SPECIAL EDITION JUNE ■ 1987

DECPRO EXTRA

- UNIX In A Business Environment
- Supporting The X Window System
- The Benefits Of Standards-Making

From the Publishers of DEC PROFESSIONAL

CLEAR THE DEC'S

OPUS220 IS HERE!

THE FIRST DEC/ANSI TERMINAL WITH WINDOWING AND FREE 24-HOUR REPLACEMENT SERVICE AT A PRICE OF $559.

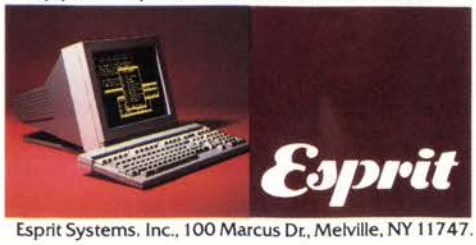
Got your eye on a DEC® VT220™ terminal? Esprit's new OPUS220 will knock it right out of the picture.

OPUS220 is the first VT220 compatible that does windows. User-friendly, PC-like "Express Windows" that allow you to suspend ongoing applications, display important information like help screens, then retrieve original screen data when you resume. Only Esprit puts such versatility at your fingertips.

For virtual terminal environments like UNIX™ and VMS™, Esprit's OPUS220 performs like no other. It provides up to four pages, each independent and supporting a different application. OPUS220 also features a 14" dark-background, high-contrast video screen that's brighter and sharper than DEC's. With "touch-tilt" and a "lazy susan" swivel that make viewing easy from every angle. Enhancements that helped the OPUS design win a 1986 Industrial Design Excellence Award for functionality as well as good looks.

Now the clincher. OPUS220 is backed by an industry first: Esprit Express 24-hour replacement service, free for a full year. No one goes as far to guarantee quality and performance. Yet OPUS220 lists for hundreds less than DEC's VT220.

The more you compare, the easier it is to see why smart terminal buyers are clearing their desks for the new OPUS220 from Esprit. For the dealer nearest you, call toll free: **(800) 645-4508**. New York State: (516) 293-5600.



Esprit Systems, Inc., 100 Marcus Dr., Melville, NY 11747.

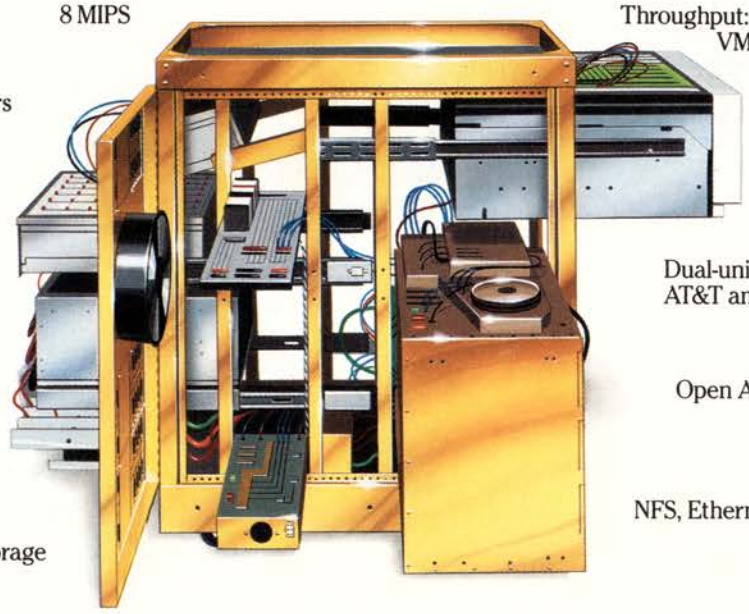
ENTER 621 ON READER CARD

I/O, I/O
IT'S OFF TO WORK
WE GO.

8 MIPS

Throughput: Two 40 MB/Sec. VMEbuses

Up to 256 Users



Up to 128 MB Memory

Dual-universe UNIX: AT&T and Berkeley

Open Architecture

NFS, Ethernet, X.25 and SNA

Up to 17 GB Storage

INTRODUCING THE HARRIS HCX-9 COMPUTER.

It's no fairy tale.

Our new UNIX™-based Harris HCX-9 eliminates your I/O bottleneck. Because it can serve more users and execute more transactions, with faster response. That should make you whistle while you work.

The HCX-9 features the industry standard VMEbus with enhancements for faster I/O throughput. It also supports standard communications through local and wide area networking, so it will fit in with your current systems.

We offer a host of systems and application software that are nothing to sneeze at.

