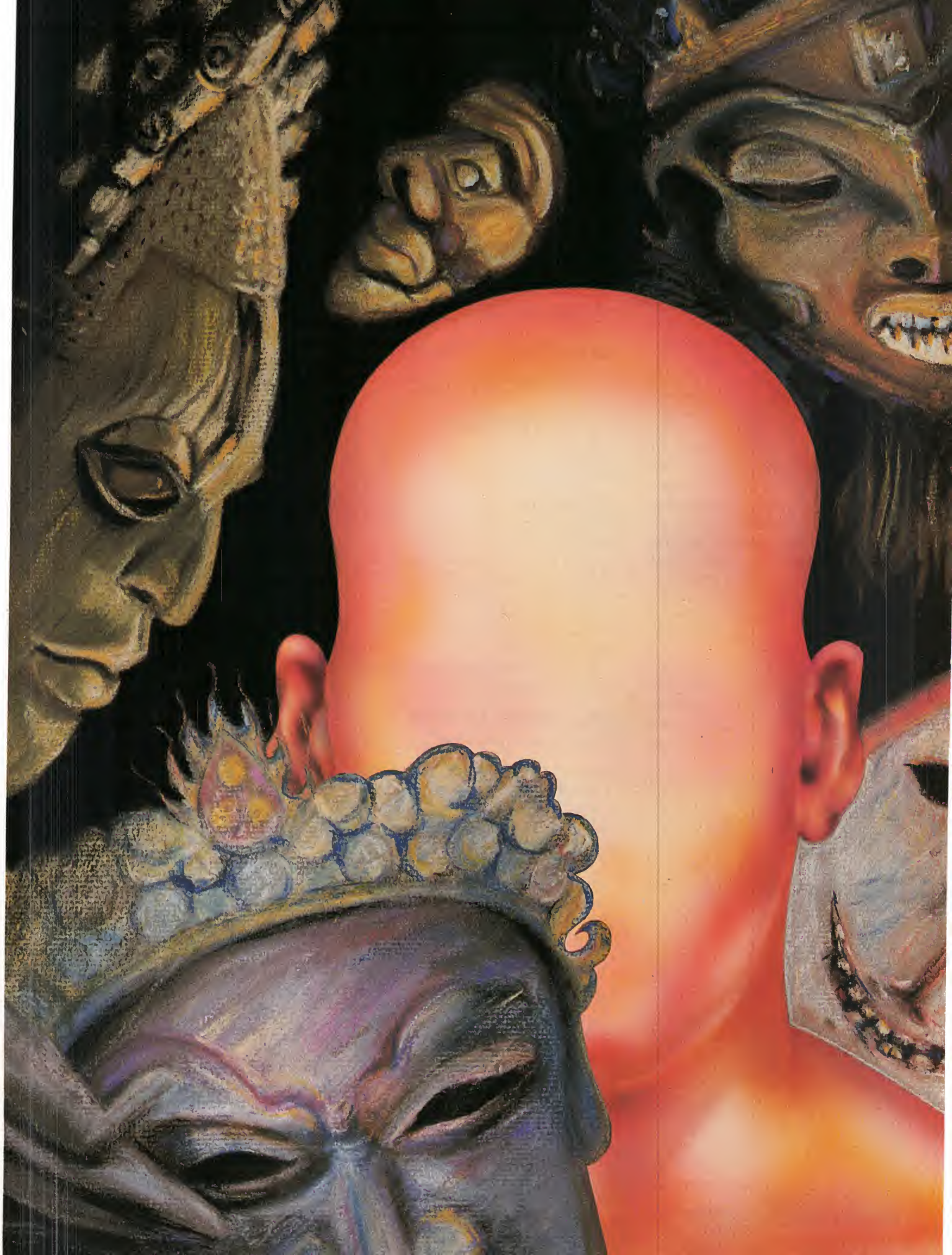
Because the UNIX system was developed in the United States, it *speaks* English (or at least *American* English) well and manages to support most of the conventions found in the US. But support for other languages and conventions does not exist. This article describes an architecture for rectifying the problem. The purpose here is to show how a new system design might be able to minimize development even while maximizing flexibility.

OBJECTIVES

What should the non-English speaking user expect from such an architecture? Among the list of features that should be supported are:

- The ability to use native language character sets.
- The ability to communicate with the UNIX system in the user's native language.
- The ability to converse in one or more native languages simultaneously.
- The ability to build *language-independent* applications on the architecture.
- Support of local conventions and habits (such as different date and time formats, sorting, and numeric representations, among others).
- The ability to adapt to new languages *without* recompiling software.
- A consistency among features that makes it possible for applications to share information.
- A level of performance compara-

Illustration by M. Kathryn Thompson



ble to that offered by current domestic releases of the system.

ARCHITECTURE

In truth, the architecture for an international version of the UNIX system should not differ much from the one the system has today. It may be advantageous, in fact, to provide the global features (pardon the pun) in a single version of the system so as to avoid duplication of work, incompatibilities, and so on. To accomplish this, the architecture of the UNIX system would need to be divided into two major components: a UNIX system base and a *national supplements layer*.

The first component simply would incorporate the traditional UNIX system product as we know it today. In addition, it would contain *hooks* for the international features included in the national supplements.

The national supplements layer, of course, would be an addition to the current UNIX system. There certainly is nothing sacred about the term itself. "National supplements" is merely the best designation I can think of to describe optional packages containing the extensions (such as tables, databases, and drivers) needed for the UNIX system to handle languages other than English and to support non-ASCII characters sets.

> In a way, building an international version of the UNIX system is very similar to building a *friendly* user interface.

A picture of this architecture is shown in Figure 1.

THE UNIX SYSTEM BASE

Changes must be made to the UNIX system base to provide new features, modify existing ones, and, in general, remove language-specific dependencies. To make this easier to describe, I've divided the discussion into four subcategories, namely:

- Eight-bit Cleanup.
- Codeset Support.
- Message Handling.
- Local Conventions.

The Eight-bit Cleanup. It should come as no surprise that the UNIX system is an ASCII-based operating system, meaning it was designed to work with the seven-bit ASCII codeset. (There are implementations of the UNIX system that use codesets other than ASCII (for example, IBM and Amdahl implementations), but for the purposes of this article, I will assume that the UNIX system is based on the ASCII codeset.) Since most computer hardware assigns eight bits to a byte, the UNIX system gives programmers an extra bit to fiddle with in their applications. Predictably, several UNIX system utilities also have taken advantage of this free bit to become more compact and efficient.

In order to support codesets other than ASCII, all eight bits of a byte are necessary. This means that all UNIX system utilities that use the most significant bit for internal purposes or that only allow for seven bits of information will have to be changed to support eight bits.

Character Set Support. An additional facility is necessary to support non-ASCII character sets. This will involve placing a structure on text files so that they can contain multiple character sets.

In addition, once a facility for supporting supplementary character sets has been provided, it

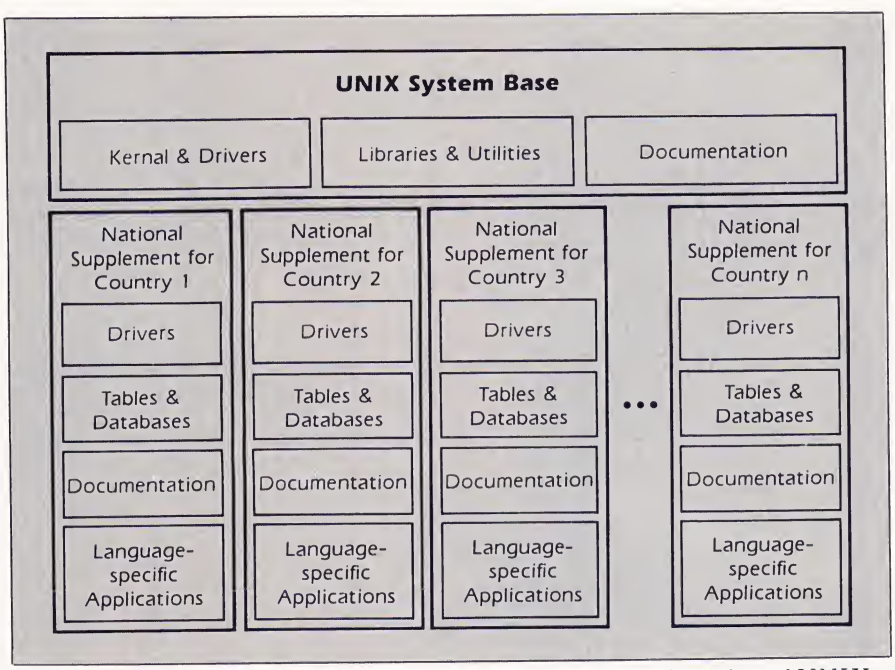


Figure 1 — *A proposed architecture for an international version of UNIX.*

also will be possible to include an additional layer of software to hide many of the details from software developers. This layer should provide for the widest character (yes, characters can be wider than eight bits!) and provide a means for determining what set a character comes from. Add to this a set of library routines that provides the same power as the C library and you have a nice, clean interface for handling multiple-byte character sets.

Message Handling. Probably one of the most visible parts of an internationalized UNIX system will be the multiple-language message handling facility it offers. This can take many forms, but the idea is to separate messages and text strings from the source code so that they later can be bound in the appropriate language. This is primarily so that error messages can be displayed in the native language, but it may also be used, for example, to print the name of the month in the native language whenever the **date** command is used. User prompting is also a concern since it's important that users be able to reply in their native language.

Local Conventions. The UNIX system will need to account for the common forms and rules applied to information communication in a variety of cultures. The aim of internationalization is to provide UNIX system applications and utilities capable of interacting with end users in a manner adapted to these conventions. At the same time, applications and utilities must be portable and easily adaptable to other conventions (which is to say that they must be shielded against the possibility of becoming hostage to any particular set of conventions). What's more, existing utilities and interfaces must be modified to support both implicit and explicit invocation of these conventions.

Among the areas that must be modified to provide for local adaptation are:

Collating Sequences. The ability to define one or more *collating sequences* for a specific codeset must be provided. Utilities that produce sorted output or require sorted input must be modified to allow the invocation of different collating sequences.

Character Classification. The ability to define character classes on a language-by-language basis must also be offered. In addition, new routines must be developed to support new capabilities (such as the ability to indicate which codeset a particular character comes from).

Date and Time Formats. The ability to enter and display date and time in native languages according to local format conventions is another requirement for internationalization.

Numeric Representation. The ability to define the rules for numeric editing (such as decimal and thousands delimiter) is essential.

Currency Representation. The ability to specify rules and formats for editing references to local currency is yet another international requirement.

NATIONAL SUPPLEMENTS

Language-specific components of the UNIX system, such as drivers, **help** databases, error messages, and online documentation, should be provided in national supplements for specific languages or territories. These national supplements could contain the necessary software to support a particular language and the various hardware peripherals found in the countries in question. All of this software could (and should) be created from tools provided in the UNIX system base.

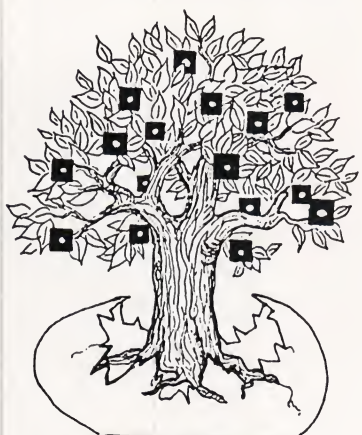
Language-specific applications could also be provided as part of the national supplements.

APPLICATIONS PACKAGING

Software developers will be able to provide *international* capabilities without significant software modifications if they build them according to the layered architecture described here. As with the system as a whole, applications can be manufactured and packaged with application-specific national supplements so that customers will be able to purchase just those supplements that they need to use with their applications. In this

Tree Shell

A Graphic Visual Shell for Unix/Xenix
End-Users and Experts Alike!



See Us At
UniForum
The International Conference of UNIX Users
February 4-7,1986
Anaheim Convention Center

COGITATE

"A Higher Form of Software"
24000 Telegraph Road
Southfield, MI 48034
(313) 352-2345
TELEX: 386581 COGITATE USA

Circle No. 8 on Inquiry Card

Introducing the HP 9000 Series 300

The computer that

Starting right now, HP is going to change your thinking on the ways that computers can change. Because now, there's a computer system so easy-to-configure that it meets today's application requirements quickly and cost-effectively, and so modular and expandable that it embraces future application needs as well. Whatever the job at hand — advanced CAD and measurement automation, or word processing, spread sheets, and database management — the new HP 9000 Series 300 is equal to the challenge.

Your pick of processing power.

The Series 300 offers you the appropriate processing power for the job, running your choice of two Motorola microprocessors: the 68010 16/32 bit and the 68020 32 bit. You can start with the 68010 and easily upgrade to the 68020 when more processing power is required. Just as important, you have complete object code compatibility across the product line. So when you change processors, there's no need to recompile.

Changing CPUs in the HP 9000 Series 300 is a snap. You simply plug in a new card set and, with object code compatibility, you shift from a 68010 running at 10 MHz to a 68020 running at 16.6 MHz.

Adding peripherals is easy.

The Series 300 has the built-in interfaces to handle HP's large, fully compatible family of peripherals. There are many compatible monitors of varying resolution, too, so you can go from 12-inch monochromatic display all the way to high-speed, high-resolution color graphics.

UNIX the operating system is a trademark of AT&T Bell Laboratories.
Ethernet TM is a trademark of Xerox Corporation.

loves changes

In addition, there are a number of HP peripherals for you to choose from: input and mass storage devices, plotters, printers, and more.

Productive programming language options.

You also have a complete set of programming language tools to work with, to help you better meet the needs of your application. For instance, the Series 300 runs HP BASIC, as well as HP-UX — HP's robust version of AT&T's System V UNIX™ operating system. And HP-UX supports industry standard programming languages, too — FORTRAN 77, Pascal, and C.

Link entire systems, not just users.

The Series 300 is designed to be linked with other systems. Your initial application may call for a simple, single-user system. But the Series 300 has what it takes to grow into a sophisticated 100-node LAN based on IEEE 802.3 or Ethernet™. With LAN, the Series 300 can share data with the Series 200 and 500 computers in the HP 9000 family, plus the popular HP 1000 and 3000 family.

Consistent HP quality.

With the HP Series 300, you can count on cost of maintenance below 4 percent, the result of exceptional HP product quality, uniformly maintained with exacting tests in temperature, shock, humidity, altitude, and many others. Couple this with our complete service and support package and you have still more reasons to go with HP.

Call us today!

Choose the system that will change to meet the application requirements of you, your users, and your customers today and tomorrow. Call your local HP sales office listed in the white pages. Or call 1-800-522-FAST (in Colorado, 223-9717 collect) for the number of the sales office nearest you.

Now, get data on-line, 24 hours a day!

For immediate information, use your computer and modem and dial 1-800-367-7646 (1200 baud, 7 bits even parity, 1 stop bit). In Colorado call 1-800-523-1724.

HEWLETT PACKARD

Circle No. 44 on Inquiry Card

way, the overall cost of an application can be kept low, and modest additional fees can be applied to each supplement.

Since the mechanisms to support international capabilities would be provided by the generic UNIX system base—rather than by the national supplements—applications would not need to be recompiled each time a national supplement is purchased. Similarly, applications would not be locked into specific national conventions. It should be possible for a customer to purchase an application for one language and then be able to use the application in a new language under a different supplement after only a simple installation process.

CONCLUSION

Building international capabilities into an operating system is not new; many computer companies have been doing it for years. In fact, native-language support is not even new to the UNIX system: there are many derivatives of the UNIX system worldwide that support native languages and supplementary codesets. But the focus of these implementations uniformly has been on solving particular problems rather than on addressing the general one.

The approach presented here has several advantages. For users, it can lead to products with standard facilities for supporting native languages—as well as the ability to provide a smooth migration path to new languages. For software developers, it represents a means for producing applications that have a much wider audience. This, in turn, should encourage the development of UNIX system software and propagate the system's use.

Gary Lindgren has been with AT&T for nine years, spending all but the last six months with Bell Laboratories. He now works with AT&T Information Systems. Mr. Lindgren's work has included a myriad of assignments, but most of these have had some contact with the UNIX system. His latest responsibilities include systems engineering for new features in UNIX System V, including work on the modifications necessary for internationalization. ■

EASIER THAN 1-2-3...

BUT DESIGNED FOR LARGER SYSTEMS

It's simple, C-CALC from DSD Corporation is more flexible, has more functions, and is easier to use than the best selling spreadsheet. We made it that way for a very simple reason, you'll get more work done and make better decisions in less time. That's what makes you successful whether you are planning for the future, forecasting trends, or analyzing profits.

The most popular spreadsheets require a great deal of time to get up and running. When we created C-CALC we kept in mind that time is your most important resource. Our On-Line Help facilities, prompts and menus allow even someone with minimal experience to see meaningful results in very little time. Our built-in training procedures let you pace your own learning with tutorial topics that range from basic to advanced. As you become more experienced, C-CALC allows you to bypass prompts and menus to save even more time.

So call DSD Corporation at (206) 822-2252. C-CALC is currently available for: **UNIX,** VMS, RSTS, RSX, IAS, P/OS, AOS, AOS/VS (Data General), IBM CSOS.

DSD CORPORATION
P.O. BOX 2669
KIRKLAND, WA 98033-0712

EFFECTIVE SOFTWARE FOR BUSINESS

C-CALC is a registered trademark of DSD Corporation. **UNIX** is a registered trademark of Bell Labs. P/OS, RSTS and RSX are registered trademarks of Digital Equipment Corporation. AOS and AOS/VS are registered trademarks of Data General Corporation.

Circle No. 9 on Inquiry Card

YOU CHOOSE:

	MLINK	CU/UUCP
Terminal Emulation Mode		
Menu-driven Interface	Yes	
Expert/brief Command Mode	Yes	Yes
Extensive Help Facility	Yes	
Directory-based Autodialing	Yes	
Automatic Logon	Yes	Yes
Programmable Function Keys	Yes	
Multiple Modem Support	Yes	Yes
File Transfer Mode		
Error Checking Protocol	Yes	Yes
Wildcard File Transfers	Yes	Yes
File Transfer Lists	Yes	Yes
XMODEM Protocol Support	Yes	
Compatible with Non-Unix Systems	Yes	
Command Language		
Conditional Instructions	Yes	
User Variables	Yes	
Labels	Yes	
Fast Interpreted Object Code	Yes	
Program Run	Yes	
Subroutines	Yes	
Arithmetic and String Instructions	Yes	
Debugger	Yes	
Miscellaneous		
Electronic Mail	Yes	Yes
Unattended Scheduling	Yes	Yes
Expandable Interface	Yes	
CP/M, MS/DOS Versions Available	Yes	

MLINK™

The choice is easy. Our MLINK Data Communications System is the most powerful and flexible telecommunications software you can buy for your Unix™ system. And it's easy to use. MLINK comes complete with all of the features listed above, a clear and comprehensive 275-page manual, and 21 applications scripts which show you how our unique script language satisfies the most demanding requirements.

Unix System V
Unix System III
Unix Version 7

BSD 4.2
Xenix
VM/CMS

MS-DOS
CP/M
and more...

Choose the best. Choose MLINK.

Altos
Arrete
AT&T
Compaq

Data General
DEC
Kaypro
Honeywell

IBM
Onyx
Plexus
and more...

MLINK is ideal for VARs and application builders. Please call or write for information.

Corporate Microsystems, Inc. P.O. Box 277, Etna, NH 03750 (603) 448-5193

MLINK is a trademark of Corporate Microsystems, Inc. Unix is a trademark of AT&T Bell Laboratories. IBM is a registered trademark of IBM Corp. MS-DOS and Xenix are trademarks of Microsoft Corp. CP/M is a registered trademark of Digital Research.

An interview
with
Jim Bell

FOREIGN EXCHANGE

It has been said that those who are unfamiliar with efforts to create an international version of UNIX cannot possibly imagine how hellish the problems are, and that those who are familiar with the problems can't bear to think about them anymore. Jim Bell has no choice. As the Group Engineering Manager for the Information Systems Group of Hewlett-Packard, he is responsible for coordinating all UNIX activities, including the internationalization of HP-UX.

From his management perch, Bell must consider more than technical hurdles alone. Marketing issues, legal considerations, political obstacles, and sociological factors also must be weighed.

This, though, is hardly the first formidable problem Bell has tackled. Prior to joining HP five years ago, he managed various engineering functions at DEC for 12 years, serving as the Corporate Director of Research from 1973 to 1980. He previously held positions with Bell Labs, IBM, SRI International, and Control Data Corporation.

So as to probe better into the world view shaped by such experience, UNIX REVIEW asked Jeff Schriebman, President of UniSoft Systems, to interview Bell. Schriebman himself is no stranger to the issue of internationalization; his company is currently under contract with AT&T to jointly develop an interface between Japanese Kanji applications and System V running on AT&T's 3B series of computers.

REVIEW: *Can you briefly summarize what Hewlett-Packard has done to accomplish UNIX internationalization? What fea-*

Photos by Randal S. Becker

tures have you found to be necessary?

BELL: For the present, we're providing full eight-bit transparency for support of both European languages and Katakana phonetic Japanese. This is being shipped in our current UNIX products. We're also actively engaged in work to produce message catalogues, file manipulations, and custom routines for things like local currency and date conventions. Another project that's underway is generating native language user documentation. Because of varying national standards in areas such as data communications, we also have been inspired to make UNIX a more open, modular system.

REVIEW: *That's a lot of topics—a lot of things to implement. How long have you been working on these projects?*

BELL: We've been working for several years. We already had a base of experience to build on because we've been providing foreign language support for our MPE commercial operating system for some time. Many of the ideas from that effort have carried over. In particular, we had something called the "Localization Cookbook", which software developers inside the company could use to make products suitable for localization. We believe that the systems software divisions and operating systems divisions need to put in hooks that can provide, once and for all, the infrastructure that will allow one to localize a given application easily. Then, within given foreign countries, we have "localization centers" that can make the necessary business decisions about which things to localize. These centers also can handle the actual localization efforts.

REVIEW: *How successful have*

As a general philosophy, we want to be compatible with standards in every case possible unless there's an extremely compelling reason not to be.

these "once and for all" hooks been? It's rather difficult to do things "once and for all", isn't it?