And it's easy to port software to the HCX-9, if it's not already in our catalog. In short, the new Harris HCX-9 is a full function system with a small footprint, an impressive growth path and a name behind it you can believe in.

So don't be bashful. Find out how the HCX-9 can meet all your computing requirements at a price that won't make you grumpy.

For information write D.S. Coller, Harris Computer Systems Division, 2101 W. Cypress Creek Road., Ft. Lauderdale, FL 33309.

Or call 1-800-4-HARRIS, ext. 4025.

Then it's off to work you go.

HIGH PERFORMANCE COMPUTER SYSTEMS FOR THE WORLD'S MOST DEMANDING USERS.

HARRIS

ENTER 612 ON READER CARD

Consider the Source... Your All-in-One Reference to the UNIX® Systems Market

/usr/group
The International Network of UNIX Users

UNIX® 1987 DIRECTORY PRODUCTS

Our coverage speaks for itself. /usr/group's 1987 "UNIX Products Directory" is the only independently published directory of UNIX products and services.

Our numbers speak for themselves. More than 3,000 UNIX-specific products and services, from over 800 companies —including packages from 170 international vendors.

1,587 vertical and horizontal software packages

383 hardware systems, from PCs to super-computers

Contact information for 36 UNIX user groups worldwide

123 consultants, including 42 designers of application software

UNIX is a registered trademark of AT&T

Member price is $25. Non-member price is $50. Overseas shipping extra. Bulk purchase rates available. To order by telephone (MasterCard or Visa), call (408) 970-4966. Or, write:

/usr/group
The International Network of UNIX Users
4655 Old Ironsides Dr., #200
Santa Clara, CA 95054

ENTER 567 ON READER CARD

CONTENTS

JUNE 1987 SPECIAL EDITION

DEPARTMENTS

We will consider for publication all submitted manuscripts and photographs, and welcome your articles, photographs and suggestions. We cannot be responsible for loss or damage. This magazine is not sponsored or approved by or connected in any way with Digital Equipment Corporation. "DEC" is a registered trademark of Digital Equipment Corporation. Digital Equipment Corporation is the owner of the trademark "DEC" and is the source of all "DEC" products. All materials presented are believed accurate, but we cannot assume responsibility for their accuracy or application. DEC PROFESSIONAL Magazine ISSN 0744-9216 is published monthly, plus two issues in the spring and fall, by Professional Press, Inc., 921 Bethlehem Pike, Spring House, PA 19477. Printing and binding by R. R. Donnelley & Sons Company. Subscriptions are complimentary for qualifed U.S. and Canadian sites. Single copy price, including postage, $4. One year subscription rate $30 in the U.S. and Canada; and $60 foreign. All orders must be prepaid. Second Class postage paid at North Wales, PA, and additional mailing offices. POSTMASTER: Send all correspondence and address changes to: DEC PROFESSIONAL, P.O. Box 503, Spring House, PA 19477-0503.

DECPRO EXTRA

Publisher: Carl B. Marbach
Editorial Director: R. D. Mallery

Editorial

MANAGING EDITOR Linda DiBiasio
ASSOCIATE EDITOR Bruce Feldman
SENIOR TECHNICAL EDITOR Al Cini
EAST COAST EDITOR Charles Connell
WEST COAST EDITOR Philip Naecker
BACK END EDITOR John C. Dvorak
C EDITOR Rex Jaeschke
DCL EDITOR Kevin G. Barkes
MICROVAX EDITOR David W. Bynon
NETWORKING EDITOR Bill Hancock
RSX EDITOR Ralph Stamerjohn
SPECIAL EDITOR Victor J. Chorney
UNIX EDITOR Lori A. Snyder
COPY EDITORS Karen Detwiler, Pamela F. Fullerton
EDITORIAL ASSISTANT Anne Schrauger
CONTRIBUTORS Vasudev K. Bhandarkar, Matt Bishop, James Brunet, Richard B. Curtis, Harry J. Foxwell, Oren Ben Kiki, Mark Krieger, P.P. Lau, P.J. Plauger, Naaman Reiz, Bob Saunders, Robert Segall, Joseph Zitt

Design

DESIGN/PRODUCTION DIRECTOR Leslie A. Caruso
DESIGN/PRODUCTION ASSOC. Ruth Ann Leiby
ART/PRODUCTION ASSOC. Timothy M. Kraft
ART ASSISTANT Sue Ann Rainey
PRODUCTION ARTIST Richard G. Kortz
SENIOR TYPESETTER Joseph E. Hohenwarter
TYPESETTING/PRODUCTION MaryEllen Springer

Circulation & Administration

VICE PRESIDENT Peg Leiby
CIRCULATION DIRECTOR Mary Wardlaw
CIRCULATION MANAGER Margie F. Pitrone
CIRCULATION FULFILLMENT Ruth Henderson, Claire Hollister, Joann Ness, Donna Schmidt
ACCOUNTING Andrea Beneke
COMPUTER SYSTEMS Kevin Kennelly, Ruth Mermelstein
ARIS MANAGER Bonnie Auclair
MARKETING SERVICES (215) 542-7008 Mary Ann Browarek, Cathy Dodies, Lori Goodson, Dan Mainieri, Patricia McCauley, Kim Slackway
ASSISTANT TO THE PUBLISHER Jan Krusen

PROFESSIONAL PRESS, INC.
For information on how to contact your sales representative, see page 65. Editorial, Advertising Sales, and Executive Offices at 921 Bethlehem Pike, Spring House, PA 19477. (215) 542-7008. TWX 910 333 9522. FAX (215) 628-2845. Easylink 62805174. ARIS (Automated Reader Information Service) (215) 542-9458. Additional Editorial Offices: East Coast Office at 5 Militia Dr., Suite 106, Lexington, MA 02173. (617) 861-1994 West Coast Office at 3011 N. Mount Curve Ave., Altadena, CA 91001. (818) 791-0945.

The Silent 700™ Data Terminal Series from Texas Instruments.

Next time you take off, take the DEC™ connection.

Talk about convenience for the DEC user. TI's TravelMate™ 1200 is display, printer and communications all rolled into one lightweight, go-anywhere package that emulates a VT-100™ video terminal. With it, you can access DEC computers from just about any remote location as long as there's a phone handy.* Perfect for on-the-go DEC communications.

The TravelMate 1200 VT-100 emulation cartridge also includes auto access features such as stored phone numbers and log-on sequences. This actually makes your TravelMate-to-DEC communications easier than using a phone. And the VT-100 emulation cartridge can be programmed to fit individual applications within your DEC operating environment.

The TravelMate 1200 with VT-100 emulation capability also carries an impeccable pedigree — a 15-year heritage of reliable, rugged design and dependable operation. After all, it's a

member of the Silent 700 Series of Portable Data Terminals family from Texas Instruments. It's definitely a terminal you can trust.

So next time you have to take off, take the DEC connection.

For more information on the Silent 700 Series, TravelMate and VT-100 emulation, call toll-free **1-800-527-3500.**

TEXAS INSTRUMENTS

ENTER 610 ON READER CARD

*In locations where a phone jack is not accessible, the optional acoustic coupler may be required.

P.J. Plauger

GUEST EDITORIAL

A Unification Of UNICES

A clowder of cats, a gaggle of geese, a pride of lions — the English language has a rich assortment of irregular names for collections of animals. Urbanization has driven many of these names into the foothills of trivia, only to be replaced by parlor neologisms such as "a tintinnabulation of teenagers."

The name "UNIX" also is a neologism, coined by Brian Kernighan in a Bell Laboratories parlor called the Murray Hill cafeteria. Originally it designated a specific PDP-7 operating system, which provided for a single user the nicer features of the multiuser system MULTICS. It grew, however, to name an irregular assortment of operating systems on different machines, each with its own rich collection of utilities. We now are reminded with cloying frequency that "UNIX is a trademark of AT&T Bell Laboratories," lest anyone suspect that it is a generic name for any operating system with a "shell" command interpreter and a "fork" system call.

"UNIX" still designates an irregular assortment of operating systems. Even if you leave out all the "UNIX-like" systems, such as IDRIS and UNOS to name just two, you still are faced with considerable variety among the "genuine" flavors. This is the heritage of all those years when small companies commercialized UNIX and added value by adding variety, before AT&T jumped into the market with both feet. The marketplace is sufficiently Balkanized that you still can't walk into a retail computer store and find a shelf of UNIX software for a 68000. There are plenty of UNIX systems on the 68000, but still too many media formats, object file encodings, and system call conventions to permit a standard packaging.

Order is emerging out of this chaos, however, thanks to standardization efforts on several fronts. The commercial UNIX users' group, /usr/group, deserves credit for pioneering these efforts. By 1984, it had agreed on what library functions must be available to an applications programmer writing in C for a system to call itself "UNIX-like." ANSI X3J11 appropriated much of this library as the standard environment for C on any operating system — after three years of hard work it has put out a C standard for formal public review. IEEE P1003 took the UNIX-specific residuum and produced POSIX, a specification for UNIX-like systems that starts with the system calls and is working steadily outward.