BELL: The efforts have been quite successful. It's actually easier to produce "once and for all" hooks than it is to repeat fundamental localization operations inside each application.

REVIEW: *As I understand it, your process is to develop a basic version of an operating system in the US, complete with appropriate hooks, and distribute this code to your international affiliates who then can tailor it for their customers' use.*

BELL: That's right. We feel that decisions weighing the needs of the local market are best made by people in the country in question. The hooks—the so-called native language support (NLS) hooks—are designed so that localization work can be done quite easily. In general, we make the hooks tabular, data-oriented, and highly encapsulated so that the people responsible for local modifications don't actually have to go through and make changes in the code itself.

REVIEW: *Do you work with standards organizations outside the US?*

BELL: Absolutely. We not only track standards closely, we also try to take an active posture in influencing their development. We've found that standards like GKS graphics and ISO networking are even more of a market factor internationally than they are domestically.

REVIEW: *Do you think it's possible to develop one UNIX standard for the entire world?*

BELL: I believe so, although some questions—such as support of large character sets—obviously have a much higher priority in

Asia than they do here.

REVIEW: *Do you think that 16-bit support is sufficient? It's not clear that it will be enough in China, for instance, where it may be necessary to offer 24 or 32-bit support.*

BELL: We feel that 16-bit support is fully adequate. We are in close touch with our affiliate in Taiwan. In fact, I suspect we may have a larger proportion of the total data processing market in Taiwan than we have in any other country in the world. We also have a joint venture going in the People's Republic of China. So we get quite a bit of feedback on our plans before executing them, and it's our impression that, for a surprising amount of the market, even 15-bit support is adequate—highly accepted, in fact—in the personal computer world. After all, a 35,000-character set suffices even for Asian languages, so 15 bits offer more than enough combinations to encode it.

REVIEW: *How does your international version of UNIX compare with the X/OPEN standard just recently released in Europe? [X/OPEN is a body made up of leading European manufacturers of UNIX systems—including Bull, Ericcson, ICL, Nixdorf, Olivetti, Philips, and Siemens—that is working to develop a common European interface to UNIX.] If you're not now compatible, will you become so in the near future?*

BELL: That's our plan. As a general philosophy, we want to be compatible with standards in every case possible unless there's an extremely compelling reason not to be. And if there are multiple standards, we may very well support them all. Even now, our base UNIX product is a superset of AT&T System V.2, but at the same time we support various other standards, including utilities developed at Berkeley.

REVIEW: *It is impossible to support multiple standards in some cases because standards conflict. How do you deal with these cases?*

BELL: We've actually found that those instances are surprisingly rare. There are strong reasons for supporting multiple standards. For example, if you consider networking, you find that there already is a set of standards that's widely used in the UNIX world and is generally expected by our UNIX-oriented customers. Although we support TCP/IP for levels three and four of the ISO model, HP customers also are accustomed to our own network services. So we support both, particularly to facilitate communications between our UNIX-based machines and our machines not running UNIX, including personal computers and larger business systems.

REVIEW: *How do you think the System V Interface Definition coming out from AT&T will interface with the X/OPEN group standard? How will they both interface with the Japanese standard recently proposed by a Japanese advisory group to AT&T? Do you think we can arrive eventually at a global user interface—a C-level interface, if you will? What do you see happening with this over the next few years?*

BELL: I think most people realize that it's in everyone's interest to have standards converge rather than diverge. Personally, I've been very pleased with some developments in that regard over the last year or two. For example, the plan to converge Xenix with System V is very helpful at the low end. The plan for a cooperative standard between Sun and AT&T is equally helpful at the high end. The IEEE standardization plans, which initially were seen as being in potential conflict with the /usr/group standards, have turned out instead to be mutually reinforcing. Given Europe's and Japan's goal of being able to import software easily, it clearly is in their interest—as well as everyone else's—to have a standard that's compatible with the US standards.

REVIEW: *What are the obstacles to compatibility? That is, how has the UNIX market in Europe and Japan differed from the UNIX market in the US?*

BELL: I think they're actually fairly similar. The three US markets that are disproportionately strong are universities, communications companies, and governmental agencies. In Europe and Japan, UNIX is also very popular in universities and in communications companies. An example of the significance of university installations may be seen by contrasting the strength of UNIX in the Netherlands, where university use is common, with its strength in Italy, where university use is less common. I think that governmental use may be a little less in Europe and Japan, although formal endorsements—by MITI [the Ministry of International Trade and Industry], for example—very well may be changing that in Japan.

There are other characteristics that differ somewhat. For example, we have found that on many of our machines the European user has tended to order smaller configurations. Europeans in general seem to be particularly price-sensitive. They also tend to place more stress on their systems by adding terminals until system performance degrades.

In the Japanese market the

technical user is more likely to be comfortable with English language versions of software than the commercial user is, so I think the ratio of technical UNIX machines to commercial UNIX machines is higher in Japan than it is in the United States.

In general, we've found that our UNIX-based machines are roughly as popular, relatively speaking, in Europe as they are in the United States. This is in contrast to the European market for many other computer products, which generally lags behind the US by one to two years. In other countries, we find that the market is less developed—with the notable exception of Japan.

REVIEW: *What about applications packages? How has the applications cottage industry, if you will, been developing in Japan and Europe?*

BELL: I think it has been developing more slowly than here, particularly in terms of indigenous applications. Most of the UNIX applications I've found are packages imported from the United States. The Japanese have had the particular tradition of providing software customized to the individual customer rather than designed for off-the-shelf sales. This has carried over into the UNIX world as well. We have found in the Japanese case that a key to increasing UNIX sales is education about UNIX and the applications available for it.

REVIEW: *Education of the end user?*

BELL: Yes, as well as education of intermediate users such as software houses. In both Japan and Europe, we have found that about half of all the calls to our systems engineers are educationally-oriented, pre-sale questions.

Actually, we've done something within Japan that we haven't done anywhere else in the world by establishing a UNIX demo and training center in the Tokyo area that offers courses to the general public. The courses are aimed at helping people understand UNIX products in general and ours in particular. The center also has a library with a comprehensive set of materials for self-study.

I think that both vendors and users recognize the importance of a standard and will embrace the AT&T standard as a base for achieving universal acceptance.

REVIEW: *Has that been well received?*

BELL: It's been extremely well received. We opened it in June of this year, and we've been fully booked ever since.

REVIEW: *How large is the Japanese market for US hardware? Moreover, what's essential for selling this hardware in Japan?*

BELL: It's difficult to answer that question in general terms. It certainly is the case that US companies face a number of obstacles in trying to sell goods of any type in Japan.

HP has a rather large joint venture in Japan, Yokagawa/Hewlett-Packard, which is in a relatively strong position in terms of sales of instruments. Hence, we have tried to use that as a base from which to build our computer sales. This conveniently dovetails with our strategy of emphasizing UNIX for the technical marketplace, so we actually have sold a rather large number of technical computers in Japan. The obstacles to selling larger, business-oriented machines in Japan are more significant, often making it valuable for US-based companies to work with Japanese partners in one way or another.

REVIEW: *Might those obstacles include things like long-term support and a company's past history?*

BELL: Yes. It's almost a cliche, but Japanese customers in gener-

al are more interested in a long-term relationship than a mere purchase. There also is the practical obstacle posed by the fact that many of the larger Japanese industrial groupings include computer companies, meaning that many of the companies will end up buying gear from a computer company that they're associated with. Additionally, there is the fact that Japanese companies naturally have been somewhat quicker than their American counterparts in providing good Japanese language support for UNIX and other products.

REVIEW: *The growth of UNIX in the US has been less than spectacular, especially during the last year. What sort of growth do you think UNIX will experience in the Japanese and European markets during the next few years?*

BELL: I believe there has been and will continue to be a steady growth in the UNIX market. Certainly, HP is seeing that now, particularly in terms of number of units sold. I think that the disappointment found in some quarters is relative to expectations that may not themselves have been realistic. There also is a snowball effect to take into account. I believe that as the applications base steadily grows, the market itself will grow naturally.

REVIEW: *I would expect that end user requirements differ in Europe, Japan, and the United States. Would you agree? And if so, can you cite the differences?*

BELL: I actually think the needs are quite similar in most cases. There are *some* differences, of course. For example, since many smaller vendors cannot offer good local support, foreign users place more emphasis on product quality. Even though HP offers worldwide support, we too have chosen to invest in quality as a competitive advantage. By writing and using a 600,000-line suite of tests and validation code, we have found and fixed over 1000 bugs that exist in most UNIX ports. Quality in the realms of documentation and training is also important internationally.

Another difference between US and international markets has to do with legal issues. For example, there's the problem of US export controls—in particular, the restrictions placed on **crypt**. Surprisingly, though, when we spoke with our domestic customers about the possibility of having a domestic version of UNIX including **crypt**, *and* an international version excluding it, we found very little interest. For a long time now, many of us have felt that security was going to become an urgent issue, but it appears to be an issue whose day has not yet come.

There is a problem with differing legal protections within countries, given AT&T's natural reluctance to see its intellectual property jeopardized in countries that provide fewer intellectual property rights than are found in the US. For example, Africa and parts of the Middle East—Kuwait, in particular—are areas of concern. AT&T's posture is that any company desiring to export to such areas must itself do the research necessary to prove that local copyright law is an adequate backup should the "shrinkwrap protections" not apply. However, AT&T then makes the final decision, based on information provided by would-be vendors.

We have been pleased with AT&T's rapid progress in the last year on administrative and legal matters. It is now more open to suggestions and more quickly responsive to requests from vendors. Requests that used to take several months to be considered are now turned around in weeks.

REVIEW: *Is the licensing of the UNIX operating system any different from licensing HP's own proprietary software in countries outside the US?*

BELL: Yes. We would be happy to be allowed to treat the licensing of UNIX with the same care that we treat the licensing of our own products. But we feel that AT&T is significantly more cautious in this regard than we would be. We have had much experience in international markets, and in general we feel that the rewards of licensing abroad far outweigh the risks.

In considering problems, however, it is important not only to look at the differences between countries, but also to remember the fact that the borders between them can cause additional logistical problems. For example, when we made our domestic upgrades from System III to System V, we asked our users to return their System III-based copies. However, in the case of our international customers, that simply wasn't feasible because the return of the earlier version would, of course, mean dealing with customs agents, who are notoriously unsophisticated about software matters. We ended up letting our customers keep copies of both System III and System V so long as they were used only on the same licensed machine.

A few years ago, a friend was having trouble carrying a tape full of software past a border guard, but resolved the problem by saying to the guard, "Don't worry about it, the tape's been used already."

REVIEW: *A number of times, requests have been made of AT&T to unbundle UNIX—to break it down into smaller run-time packages and reduce the*

licensing fees accordingly. In the international arena, would this also help sell more UNIX products?

BELL: I think flexibility is helpful in general, but I don't think this is a key obstacle to the acceptance of UNIX.

REVIEW: *What are the key obstacles?*

BELL: I think that two central issues are standards and applications. And, in fact, these are not really separate issues at all because standards provide a key for porting applications.

The frequently noted lack of user friendliness remains an issue, albeit one where steady progress is being made. A final obstacle is the "UNIX is only a toy" sales pitch from companies with a weak or nonexistent UNIX offering. This obstacle will vaporize as UNIX begins to make significant inroads into the mainframe market.

REVIEW: *Do you think AT&T will set the standards or merely guide them?*

BELL: I think the whole UNIX community looks to AT&T for leadership. But a number of other companies, universities, user groups, and individuals can also play roles, including leadership roles.

REVIEW: *Are other companies going to accept what AT&T dictates, or are they going to wish to have a voice in setting the standard? And if so, how will they best be able to convey their views?*

BELL: I think that both vendors and users recognize the importance of a standard and will embrace the AT&T standard as a base for achieving universal acceptance. The most difficult issue here is that standards bodies—even AT&T itself—may not be able to move fast enough to keep up with customer requirements.

We have very clearly specified our own priorities. First, we want to allow easy *importability* of applications, which means using the AT&T UNIX standard as well as appropriate official or de facto standards in other areas, such as languages and communications. Our second goal, which in no way is inconsistent with the first, is to provide for compatibility among our own machines. Our third goal is to add value by offering capabilities that go beyond what is currently embodied in AT&T's implementations or in any external standard.

We recognize that there's a risk in the latter goal. For example, in the case of real-time extensions to UNIX, there's always the danger that future AT&T or external standards may differ somewhat from what we have done. But the alternative of waiting until a standard emerges is unacceptable. We are anxious to link our UNIX-based machines more tightly with our factory floor and instrument control applications. We simply can't afford to wait.

REVIEW: *Besides real time, are there other issues on which developers would like AT&T to move more quickly?*

BELL: Yes. I think graphics is one area that should be moving faster.

Circle No. 11 on Inquiry Card

Database management is another. Of course, native language support must also be offered.

REVIEW: *How is support of an international version of UNIX different from support of a domestic one? Are the differences mostly logistical?*

BELL: Certainly there are logistical differences. If you think about the time differences, for example, you can see that it would be highly inconvenient for an Oriental or European customer to rely on the working day overlap with a US-only company.

REVIEW: *What are the advertising and promotion differences? What problems have you encountered in Europe and Japan?*

BELL: In the case of both Europe and the Orient, it is not possible to have individual advertisements that cover the whole market. Advertising and product emphasis often vary between countries. For example, UNIX, MS-DOS, and our own MPE operating system each play an important role for us in European countries and the US. But in Japan, we've chosen to emphasize UNIX strongly.

REVIEW: *In general, is your emphasis on UNIX greater outside the US than within it?*

BELL: Our base strategy is fundamentally the same worldwide. That is, we emphasize UNIX for engineering, software development, and technical applications. And we're also in the process of greatly increasing our emphasis on manufacturing and real-time UNIX applications. A bit further down the line, we expect to promote more use of UNIX on personal machines and in small business applications. The area where we are not emphasizing UNIX is in the realm of medium to larger business machines.

REVIEW: *UNIX frequently has been proposed as the lingua franca of computing. What are your views on that?*

BELL: UNIX certainly is pervasive. One of the reasons that I'm confident of the continued growth and increased success of UNIX is that it currently has no effective competition in certain markets, such as the high-end workstation market and the multiuser personal machine market. Additionally, UNIX has particular advantages for certain classes of vendors. For example, some of the mainframe makers who have been trapped in narrow, private markets can use UNIX to break free. High-volume, small machine-oriented vendors, such as the Japanese, can use UNIX as a lever for breaking into the American market. Small, new hardware companies can use UNIX to rapidly provide a software base that would be impossible to provide by proprietary means.

In the final analysis, however, the success of UNIX will be based on the advantages it gives to the customer rather than the vendor. No previous operating environment has spanned so wide a variety of markets and machine sizes. Indeed, it is this breadth of applicability and utility that augurs well for an increasingly important role for UNIX in computer markets throughout the world. ■

Break through the PC/UNIX* barrier!

At last. A product that guarantees clear passage from IBM PC's and PC-Compatibles to UNIX — without unnecessary roadblocks.

DaTapaSS is quick and simple. No clutter, no confusion: Just a direct path to the features you need most. Features like terminal emulation. Full access to DOS functions. Error-free uploading and downloading across public networks, LAN's, or phone lines. Automatic restart and recovery of interrupted transfers. Key-selectable signon sequences. Softkeys you can program from DOS or UNIX to automate repeat transactions. It even lets you talk to other PC's . . . and Honeywell mainframes, too.

Interested? Give us a call for more information about DaTapaSS: your best route to error-free PC-to-UNIX communications.

DTSS Incorporated
Buck Road • Box 70 • Hanover, NH 03755 • 603 643-6600
A Subsidiary of Metropolitan Life Insurance Company

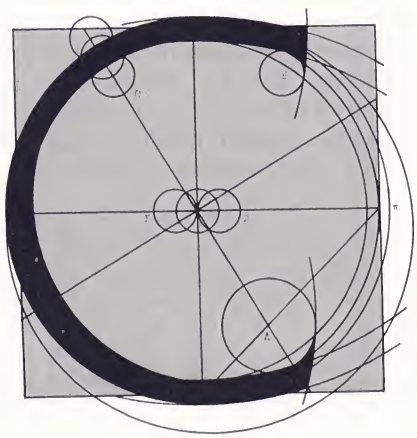
* UNIX is a trademark of AT&T Bell Laboratories. IBM is a trademark of International Business Machines Corporation.

Circle No. 10 on Inquiry Card

C ADVISOR

Debugging with **adb**

by Bill Tuthill



In many ways, effective debugging is as critical as intelligent programming. The UNIX operating system has achieved a certain stability in the marketplace, making debugging skills all the more important as consolidation of various versions of the system proceeds. Application software also demands debugging since there is at least one program for every known application, and at least one bug for every page of program source. Some bugs are merely a nuisance; others prevent programs from working altogether.

Debugging isn't glamorous. The joy of creating something original is largely absent. Afterwards, there's no new program to show your users. Nor can you produce pages of code to impress your manager. All you can say is that the software you've tended to works better than it did before.

To my knowledge, no university offers a course on debugging, and no textbook exists that purports to teach debugging. This is too bad, because debugging requires much skill and programmers would benefit greatly from training on the subject.

Even after Fred Brooks counseled against it in *The Mythical Man Month*, novice programmers have continued to be assigned debugging chores while more experienced programmers have been allowed to write new code. It is best if programmers maintain the code they write, but this does not usually happen because talented programmers like to move on to new challenges.

On the other hand, some programmers actually enjoy debugging. Programmers who are good at the task usually fall into three categories: 1) those with good intuition and a grasp of the "big picture"; 2) those with great patience and attention to detail; and 3) those with a good debugger. It is hard to find programmers in the first two groups. And, unfortunately, the standard UNIX debugger, **adb**, does not qualify its users for the third group. [It should be mentioned, though, that other debuggers *are* available under UNIX—**sdb**, **dbx**, and **cdb** in particular.]

All too often programmers use **printf()** statements instead of employing a debugger. This is a slow method, because code must be recompiled at every step. Intelligent use of a good debugger can yield better production. This article is the first of a series that describes the UNIX debuggers.

The most widely propagated UNIX debugger is **adb**, which first appeared on Version 7 and has been on every major UNIX release since. One reason why UNIX programmers use **printf()** statements instead of a debugger is that **adb** is so limited. It may have the worst user interface of any UNIX program. Furthermore, it is not symbolic, so you can't display C source code as you debug. Better debuggers are provided on other systems, including VMS and MS-DOS. This is embarrassing for an operating system that is supposed to be the best software development environment available.

Like compilers, debuggers are not portable. Since both deal with machine instructions and subroutine calling sequences, they have to be changed when the UNIX system is moved to a new processor or even when a different implementation is used on the same processor. Consequently, **adb** is not the same under every implementation. The examples here were taken from an MC68000-based machine; you may see slightly different results on different

UltraCalc®

PEOPLE WHO DO HANDS-ON COMPARISON OF ELECTRONIC SPREADSHEETS CHOOSE ULTRACALC HANDS DOWN

As you may have discovered by now, choosing a software package by its list of features or by the claims made by its manufacturer can be very disappointing. The best way to judge software is to use it. The next best way is to talk to people who have used it. Good software gets a reputation for being good.

UltraCalc is an electronic spreadsheet of unusual quality. It's reputation is getting around. We don't expect you to take our word for it. We want you to judge its features for yourself. That's why we're offering free evaluation copies of UltraCalc on selected machines.

UltraCalc is a virtual electronic spreadsheet program. You won't run out of memory even on limited memory machines, and it's FAST! If you compare, UltraCalc will outshine the competition. Lots of OEM's and end-users are turning to UltraCalc for its speed, versatility, and ease of use, including IBM and AT&T.

UltraCalc was written specifically for UNIX, so it takes full advantage of UNIX's capabilities. It also runs on XENIX/UNIX 8086 and 80286 machines. Not only does UltraCalc run, but it runs circles around the competition. In spreadsheet size, recalculation speed, and ease of use, UltraCalc is the clear winner.

UltraCalc is earning its reputation. We know that sonner or later you will hear about it. If you want to hear about it sooner than later, call or write us and we'll arrange for a hands-on comparision.

Some of UltraCalc's advanced features:

- Virtually unlimited spreadsheet size.
- Full, multi-level, context sensitive, cross-referenced, on-line help.
- Macro command files.
- Report generating print facility.
- Auto-linking of spreadsheets.
- Automatic consolidation facilities.
- Multiple key sorting on rows OR columns.
- Escape to UNIX without leaving UltraCalc.
- Importing/exporting of data, even through pipes.
- Color/business graphic character support.
- Vertical and horizontal bar charts.
- Command recall.
- Named cells and regions, use of names in commands and formulas.

UltraCalc is available from Olympus Software on the following machines:

IBM PC/AT & PC/XT (XENIX, PC/IX)
Altos, 486, 586, 986 (XENIX 3.0 & 2.3)
Plexus P60 & P35
NCR Tower
AT&T 3B2 & 3B5 (Sys V Rev 2)
Sun Microsystems (Sunwindows)
All UNIPLUS+ systems
Intel 310
Motorola VME10
All XENIX/286 3.0 machines
Tandy 6000 & Model 16
Fortune 32:16
Zilog S8000 (all models)
DEC PRO 350 & 380 (VENIX)
DEC VAX (all models, ULTRIX, BSD 4.2)
SCI 1000 & 2000
Integrated Solutions (all models)
Pyramid
Convergent Technologies

UNIX is a trademark of Bell Laboratories.
XENIX is a trademark of Microsoft.

SOFTWARE, INC.

1733 South 1100 East • Salt Lake City, Utah 84105
(801) 487-4534

Circle No. 60 on Inquiry Card

Available NOW from IBM as Interactive Executive UltraCalc.
Also available from AT&T as UltraCalc.

It is best if programmers maintain the code they write, but this does not usually happen because talented programmers like to move on to new challenges.

machines. Not all features of **adb** work on every processor.

THE adb DEBUGGER

Most of the time, programmers use **adb** to find out why a program dumped core (stack backtrace). To ensure valuable output, it's first necessary to check that the program hasn't been stripped of its symbol tables. If it has, few of **adb**'s features will work. Invoke the debugger as follows, where *program* is the pathname of the executable file that dumped core:

```
$ adb program core
```

Also, consider the program listed in Figure 1,

```
#include <stdio.h>
#define LIMIT 5
main()        /* print message and die */
{
        int i;
        for (i = 1; i <= 10 ; i++) {
                printf("Goodbye world!\n");
                dumpcore(i);
        }
        exit(0);
}
dumpcore(lim)        /* de-reference NULL pointer */
int lim;
{
        int *ip;
        if (lim >= LIMIT) {
                ip = NULL;
                *ip = lim;
        }
}
```

Figure 1 — *A program that de-references a NULL pointer.*

which de-references (references through) a NULL pointer. This is a common (but illegal) operation on VAX/UNIX, but causes a core dump on MC68000-based UNIX systems.

On many machines, an assignment to address zero will cause a core dump due to a segmentation violation or memory fault. Here's how you could find out why the program died:

```
$ adb
core file = core / program = a.out
memory fault
$c
_dumpcore[80b8](5) + 26
_main[8074](1,fffd84,fffd8c) + 2e
$C
_dumpcore[80b8](5) + 26
        ip:     0
_main[8074](1,fffd84,fffd8c) + 2e
        i:      5
```

The request *$c* yields a C stack trace, while *$C* yields a stack trace and also prints the value of all local variables. Other useful requests are *$r* to print the contents of all registers, *$e* to print the value of external variables, and *$m* to print out the memory maps. Note that the values of the local variables *ip* and *i* are just what we would expect—0 and 5. You can print the values of local variables in active procedures (ones that actually are located on the stack) by typing the procedure name, a period, and then the variable name, followed by a slash:

```
main.i/
fffd68:         5     =     orb   #0,d0
dumpcore.ip/
fffd58:         0     =     ???
```

The value of *ip* in the **dumpcore()** procedure is suspicious because it doesn't point to anything. The three question marks are an indication that something is amiss. If you are an assembly language buff, you can see the assembler instructions at the beginning of *main()* by typing:

```
main,5?i
_main:
_main:          link    a6,#0
                addl    #-4,a7
                moveml  #(),sp@
                movl    #1,a6@(-4)
                cmpl    #a,a6@(-4)
```

Now you'll probably want to edit the program. To get out of **adb**, type CTRL-D or use the *$q* request. Since

Some adb Format Letters	
Letter	*Description*
c	one byte as a character
o	one short word in octal
d	one short word in decimal
x	one short word in hexadecimal
O	one long word in octal
D	one long word in decimal
X	one long word in hexadecimal
f	single-precision floating point
F	double-precision floating point
i	machine instruction
s	a null terminated character string
a	the value of dot (the address)
n	print a newline
t	print a tab
^	decrement dot (not really a format)

Figure 2 — *A table of formats for the* ***adb*** *debugger.*

adb traps signals, you can't interrupt out of it.

SYNTAX SUMMARY

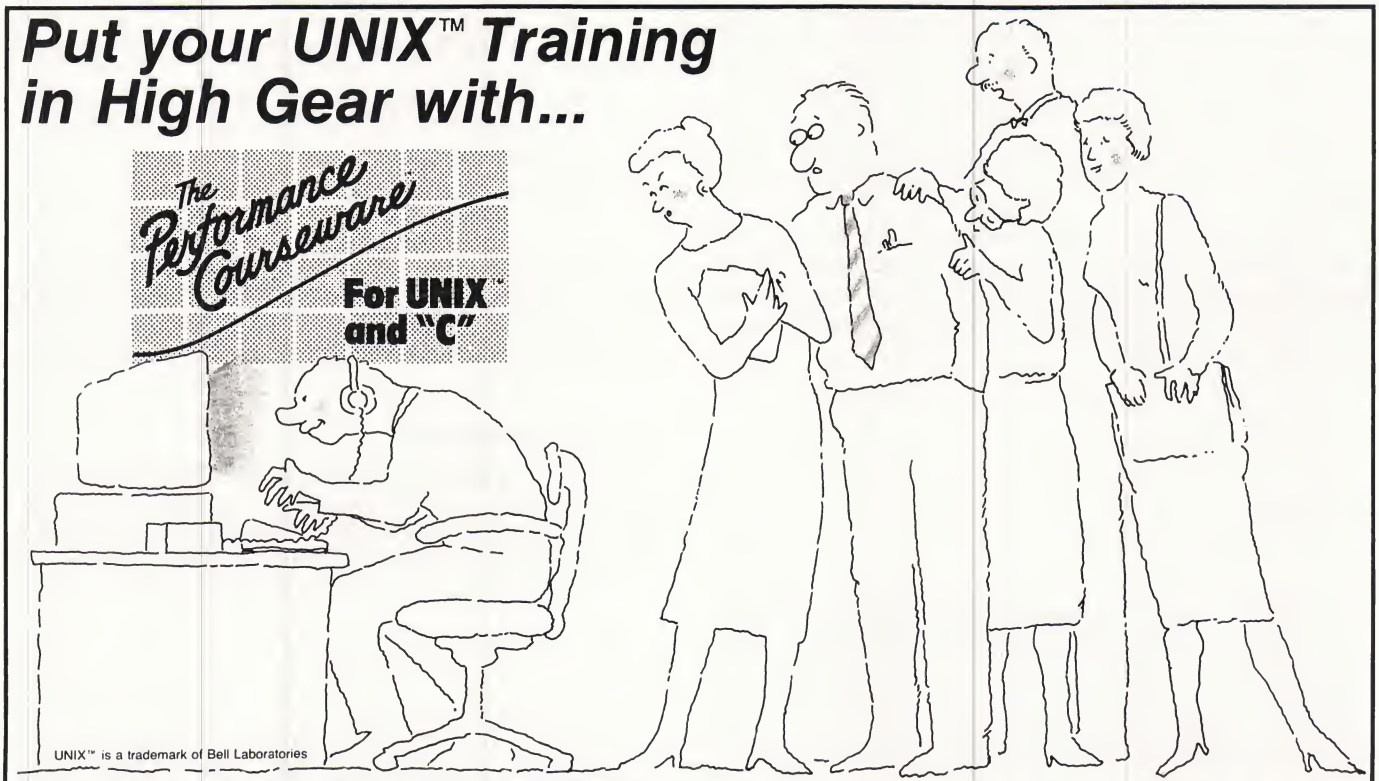
You can examine locations in an executable file with the *?* request, or locations in a core file with the */* request. These requests take the form:

```
address ? format
address / format
```

The address may be a number or a symbol. The current address, called *dot*, is set when you specify an explicit address, and can be advanced by pressing RETURN. A table of formats is given in Figure 2. Your system remembers these formats, so once you give an address and format, RETURN advances through memory in the same format. Note that capital letters indicate added length, as in the difference between *short* word and *long* word.

Requests are different from formats because they cause **adb** to react, rather than simply to print data.

Put your UNIX™ Training in High Gear with...



Learning that gets your attention.

USER TRAINING CORPORATION

(408) 370-9710

See us at UniForum Booth # 1317

130-B Knowles Drive • Los Gatos, CA 95030

Circle No. 15 on Inquiry Card

The **adb** dubugger may have the worst user interface of any UNIX program.

The general form of a request is:

```
address,count command modifier
```

This sets dot to *address* and executes *command count* times. Figure 3 lists the meaning of various **adb** commands. The useful commands presented there—*$c* for a stack trace, *$r* for the registers, and *$e* for the externals—are all considered miscellaneous requests.

SETTING BREAKPOINTS

Many programmers are intimidated by the **adb** documentation for breakpoints, but it isn't hard to learn and is well worth the effort. The main problem is that **adb** can set breakpoints only at the subroutine level—but not at the statement level (this *is* possible, however, with the 4.2BSD debugger **dbx**.)

When you invoke **adb**, give a dash as the second argument to indicate that the core file should be ignored. (On some systems a second argument is not necessary.) This will let you run the program under the control of **adb**:

```
$ adb a.out -
dumpcore+4:b
$b
breakpoints
count   bkpt        command
1       _dumpcore+4
```

Some* adb *Commands

Command	*Description*
?	print contents from *a.out* file
/	print contents from *core* file
=	print value of dot
:	breakpoint control
$	miscellaneous requests
:	request separator
!	escape to shell

Figure 3 — *The meaning of various* ***adb*** *commands.*

On an MC68000, set the breakpoint at the subroutine plus 4 (the first instruction sets up the stack frame pointer) and then list the breakpoints with the *$b* request. To run the program, enter *:r*. To continue the program after the breakpoint, enter *:c*. Do this five times, printing the variable *i* to make sure it works:

```
:r
Goodbye world!
breakpoint     _dumpcore+4:     addl  #-4,a7
main.i/
fffd68:        1            =   orb   #0,d0
:c
Goodbye world!
breakpoint     _dumpcore+4:     addl  #-4,a7
main.i/
fffd68:        2            =   orb   #0,d0
```

When the value of *i* reaches 5, the program will have a memory fault. This is because the NULL pointer is de-referenced only when *lim* becomes 5 or greater:

```
main.i/
fffd68:        5            =   orb   #0,d0
:c
memory fault
stopped at     _dumpcore+26:    moveml  a6@(-4),#()
```

Now you know exactly how the program got to the point where it core-dumped. If single-stepping proceeds too slowly, you can remove breakpoints with the *:d* request, which has the same syntax as *:b*.

SOME ANOMALIES

Like **ed**, **adb** issues no prompt. When it doesn't understand a request, it types its name back at you. For example, if you forget to put the slash after a variable name, you'll see something like this:

```
main.i
adb
```

This is the same response you'll get if you try to interrupt it. You can print external variables simply by giving their name before the slash. Local variables, however, must be preceded by their function name. If you forget to do this, you'll see the following message:

```
i/
symbol not found
```

Many programmers are intimidated by the **adb** documentation for breakpoints, but it isn't hard to learn and is well worth the effort.

If everything fails, you are probably trying to debug an executable file that has been stripped of its symbol tables. Make sure the **file** command reports that it is "executable not stripped" before starting to worry. If the file has been stripped, recompile the program and try to duplicate the bug that caused it to dump core in the first place. If you cannot recompile the program, you're out of luck.

It is possible to use **adb** to examine arbitrary binary files, to disassemble object code, and to patch binary files. The first task can also be done with **od**, but **adb** provides lots of control over the output format. You can even write **adb** scripts for non-interactive use. The patching of binary files is outside the scope of this article, but can be of great use for binary UNIX licensees.

Next month, we will discuss a more powerful debugger, available on 4.2BSD—**dbx**. It is a symbolic debugger, so you can display code as you're debugging. You can also set breakpoints at source statement boundaries, which is extremely useful.

Bill Tuthill was a leading UNIX and C consultant at UC Berkeley for four years prior to becoming a member of the technical staff at Sun Microsystems. He enjoys a solid reputation in the UNIX community earned as part of the Berkeley team that enhanced Version 7 (4.0, 4.1, and 4.2BSD). ■

Q-CALC

A superior spreadsheet on UNIX*

As powerful as Lotus 1–2–3*

- large spreadsheet
- many business functions
- complete GRAPHICS package
- translates 1–2–3 models into Q–CALC
- already ported to: VAX, Callan, Fortune, 3B2, Cyb, Plexus, Codata, Cadmus, Masscomp, Sun, etc.
- Ideal for VARs/ISVs

Available since Jan. '84
For more information write/call
Quality software Products
348 S. Clark Drive
Beverly Hills, CA 90211
213–659–1560

*Lotus 1–2–3 is a trademark of Lotus Development Corp. UNIX is a trademark of AT&T.

Circle No. 16 on Inquiry Card

68000 68010 **68020**
SOFTWARE TOOLS

WE ARE PROUD TO ANNOUNCE THE BIRTH OF THE NEWEST MEMBERS OF OUR 68000 FAMILY . . . YOUR 68020 TOOLS ARE HERE!

TOOL KIT

- 68000/10/20 Assembler Package:
  - Macro Cross/Native Assembler
  - Linker and Librarian
  - Cross Reference Facility
  - Symbol Formatter Utility
  - Object Module Translator
- Green Hills C 68000/10/20 Optimizing Compilers
- Symbolic Debuggers

AVAILABILITY

VAX, microVAX, 8600, Sun, Pyramid, Masscomp, IBM/PC, OASYS Attached Processors for VAX and PC, others. Runs under VMS, Bsd 4.2, System V, MS/DOS, dozens more.

You name it . . .

We provide a "One-Stop Shopping" service for more than 100 products running on, and/or targeting to, the most popular 32-, 16- and 8-bit micros and operating systems.

FEATURES

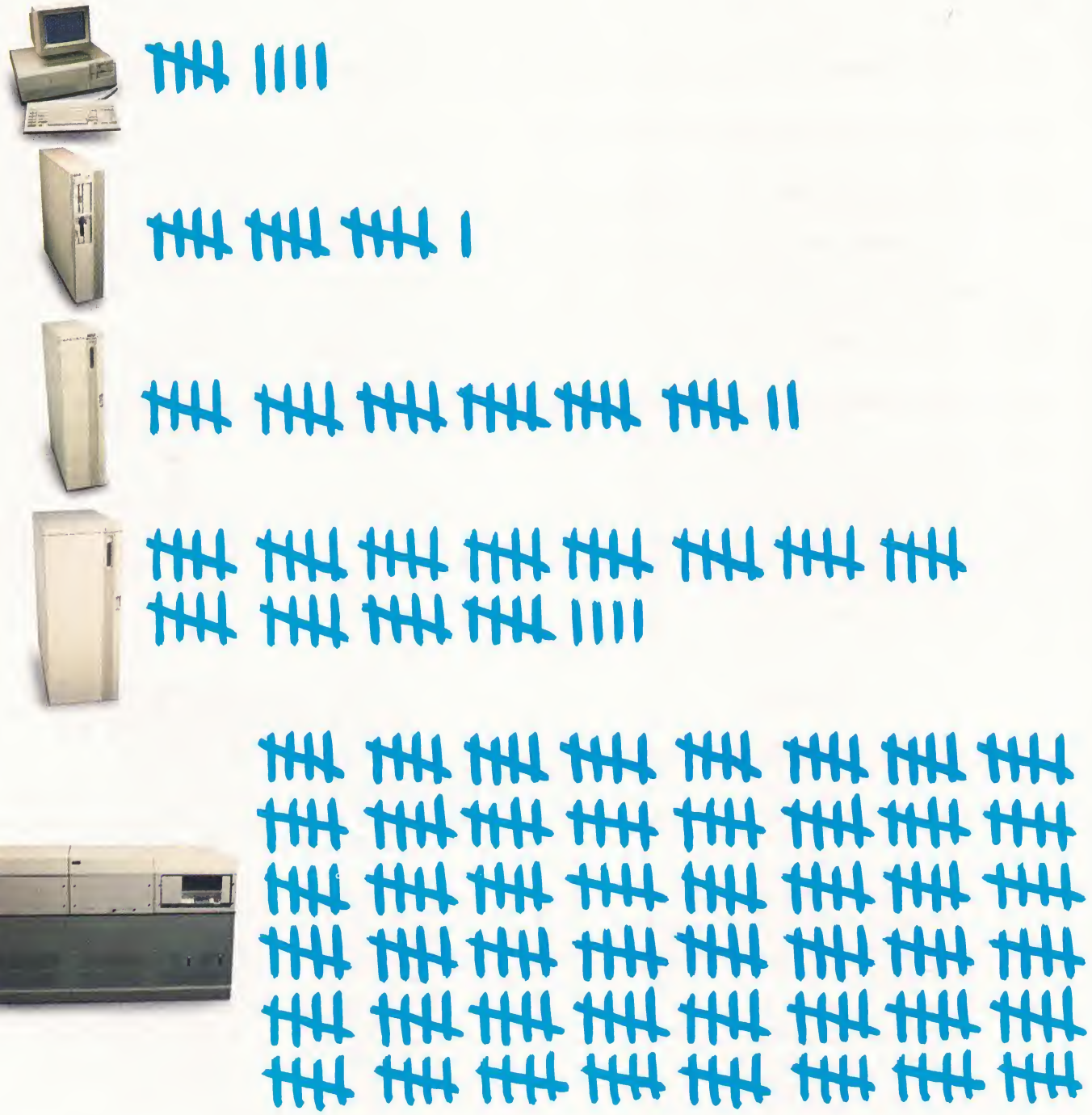
- Written in C; fast, accurate, portable.
- Supports 68000 and 68010.
- 5,000 line test suite included.
- EXORmacs compatible.
- Produces full listings and maps.
- Outputs S-records and Tek-Hex formats.
- Runs native or cross. Extensive libraries.
- Supports OASYS compilers.
- Generates PROMable output and PIC.
- Full Floating Point support.

Over 100 Other OASYS software tools to choose from.

A DIVISION OF XEL **oasys**

60 Aberdeen Avenue, Cambridge, MA 02138 **(617) 491-4180**

Circle No. 17 on Inquiry Card



One company. One OS. One to 240 users.

A singular statement. With multiple benefits for you.

Software transportability. Vertically, as well as horizontally.

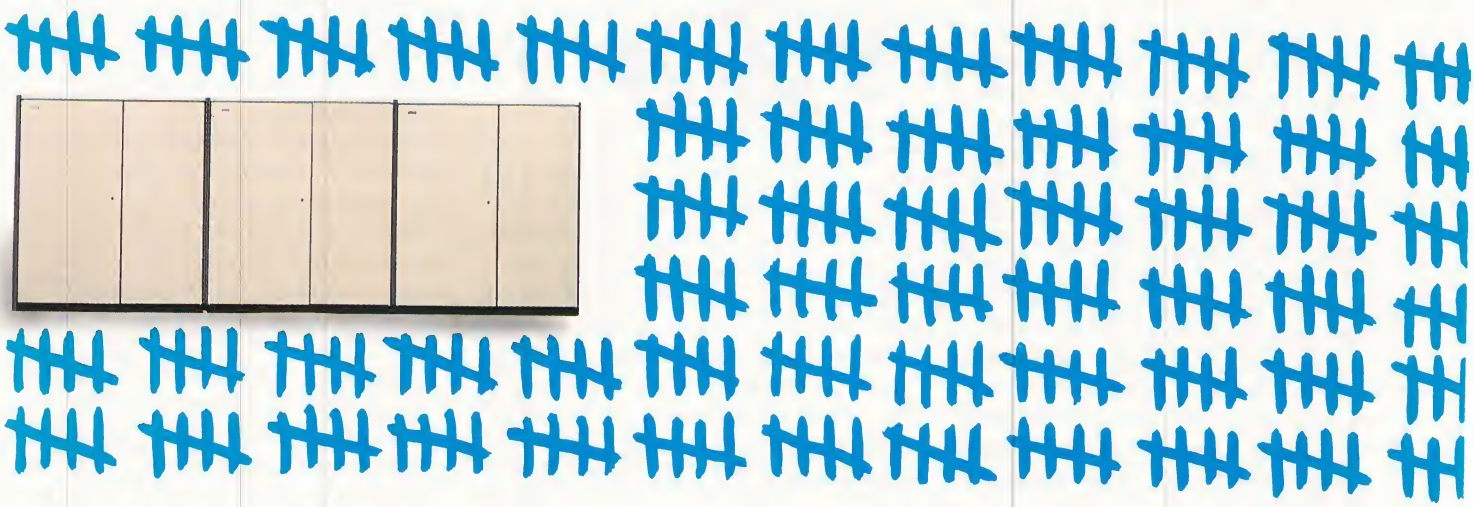
Open channels of information, communication and the creative energy that results when many users can share the same ideas, address the same problems.

A growth path that is impossible to outgrow. And therefore protects your investment, by enhancing it.

And the security of dealing with a single supplier who understands more about what makes your business special than any collection of suppliers could ever hope to.

Because Sperry is the world's largest supplier of hardware running the UNIX™ O/S, it's the only company that can offer you all the benefits of the UNIX Operating System in such a range of superior products, including:

Desktop PCs.
Microcomputers.
Minicomputers.
Superminis.



And up.

And Mainframes.

A single call will bring you a *free* Sperry Information Kit on our Micro-to-Mainframe line featuring the UNIX™ Operating System.

For your copy, or to arrange for a demonstration at a Sperry Productivity Center near you, telephone toll-free: **1-800-547-8362, ext. 61.** Or write Sperry Corporation, P.O. Box 500, Blue Bell, PA 19424-0024.

Resellers: Call Sperry at 1-800-547-8362, ext. 125 to carry the only complete UNIX O/S micro-to-mainframe line.

SPERRY

Circle No. 63 on Inquiry Card

*UNIX is a trademark of AT&T Bell Laboratories.

©Sperry Corporation 1985

RULES OF THE GAME

Law and motion

by Glenn Groenewold

By now nearly everyone who reads the business pages must be aware that Steven Jobs, co-founder of Apple Computer, severed his connection with that company this past September. Most of these readers probably also know that Apple quickly responded by filing a lawsuit against its former leader, claiming that by planning and starting a new enterprise, Jobs had violated his obligations to the company and thus had harmed it.

It's too soon, of course, to know what will come of this controversy. Possibly the entire matter will end up being settled quietly. If so, that would be that. But if this doesn't happen, the dispute has the potential to generate a landmark court decision—one which could have a major impact on the future of the computing industry.

Jobs, after all, can be considered the archetypical computer entrepreneur. He started a business with Stephen Wozniak in a garage in 1976, and then presided over the endeavor's growth into a Fortune 500 company. In the process he not only became a rich man, but, as much as anyone, helped bring the computer out of the domain of giant corporations and into America's schools and homes.

Though the Jobs and Wozniak enterprise was perhaps the most spectacular of the Cinderella stories emerging from what came to be referred to generically as Silicon Valley, Apple was scarcely alone in its success. These days the observation is made with increasing frequency that none of this achievement would have been possible in the absence of the free-wheeling atmosphere that prevailed in the computing field during the '70s. It was a time when hardware and software were viewed as things to be used, improved upon, and then left behind as dazzling leaps were taken into superior technology.

This was all happening too fast for anyone to pay much attention to legal niceties such as licensing requirements and ownership of derivative creations. Who really cared about protecting a proprietary interest in something that soon would be obsolete? The result was a field day for hackers.

A new industry typified by such rapid technological innovation at the hands of a large number of gifted—and generally quite young—individuals is probably without parallel in our history. Though the infant motion picture industry had its rough-and-tumble beginning, film technology stabilized relatively quickly, and control over theaters and escalating costs of film production soon limited access for new participants.