Meanwhile, back at the ranch, AT&T has produced a "System V Interface Definition," and is working to enforce a broader base of commonality among releases and implementations of System V UNIX. And in Europe there is X/OPEN, a standard that begins with UNIX and goes well beyond it into the realm of internationalization issues.

Having an assortment of standards is worse than having an assortment of implementations, of course. But there has been a remarkable degree of cooperation among the various groups named above. Where there is overlap, there is very good agreement. And the very multiplicity of active parties keeps any one group from dictating what shall be standard UNIX — even AT&T lacks the clout to evolve its SVID in a vacuum.

I will not opine that the era of abundant applications for UNIX is just around the corner — I have heard that speech made annually for the last decade. I have observed, however, that these standards are stabilizing the marketplace in a way that permits a healthy assortment of UNIX variants to thrive. By defining clearly what we all agree constitutes a UNIX-like system, we bless the useful variations that stay within the rules.

Thus, DEC can sell ULTRIX, its own blend of Berkeley enhancements and VAX-specific packaging, with no need to apologize. I find that it is the system of choice for many dedicated DEC customers who prefer one-stop shopping. And we at Whitesmiths can continue to enhance IDRIS for real-time and embedded applications, knowing what we must keep constant to stay in the UNIX community. If customers know they can port C source codes and programmer expertise among standard conforming systems, they breathe easier.

UNIX well may remain a collective noun. It continues to inhabit a market niche that rewards innovation more than uniformity. But standardizing at the current levels already is producing a new collective — a unification of UNICES. ■

Dr. P.J. Plauger is the founder and president of Whitesmiths, Ltd., a developer of compilers and computer operating systems. A former member of the Bell Laboratories technical staff during the development of C and UNIX, Dr. Plauger is the co-author of several programming texts, including Elements of Programming Style *and* Software Tools. *He currently is secretary of the ANSI X3J11 C Standards Committee.*

CONNECTIVITY NOW.

NOW THERE'S AN EASY, LOW COST WAY to connect all your computers and share any data anyone needs... all with one powerful software package.

BLAST IS HIGH-PERFORMANCE COMMUNICATIONS software for 100% error-free connectivity among many different computers and operating systems. More than just VT100 terminal emulation and file transfer, BLAST gives you the fast, automated data transfer your company needs NOW.

FROM PC's TO MINIS TO MAINFRAMES ...from MS-DOS to VAX to UNIX or XENIX, from laptops or Macs to big IBMs, BLAST lets you send any data... binary or text... even unattended at night, and always guaranteed 100% accurate!

EASY-TO-USE NEW PC BLAST II is a snap for everyone in your company, with Lotus-style menus and simple scripts to automate functions so that just a keystroke moves data between computers.

BLAST'S "INDUSTRIAL STRENGTH" GUARANTEES faster, more efficient, and more reliable data transfer than Kermit or Xmodem products, especially through today's obstacle courses of line noise, satellites, PBXs, and X.25 networks, reducing costs as much as 50%.

CREATE YOUR OWN NETWORK for any project needing data transfer or distributed processing... like order entry, sales data collection, financial,scientific, medical, or others. BLAST's scripts set up automated or repetitive tasks.

BLAST®

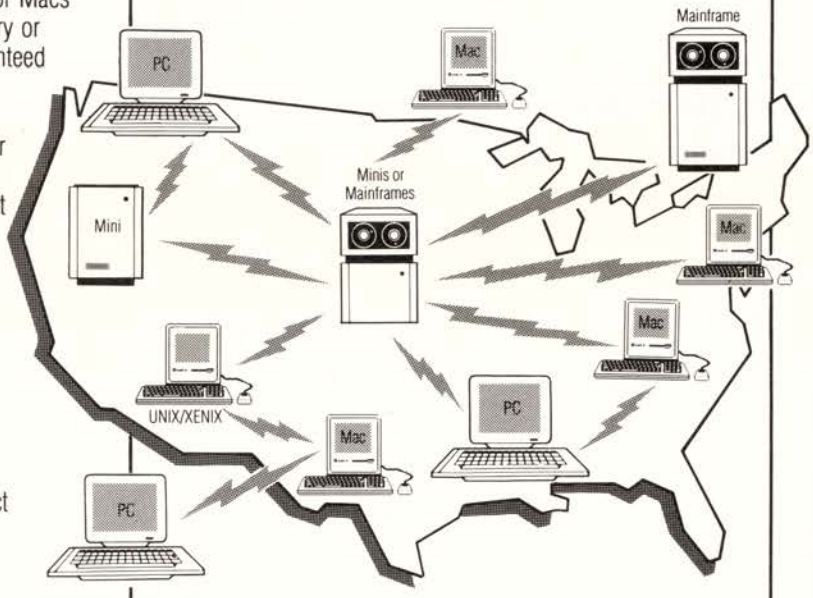
(800)-24-BLAST

or (504) 923-0888; Telex 759985

Over 20,000 users rely on BLAST in Fortune 1000 companies such as RCA, EXXON, GM, GE, NASA, and FORD.