By contrast, computer technology continues to advance at dizzying speed, and no one can predict where it might take us. Moreover, unlike other 20th Century industries founded on new technologies—automobile and aircraft production, for instance—computing has continued to provide nearly limitless opportunities for perceptive individuals and small concerns. Requirements of scale and financing precluded this sort of individual entrepreneurship in other fields.

But is the creativity that has characterized the computing industry now in danger of being stifled by the application of legal concepts that tend to favor established enterprises? Questions of this sort are being asked more and more often. Steve Jobs himself has been quoted as saying, "With five other people I want to go start a company. . .and they

NEW VERSION

Come to TERM with your Unix/Xenix communications problems.

TERM - More Powerful. Easier To Use.

Compare These Special Features:

- Easy to remember mnemonic commands
- Online user's manual for instant help
- Menu driven interface
- Fast - 9600 baud file transfers
- Self installing
- Powerful scripting language with variables
- Wildcard file send/receive capability
- Automatic error-checking and re-transmission
- Xon/Xoff, Etx/Ack, Line and character protocols for communications with non-TERM systems
- Xmodem protocol for remote bulletin boards
- Full/half duplex emulation modes
- Automatic login and logout
- Auto-dial, auto-redial, answer and hangup modem support
- Unlimited phone number directory for auto-dialing
- Unattended file transfers
- Remote maintenance capability
- Sample scripts included
- MS-DOS and CP/M versions available

TERM - Powerful Communications.

TERM - Unix/Xenix's most powerful communications program. TERM Communications Software provides a full-featured, programmable communications tool under the Unix/Xenix environment.

You'll appreciate TERM's ease of use, compatibility with a wide user base, and ability to talk to most other systems. TERM is both a smart terminal and file transfer program. It has extras you won't find in other Unix communications programs: On-line HELP, character translation, efficient error-checking protocols and file transfers for text and binary data.

TERM provides full modem control, an extensive script language, auto-login and logout functions, and can be run unattended for remote maintenance.

TERM is available **NOW** on the Altos 586, 2086, IBM AT, Tandy Model 16, 6000, AT&T 3B2, IBM PC/XT, and many others. Find out how easy it is to get your Unix, Xenix and MSDOS machines all talking together.

$ 295.00

TERM

COMMUNICATIONS SOFTWARE

Call or write for more information.

CENTURY SOFTWARE

9558 South Pinedale
Salt Lake City, Utah 84092
(801)943-8386

VISA / MC

Unix is a trademark of AT&T Bell Laboratories. MS-DOS and Xenix are trademarks of Microsoft Corp. CP/M is a registered trademark of Digital Research Inc.

Circle No. 56 on Inquiry Card

won't let me. If this hadn't happened before, how could there ever have been a Silicon Valley?"

How indeed? And yet there have to be rules of the game for the computer industry, as for any other. Its future may depend on whatever the courts decide these are to be.

THE "CASE" AGAINST STEVEN JOBS

Regular readers of this column should have little difficulty understanding the grounds on which Apple's suit against its former *wunderkind* is based. To begin with, Jobs, like any employee, owed the company a *fiduciary* duty. But Jobs was, of course, more than an ordinary employee. He had been Apple's vice-president until he was pushed out of that position last May. Even afterwards he remained chairman of the company's board of directors. In these positions, the law dictates that he had a greater responsibility to serve the interests of the company than even a *key* employee would have.

In whichever capacity, Jobs certainly could have been expected to acquire intimate knowledge of Apple's operations—at least through last April. It would be surprising if this knowledge did not encompass many, if not all, of Apple's trade secrets. In addition, it seems reasonable to assume that Jobs would have become familiar with a great deal of proprietary information which, though lacking the exalted status of a trade secret, nevertheless remains valuable to Apple in maintaining its competitive position.

Apple's suit against its former chairman was triggered by the disclosure that he intended to launch a new company oriented toward the university market, and that five of Apple's employees would be joining him in the venture. As might be expected, each side has its own version of the facts beyond these bare details.

Apple contends that the former employees who will be associated with Jobs' new company, Next, Inc., held key technical, financial, and marketing positions. Specifically, Apple's suit names Richard A. Page, who while employed at Apple allegedly worked on the type of technology that it's speculated Next intends to utilize.

> Who really cared about protecting a proprietary interest in something that soon would be obsolete?

In its lawsuit, Apple asks that Jobs and the other five former employees be prevented from using the assertedly confidential information they acquired while at Apple, that they be precluded from hiring any more of Apple's employees, and that they be barred from competing with their former company.

For his part, Jobs denies that Next intends to use Apple's technology or to enter into direct competition with it. Whether the employees who left to join him were key employees or not "depends on what your definition of 'key' is", he stated. And Jobs insists that Next will not be doing anything similar to the project Page was working on at Apple, noting that protection of Apple's trade secret prevents him from discussing the matter further.

WHAT'S AT ISSUE

The Apple-Steve Jobs controversy brings to a head the question of how much use departing employees can make of knowledge and contacts acquired during employment. We discussed these problems in some detail in "What can you take when you clean out your desk?" in October, 1984. There, it was indicated that the trend is for the courts to protect employee mobility when possible. It also was pointed out that blanket prohibitions against competing with a former employer are void in California, which is the site of Apple's present lawsuit.

A great deal will depend on the provisions of the employment contracts and agreements signed by Jobs and the other departing employees while they still were at Apple. The circumstances under which they acquired their knowledge of technical matters also will be significant. (Employment agreements were the subject of last September's column, while the general area of lawsuits between former employees and employers was explored last month.)

As we've suggested in many of these earlier columns, the legal concepts that apply to employer-employee relationships can be simply stated, but in practice are difficult to apply to the business of computing, both because of the nature of the technology and because of the relatively unstructured work environments characteristic of the industry. For instance, press accounts indicate that Apple became interested in RISC technology earlier this year, and that Richard Page supposedly "participated" in "several" meetings concerning it. Even taking this to be the case, should Page be precluded from working with this technology for the remainder of

his professional career?

The Apple-Jobs controversy raises fundamental questions regarding the application of established legal concepts to this industry. Exactly what *is* a trade secret in the computing field, anyhow? How deeply does a company have to be involved in a technology in order to claim a proprietary interest in it? Just when is an employee considered to be on his or her own time? Where does the employee's duty to an employer begin and where does it end? How much weight should the courts give to agreements that attempt to limit an employee's use of information acquired during employment?

Another factor, unique to computing, enters the picture. Aside from entertainment and sports, there probably is no other industry where such a great number of individuals reach the top of their profession near the beginning of their working lives. Steve Jobs illustrates this; as he wrote in his letter of resignation, "I am but 30 and still want to contribute and achieve." And while entertainment and sports careers often depend on physical attributes that do not survive the passing of time, there is no comparable limitation on computing skills. It's difficult to conceive that a court will tell Jobs that he must spend the remainder of his days tending roses in his garden because he knows too much to be allowed to continue in his chosen profession.

If *Apple vs. Jobs* does not resolve these questions, sooner or later we shall see other lawsuits that will. The entire industry has a stake in their outcome.

BY THE WAY. . .

Apple Computer has been providing quite a bit of subject material for this column. In July, 1984, we considered the landmark court decision of *Apple Computer, Inc. vs. Franklin Computer Corporation*. In that case, as readers will recall, Apple obtained a decision from a US Court of Appeals establishing that Franklin had infringed Apple's copyrights by copying 14 of its operating programs for the Apple II. However, the court in its opinion also put forth the scary notion that if Franklin could prove there was no other way these programs could have been written to achieve their purpose, Apple's copyrights would be invalidated. Following this decision, Apple and Franklin settled the suit.

Then, in September of this year Franklin unveiled its new personal computer, which it says is compatible with software designed for the Apple II. Presumably, Franklin was able to come up with alternate operating programs of its own, though Franklin's CEO is quoted as complaining that the Apple "was designed in a manner to make it difficult to build a computer and not infringe on the company's copyright."

Questions on legal subjects from readers of this column are most welcome. Individual response is usually not possible, but queries dealing with areas that are of general interest to the UNIX community are used as the basis for future columns. Any questions (or comments) should be sent in care of UNIX REVIEW, 500 Howard Street, San Francisco, CA 94105.

Glenn Groenewold is a California attorney who devotes his time to computer law. He has served as an administrative law judge, has been active in trial and appellate work, and has argued cases before the state Supreme Court. ■

COSI

Much More than *cu* and *uucp*—

OUTBOUND™

Simplicity for the End User

- Menu control of communications link
- Local scripting for batch terminal sessions

Flexibility for Outside Communications

- Terminal emulation
- X.PC and X.MODEM protocols for public data access

Security Provisions for Management Control

- Remote host control over local machine
- User/group level security
- Automatic audit trails for traffic/cost of communications

For further information, contact:

COSI, Incorporated
313 North First Street
Ann Arbor, Michigan 48103
Phone: (313) 665-8778
Telex: 466568 COSI CI

Outbound is a trademark of COSI, Inc.

Circle No. 18 on Inquiry Card

THE DEVIL'S ADVOCATE

Goodspels, badspels. . .

by Stan Kelly-Bootle

Those of you blissfully unversed in contemporary Biblical scholarship still must be basing your Christmas activities hopefully and innocently on the King James Version of the Gospels. In particular:

Glory to God in the highest,
and on earth peace,
good will toward men.
Luke 2:14 (KJV)

Yet, most reputable scholars, suitably aided with Biblical and Dead Sea Scroll databases, are now convinced that the Greek word *eudoxia* is better translated in the above context as "good pleasure" or "favor" rather than as "good will". Further, the preferred Lucan manuscripts have *eudoxias*, which is the genitive form: "of (God's) good pleasure". Cutting through many years of heated exegesis (see, for example, *The Anchor Bible*, Volume 28, pp. 410-412, Joseph A. Fitzmyer, Doubleday, NY), we can report that the heavenly hosts undoubtedly proclaimed to the frightened shepherds:

Glory in highest heaven
to God;
and on earth peace for
people whom he favors.

The good news is that *anthropoi*, formerly rendered in the

macho modal ("men"), becomes "people", thereby subsuming the ladies. The not-so-good news is that the Almighty is still male (to offset this, though, note that Commodore's Amiga is a señorita) and that the target group for peace is restricted to those whom God favors or to whom he "manifests his predilection" (op. cit. p. 411).

Whether this spoils your holiday or not I must leave to you and your conscience. Other seasonable scriptures, unchanged as far as I can determine at the time of this essay, stress the advantages of giving over receiving, a precept that I wish more of my friends would observe! The same secular tendencies that label Christmas as "Xmas" or "Bah-humbug-time" also replace the pure joy of giving with a cold, annual calculation known as the *exchange* of gifts, whereby December 26 is spent unwrapping the new PC and computing VAL_GIFTS_IN and VAL_GIFTS_OUT. Any imbalance is carefully noted and used to prepare next year's shopping lists. If you doubt me, look up "Present Value" in the index of any book on Actuarial Algorithms.

"What's in all this for me, me, me?" I hear you bleat. Well, I seem to have talked myself into a generous frame of mind, and many of you will benefit as I discharge my merry sleigh.

True, some of my gifts this year will be of an advisory disposition, but remember that computer consultants are so exorbitant nowadays that it is patently crazy to ignore their advice. With mine, which is free, you can take it or leave it; see if I care. I am even prepared to offer a free *second* opinion.

The first present out the bag goes to the semiconductor industry—my suggestion is that it immediately adopt a sensible supply/demand mechanism. It appears obvious to me that the only solution to the present roller-coaster instability is to make *demand* continuously responsive to *supply*, rather than the other way round. You see how simple things can be when a fresh, uncluttered mind approaches a problem from the outside?

Ah, and here we find a game for all those who consider Raymond Smullyan's puzzles too difficult. I call it the "Towers of Cracow". You have to transfer the hoop from A to B without using C. (See Figure 1.)

Next, for non-accountant computer scientists who have been forced to use spreadsheets, I offer binary and hex versions of a well-known package, which I have named Lotus 01-10-11 and Lotus H01-H02-H03, respectively. These help spreadsheets look like the more familiar and tractable core dumps.

I have something special for all ye who labor in vain. Whenever your genius slips by unrecognized, re-read the following *complete* review of the first-ever performance of Mozart's opera *Idomeneo* in 1781, as reported in the local Munich newspaper:

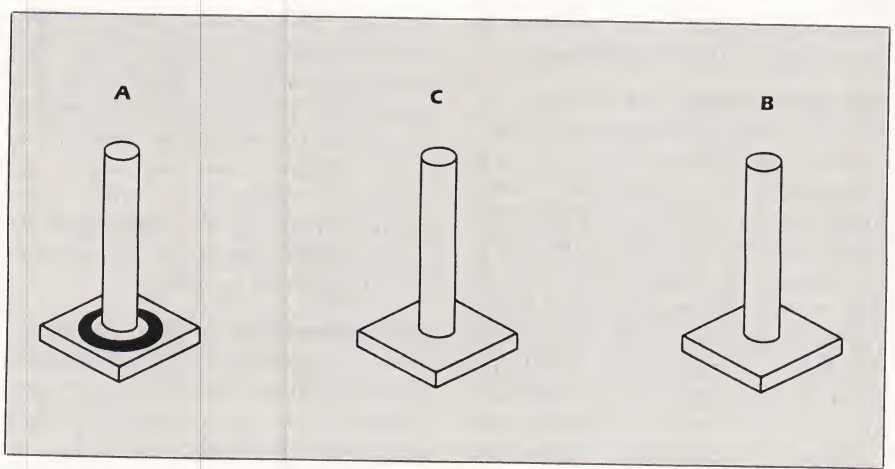
On the 29th of the month, the opera Idomeneo was performed for the first time in our new opera house. Libretto, music and translation originated in Salzburg. The decorations—of which the most inspiring are the view of the seaport and the temple of Neptune—are masterpieces by our architect, Mr. Corent Quaglio, and aroused the admiration of all.

Is there a Mr. (or maybe a Mrs.) Quaglio in *your* life, grabbing all the glory?

Everyone it seems—including such culprits as glossarist Steve Rosenthal — keeps complaining about the impossible data flow generated by the computer industry—this daily accumulative assault on the finite bandwidth of our comprehension. Our numbed channels, they claim, can no longer distinguish useful signals from noisy hype.

So, for all of us, God-favored or not, I whisper a heartfelt prayer for peace and a well-earned break from the dyna-quo.

Liverpool-born Stan Kelly-Bootle has been computing, on and off, at most levels since the pioneering EDSAC I days in the early 1950s at Cambridge University. After graduating from there in Pure Mathematics, he gained the world's first post-graduate diploma in Computer Science. Between authoring such books as The Devil's DP Dictionary and The MC68000 Primer, he has also served as Chairperson of the Biblical Studies Special Interest Group for the Association of Literary and Linguistic Computing. ■



The "Towers of Cracow" game—what a great gift idea!

COSI

PC to UNIX™ conversations *open* with—

TANGO™

Distributed processing

- Dynamically distribute data and programs to PCs
- Control all PC access to corporate data with UNIX

Reliable Systems

- Monitor with UNIX
- Investigate PC environment with UNIX utilities

DP Management

- Audit, security, and integrity functions

TANGO FEATURES:

- Terminal Emulation
- File Transfer
- Full Host Control

For further information, contact:

COSI, Incorporated
313 North First Street
Ann Arbor, Michigan 48103
Phone: (313) 665-8778
Telex: 466568 COSI CI

UNIX is a trademark of AT&T Bell Laboratories.
Tango is a trademark of COSI, Inc..

Circle No. 19 on Inquiry Card

THE UNIX GLOSSARY

Worldly words

by Steve Rosenthal

Note: only those terms related to the general topic of UNIX internationalization have been included in this listing.

ANSI X3.64 — the prevailing standard for character encoding and display in the non-IBM minicomputer and mainframe world. A superset of ASCII, it includes escape sequences for communications and output formatting.

attribute—an extra unit of information attached to a character, word, or region indicating how it should be displayed or processed. Many programs in UNIX use the eighth bit in each byte to convey attributes (the shell uses it to "quote characters" conveniently, for example), but this makes for trouble when the seven-bit ASCII character set is expanded to eight bits to accommodate international alphabets.

Berne Convention—a major international treaty for the protection of intellectual property. For protection under the Berne Convention, software must first be published in a Berne Convention country. The US is not one of these countries.

binding—the process of assigning a system resource—such as the text of error messages—to a program where it can be changed

from its general form to a specific code in order to be executed. Ideally, programs for international use should be written in a language-independent way with hooks that allow them to be *bound* to appropriate localization packages. This process is sometimes called *assignment*.

character class—a construction used to specify a range of characters acceptable for use in a token attached to a particular command or instruction. In the UNIX shell notation, a character class is created by enclosing a specified range in square brackets. For international use, character classes present thorny problems because they depend on an equivalence between the collating sequence of underlying codes and the dictionary collating sequence of visible characters—an equivalence that does not occur in many languages.

character set—the complete collection of characters a computer is capable of representing. Most UNIX machines use the ASCII character set, which defines 128 characters by using seven-bit code values that are part of eight-bit bytes, in which the high bit commonly is ignored or set to zero. However, for international use, larger character sets using as many as 32 bits to encode each symbol may prove to be necessary.

code expansion—the enlargement of a character set by increasing the number of bits used to denote each character. Although this theoretically is the cleanest way to make room for more characters, it often creates a multitude of practical difficulties. In UNIX, for example, many system utilities assume seven or eight-bit characters, and thus would need to be rewritten in order to accommodate characters of a longer bit length.

code extension—the use of character sequences to represent symbols that cannot be accommodated in the normal codeset. The most frequent method is to define an "escape" character

that introduces such sequences and then, as needed, ascribe meanings to those extended character sequences that seem useful. Codes also can be extended by using "escape" sequences to shift between two or more subsets; this is the method most commonly used to extend the seven-bit ASCII code for international use.

codeset—the collection of numbers available to represent characters or other information in a given computer system or program. Most codesets are less familiar to users than their corresponding character sets; consider the ASCII code and character set, for instance.

collating sequence—the order in which items are sorted. For example, alphabetical order is the most common collating sequence for entries in a dictionary. Because most sorting on computer systems is based on a simple comparison of the codes that represent characters—rather than on the true lexical relationships between characters—the results of a computer sort can seem illogical to the lay observer. Collating sequences are problematic when the order of a codeset differs from that of corresponding symbols: some UNIX utilities can handle these sorts of irregularities (as in the case of the difference between lower and upper case letters), but the problem is more acute with many non-English alphabets.

cursive—a writing style that uses letters that are connected by flowing lines. Cursive character sets are difficult to implement because letter forms often differ depending on preceding or succeeding characters, and on the position a character maintains within a word.

date format—the way a date is represented in written form. In the US, the abbreviated date format is *mm/dd/yy* (with *mm* standing for month, *dd* for day, and *yy* for year). In other regions of the world, the positions of days and months often are transposed. See *European date*.

dead-key—a key on a typewriter or other printing apparatus that sets a mechanism to prevent a normal advance to the following print position. For European languages that use accents, electronic systems sometimes use the electronic equivalent of a dead-key to produce a composite character out of a normal letter and an accent mark.

diacritical mark—an accent or other modifying symbol attached to a letter. In some languages, diacritical marks serve only to alter pronunciation, while in others the mark changes the nature and alphabetical order of the letter. Standard UNIX does not support diacritical marks. Many utilities, in fact, cannot handle the two and three-byte combinations sometimes used to represent characters with such marks.

European date—a date given in the *dd/mm/yy* form rather than the *mm/dd/yy* form used in the United States. See *date format*.

GMT—the English-language abbreviation for "Greenwich Mean Time", the traditional standard for solar time obtained by using the local time in Greenwich, England, as a reference. In more recent documentation, this same value has come to be referred to as "Universal Coordinated Time".

hard-coded—as applied to messages and instructions in programs, "hard-coded" refers to instructions embedded within program code instead of those that are linked by way of a connection to a separate module

COSI

When UNIX™ is *too much*, And DBMS is *too little*—

VISUAL/menu™

Standardized End-User Interface

- Full-screen interface
- Identical interface for transactions

Increased Programmer/Analyst Productivity

- Interactive, full-screen (graphic) interface to set of tools
- Incremental specification of transactions, screens, and help text
- Incremental development using *top-down* design, *bottom-up* implementation

Improved Management Control

- Security definable by user and transaction
- Automatic transaction logging

For further information, contact:

COSI, Incorporated
313 North First Street
Ann Arbor, Michigan 48103
Phone: (313) 665-8778
Telex: 466568 COSI CI

UNIX is a trademark of AT&T Bell Laboratories.
VISUAL/menu is a trademark of COSI, Inc.

Circle No. 22 on Inquiry Card

or file.

intellectual property—the legal term used to refer to the rights that authors of programs and other intellectual works can use to protect the fruits of their labor. Unfortunately, the rights accorded to intellectual property vary greatly from country to country, thus complicating the efforts of those who would like to sell UNIX programs internationally.

international characters–a collection of letters from the ordinary Roman alphabet that bear the diacritical marks needed to print various European languages. All countries, of course, have a different view of what an "international character" is and what a "national character" is.

international character set—a collection of letter forms used in various languages—in particular, the currency symbols and Roman alphabet members required for the printing of various European languages. See also *ISO character set*.

ISO character set—the various alphabets and symbols used in different countries, as standardized by the International Standards Organization. ASCII is ISO Alphabet Number 5.

JIS—an abbreviation for Japanese Industrial Standard, a Japanese equivalent to the US ANSI (American National Standards Institute) or ECMA (European Computer Manufacturing Association) standards. JIS values are used by most Japanese companies in the production of terminals and printers, including many that ultimately are used with UNIX systems.

Kanji—the Japanese pictographic character set, consisting of thousands of different characters. Various attempts have been made to produce versions of UNIX that use Kanji, with most of these using two-byte (16-bit) codes for each character.

Katakana—a phonetic rendering of the Japanese language that can be expressed in well under 100 characters, making it

Basmark BASIC

The first IBM®-PC Compatible BASIC Compiler for UNIX®

Available now:

- IBM/370, Amdahl
- VAX®, PDP®-11
- Intel® 80286, 8086
- Motorola® 68000
- System V, System III
- 4.xBSD, Xenix®

- AT&T 3B2 (in progress)

– *Binary/Source licenses available*
OEM/Distributor inquiries invited

Basmark Corporation

1717 East Ninth • Cleveland, Ohio 44114
(216) 621-7650

Trademarks: Motorola - Motorola Inc., VAX, PDP - Digital Equipment Corp., IBM - International Business Machines Corp., UNIX - AT&T Bell Laboratories Inc., Intel - Intel Corp., Xenix - Microsoft Corp.

Circle No. 20 on Inquiry Card

suitable for use with keyboards and display screens. Unfortunately, Katakana is regarded as being more crude than full Kanji, which, of course, is much more difficult to implement on computers.

language-independent—said of programs or systems that work equally well under different languages, at least once their message and help screens have been translated. Much of UNIX currently is not language-independent, especially those utilities concerned with text processing, sorting, and file formatting.

lexical—referring to the way that elements of a language or code are constructed, rather than to the meanings (semantics) these elements have or the way they are combined (grammatics). Lexical issues, such as the means by which languages with symbol repertoires larger than 128 or 256 characters are represented, are a major concern in efforts to internationalize UNIX and other software products.

localize—to change a program or system into a form suitable for a particular country, starting either from a universal version or from a different national implementation. Most vendors agree that UNIX and its attendant software will need to be localized if it is to be successfully sold and supported internationally.

Pan American Convention—a set of major treaties that protect intellectual property rights in the Americas. The US is a party to the Buenos Aires Convention in this series. Among its other requirements, the convention specifies that the phrase "All rights reserved" (or "Todos los derechos reservados") must appear on all materials to be protected.

repertoire—the total group of symbols to be included in a character set, providing there's room. In most cases, a smaller subset is chosen for inclusion.

transborder data flow — the transmission of information, particularly computer information, across national boundaries. Many nations restrict such transborder exchanges, either for economic or privacy-protection reasons. The UNIX community, on the other hand, expects to send data back and forth freely using such networks as Usenet and EUnet.

Universal Coordinated Time—the more formal and contemporary name for "Greenwich Mean Time", the mean solar time described by using the local time in Greenwich, England, as a reference. By historical convention, this is the most popular time standard used internationally.

Universal Copyright Convention—the principal international treaty protecting intellectual property. In essence, it gives the residents of all treaty countries the same rights internationally that they enjoy as citizens in their own countries—providing, of course, that certain standard procedures are followed. The US is a member of this convention.

Zulu Time—the name given to expressions of Greenwich Mean Time that use a 24-hour clock. Zulu Time is often used by international networks and documents.

If you have comments, questions, or corrections to offer, please send them to Rosenthal's UNIX Glossary, Box 9291, Berkeley, CA 94709.

Steve Rosenthal is a lexicographer and writer living in Berkeley. His columns regularly appear in six microcomputer magazines. ■

COSI

Tape Management and Backup for UNIX™

REEL™

Automatic Backup Operations

- Full-screen operator interface
- Automatic configuration of different backups
- Error handling
- Automatic logging of tape events

Standardized Interfaces to Tape Devices

- IBM and ISO/ANSI tape label formats
- Management mount/dismount queues and operator interfaces
- Access/update tape system library database

Improved Disk Utilization

- User-oriented utilities to request file backup
- Reports showing file and tape usage

For further information, contact:

COSI, Incorporated
313 North First Street
Ann Arbor, Michigan 48103
Phone: (313) 665-8778
Telex: 466568 COSI CI

UNIX is a trademark of AT&T Bell Laboratories.
REEL is a trademark of COSI, Inc.

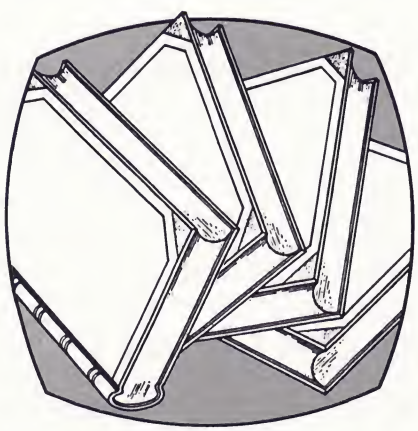
Circle No. 21 on Inquiry Card

FIT TO PRINT

High tide in C literature

by August Mohr

For years, *the* notable reference on the C language was *The C Programming Language*, by Brian W. Kernighan and Dennis M. Ritchie. Although the past couple of years have seen several additions to C's bibliography, it wasn't until a few months ago that the floodgates truly opened. This month's commentary looks at three of the books included in the recent torrent: *The C Answer Book*, by Clovis L. Tondo and Scott E. Gimpel; *C Made Easy*, by Herbert Schildt; and *Reliable Data Structures in C*, by Thomas Plum.



The C Answer Book

Clovis L. Tondo and Scott E. Gimpel
209+vii pp. ISBN 0-13-109877-2
Prentice-Hall, Inc., 1985
Englewood Cliffs, NJ 07632
$14.95 (paper)

The C Programming Language has become such a "bible" among C programmers that it commonly is referred to simply as "the white book". *The C Answer Book* is a long-awaited companion to this old favorite. Tondo and Gimpel have provided answers to the exercises presented by Kernighan and Ritchie. In describing the premise of their book, they note:

Careful study of The C Answer Book, used in conjunction with K&R, will help you understand C and teach you good C programming skills.

Use K&R to learn C, work the exercises, then study the solutions presented here. We built our solutions using the language constructions known at the time the exercises appeared in K&R. The intent is to follow the pace of K&R. Later, when you learn more language constructs, you will be able to provide possibly better solutions.

Since the avowed emphasis of the book is on solutions to exercises, it's hardly surprising that it contains more program than prose. Basic concepts of the C language are not the concern of this work. Such explanations are properly left to K&R.

All that is repeated from K&R are the problem statements (complete with page references). The solutions to these problems are explained in detail, if a bit dryly. For example:

Exercise 4-9: (page 87 K&R)

Define a macro swap(x, y) which interchanges its two arguments. (Block structure will help.)

```
#define swap(x, y) { int _z; _z = y; y = x; x = _z; }
```

The swap macro works if neither of the arguments is _z. If one of the arguments has the name _z, then when the macro is expanded it becomes

```
{ int _z; _z = _z; _z = x; x = _z; }
```

and the result is undefined. The assumption

NAME THE MOST WIDELY USED INTEGRATED OFFICE AUTOMATION SOFTWARE FOR UNIX™ SYSTEMS.

"UNIPLEX II"™

YOU'VE GOT IT!

User satisfaction is the primary reason no other product can make this claim. Already in its second generation, UNIPLEX II offers features designed to meet the requirements of the most demanding user.

The beauty of UNIPLEX II is its simplicity. One personality and one command structure throughout the program provide an ease of use never before experienced with UNIX application software.

UNIPLEX II integrates sophisticated word processing, spreadsheet, and relational database applications into a powerful one-product solution.

UNIPLEX II uses termcap, so it can run on virtually any computer terminal. "Softkeys" allow the user to define function keys which are displayed on the 25th line of most terminals to provide versatility and ease of use.

All this at a price you'd normally pay for a single application software package.

UNIPLEX II is available immediately from UniPress Software, the company that's been at the forefront of quality UNIX software products longer than anyone else.

Call today! Once you've got it, you'll see why UNIPLEX II is the most widely used integrated office automation software for UNIX-based systems.

OEM terms available. Mastercard and Visa accepted!

Write to: UniPress Software, 2025 Lincoln Hwy., Edison, NJ 08817 or call: 1-800-222-0550 (outside NJ) or 201-985-8000 (in NJ); Telex: 709418. European Distributor: Modulator SA, Switzerland 41 31 59 22 22, Telex: 911859.

UNIX is a trademark of AT&T Bell Laboratories. Uniplex II is a trademark of Uniplex Integration Systems.

NOW AVAILABLE ON THE AT&T UNIX PC 7300 & 3B SERIES!

UniPress Software

Your Leading Source for UNIX™ Software

Circle No. 47 on Inquiry Card

This is an excellent, sorely needed book.

is made that _z will not be used as a variable name.

```
#define swap(x, y) { x ^= y; y ^= x; x ^= y; }
```

This solution uses the bitwise exclusive OR operator (^). The following table shows the resulting value of z for different values of x and y:

```
x ^ y = z
-   -   -
0   0   0
0   1   1
1   0   1
1   1   0
```

x and y are swapped by exclusive OR'ing x and y three times:

```
x = x ^ y;
y = x ^ y;
x = x ^ y;
```

The first exclusive OR operation sets x equal to a mask. A bit in this mask is equal to 1 if both bits in the original x and the original y differ, and a bit is equal to 0 if both bits in the original x and the original y are equal. The second exclusive OR operation sets y equal to the original x from the information in the mask and from the original y. The third exclusive OR operation sets x equal to the original y from the information in the mask and from the new y (original x).

The C Answer Book is clearly typeset, with a nice mono-spaced font used for the C code itself. My only complaint is that the zero is hard to distinguish from the "O" character, but the difference is usually clear in context.

The index also seems very complete. The macro problem listed above is indexed under "^ bitwise exclusive OR operator", "bitwise exclusive OR operator ^", "block structure", "exclusive OR operator, bitwise ^", "macro *swap*", "operator, bitwise exclusive OR ^", "^OR operator, bitwise exclusive ^", and "*swap* macro".

This is an excellent, sorely needed book.

C Made Easy

Herbert Schildt
292+x pp. ISBN 0-07-881178-3
Osborne McGraw-Hill, 1985
2600 Tenth Street
Berkeley, CA, 94710
$18.95 (paper)

Herbert Schildt has produced a reasonable introduction to C programming. Topics are presented in a straightforward manner, and the explanations he offers are clear. It's unfortunate that the book suffers from several disappointing flaws.

Schildt starts with the presumption that his reader can program on a microcomputer and already has some experience with BASIC. Almost every chapter has example programs that are presented with their "equivalent" in BASIC. Unfortunately, the equivalences are not strict, and **printf()** is used both with and without a newline as if it were equivalent to BASIC's **PRINT** command. There is also the presumption, repeated frequently in the text and example programs, that output lines are terminated with both a carriage-return and a line-feed. The difference between this and standard UNIX usage is not mentioned, although Schildt does recommend using a "*UNIX-compatible*" C compiler.

The typesetting of the program examples usually uses an unambiguous, mono-spaced font, but the book is not consistent in this respect. Frequently, *C Made Easy* suffers from the all-too-common mistake of using open and close-quotes when C's syntax demands quotes of the same type. For instance:

> *First, individual characters that use the* ***%c*** *format command must be enclosed between single quotes; for example,* ***'c'****. Second, strings of characters that use the* ***%s*** *format command are enclosed between double quotes; for example,* ***"this is a string"****.*

This kind of inaccuracy can be glossed over by the experienced programmer, as no doubt it was by the author, but it is inconsistent with the otherwise careful tone Schildt takes toward his beginning audience.

The book also has other limitations, such as an index that's too sparse. But there's no need to

Another in a series of productivity notes on software from UniPress.

NEW RELEASE

UNIPRESS EMACS™

EDITOR FOR: UNIX™/ VMS™/ MS-DOS™

Subject: Multi-window, full screen editor.

Multi-window, full screen editor provides extraordinary text editing. Several files can be edited simultaneously, giving far greater programming productivity than vi. The built-in MLISP™ programming language provides great extensibility to the editor.

New Features:

- *EMACS is now smaller and faster.*
- *Sun windows with fonts and mouse control are now provided.*
- *Extensive on-line help for all commands.*
- *Overstrike mode option to complement insert mode.*
- *New arithmetic functions and user definable variables.*
- *New manual set, both tutorial and MLISP guide.*
- *Better terminal support, including the option of not using unneeded terminal drivers.*
- *EMACS automatically uses terminal's function and arrow keys from termcap and now handles terminals which use xon/xoff control.*
- *More emulation-TOPS20 for compatibility with other EMACS versions, EDT and simple Wordstar™ emulation.*

Features:

- *Multi-window, full screen editor for a wide range of UNIX, VMS and MS-DOS machines.*
- *"Shell windows" are supported, allowing command execution at anytime during an edit session.*
- *MLISP programming language offers extensibility for making custom editor commands! Keyboard and named macros, too.*
- *"Key bindings" give full freedom for defining keys.*
- *Programming aids for C, Pascal and MLISP: EMACS checks for balanced parenthesis and braces, automatically indents and reformats code as needed. C mode produces template of control flow, in three different C styles.*
- *Available for the VAX™ (UNIX and VMS), a wide range of 68000 machines, AT&T family, Pyramid,™ Gould,™ IBM-PC,™ Rainbow™ 100+ and many more.*

Price:

	Binary	Source
VAX/UNIX		$995
VAX/VMS	$2500	7000
68000/UNIX	395	995
MS-DOS	325	995

*For our **Free Catalogue** and more information on these and other UNIX software products, call or write:*
UniPress Software, Inc.,
2025 Lincoln Hwy.,
Edison, NJ 08817.
Telephone: (201) 985-8000.
***Order Desk: (800) 222-0550 (Outside NJ).** Telex: 709418.*
European Distributor:
Modulator SA, Switzerland
Telephone: 41 31 59 22 22,
Telex: 911859.

OEM terms available.
Mastercard/Visa accepted.

Trademarks of: UniPress EMACS & MLISP, UniPress Software, Inc.; UNIX, AT&T Bell Laboratories; VAX/VMS & Rainbow 100+, Digital Equipment Corp.; MS DOS, Microsoft Corp.; WordStar, MicroPro; Pyramid, Pyramid; Gould, Gould

UniPress Software

Your Leading Source for UNIX™ Software.

Circle No. 48 on Inquiry Card

belabor the problems here. Suffice it to say that the clear explanations and well-written, readable prose should have been given a better presentation. With more careful editing and typesetting—and a good index—this could have been an excellent book.

Reliable Data Structures in C

Thomas Plum
Pages by chapter. ISBN 0-911537-04-X
Plum Hall Inc., 1985
1 Spruce Ave.
Cardiff, NJ 08232
$25.00 (paper)

Plum's new offering, in some ways, is a sequel to the author's 1983 book, *Learning to Program in C*. Many of the sections begin with the notation "*Topics assumed from previous book*", and proceed to list topics with chapter and page references from that book. The explicit listing of those topics that assume background knowledge is admirable, and make the book valuable to readers from a wide variety of backgrounds.

Plum has given us more than a book on data structures.

Plum has given us more than a book on data structures. *Reliable Data Structures in C* also covers reliability and clear programming, and provides advanced tutelage on programming in C.

Plum writes clearly and shows respect for the intelligence of his reader. Over and again he points out potential traps and warns of bugs that are hard to find. What's more, he takes care to label examples of poor programming:

> *Just giving a constant a name is not enough to ensure modifiability; you must be careful always to use the name, and remember that that value could change. One project had difficulties changing the value of BUFSIZ because some programmers had written*
>
> ```
> nblocks = nbytes >> 9; hard to modify, uses ''magic number''
> ```
>
> *in a number of places where*
>
> ```
> nblocks = nbytes / BUFSIZ;
> ```
>
> *was needed. The programmers figured that "everyone knows that BUFSIZ equals 512." and right-shifting nine bits is the same (for positive numbers) as dividing by 512. But when BUFSIZ changed to 1024 on some systems, modifications were difficult. Hence, this rule:*
>
> *Rule 1-5: If a value is given for a #defined name, do not defeat its modifiability by assuming its value in expressions.*

Plum is very conscientious in his references to the idiosyncracies of different compilers. He also makes reference, as appropriate, to the evolving ANSI X3J11 standard for the C language. For instance, he covers the new (ANSI C) "generic pointers", but is careful to emphasize that the standard is still in its draft form. This is just the sort of precise reference one would expect from a member of the committee, as Dr. Plum is.

The book covers standard data structure techniques like stacks, queues, double-ended queues, and trees. Plum's discussion of I/O explains standard files, direct screen output, and binary records. The syntactic subtleties of **#if** (conditional compilation), **typedef**, **void** functions, **enum**, pointer types, pointers to functions, unions, bit-fields, dynamic allocation (**malloc**, **calloc**, and **free**), and cross-module (**lint**) type-checking are also given attention. The chapter on structures includes a case study of a menu processor and menu generator.

Reliable Data Structures in C is well-indexed, endowed with a good bibliography on C, and supplemented by an appendix that collects together all the rules of C programming that Plum has formulated.

Graphically, the book is very crowded, with diagrams done in the same font used for program examples. This may have been easier for the author, but I object to the use of *A*s and *V*s for up and down arrows.

With a background in both computer science and publishing, August Mohr formerly served the international UNIX users' organization /usr/group as the founding editor of its newsletter/magazine CommUNIXations. He also compiled and produced the UNIX Products Catalog. As a consultant, Mr. Mohr continues to maintain an active role in the computer and desktop publishing communities. ■

It's in the stars: 0 = kitty litter; 1 = take it if it's free; 2 = worth the cover price; 3 = well worth reading; 4 = get the leather-bound edition.

AN EXTENSIVE LINE OF PROGRAMMING LANGUAGES FOR NATIVE AND CROSS DEVELOPMENT WORK

SVS™ FORTRAN 77/ Pascal/BASIC-PLUS/C

SVS family of native mode compilers for MC68000™ UNIX™ machines. Full ANSI standard, symbolic debugger and optimized code generator with high speed optimization. Support for IEEE floating point, both single and double precision. SVS languages give excellent performance.

u4th

FORTH programming language for UNIX. Largely compatible with the FORTH-83 standard. u4th is interactive and allows full UNIX system call interface, as well as UNIX command pass-through. Permits C primitives and FORTH words to be loaded into a new kernel image. Used frequently in Artificial Intelligence work.

SSI TOOLKIT

SSI Toolkit is a set of Intel-style cross development tools for UNIX and VMS.™ Package includes macro cross assembler compatible with Intel ASM-86/87/88/186/188, linker, locator and librarian.

LATTICE® C CROSS COMPILER

Use your VAX™ (UNIX or VMS) or other UNIX machine to create standard MS-DOS™ object code for 8086™ and 186.™ The Lattice package includes compiler, linker, librarian, disassembler and 8087™ floating point support. Optional SSI Intel-style cross development tools can be used in conjunction with Lattice for native mode 8086 applications.

AMSTERDAM COMPILER KIT

Now includes BASIC, additional back-ends and simpler licensing!

A package of C, Pascal and BASIC (native and cross) compilers for UNIX machines. Hosted on VAX, PDP-11,™ MC68000,™ Z8000™ and 8086. Targets for VAX, PDP-11, MC68000,™ 6500™/6502,™ Z8000, 8086, NSC16032™ and 8080/Z80.™ Cross assemblers provided for MC6800™/6805™/6809,™ NSC16032 and Signetics 2650.™ Package contains complete sources.

PRICE:

SVS Languages	
FORTRAN 77, Pascal, and C	$995 each
BASIC-PLUS	750
u4th	
IBM-PC AT,™ Intel 286/310™	195
MC68000	495
VAX	1500
SSI Toolkit	
VAX (UNIX and VMS)	8000
MC68000	5000
Lattice C Cross Compiler	
VAX (UNIX and VMS)	5000
MC68000	3000
Amsterdam Compiler Kit	
Commercial users	9950
Educational Institutions	995

For our free catalogue and more information on these and other UNIX software products, call or write:
UniPress Software, Inc.,
2025 Lincoln Hwy.,
Edison, NJ 08817.
Telephone: (201) 985-8000.
Order Desk: (800) 222-0550 (Outside NJ).
Telex: 709418.
European Distributor:
Modulator SA, Switzerland.
Telephone: 41 31 59 22 22,
Telex: 911859

OEM terms available.
Mastercard and Visa accepted.

Trademarks of: UNIX, AT&T Bell Laboratories; VAX, VMS, and PDP-11, Digital Equipment Corp.; 8086/8087/186/286/310, Intel Corp.; MS-DOS, Microsoft; Lattice, Lattice, Inc.; MC6000, 6500, 6502, 6800, 6805, 6809, Motorola Corp.; NSC16032, National Semiconductor Corp.; Signetics 2650, Signetics Corp.; SVS, Silicon Valley Software, Inc.

UniPress Software
Your Leading Source for UNIX Software.

Circle No. 49 on Inquiry Card

RECENT RELEASES

286×4 = 5 VAXen

"I'm not so sure we're going after the mini market so much, but the distinction is becoming a little more blurred as to what kind of performance represents what", said Jeff Paine of Intel regarding the company's market focus for its APEX series of micros based on the use of multiple 80286 processors.

The point Paine was trying to make is that though the DEC VAX family of minicomputers continues to serve as a standard by which to assess the processing power of other minis and micros, hardware advances are allowing smaller machines to achieve what has traditionally been considered to be minicomputer-level power. This evolution, not surprisingly, has served to mutate the distinctions that historically have applied.

Intel actually has released two series of micros, the AP (Advanced Processor) and the APEX (Advanced Processor Extension), both using Multibus and running Xenix 3.0. Targeted at the high-end needs of the office automation market (including the health care and government vertical markets already penetrated by the company), Intel does not foresee any adverse effects on the sales of other 80286-based machines (such as the IBM PC-AT). "We're pretty comfortable that the APEX is something that nobody else is offering", Paine said.

The AP series is a faster single-CPU version of the company's current 286/310 product line, and the APEX machines (the top configuration of which, Intel claims, runs at 5 MIPS—or five times a VAX) use multiple 80286 CPUs in a 286/310 chassis. Both series achieve a performance increase by making use of an 80286 CPU running at 8 MHz (instead of 6), affording zero-wait-state access to all main memory, and offering an enhanced mass storage I/O subsystem using track caching.

An APEX multiprocessing supermicro, one of a new series from Intel.

All the AP configurations include a matched 80287 numeric co-processor, a 320 KB floppy, and support for a 60 MB ½-inch streaming tape cartridge drive and a 40 or 140 MB (unformatted) Winchester. AP systems come with 1 or 2 MB of RAM, expandable up to 9 MB. The entry-level system, the 310 AP-44 (for four users), is priced at $10,565 in OEM quantities.