NOW YOU CAN HAVE A BLAST TOO! BLAST is now available from leading software dealers and distributors.

Prices start at $250/PC (Site licensing available). Just call your local dealer to order BLAST. Or, call us at (800)- 24- BLAST for dealer information.



PCs & Laptops- MS-DOS, CP/M, Macintosh, Xenix. **Minicomputers and Mainframes- IBM** VM/CMS, MVS/TSO; **DIGITAL (DEC)**- VAX/VMS, PDP/RSX, RT-11, TSX; **WANG VS; DATA GENERAL**- AOS/VS, RDOS, etc; **HEWLETT PACKARD** 3000/1000; **PRIME**/PRIMOS; **BURROUGHS**- BTOS; **CONVERGENT TECH.**-CTOS; **UNIX/XENIX**; Sun, Harris, AT&T, etc.

® BLocked ASynchronous Transmission

ENTER 542 ON READER CARD

Future Directions Of UNIX

By Mark Krieger

UNIX Will Continue To Struggle In The Commercial Computing Sector, But Dominate Certain Key Markets.

Perhaps the single, most dramatic, long-term development in the future of AT&T's UNIX operating system is that, after years at the center of controversy, UNIX soon will become a non-issue.

Quite simply, UNIX will become an important, permanent fixture in the computer marketplace, and the long, acrimonious debate over its merits and demerits will die away. What will remain is a single, standardized version of UNIX that will be a dominant influence on certain key markets, but which will continue to struggle to make headway in the commercial computing sector.

Along with the cessation of hostilities, the following developments are likely to occur:

1. UNIX will become a commodity item.
2. UNIX's marketing strength and market share strength will be in four primary markets — technical, university, OEM and federal government computers, and, especially, in sophisticated technical workstations.
3. UNIX will not fare as well in the commercial sector, although it will come to some kind of compromise with DOS.
4. A UNIX standard will be achieved, based on a blending of the more popular versions.
5. The key players in the UNIX market will (not surprisingly) be Digital Equipment Corporation, AT&T, IBM, and Sun — but not necessarily in that order!

UNIX As A 'Commodity' Item

As incredible as that might seem to some, it's true. Market forces and user demand are standardizing UNIX to such a degree as to make it a prepackaged, one-size-fits-all operating system. In this respect, UNIX will become no different from many of the other components of today's computers, including disk drives, microprocessors, and memory chips.

Examples already abound. Sun Microsystems, Inc., the Mountain View, California, technical workstations manufacturer, probably is the most high-profile example of this emerging "off-the-shelf" attitude.

Sun designers decided on the following technical characteristics for their first family of workstations: an open architecture, strong industry standard networking, and strong graphics capabilities. In addition, Sun needed to make the product appealing to the CAD/CAM environment, have a wide variety of programmer productivity tools available, and possess a built-in appeal to OEMs.

Inevitably, the technical and marketing criteria lead them to UNIX. And, in keeping with the off-the-shelf theme, an outside independent porting house supplied Sun with a UNIX Version 7 operating system for its first product, the Sun-1. Of course, the current Sun-2 and Sun-3 systems run an internally developed Berkeley-devised UNIX, including many extras, such as Sun's Network File (NFS) and support for TCP/IP. But even those are closely allied with UNIX.

The commercial market continues to be the toughest nut for UNIX to crack.

Sun certainly is not alone in realizing this strategy. Masscomp — which adds special emphasis on real-time applications — Pyramid Technology, Convex Computer, and at least a dozen other computer vendors saw UNIX the same way: It gave them a quick, easy, and standard platform upon which to build their products.

You see, the fact that it's UNIX that these firms are using will become less important. UNIX is a means to an end, that end being a low-cost, easy-entry vehicle into the computer market. This utilitarian approach to UNIX will only increase over time.

Major Markets, Product Catagories

The commercial market continues to be the toughest nut for UNIX to crack. But, in the end, UNIX's success or failure in commercial computing may not matter, as the technical, university, and federal government computer markets take to UNIX on a grand scale. In particular, technical workstations will continue to be the most promising market for UNIX-based systems.