The APEX series is designed for processing applications that either provide for a large number of users or require CPU-intensive work. The APEX-2, APEX-3, and APEX-4 add one, two, and three CPUs, respectively, to a basic 286/310 AP; each CPU board comes with its own RAM. The main CPU dispatches applications to APEX processors according to a system load-balancing algorithm, and each APEX board manages its own tasks. Although these machines do not perform parallel processing in the sense that individual applications are subdivided over all the processors, APEX system users can override the load-balancing algorithm and assign—or "force allocate"—specific applications to specific APEX processors.

APEX series kits for upgrading AP systems are now available and start at $6995. Also currently available is the APEX-2 system, beginning at $16,500 in OEM quantities. The APEX-3 and APEX-4, both requiring a system expansion chassis, are due out in the first quarter of 1986. A high-end configuration of the APEX-4, with 1 MB of RAM, a 140 MB hard disk drive, a floppy disk drive, and a tape drive, sells for about $35,000 in OEM quantities.

Intel Corp., 2402 W. Beardsley Rd., Phoenix, AZ 85027. 602/869-3825.

Circle No. 23 on Inquiry Card

HAVE DATABASE, WILL TRAVEL

Unify Corp. has signed agreements with eight firms to distrib-

ute the Unify Relational Database Management System internationally. This brings to 18 the number of international distributors carrying the system.

Computerlink Design Pty. Ltd. and Datacraft Office Systems Pty. Ltd. distribute the product in Australia. In Taiwan, Hong Kong, and Singapore, Unify is available from Asiatek, Inc.—based in Taiwan—which is developing a Chinese language version. Other distributors signed include HN Systems of Denmark; Jersoft OY of Finland; Systems Technology Research Corp. of the Philippines; Datatech of Switzerland; and Digitus Ltd. of the UK. Other countries in which Unify was made available by prior agreements include Japan, France, Germany, Holland, Italy, and Sweden.

Distribution through the recent agreements has already begun. The interface in each case has undergone translation to the language of the given country, including the development of a Japanese port using Kanji (there is even a Greek Unify). A separate license is needed for each translated version. The market consists primarily of professional applications developers, such as the European software houses and US and foreign OEMs. According to William Merchant, Marketing Communications Manager at Unify, a wide variety of applications have already been developed by organizations ranging from Sumitomo Electric Co. to the Finnish PTT to the British Library System (the counterpart of the US Library of Congress).

Prices for Unify vary according to the hardware on which it is implemented. On a PC-AT with Xenix, the RDBMS costs $1500; on a VAX 11/780, $15,000; and if you are so inclined (and equipped), on an Amdahl 580, $80,000.

Unify Corp., 4000 Kruse Way Place, Bldg.2, Lake Oswego, OR 97034. 503/635-6265.

Circle No. 24 on Inquiry Card

R YOU READY TO TRANSLATE?

R Systems, Inc., has released a language conversion program, R Linguist, for its line of office automation software. Designed to be used mainly by R System distributors and manufacturers whose machines run R Systems software, this standalone program sells as a separate package for $3995.

R Linguist operates on R Systems' three office automation software packages running on UNIX or Xenix-based machines: R Office, an integrated OA package; R Word, a production word processing program; and R Desk, a multifunction desk organizer. R Linguist displays English prompts, commands, and messages from the given software on the screen. The user then can fill in the appropriate equivalent in the desired language. This creates a language table that displays on-screen prompts in the new language. More than one language can be used on a given machine.

Bill Kiely, president of R Systems, said that the language conversion program translates on-screen English into any foreign language requiring only one-bit characters. R Linguist has already been used to produce programs in French, German, Portuguese, and Norwegian.

R Systems, Inc., 11450 Page Mill Rd., Dallas, TX 75243. 214/343-9188.

Circle No. 25 on Inquiry Card

CARRY THE TORCH: UK TO USA

"God save the queen and Torch

adax

X.25 FOR UNIX* Communications System

- Efficient, error-free data transmission to multiple hosts via international standard X.25, the only fully certified error-free public networking system used world-wide.
- User utilities
  - Remote user login
  - Remote mail service
  - Remote file transfer
- Compatible with widest number of host computers.
- Hardware available for VME, Multibus, IBM PC and others.
- Previously certified on TELENET, TYMNET and UNINET networks.
- Lowest cost per node.

Adax, Inc.

737 Dwight Way
Berkeley, CA 94710
(415) 548-7047

* UNIX is a trademark of Bell Laboratories.

Circle No. 26 on Inquiry Card

MULTIUSER SYSTEMS?

USE THE MULTIPORT™ SOLUTION

- Connect up to 8 terminals to an IBM PC for under $100 per connection
- Network PC's together using inexpensive serial ports instead of high-priced cards
- Kits compatible with XENIX, PICK, BOS, THEOS, VENIX/86, PC-SHARE, & EasyLAN

FREE Technical Assistance!
Call (615) 254-0646

ARNET
The Multiuser Experts
476 Woodycrest Ave./Nashville, TN 37210/TELEX 332762

Circle No. 27 on Inquiry Card

UNIX* JOBS REGISTRY

National registry of candidates and jobs in the Unix field. Please give us a call; send a resume; or request a free Resume Workbook & Career Planner. We are a professional employment firm managed by graduate engineers.

800-231-5920
P. O. Box 19949, Dept. UR
Houston, TX 77224
713-496-6100

Scientific Placement, Inc.

*Unix is a trademark of Bell Labs

Circle No. 28 on Inquiry Card

the prime minister!" Ahem. Well, Prime Minister Thatcher already *has* been Torched, if you really must know. She is aided daily, in fact, by a Torch desktop computer.

Torch Computers Ltd., of Cambridge, England, released its newest product, the Triple X, at the end of October in Britain and is actively seeking to penetrate the North American market.

The Triple X runs under UniSoft's UniPlus System V, and with Torch's own Man-Machine Interface (MMI) it now is being marketed not only to the scientific/technical consumer, but to the non-technical (business) user as well. A single-board, 32-bit, single-user PC with a 68010 processor running at 8 MHz, the Triple X comes with a 68450 DMA controller, a 68451 MMU, 1 MB of memory, and either a 20 or 40 MB hard disk. Among its features are a color bitmapped display, a multi-window environment, a mouse, icons, and a pull-down menu system—all on a monitor designed for Torch by Sony of Japan. The machine uses standard peripheral interfaces (Ethernet, SCSI, X.25, VME, RS232). The MMI itself is actually part of the UNIX kernel, and so operates at higher speeds than a separate process being swapped to disk.

Torch has strong links with Cambridge University, and over the past four years has sold more than 14,000 systems to the Ministry of Defence, various other government agencies, British Telecom, and various European PTTs. The Torch Unicorn, an earlier UNIX-based PC model launched in early 1984, is now the UK's best UNIX seller.

Torch Computers Ltd., Abberley House, Great Shelford, Cambridge, CB2 5LQ, England. 44-223-841000.

Circle No. 29 on Inquiry Card

THRIFTY, BRAVE, AND A GOOD HACKER

In this the Yuletide season, let it be noted that there are good investments in the future that are not of the strictly financial kind. The Boy Scouts of America (BSA), in cooperation with the Data Processing Management Association (DPMA), have announced strong growth in the Computers Merit Badge Program, "a popular, unique educational opportunity for thousands of young boys".

Since the Program's 1968 introduction, 87,792 scouts have earned a Computers Merit Badge, owing in no small part to the program's growth over the past three years: in 1982, 739 scouts earned the badge; in 1983, more than 13,000; and in 1984, 15,157. For successful completion of the program, scouts must be able to fulfill a number of requirements. These include: demonstrable familiarity with three programming languages; knowledge of several terms in computer hardware, software, I/O, storage, systems analysis, and design; and a grasp of the types of jobs and careers available in the computer field. Candidates for the badge also must develop a computer program that lists the names and phone numbers of troop members.

Requirements were updated in 1984 to reflect and incorporate new terminology and advanced computer technology. [BSA officials did not comment on whether this included an oral explanation and comparison of System V and 4.2BSD system calls.] IBM, Apple, and Hewlett-Packard sponsored the Computers Merit Badge demonstration at the last Boy Scout Jamboree, an annual event attracting scouts from all over the world.

DPMA, Chapter Member Services Coordinator, 505 Busse

Highway, Park Ridge, IL 60068-3191. 312/825-8124. Or contact your local Boy Scout Council.

Circle No. 32 on Inquiry Card

SQUEEZE FOR MORE RAM

The base price of a DEC MicroVAX II is around $24,000. This buys, among other things, 1 MB of memory. For $5995, Chrislin Industries, Inc., offers the Squeeze, a single memory card built for use on the DEC machine that holds 8 MB of memory.

A MicroVAX owner has the option of acquiring two 4 MB cards from DEC or some other vendor. But Edward Ross, Chrislin National Sales Manager, said that by using a Squeeze card in the place of one of the 4 MB add-ons, a customer could have a configuration comprising the 1 MB on the CPU chip, a 4 MB card, and the 8 MB Squeeze—adding up to a total memory capacity of 13 MB.

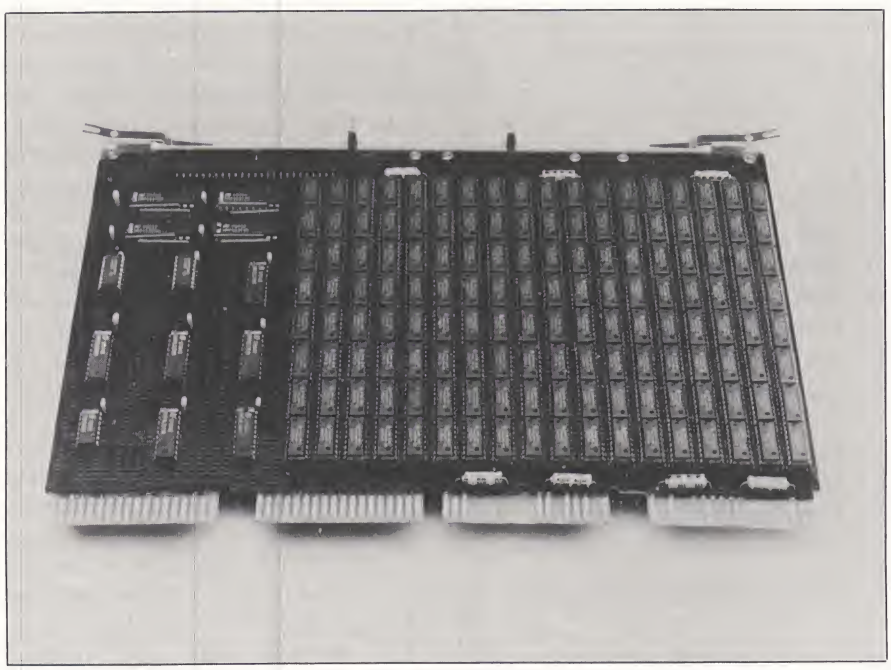
The Squeeze module has a maximum power consumption of 1.7 amps at +5 volts. The word size is 32-bit, and the board is completely hardware and software-compatible with the MicroVAX II. A five-year parts and labor warranty and a 24-hour repair/replacement policy come with the Squeeze card. Chrislin also manufactures memory cards for various Multibus-based systems, and a RAM card for the IBM PC.

Chrislin Industries, Inc., Computer Products Division, 31352 Via Colinas, #101, Westlake Village, CA 91362. 818/991-2254.

Circle No. 30 on Inquiry Card

DATABASE APPLICATIONS CAN NOW GO FOR A RIDE

Marc Rochkind, erstwhile a member of the Programmer's Workbench design team at Bell Laboratories, the inventor of SCCS, and author of the book, *Advanced UNIX Programming* (Prentice-Hall, 1985), is also the founder of Rochkind Software Corp. Having created the RIDE programming language (in-



Chrislin Industries' Squeeze memory card for the MicroVAX II.

The Truth of the Matter is...

Prevail is a UNIX-based office automation and application development solution which can be shipped to you *today*. If you are looking for office automation software or need a fourth-generation language, look to Prevail—an A.T.&T. co-labeled product.

Prevail has seven components which will meet your needs.

- Word Processing
- Spreadsheet
- Database Management System
- Window Manager and User Interface
- Report Writer
- Applications Development Language
- Telecommunications

Prevail is available on AT&T 3B series, AT&T Unix PC Model 7300, NCR Tower, DEC VAX and MicroVax II series, Sun Microsystems computers, and Masscomp computers.

Inspiration Systems, Inc.
400 Cummings Park, Suite 4300
Woburn, MA 01801
617 938-1160

Circle No. 31 on Inquiry Card

troduced last March), and having released four UNIX versions for specific computer models from three hardware manufacturers, Rochkind now has also released RIDE/DM, a version of the language that enables programmers to develop multiuser database applications for networked PCs.

RIDE is a high-level language similar to C, but Rochkind claims it is a business applications language superior to COBOL. COBOL, for example, doesn't boast of database applications, while RIDE contains 118 built-in functions, including a full set of mathematical operations (dollars and cents arithmetic to $2.1 billion), a screen forms manager, and the ability to read dBase II and dBase III files directly. The objective is to keep programmers from having to choose between a programming language and a significantly distinct database system.

Versions of RIDE are now available for the VAX 11/780, running under 4.2BSD (selling for $1495); the AT&T UNIX PC and 3B2, running under System V (retailing for $495); and Tandy's Models 16 and 6000, running under Xenix (also selling for $495).

RIDE/DM was designed specifically for The Database Machine, Cogent Technologies' PC board for the IBM PC, XT, and AT. The Database Machine acts as both file server and hard disk controller for the network, and also gives networked PCs simultaneous access to shared data files. Instead of waiting for off-the-shelf software or using conventional languages unsuited for business programming, RIDE/DM allows creation of specifically configured software for PC networks. The language package sells for $495.

Rochkind Software Corporation, 3080 Valmont Rd., Boulder, CO 80301. 303/442-4981.

Circle No. 33 on Inquiry Card

THE MINISTRY OF ADMINISTRATION

The challenges posed by UNIX system administration can be viewed as mere inconveniences by some (mostly those who've never tried it), but to others these challenges are the stuff of nightmares. Wherever you place yourself on this spectrum, there are two packages from Unitech Software to consider if you have a system that needs to be administered.

The first is UBACKUP, a backup, restore, and media management package. It offers system-wide, selective, full, and incremental dumps. Simple configuration files are defined initially to tailor both dump cycles and operational procedures to specific installation-dependent requirements. An integral media management feature controls volume rotation, and backed-up volumes can be kept for specific periods of time and then released for re-use. The software features an online catalog for rapid location of data, a log to track utilization and error conditions, and a file system usage report feature that identifies and summarizes high overhead points of interest such as large or underused files.

UBACKUP supports a variety of data storage media such as floppy diskette, cartridge tape, and 9-track tape, and verifies media after it has been tran-

Complete Unix ™ Development System from $2995

Fortune™ 32:16 System 10 with Multi-User Operating System,*
512K Main Memory
(expandable to 1MB),
10MB Hard Disk,
5.25" Floppy Drive,
Monitor & Keyboard,
IDS Prism 80 Printer, 150 CPS.
ALL FOR $2995!
While Quantities Last

We also have systems with your choice of 256K, 512K or 1MB Main Memory, and 10, 20, 30, or 45MB Hard Disks. We carry NEC, IDS and Genicom printers, and printer options.**

These are Fortune™ reconditioned machines which carry a 30 day Limited Warranty.

San Carlos Computer Supply
"Reconditioned & Reasonable"
1571 Industrial Road
San Carlos, CA 94070
(415) 593-2653

Reconditioned - 30 day Warranty
"As Is" - No Warranty

*Includes C Programming Language.
**Call for a price list.

Unix is a TM of AT&T Bell Labs
Fortune is a TM of Fortune Systems

Circle No. 35 on Inquiry Card

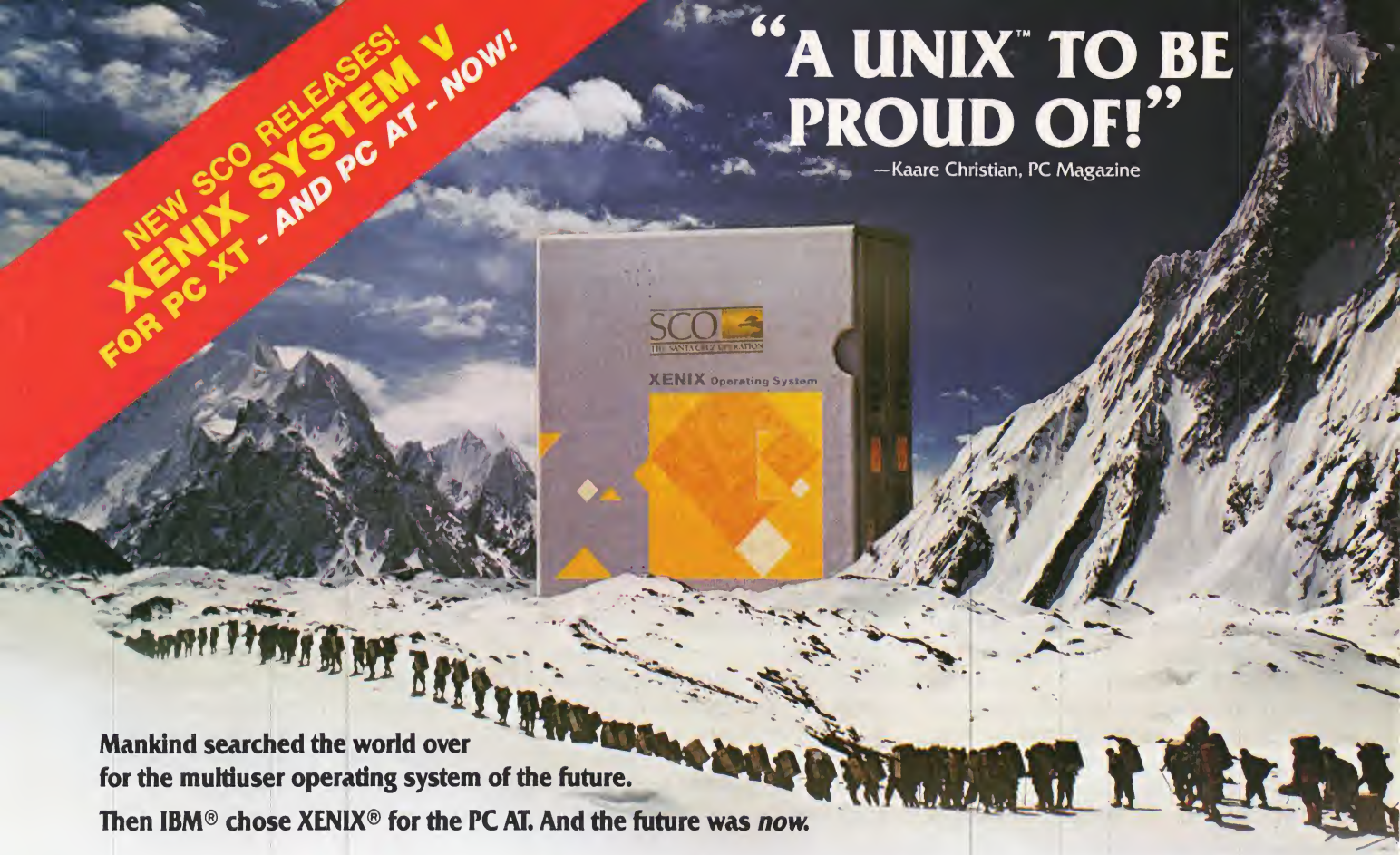
FRANZ
THE FIRST NAME IN
LISP

Franz LISP from Franz Inc. is currently available under UNIX and VMS. Now with Flavors and Common LISP compatibility. Franz sets the standard for LISP.

Franz Inc.
1141 Harbor Bay Parkway
Alameda, California 94501
(415) 769-5656

UNIX is a trademark of Bell Labs. VMS is a trademark of Digital Equipment Corporation.

Circle No. 36 on Inquiry Card



Mankind searched the world over for the multiuser operating system of the future.

Then IBM® chose XENIX® for the PC AT. And the future was *now*.

THE SANTA CRUZ OPERATION PRESENTS

XENIX NOW!

AN SCO PRODUCTION IN EXCLUSIVE ASSOCIATION WITH MICROSOFT CORPORATION
THE MULTIUSER, MULTITASKING PC BLOCKBUSTER "XENIX NOW!"
STARRING VISUAL SHELL • MULTISCREEN™ • MICNET • THE BERKELEY ENHANCEMENTS
AND INTRODUCING C-MERGE AS THE MS-DOS DEVELOPMENT ENVIRONMENT
FEATURING WORLD FAMOUS SCO TRAINING AND SUPPORT FOR DEALERS • END USERS • ISVs • OEMs
AND AN INTERNATIONAL CAST OF HUNDREDS OF XENIX APPLICATIONS
INCLUDING LYRIX™ AS THE UNIX/XENIX WORD PROCESSING SYSTEM
PRODUCED AND DIRECTED BY THE SANTA CRUZ OPERATION
SCREENPLAY ADAPTED BY THE SANTA CRUZ OPERATION FROM ORIGINAL STORIES BY MICROSOFT AND AT&T
IN BREATHTAKING SELECTABLE COLOR
NOMINATED FOR ★ BEST DOCUMENTATION! ★ BEST SUPPORT! ★ BEST TRAINING!
★ BEST ELECTRONIC MAIL AND NETWORKING! ★ MOST APPLICATIONS!
★ MOST COMPLETE UNIX SYSTEM!

SCO
THE SANTA CRUZ OPERATION

RELEASED FOR MOST POPULAR PERSONAL COMPUTERS.
APPLICATIONS ALSO AVAILABLE: LYRIX, MULTIPLAN®, INFORMIX®,
LEVEL II COBOL™, 3270 MAINFRAME COMMUNICATIONS.

(408) 425-7222
TWX: 910-598-4510 SCO SACZ

Circle No. 45 on Inquiry Card

M MULTIUSER OPERATION SUGGESTED
XENIX WILL TURN YOUR PC INTO A REAL COMPUTER

©MCMLXXXIV The Santa Cruz Operation, Inc.
The Santa Cruz Operation, Inc., 500 Chestnut Street, P.O. Box 1900, Santa Cruz, CA 95061 (408) 425-7222

UNIX is a trademark of AT&T Bell Laboratories • Lyrix and Multiscreen are trademarks of The Santa Cruz Operation, Inc. • IBM is a registered trademark of International Business Machines Corporation • XENIX and Multiplan are registered trademarks of Microsoft Corporation • Informix is a registered trademark of Relational Database Systems, Inc. • LEVEL II COBOL is a trademark of Micro Focus, Ltd.

scribed. The package requires an instruction space of less than 64 KB.

The second software package is USECURE, a menu-driven package to provide enhanced system security. Among other services, the software copies or moves files and directories based on specific selection criteria; maintains *.profile*, *umask*, user, and group information; and maintains an online log file that provides an audit trail of all changes made to the system via USECURE, including a record of system startup and shutdown activities. USECURE requires at least 128 KB of memory space, but this is because it runs as an application of SSL, Unitech's screen manager and application development system.

USECURE is now available and ranges in price, depending on configuration and the size of the computer on which it runs, from $146 to $850. UBACKUP is due out by year's end and has a price range from $390 to $4500. Deals are being negotiated with OEMs, but end users are the primary market for these products. As such, Unitech claims the packages can be user-installed and that customer support can be obtained by phone. Packages are tailored by Unitech to run under specific versions of UNIX, including old and new ones from AT&T, BSD, and various look-alike flavors from other companies.

Unitech Software, Inc., PO Box 7490, McLean, VA 22106-7490. 703/734-9844.

Circle No. 34 on Inquiry Card

EMANCIPATION PROCLAMATION

The first image that usually comes to mind upon hearing the term "network" may be of several smaller computers electronically linked to a larger computer. In this scenario, all information flowing in the network is passed through the host computer. The relationship of mainframe host to linked mini/micro does not have to be one of master to slave, however. The Orion Group has announced the sna62 Peer Com-

MORE NEWS FROM

4.3BSD Source or Binary with Bug Fixes, Maintenance, Enhancements

BERKELEY, CA—The 4.3BSD release of "Berkeley UNIX" from the University is due out soon. It adds significant performance improvements and new features to the capabilities which have made 4.2BSD the standard for advanced UNIX applications.

Maintained Source

Rather than obtaining a University distribution, sites now running 4.2BSD source can buy their upgrade to 4.3BSD-based source from MT XINU. Unlike the unsupported University release, MT XINU's 4.3-based source comes with ongoing maintenance and bug fixes, MT XINU's own enhancements, and access to expert help when problems arise. New sites can begin with either 4.2 or 4.3BSD source and also avoid spending time on routine source maintenance instead of more important work.

Supported Binary

VAX users who don't need source can buy a fully-supported MT XINU 4.2BSD-based binary now and upgrade to 4.3 when available.

WE KNOW UNIX™ BACKWARDS AND FORWARDS

TRADEMARKS: Unix is a trademark of AT&T Bell Laboratories. VAX is a trademark of Digital Equipment Corp. SUN is a trademark of Sun Microsystems, Inc.

munications Facility, a software package of protocols that enables micros, minis, and mainframes—as well as terminals, file servers, and other devices—to exchange information directly, without having to go through a central computer host.

Orion sna62 is a third-party implementation of the LU 6.2 network, the latest enhancement of IBM's System Network Architecture (SNA). Previously, the set of network functions needed to control the hardware in an SNA-type network could only address IBM 370-based architecture in a hierarchical master/slave arrangement. The new emulation enables peer-to-peer communication between a wide range of systems.

Orion developed the SNA Peer Network Facility under a contract with Apple Computer that called for it to link Apple's in-house UNIX-based computers to an IBM mainframe. The software allows UNIX machines to communicate with each other as peers, as well as with IBM systems. The facility is written in C (the interface works through a library of C function calls) under System V, and is a complete software implementation requiring no custom hardware. Developers of local area networks can also use the facility as a gateway to connect two LANs together.

Orion's principal market comprises hardware vendors and VARs, so sna62 is offered on a license and royalty basis.

The Orion Group, Inc., 1912 Bonita Way, Berkeley, CA 94704. 415/548-0947.

Circle No. 37 on Inquiry Card

UNDER LOCK AND (CHIP) KEY

Enigma Logic has made available SafeWord UNIX-Safe, a combined hardware and software security system for UNIX-based computers. Designed to control access to all sizes of computers, the security system can also lock specific files, accounts, databases, dial-up lines, or any combination of programs.

SafeWord software is compiled

BERKELEY

Network File System for VAX™ Increases Power of UNIX Systems

BERKELEY, CA—MT XINU's VAX Network File System lets VAX UNIX users create common file systems distributed over a network of machines including Sun Workstations™, multiple VAXen, and Gould and Pyramid computers.

The Network File System, conceived and developed by Sun Microsystems, was implemented for VAX by MT XINU. It extends the power of 4.2BSD ("Berkeley UNIX") networking by making remote file sharing transparent. An authorized user accesses a file or directory anywhere on an NFS network just by specifying a local pathname.

NFS saves network storage space and improves data consistency by eliminating unnecessary file copies, and it simplifies network system administration.

VAX NFS is available now from MT XINU as an addition to MT XINU's 4.2BSD-based source or binary system software. Upgrades to 4.3BSD will be provided after 4.3 is released by Berkeley. For independent 4.3BSD source sites, NFS will be available in the form of linkable object modules.

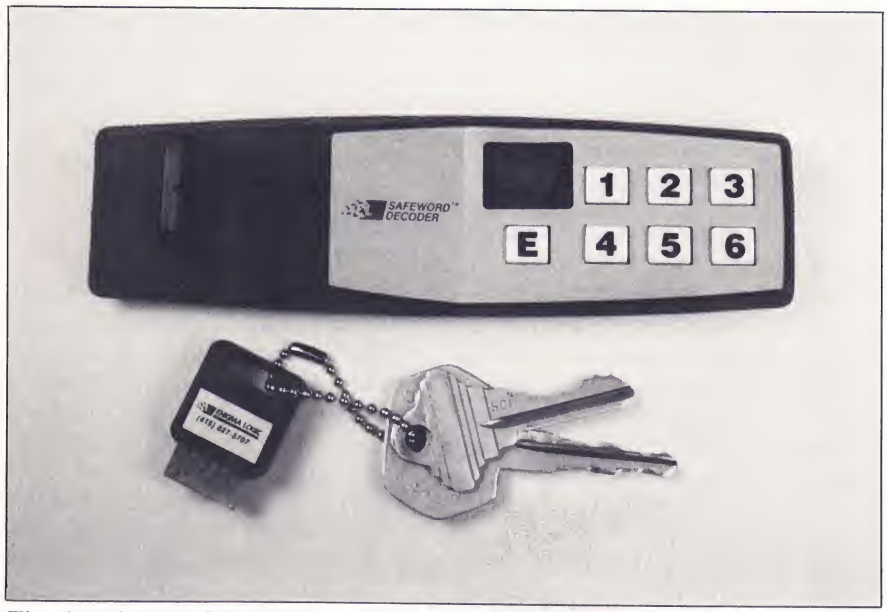
MT XINU™

UNIX™ SUPPORT FROM BERKELEY

MT XINU 415-644-0146

2910 SEVENTH STREET SUITE 210 BERKELEY, CA 94710

Circle No. 38 on Inquiry Card



The decoder and key "password dispenser", part of Enigma Logic's UNIX-Safe security system.

into the protected computer's operating system. Legitimate users are provided with a unique, calculator-like SafeWord decoder and key (with an electronic chip on it), which act as a "password dispenser". When a user wishes to login, the UNIX system requests the account name and regular password, as usual. According to vice president of marketing Dr. J-C. Spender, the system—depending on how the customer has it configured—can then proceed in one of two ways.

The first is a duplex procedure. The system asks for a second password (which will be used only this one time) in the following fashion: the system selects a seven-digit number from a random number mill and displays it on the terminal screen; the user enters this number on the hand-held decoder (which serves as a logical rather than electrical interface) and presses the "enter" key"; the decoder, with key inserted, calculates a response, and this serves as the one-time password that the user can then enter on the terminal keyboard to gain entry into the computer system.

The second type of configuration is a simplex procedure. After the login name and regular password are entered, the system waits for the user to generate the second password. But, instead of being a response to a random number challenge, this second password is derived from past usage. The user gets this password by inserting the key (which has data files in its chip) into the decoder and pressing "enter"; the new password appears on the decoder screen, and the user enters this password into the computer. Computer software contains a password history and calculates the new password. The password generated separately by the decoder and the computer should match, and the user should then be logged in.

Enigma Logic is working on a family of other interfaces, including a port-mounted version not requiring the hand-held decoder. (The hand-held model, however, has the advantage of terminal independence.) Though negotiating with OEMs (including AT&T), the company sells directly to end users. UNIX-Safe software is supplied on tape or diskette, with documentation, and can be installed by the customer or with company support. Price varies with functionality and starts at $3500 for the software. Decoder and key costs also vary with functionality and volume, but are typically about $150 per user.

Enigma Logic, Inc., 2151 Salvio, Suite 301, Concord, CA 94520. 415/827-5707.

Circle No. 64 on Inquiry Card

KERNEL DOES GRAPHICS RIGHT

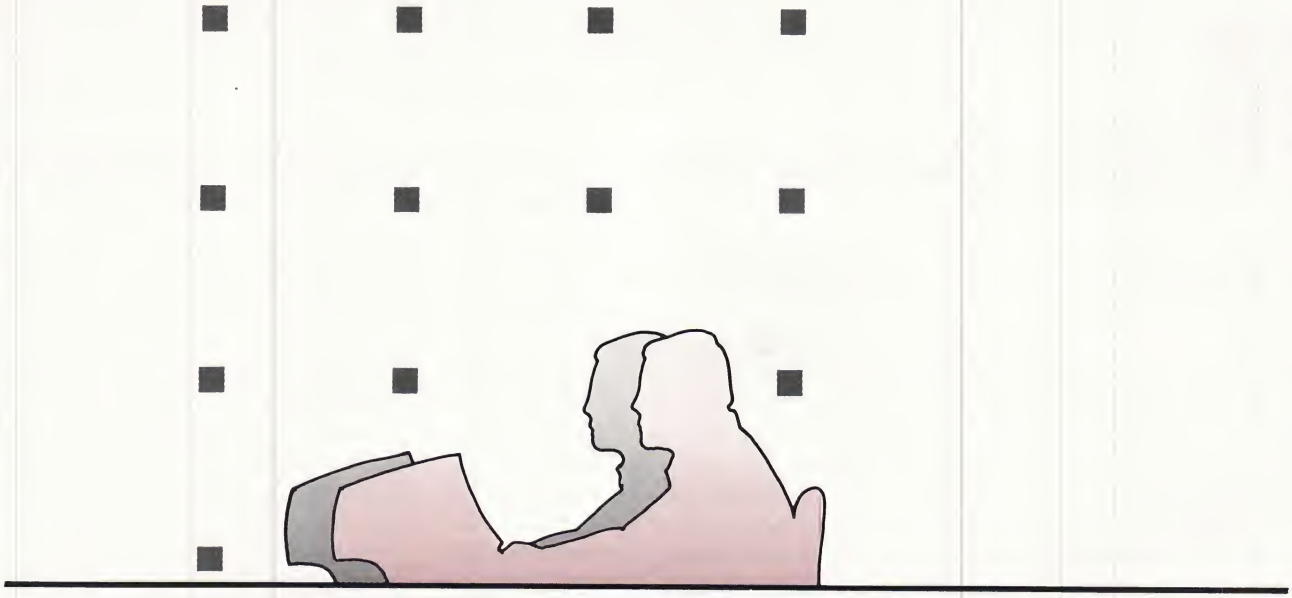
The GSS-Toolkit kernel system developed by Graphic Software Systems is an implementation of the ANSI and ISO Graphical Kernel System (GKS) Specification Level 2b. Programmers choose between C, Fortran, and Basic compilers to develop their GKS applications.

The toolkit kernel system runs on UNIX System III, System V, Version 7, and Berkeley 4.2. On a retail basis, it is available through Lattice, Inc. at a price of $495. OEM agreements can be made directly through GSS.

There are many characteristics separating this product from the company's Level mb software. Bundled attributes, ISO GSK metafile support, the full text model including stroke precision text, workstation control, cell array primitive, control over deferring changes to the workstations, and the ability to allow the simultaneous opening of multiple workstations have been added.

Graphic Software Systems, 25117 SW Parkway, PO Box 673, Wilsonville, OR 97070, 503/682-1606.

Circle No. 65 on Inquiry Card



THE International UNIX* Event of the Year!

February 4-7, 1986 / Anaheim Convention Center / Anaheim, California

UniForum 1986, the International Conference of UNIX Users, has established itself as THE premier UNIX conference/trade show.

■ Over 200 major vendors exhibiting their newest UNIX-based hardware, software, systems, services and peripherals.

■ A complete tutorial and conference program. Multiple sessions target the connectivity of UNIX between the user and technical environments; the hardware and software technology for office systems and workstations; the interface between man and machine, machine and machine, and much more. Day-long tutorials provide intensive, focused material on specific subjects in UNIX...while the conference sessions highlight the latest developments in the continuing evolution of UNIX.

*UNIX is a trademark of AT&T Bell Laboratories.

■ **FREE** UNIX Introductory Workshops.

■ Birds-of-a-Feather impromptu sessions.

Call us NOW for your **FREE** Show-Only Badge *and* complete registration information on the dynamic conference/tutorial program.

The power and potential of UNIX are important to you. UniForum 1986 is THE computer event you can't afford to miss!

UniForum™
The International Conference of UNIX Users

For Complete Information, Call:

800-323-5155

(In Illinois, Call: 312-299-3131)

Or Write:

UniForum
2400 East Devon Avenue
Suite 205
Des Plaines, IL 60018

Sponsored by

/usr/group
The International Network of UNIX Users

Circle No. 61 on Inquiry Card

GLOBAL VIEW

Continued from Page 35

question, "Why internationalize UNIX?" is the question, "Why should I be concerned with all of this?" If neither you nor your organization plan to be involved in the international marketplace at any time in the foreseeable future, why bother with the problems and issues that face the international community of UNIX users? Again, the most straightforward answer springs directly from the UNIX philosophy, which itself is in keeping with the *Unification Philosophy* underlying much of the progress made in science and technology over the last 500 years. Fundamentally, we are in the same position as those medieval astronomers locked into the classical Universe according to Ptolemy. We simply have made North American seven-bit ASCII the center of our universe—because it works. That is, it works as well as a flat earth: sailing off the edge can be handled as an "exception", as can the kludges necessary to get the planets and the calendar to come out correctly. As we get more data we can add to the exceptions—until we are rationally forced to accept a more inclusive model.

Within the US itself, how does the telephone directory list */usr/group*, *3Com*, or *7-Eleven*? For that matter, where do initials sort and at what point should the perennially lower-case *e e cummings* be listed in the index of *20th Century Poets*? When I'm rushing to find a flight, should I look under *St. Louis* or *Saint Louis* in my trusty *Official Airlines Guide*? Why does the ASCII underscore print as a left-arrow on the KSR-33 Teletype from which ASCII was lifted? Why does a dictionary sort *sanatorium* right after *San Antonio*? How should text scanners interpret the last period in an ellipsis following an abbreviation at the end of a sentence? Or how should text scanners cope with a sentence beginning ".357 magnums will not be worn in class..."? When numerics become strings,

Only one word processing program for these UNIX®-based systems isn't just a lot of talk.

Many companies are promising UNIX-compatible word processing software. But only WordMARC™ is being used successfully right now on such major UNIX-based systems as DEC,® HP,® SUN,® AT&T,® MASSCOMP,® PYRAMID® and NCR®

With WordMARC, you'll have a single, full-featured program that will end the proliferation of word processing software. Training time will be cut because the *identical* program runs on all kinds of computers. So users can easily switch terminals or systems. And with its optional LinkMARC feature, text created on your UNIX-based system can be transferred to and shared by superminis and personal computers.

UNIX, DEC, HP, SUN, AT&T, MASSCOMP, PYRAMID and NCR are registered trademarks of, respectively, AT&TBell Laboratories, Digital Equipment Corporation, Hewlett-Packard Co., Sun Microsystems, Inc., AT&TBell Laboratories, Massachusetts Compurer Company, Pyramid Technology Corporation and NCR Corporation.

isn't it bothersome that "123" comes between "12" and "13" and that an underscored character in your favorite text editor or regular-expression analyzer either *does* or *does not* match its naked equivalent—whichever is least convenient?

As a child meeting trigonometry for the first time, my reaction to the *Law of Cosines* was that it was an exceptional case of the familiar Pythagorean Theorem with an irritating "*-2 a b cosine theta*" thrown in. Ditto for the nasty Lorentz-Fitzgerald term Einstein tagged onto Newton's beautiful laws. What becomes clear after a little while, though, is that *in the world at large, exceptions are the rule* and that we cling to the simplifying special case like a drowning man to a life ring. But if we back off a bit, the UNIX Philosophy can do for us what Tycho's calculations did for Kepler.

Fundamentally, we are in the same position as those medieval astronomers locked into the classical Universe according to Ptolemy.

Besides chairing the /usr/group Technical Advisory Committee on Internationalization, Brian Boyle is the Director of Research at NOVON Research, a San Francisco-based organization investigating emerging technologies related to UNIX systems and software, supercomputing, integrated voice and data, and artificial intelligence. Prior to founding NOVON, he was the managing analyst of the Systems and Software Group at Gnostic Concepts. Mr. Boyle holds a Ph.D. in Medical Information Systems, although he claims the degree's warranty has run out. ■

PYRAMID

SUN

AT&T

In addition to being compatible with all kinds of computers, WordMARC also gets along with all kinds of users.

Its documentation is written specifically for the computer system it will operate on. Its self-teaching guide helps novice users get quickly up to speed. And it's supported by a special "800" number hotline.

WordMARC's many versatile features include technical and scientific symbols, foreign language characters, a what-you-see-is-what-you-get screen, and menu-driven operation with convenient function keys.

WordMARC can also be integrated with other popular applications software.

So get the UNIX-compatible word processing system that's up and running now—and put your word processing software resources back under control. With WordMARC. The Uncommon Denominator.

Contact MARC Software for more information. 260 Sheridan Avenue, Suite 200, Palo Alto, California, 94306.

MARC SOFTWARE INTERNATIONAL, INC.
1-800-831-2400. In California 1-800-437-9900.

Circle No. 5 on Inquiry Card

WordMARC
The Uncommon Denominator

WordMARC is a trademark of MARC SOFTWARE INTERNATIONAL, INC. © 1985, MSI, INC.

CALENDAR

EVENTS

DECEMBER

December 12-13 Monterey, CA: Usenix second annual graphics workshop. Contact: Usenix Conference Office, PO Box 385, Sunset Beach, CA 90742. 213/592-3243.

JANUARY 1986

January 15-17 Denver: Winter '86 Usenix Technical Conference. Contact: Usenix Conference Office (see above).

FEBRUARY

February 4-7 Anaheim, CA: UniForum International Conference of UNIX users, sponsored by /usr/group. Contact: UniForum 1986, 2400 E. Devon Ave., Suite 205, Des Plaines, IL 60018. 312/299-3131.

TRAINING

Note: Below are listed the dates, locations, titles, and contacts for UNIX-related training courses. For registration and further information on particular courses, contact the firm cited. Training firm addresses and phone numbers are listed alphabetically at the end of the calendar.

DECEMBER

December 2 New York: "Mainframe-to-Mini-to-Micro Links". Contact Interactive.
December 2 Dayton, OH and Houston: "UNIX Operating System". Contact NCR.
December 2-3 New York and Washington, DC: "Advanced C Programming Workshop". Contact CTG.
December 2-3 Menlo Park, CA: "Basic Informix". Contact RDS.
December 2-3 New York: "Shell Programming Workshop". Contact Structured Methods.
December 2-4 Washington, DC: "UNIX for Users, Including Shell Programming". Contact AIQR.
December 2-4 Edison, NJ: "Advanced C Language Programming". Contact AUXCO.
December 2-4 Philadelphia: "UNIX and C: A Hands-on Workshop". Contact Drexel University.
December 2-4 Santa Monica, CA: "UNIX Fundamentals". Contact Interactive.
December 2-5 Union, NJ: "Advanced UNIX". Contact Asidor.
December 2-5 Callaway Gardens, GA: "UNIX OS: The First Step". Contact AT&T.
December 2-6 Dallas and San Francisco: "UNIX Internals". Contact CTG.
December 2-6 London: "UNIX for Programmers". Contact The Instruction Set.
December 2-6 Cincinnati: "UNIX for End Users". Contact ITDC.
December 2-6 Absecon, NJ: "C Programming Workshop". Contact Plum Hall.
December 2-6 Washington, DC: "C Language Programming". Contact Webco.
December 3 London: "UNIX Overview". Contact CTG.
December 3 Los Angeles: "UNIX Management Overview". Contact NCR.
December 3-4 Menlo Park, CA: "Basic Informix". Contact RDS.
December 3-5 San Francisco: "SNA Architecture and Implementation". Contact CSI.
December 4 Tampa: "UNIX Management Workshop". Contact NCR.
December 4-6 London: "UNIX Fundamentals for Non-Programmers". Contact CTG.
December 4-6 New York and Washington, DC: "Advanced C Programming Under UNIX". Contact CTG.
December 5-6 Washington, DC: "C Programming". Contact AIQR.
December 5-6 Edison, NJ: "C Language Debugging". Contact AUXCO.
December 5-6 Santa Monica, CA: "Using the Shell". Contact Interactive.
December 5-6 Menlo Park, CA: "Basic Informix-SQL". Contact RDS.
December 9 New York: "Principles of Computer Graphics". Contact LUCID.
December 9 Dayton, OH: "C Programming". Contact NCR.
December 9 Dayton, OH: "UNIX System Administration". Contact NCR.
December 9 Tampa: "UNIX Operating System". Contact NCR.
December 9-10 Santa Monica, CA: "System Administrator's Overview". Contact Interactive.
December 9-10 Anaheim, CA: "The Concepts of Object-Oriented Programming". Contact PPI.
December 9-11 Edison, NJ: "Advanced Shell". Contact AUXCO.
December 9-11 London: "UNIX Fundamentals for Programmers". Contact CTG.
December 9-12 Union, NJ: "Introduction to C—A Hands-on Workshop". Contact Asidor.
December 9-13 Trumbull, CT: "Intro to UNIX". Contact Bunker Ramo.
December 9-13 New York and Washington, DC: "Berkeley Fundamentals and **csh** Shell". Contact CTG.
December 9-13 London: "Advanced UNIX". Contact The Instruction Set.
December 9-13 London: "C Programming Language". Contact The Instruction Set.
December 9-13 London: "Device Drivers and Kernel Overview". Contact The Instruction Set.
December 9-13 Washington, DC: "The UNIX System for the DP Professional". Contact Webco.