Quite simply, the reason is performance. For many technical applications, an IBM PC AT simply isn't enough. IBM itself acknowledged this need early last year, with the introduction of its own UNIX-based, technical workstation, the IBM RT PC.

More important for UNIX's future is the continuing demand for higher performance workstations, even among more mainstream computer users. It is not unusual, for instance, to find UNIX workstations in some Wall Street brokerage firms, because the application demands more processing power than an AT delivers. This need for ever more powerful workstations assures UNIX a primary role, as DOS-based systems cannot deliver the same power and complexity that the sophisticated workstation user is beginning to demand. The development of *Lotus 1-2-3*-like packages for UNIX workstations will accelerate this trend.

UNIX In The Lab

UNIX's appeal to the technical computing user is broader than just workstations. The operating system's networking capabilities, development tools, and portability have won it a leading position in the technical computing market.

AT&T planted the seeds for UNIX's popularity in the technical community some 10 years ago, when it made the first low-cost licenses available to universities. University graduates have carried their taste for UNIX into the scientific and engineering research laboratories of major U.S. corporations, where it has become a virtual standard.

In the near future, the university market will contribute tremendously to UNIX in the areas of windowing and user interface technologies. For example, a sophisticated UNIX windowing system, called X Windows, has been developed jointly by MIT and DEC and adopted by many manufacturers.

UNIX is not winning the hearts and minds of the business user. From

Not suprisingly, the UNIX market will belong to the companies with three initials: DEC, AT&T, and IBM.

our vantage point as software distributors, we at UniPress Software see little activity in such applications as UNIX-based accounting systems.

However, UNIX's success in the technical market ultimately could have a coattail effect into commercial markets, though this possibility remains unclear.

UNIX In Business

History, however, appears to be on UNIX's side. Many major developments in computer technology originated in the technical community, then gradually found their way either in part or in whole into government, small business, and the Fortune 1,000. The minicomputer and relational database management systems (RDBMS) are two prime examples. Even the concept of the PC can be traced to the technical community, where scientists and engineers frequently have dedicated minicomputers at their disposal.

The secret for success on its own in the commercial market will be support, or making UNIX so simple that little if any support is needed. A great deal of hand-holding from local sales or support representatives, combined with a front-end menu system, also would help. Unfortunately, we see neither development occurring within the UNIX community in a big way very soon.

Only Tandy, with its Model 16 and Model 6000, Altos, with its line of XENIX-based systems, and NCR with its Tower, have succeeded with commercially-oriented UNIX systems. Tandy, for one, apparently has done well at providing the level of service and support required.

UNIX may not need to do accounting to be successful in the commercial computing environment, however. There remains one other approach to the business user that some commercially-oriented UNIX vendors will try.

UNIX will form a background operating system for heterogeneous networking environments. In this mixed environment, most users would have DOS-based PCs on their desks, while a UNIX-based supermicro or mini would operate in the background, providing PC users with network, file sharing, printing, and other auxiliary computing services as needed.

Such a scenario might be the best of both worlds. UNIX's networking capabilities are far superior to those available under DOS, and include such industry standards as Ethernet and TCP/IP. Moreover, UNIX always has been quite adept at iterative numeric processing — just the kind of tasks that would be offloaded onto the UNIX-based background servers.

This mixed environment also would allow for low-cost incremental growth of the network, as users could mix and match processor types and sizes. Everything from micros to supercomputers could be added as needed. And because UNIX systems are inherently multiuser, sometimes it only would be necessary to add a new terminal, not a whole new system. DOS-based networks do not provide these features, and likely never will.

On the other hand, DOS will continue to hold its firm grip on the PC and desktop workstation for business users.

First, applications software for the DOS world is still more sophisticated than for UNIX. Second, the big players in the PC game have done virtually zero to promote the use of UNIX or XENIX (Microsoft's version of UNIX) on PCs. Third, DOS is growing up and getting more UNIX-like with every release.

Uncle Sam And UNIX

The commercial market may never take to UNIX, but the federal government is going for it in a big way. In fact, it soon will shape up as one of the major markets for UNIX hardware and software.

The federal government's interest in UNIX is just beginning to percolate. More and more, government bids specify UNIX as a requirement. The list of government UNIX users reads like a *Who's Who* of federal agencies, ARPA, NSA, the U.S. Army, the Bureau of Standards, the FBI, to name a few.

In some cases, federal agencies have chosen to go with UNIX over DOS. Some AT users at the NSA, for example, are said to use XENIX rather than DOS because the users desire the more sophisticated facilities available on a UNIX system.