December 10-12 Dallas and San Francisco: "UNIX Administration". Contact CTG.
December 11-13 Santa Monica, CA: "Interactive Networking Tools". Contact Interactive.
December 12-13 London: "Shell as a Command Language". Contact CTG.
December 14 San Francisco: "Inside 4.2BSD Networking". Contact Uni-Ops.
December 16 New York: "C Programming". Contact NCR.
December 16-17 Dallas and San Francisco: "Advanced C Programming Workshop". Contact CTG.
December 16-17 Santa Monica, CA: "Advanced Commands for Programmers". Contact Interactive.
December 16-19 Union, NJ: "Fundamentals of UNIX". Contact Asidor.
December 16-20 Edison, NJ: "Introduction to C Language Programming". Contact AUXCO.
December 16-20 Trumbull, CT: "C Programming". Contact Bunker Ramo.
December 16-20 London: "C Language Programming". Contact CTG.
December 16-20 Washington, DC: "C Programming Workshop". Contact Plum Hall.
December 16-20 London: "Advanced C". Contact The Instruction Set.
December 17 Boston and Washington, DC: "UNIX Overview". Contact CTG.
December 17 Palo Alto, CA: SV Net Monthly Meeting. Contact SV Net.
December 18-20 Boston and Washington, DC: "UNIX Fundamentals for Non-Programmers". Contact CTG.
December 18-20 Dallas and San Francisco: "Advanced C Programming Under UNIX". Contact CTG.
December 18-20 Santa Monica, CA: "UNIX Architecture—A Conceptual Overview". Contact Interactive.

JANUARY 1986

January 6 Dayton, OH and Los Angeles: "UNIX Operating System". Contact NCR.
January 6-9 Union, NJ: "Shell Programming". Contact Asidor.
January 7 Dayton, OH: "UNIX Overview with Workshop". Contact NCR.
January 13 Dallas: "UNIX Operating System". Contact NCR.
January 13 Dayton, OH: "C Programming—Advanced". Contact NCR.
January 13 Los Angeles: "UNIX System Administration". Contact NCR.
January 13-16 Union, NJ: "Advanced C Programming". Contact Asidor.
January 13-17 London: "UNIX for Programmers". Contact The Instruction Set.
January 14 Dayton, OH: "Unify Database Maintenance System". Contact NCR.
January 14 Tampa: "UNIX System Administration". Contact NCR.
January 14-17 Washington, DC: "UNIX: A Comprehensive Introduction". Contact ICS.

Please send announcements about training or events of interest to: UNIX REVIEW Calendar, 500 Howard Street, San Francisco, CA 94105. Include the sponsor, date and location of event, address of contact, and relevant background information. ■

CONTACT INFORMATION

American Institute for Quality and Reliability (AIQR), 1494 Hamilton Ave., Suite 104, San Jose, CA 95125. 800/621-0854 ext.290, or in CA, 408/978-2911.

Asidor Training Institute, 2143 Morris Ave., Suite 5, Union, NJ 07083. 201/888-0241.

AT&T Information Systems, Institute for Communications and Information Management, PO Box 8, Pine Mountain, GA 31822-0008. 800/247-1212.

Auxton Computer Enterprises, Inc. (AUXCO), 2 Kilmer Rd., Edison, NJ 08817. 201/572-5075.

Bunker Ramo Information Systems, Trumbull Industrial Park, Trumbull, CT 06609. 203/386-2000.

Computer Technology Group (CTG), 310 S. Michigan Avenue, Chicago, IL 60604. 800/323-UNIX, or in IL, 312/987-4082.

Communications Solutions, Inc. (CSI), 992 S. Saratoga-Sunnyvale Road, San Jose, CA 95129. 408/725-1568.

Drexel University, 32nd and Chestnut Streets, Philadelphia, PA 19104. 215/895-2153.

Instruction Set, The, 152-156 Kentish Town Road, London, NW1 3YP, England. 01-482 2525.

Integrated Computer Systems (ICS), PO Box 45405, Los Angeles, CA 90045. 800/421-8166, or in CA, 800/352-8251.

Interactive Systems Corp., 2401 Colorado Avenue, 3rd floor, Santa Monica, CA 90404. 213/453-8649.

LUCID, 260 Fifth Avenue, Suite 901, New York, NY 10001. 212/807-9444.

NCR Corp., Customer Services Division, 101 W. Schantz Avenue, Dayton, OH 45479. 513/445-3798.

Plum Hall, 1 Spruce Avenue, Cardiff, NJ 08232. 609/927-3770.

Productivity Products International, Inc. (PPI), 27 Glen Road, Sandy Hook, CT 06482. 203/426-1875.

Relational Database Systems, Inc. (RDS), 4100 Bohannon Drive, Menlo Park, CA 94025. 415/322-4100.

Silicon Valley Net (SV Net), PO Box 700251, San Jose, CA 95170-0251. 415/594-2821 (Grant Rostig).

Structured Methods, Inc., 7 W. 18th St., New York, NY 10011. 800/221-8274.

Uni-Ops, PO Box 27097, Concord, CA 94527-0097. 415/945-0448.

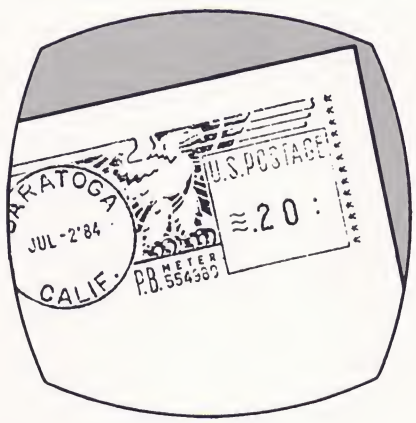
Webco Industries, Inc., 14918 Laurel Oaks Lane, Laurel, MD 20707. 301/498-0722.

THE LAST WORD

Letters to the editor

SO WHAT'S NEW?

Dear UNIX REVIEW,



Ho hum. The UNIX REVIEW issue on Languages [September] is worthy of a yawn. Fortran. Yep, UNIX has Fortran and so did the SOS 7094 two decades ago. C, for "Caveman language", that's on UNIX too, and has been for over a decade. Now there's talk of Lisp, roughly of Fortran vintage. So where's the news?

The issue does talk about Ada (which is unusable) and also Modula-2 (which is a breath of fresh air). OK, so Modula-2 is the one newsworthy language that gets reasonable coverage in the issue. But hold on, Modula-2 isn't the only exciting language under UNIX. There are hundreds of UNIX installations using the Concurrent Euclid language, which is thought by some (like me) to be the world's best implementation language. There's an implementation of UNIX written in Concurrent Euclid. And there's Turing, being used by thousands of people on UNIX and PCs, which is thought by some (like me) to be the ideal general-purpose language for use under UNIX.

So how about the real news, eh? How about covering the exciting UNIX languages of the '80s—not the '60s and '70s—in an issue on languages?

R.C. Holt
Computer Systems Research Institute
Toronto, Ontario, Canada

Mr. Holt is the author of Concurrent Euclid, The UNIX System, and Tunis (Addison-Wesley, 1983). *Editor*

Dear UNIX REVIEW,

In the September issue of UNIX REVIEW, Joel McCormack laments:

> *The lack of type checking is, I hazard, the largest source of productivity loss for C programmers.*

But lack of type checking is not inherent in C. It was in an early issue of this very magazine (UNIX REVIEW, February/March, 1984) that I first wrote about an implementation of C that performs inter-module type checking. The implementation was called the Safe C Compiler and has since been renamed the Safe C Runtime Analyzer.

The Analyzer performs checking of function parameters at runtime, thus inter-module function calls are as easy to check as intra-module. It also detects errors in the use of standard library routines including **printf/scanf**. So an error like McCormack's:

```
scanf("%d",i)
```

where *i* is an integer, is caught.

In addition, counter to McCormack's claim (on the bottom of page 28) that, "C is left with no way to perform runtime checking", the Analyzer detects array indexing errors and indirection through stray pointers.

C certainly has its dangerous points, but let's not complain about daggers that have been sheathed.

Alan Feuer
Catalytix Corp.
Cambridge, MA

LET'S TRY THAT AGAIN

Dear UNIX REVIEW,

In the September, 1985, issue, *C Advisor* by Mr.

Try listening to UNIX

That's what we do in UNIX REVIEW
We listen, and look, and probe
It's surprising how much there is that's fresh and new.
Just waiting to be shared
It's often that way with things that are simply elegant.
Like a shell. Like UNIX

UNIX™ REVIEW

THE MAGAZINE FOR THE UNIX COMMUNITY

UNIX is the trademark of Bell Laboratories, Inc.
UNIX REVIEW is not affiliated with or sponsored by Bell Laboratories.

Bill Tuthill indicates that one can combine C and Fortran as well as Pascal. That's nice to know. Unfortunately, there are a few bugs in the article.

The number of arguments in the Fortran subroutine call does not match C parameter definition. Therefore, the C program in Figure 2 [page 69] cannot work properly. I suggest the following:

```
#include <stdio.h>
long unixc_(cmd)
char *cmd;
{
    char buf[BUFSIZ];
    int i;

    sprintf(buf,"%s",cmd);
    /* Return value only for algorithmic purpose */
    for (i = 0;*(cmd+i) != '\0';i++);
    if(i >= 256)     /* UNIX command line length */
        return(-2L);
    if(system(buf) == 127)
        return(-1L);
    return(0L);
}
```

U. S. Postal Service STATEMENT OF OWNERSHIP, MANAGEMENT, AND CIRCULATION Required by 39 U.S.C. 3685.
1A. TITLE OF PUBLICATION: UNIX REVIEW. 1B. PUBLICATION NO. 07423136. 2. DATE OF FILING: 10/1/85. 3. FREQUENCY OF ISSUE: Monthly. 3A. NUMBER OF ISSUES PUBLISHED ANNUALLY: 12. 3B. ANNUAL SUBSCRIPTION PRICE: $28.00 4. COMPLETE MAILING ADDRESS OF KNOWN OFFICE OF PUBLICATION: 901 South Third Street, Renton, Washington 98055 U.S.A. 5. COMPLETE MAILING ADDRESS OF THE HEADQUARTERS OF GENERAL BUSINESS OFFICES OF THE PUBLISHER: 901 South Third Street, Renton, Washington 98055 U.S.A. 6. FULL NAMES AND COMPLETE MAILING ADDRESS OF PUBLISHER, EDITOR, AND MANAGING EDITOR: PUBLISHER Pamela J. McKee, 901 South Third Street, Renton, Washington 98055 U.S.A. EDITOR Mark Compton, 500 Howard Street, San Francisco, California 94105 U.S.A.; MANAGING EDITOR None 7. OWNER: Review Publications, 901 South Third Street, Renton, Washington 98055 U.S.A.; Pamela J. McKee, George C. Arrowsmith, Stuart M. Stern, Ronald P. King, all at 901 South Third Street, Renton, Washington 98055 U.S.A.; Miller Freeman Publications, 500 Howard Street, San Francisco, California 94105 U.S.A., a wholly-owned subsidiary of United Newspapers, plc., 23-27 Tudor Street, London EC4Y 0HR, England
8. There are no known bondholders, mortgagees, and other security holders owning or holding 1 percent or more of total amount of bonds, mortgages, or other securities.
9. Does not apply
10. EXTENT AND NATURE OF CIRCULATION—AVERAGE NO. COPIES EACH ISSUE DURING PRECEEDING 12 MONTHS. A. TOTAL NO. COPIES (Net Press Run): 22,991. B. PAID AND/OR REQUESTED CIRCULATION: 1. Sales through dealers and carriers, street vendors and counter sales: 2317. 2. Mail Subscription (Paid and/or requested): 14,130. C. TOTAL PAID AND/OR REQUESTED CIRCULATION (Sum of 10B1 and 10B2): 16,447. D. FREE DISTRIBUTION BY MAIL, CARRIER OR OTHER MEANS SAMPLES, COMPLIMENTARY, AND OTHER FREE COPIES: 1063. E. TOTAL DISTRIBUTION (Sum of C and D): 17,510. F. COPIES NOT DISTRIBUTED 1. Office use, left over, unaccounted, spoiled after printing: 4318. 2. Return from News Agents: 1163. G. TOTAL (Sum of E, F1 and F2—should equal net press run shown in A): 22,991.
ACTUAL NO. COPIES OF SINGLE ISSUE PUBLISHED NEAREST TO FILING DATE. A. TOTAL NO. COPIES (Net Press Run): 37,000. B. PAID AND/OR REQUESTED CIRCULATION. 1. Sales through dealers and carriers, street vendors and counter sales: 1950. 2. Mail Subscription (Paid and/or requested): 28,164. C. TOTAL PAID AND/OR REQUESTED CIRCULATION (Sum of 10B1 and 10B2): 30,114. D. FREE DISTRIBUTION BY MAIL, CARRIER OR OTHER MEANS SAMPLES, COMPLIMENTARY, AND OTHER FREE COPIES: 1246. E. TOTAL DISTRIBUTION (Sum of C and D): 31,360. F. COPIES NOT DISTRIBUTED 1. Office use, left over, unaccounted, spoiled after printing: 4590. 2. Return from News Agents: 1050. G. TOTAL (Sum of E, F1 and F2—should equal net press run shown in A): 37,000
11. I certify that the statements made by me above are correct and complete (signed) Pamela J. McKee, Publisher.

In Fortran use:

```
call unixc('ls -l')
```

Please note that the Version 7 Fortran compiler gives a warning if the subroutine name is more than six characters long.

David Tak-Shen Chen
The New York Blood Center
Elmont, NY

COMMITTEE REPORT

Dear UNIX REVIEW,

In your *Monthly Report* for July, you combined two different IEEE Committees into one. The P1003 Committee is focusing strictly on the Operating System Standards and, in fact, more specifically on the system call interface. This effort will hopefully go to ballot this Winter, reflecting heavily the 1984 /usr/group Standard with input from System V and also a limited number of Berkeley extensions such as **mkdir**.

The second IEEE Committee is addressing Open System Architecture. This is being chaired by Paul Borrill, Secretary of the IEEE Computer Society, and represents a substantially longer-term effort. Operating System Standards will be just one of the areas addressed by the OSA effort.

Marc Rochkind's comments on the Standards effort [also in July's *Report*] were slightly misleading. For better or worse the Standard work to date from /usr/group and IEEE has been based on System III and to some degree on System V, without a strong focus on any Berkeley extensions with 4.1 or 4.2. He is correct that Interprocess Communications (IPC) is not addressed in the current effort. This has been delegated to a real-time subcommittee which will be addressing a number of real-time related capabilities. Terminal interface control was not addressed in the /usr/group Standard. However, a set of **termio** facilities have been specified in the IEEE drafts. Record and file locking capabilities have been included in both the /usr/group document and the IEEE document. This capability has also been adopted by AT&T as part of the *System V Interface Definition*. A subcommittee has been formed in the IEEE group to address issues of record locking and atomic operations for interaction and database work. We are seeking experts in this area to help with this subcommittee.

I hope this helps to bring your readers up to date.

James Isaak
Chairperson, IEEE/CS P1003
Charles River Data Systems
Framingham, MA

COMPLETE YOUR UNIX REVIEW LIBRARY!

June/July 1983—UNIX on the IBM/PC .... ☐
August/September 1983—Sritek and Venix . ☐
October/November 1983—UNIX Typesetting ☐
December/January 1984—Vi and Emacs ... ☐
February/March 1984—UNIX Databases ... ☐
April/May 1984—Menu-based User Interfaces ☐
June 1984—Big Blue UNIX ..... ☐
July 1984—The AT&T Family ..... ☐
August 1984—Documentation ..... ☐
September 1984—System Administration ... ☐
October 1984—UNIX on Big Iron ..... ☐
November 1984—User Friendly UNIX ..... ☐
December 1984—Low Cost UNIX ..... ☐
January 1985—Evolution of UNIX ..... ☐
February 1985—UNIX Portability ..... ☐
March 1985—Performance ..... ☐
April 1985—UNIX Networking ..... ☐
May 1985—Distributed Resource Sharing ... ☐
June 1985—UNIX Applications ..... ☐
July 1985—Office Automation ..... ☐
August 1985—Database Intricacies ..... ☐
September 1985—Languages ..... ☐
October 1985—UNIX and Universities ..... ☐
November 1985—Scientific Applications .... ☐

Back issues are $4.95 each including postage. Payment in advance is required. Send this order form with check (US funds payable at US bank only) or credit card information to: REVIEW Publications, 901 S. 3rd St., Renton, WA 98055. Additional $1.00/issue for foreign mail.

Name _____
Company _____
Address _____
City _____ State ___ Zip _____
M/C or VISA # _____
Exp. Date _____

ADVERTISERS' INDEX

Adax Inc. 91
Arnet Control 92
AT&T Information Systems 43
Avalon Computer Systems, Inc. 25
B.A.S.I.S. 64
Basmark Corp. 82
bbj Computer Services 47
Bell Technologies 49
Blast/Communications Research 17
Ceegen Corp. 39
Celerity 13
Century Software 75
C-Line 29
Cogitate 53
Corporate Microsystems 57
COSI 77,79,81,83
DSD Corp. 56
DTSS, Inc. 65
Emerald Systems Corp. 35
Emerging Technology 9
Franz, Inc. 94
Handle Technologies . . . . . Cover II
Hewlett-Packard . . . . . Centerspread
Image Network 49
Inspiration Systems 93
Interactive Systems Corp. 41
Marc Software 100,101
Mt. XINU 96,97
Oasys 71
Olympus Software, Inc. 67
Overland Data, Inc. 33
Quality Software Products 71
Radio Shack 27
Rapitech Systems, Inc. 19
Relational Database Systems 1,2,3
San Carlos Computer Supply 94
Santa Cruz Operation 95
Scientific Placement 92
Sperry Corp. 72,73
UniForum 99
Unify Corp. 10,11
Unipress Software 85,87,89
Unitech Software Inc. 15
Uniworks 21
User Training Corp. 69
UX Software . . . . . Cover IV
Webco Industries 17
XED/Computer Methods 7
Zanthe . . . . . Cover III

COMING UP IN JANUARY

Global Communications

- **The anarchy of long-haul networking**
- **Names, addresses, and routes**
- **A user's view on global communications**
- **The user's dilemma**
- **Technology futures**

ZIM 2.5 A DBMS REVOLUTION

Have you been looking for perfect data management that you can enjoy on your own terms? Then you've probably already heard of ZIM 2.4 — the most powerful database system available. Until now. Because ZIM 2.5 is here.

ZIM 2.5 is a fourth generation application development tool which makes it possible to expand the capabilities of your micro beyond what you've ever imagined. ZIM mirrors the complexities of the real world by letting you develop as many and as varied applications as you could possibly need.

"ZIM is...a successful migration of mainframe ideas and needs to a micro. (ZIM) proves not only that the job can be done but also that it can be done well. ZIM provides a reference against which current and future data bases can be judged." James Creane, Data Based Advisor/July 1985.

Speed
ZIM breaks the speed limit — between 3 and 50 times faster than industry leaders in sorting and joining files within the data-base. ZIM's internal architecture, and the implementation of its strategy analyzer and priority-driven buffering ability, ensure that data is processed in the most efficient manner possible.

Portability
ZIM is the only database management system with 100% application portability for single-user and multi-user configurations. ZIM is available under PC-DOS, Concurrent PC-DOS, UNIX, XENIX, and QNX. Never again will you be required to re-write your applications for different operating systems environments.

Power
ZIM's high-level language lets you build user commands which implement applications without the necessity and cost of additional programming tools. ZIM's forms facility and extensive report generator permit completely menu-driven applications. Completed compiled, applications use the Runtime System, leading to fast execution, preventing unauthorized access or modifications, and decreasing cost and memory requirements.

Flexibility
ZIM gives you unprecedented simplicity and flexibility. ZIM commands parallel simple English sentences, making it easy to learn and use. Other features include automatic updates of all indexes, multi-user support, and an extensive range of validation, editing and masking facilities. ZIM's limits are only those of your hardware, operating system and imagination. And with ZIM 2.5, your database is no longer limited to a single hard disk.

"ZIM is (a) well-conceived, soundly-implemented, thoroughly professional system. Its design evidences a strong commitment to consistency and to the goal of natural nonprocedural user interaction."

Richard M. Foard, PC Tech Journal, October 1985.

ZIM 2.5 — DATA MANAGEMENT AT ITS BEST

ZIM
The Information Interface

Circle No. 42 on Inquiry Card

1785 Woodward Drive
Ottawa, Ontario K2C 0R1
(613) 727-1397

ZANTHE
INFORMATION INC.

Available now from IBM as Interactive Executive (IX) Basic on the IBM Interactive Executive for System/370 (IX/370)

The Language for a New Generation

Portability. UX-Basic™ application programs execute unchanged on any UNIX™ machine and are completely device independent.

Power. UX-Basic contains the building blocks for efficient application program development. It also allows you to tap the full power of UNIX and gives you direct access to data bases.

Productivity. UX-Basic is friendly and easy to learn and use. The interactive programming environment provides syntax checking as well as real-time debugging and testing.

Performance. UX-Basic gives you speed when you need it with our efficient pseudo-code compiler/runtime package. We are constantly working to keep UX-Basic's performance at the leading edge.

Profit. UX-Basic programs are structured, modular and readable. Maintenance and support are easy.

Perfect for UNIX... a new generation of computers... a new generation of computer users.

UX Software, Inc.

10 St. Mary Street, Toronto, Canada M4Y 1P9
Tel: (416) 964-6909 TLX: 065-24099

Available from major computer manufacturers such as Altos, AT&T, Siemens and an international network of distributors.

UniForum.
The International Conference of UNIX Users
February 4-7, 1986

UNIX is a trademark of AT&T Laboratories.
UX-Basic is a registered Trademark of UX Software, Inc.

Circle No. 54 on Inquiry Card