The federal government's desire for UNIX is rooted in two basic attributes, the same two that appeal to many other UNIX users — portability and standardization. These two issues, more than anything, helped the federal government to decide on UNIX.

Further developments on the UNIX standards efforts also will occur. At this time, the three major versions of UNIX, AT&T's System V, Berkeley 4X, and Microsoft's XENIX, are closer together than ever before. Moreover, they continue to move closer with each major new release.

The inevitability of the movement toward a single UNIX standard was sealed last year, when Sun announced it would work with AT&T to make its Berkeley 4X UNIX compatible with the

emerging System V standard. Since that time, two standards, POSIX in the U.S. and X-OPEN based in Europe, have been working towards a single UNIX standard.

It also is clear that the fertilization will work both ways. AT&T is picking up some of the features and utilities in Berkeley UNIX that have made it popular with technical users, while Berkeley UNIX in turn is moving toward System V, so its users will have access to System V-based software.

Despite the availability of a System V-compatible version of XENIX, called XENIX V, many users will go on believing that UNIX and XENIX are not the same. But that's no matter. The differences between XENIX and UNIX are minimal and grow less every day.

Porting from one version to another, thereafter, will become increasingly easier for the software developer over time. Meanwhile, end users are not likely to be able to tell the difference between versions for much longer, as user interfaces are standardizing as well.

Not surprisingly, the UNIX market will belong to the companies with three initials: DEC, AT&T, and IBM. But not necessarily in that order!

For its part, DEC will remain a passive player in the UNIX market, retaining its preference for its proprietary VMS operating system. It will continue to be a major player in the UNIX market, nevertheless, but more out of tradition and user preference than as the result of a direct, concerted marketing effort. That stance appears to belong to IBM right now.

AT&T, of course, will remain an important and dominant force in UNIX as well. After all, it invented UNIX. But one suspects that for AT&T, UNIX is more of a means to an end, that end being a fast entry into the computer market, and not much more.

UNIX And IBM

Technical workstations are key, for it is there that IBM has launched its first concerted effort with the RT PC.

Several outward signs indicate IBM's inward determination. Since the RT introduction, IBM has made a strong presence at important UNIX trade shows.

Most impressive of all has been IBM's willingness to provide technical data on the RT. A detailed technical journal that describes many of the design criteria and inner workings of the RT is available from any sales office at a nominal price. The book provides a level of technical detail not normally provided by competing vendors.

Finally, IBM has priced the RT aggressively, although not as aggressively as Sun. Still, an RT development system with multiuser capabilities, a 150-MB hard disk, a streaming tape drive and graphics capabilities, costs not much more than a comparable Sun, Apollo, MICROVAX II, Masscomp, etc.

Still, the RT is not without need for some improvements. Networking is one area where IBM lags far behind market leaders Sun and Apollo. Windowing is another. IBM is aware of these shortcomings, and we expect that the computer giant will address these needs shortly.

THOSE WHO THINK that UNIX needs the commercial user to succeed are dead wrong. UNIX is here to stay, and will continue to have a profound influence on the technical, university, and government computer markets.

Nevertheless, UNIX offers the commercial user certain abilities that DOS never will. Among them is the ability to mix and match processor types and sizes across the network; DOS-based networks simply don't work that way.

UNIX's dynamic networking capability alone will encourage some commercial computing sites to use UNIX-based systems as a back-end network, at the very least. So, in a way, UNIX may succeed in commercial computing after all, only in a much quieter way than its proponents originally had assumed. *—Mark Krieger is president of UniPress Software, Edison, New Jersey.*

TEN/PLUS

By James Brunet

A UNIX Environment For The Rest Of Us.

UNIX has been touted as a superior operating system on one hand, and on the other, portrayed as a hard-to-use, unfriendly system totally unsuitable for use in environments with large numbers of casual users.

TEN/PLUS, a software product from Interactive Systems Corporation, may decisively tip the balance of the debate by providing a UNIX environment that is easy for novices to use, yet supplies all the power demanded by the most sophisticated user.

What Is *TEN/PLUS*?

TEN/PLUS consists of a user interface that includes a full-screen editor and several applications, plus a User's Toolkit Package and a Programmer's Tool Kit, which together comprise the *TEN/PLUS* Development Tools. Additional optional *TEN/PLUS* applications include electronic mail, networking, and multiterminal emulator packages.

The *TEN/PLUS* Editor

The user interface is the heart of *TEN/PLUS* and, of its components, the full-screen editor is most important. The editor's prominence is based on the assumption that the most common user tasks will revolve around various forms of text editing. Based on the "Rand" editor, the *TEN/PLUS* editor presents as many as 20 lines of text, 78 characters wide, within an editing field delineated by a rectangular box drawn on the screen. (See Screen 1.)

The most common text editing tasks are invoked by using the *TEN/PLUS* special functions. (See Figure 1.) The *TEN/PLUS* functions are intuitive and easy to use; casual users can depend on these functions to perform the bulk of their duties. As an example, you can use the ZOOM-IN and ZOOM-OUT functions to move through files and directories, use PICK-COPY to pick up a block of text, return to the original file with a combination of ZOOM-IN and ZOOM-OUT, and use PUT-COPY to drop the "picked-up" block of text into the original document. (See Screens 2 and 3.)

More sophisticated users still can invoke a wider range of commands, including UNIX shell commands, directly. Invoking the ENTER command will cause a pop-up box — a small rectangular window — to appear, in which shell commands or advanced *TEN/PLUS* commands can be entered.

Two other methods of gaining access to UNIX shell commands also are available. First, operations can be defined on a general menu, which can be called up at any time. The menu can be customized by the user, or by the MIS staff for novice and casual users. New entries can be defined and old entries modified or deleted by editing a user profile form that is maintained on-line. The form defines the text that will appear in the menu entry, such as "Go to Home Directory," the entry type, and

the shell command sequence to be executed when the entry is selected. Therefore, any operation that can be accomplished by a sequence of shell commands can be defined as a menu item. Once the menu is set up, even novice users can make use of the most powerful UNIX facilities.

The other way to gain access to *TEN/PLUS* UNIX facilities is to invoke a subshell. Subshells can be invoked from the menu — the ability to invoke a subshell is a standard menu item — and the subshell replaces the screen editor until the subshell is killed.

Command Consistency

One of the most appealing features of the *TEN/PLUS* user interface is the command consistency. The meaning and method of invoking each command remains constant, regardless of the application or context. Thus, text formatting and manipulation is the same whether performed in a data file, a text file, or an electronic mail file. The user who masters *TEN/PLUS* in one easy application can transfer that knowledge to subsequent and more sophisticated applications.

Other User Interface Features

In addition to the editor, the user interface features a number of other useful components, including:

1. A profile helper. A helper is an application program, usually written in C or the *TEN/PLUS* Forms Language, that extends the editor so it can manage tasks specific to a certain kind of data. The profile helper lets you "tailor" *TEN/PLUS* to your specific needs and desires. One example might be the suppression of backup and utility files so that they don't clutter up your directories, yet still are available when needed. Another example might be if, at log-in, the electronic mailbox is opened and new mail is displayed.
2. A print helper, which allows you to format and print files on any printer available to the system, or alternately copy the file to an ASCII text file.
3. A file manager, which has the ability to view, create, edit, delete, and move files and directories.
4. A history display, which provides information relating to previous states of a directory or file and permits you to recreate it as it existed at a previous point in time.

The profile helper is particularly useful, letting you adapt the system to your needs and tasks instead of the reverse, as too often is the case.

Development Tools And Options

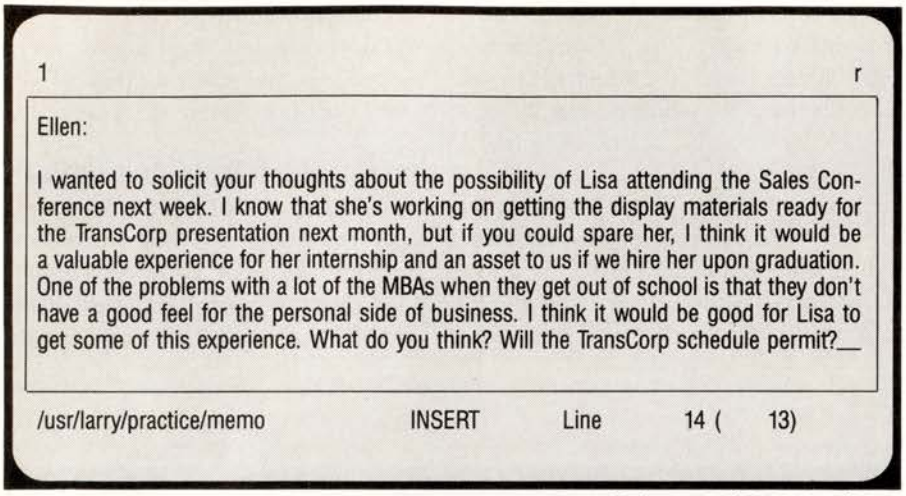
In evaluating *TEN/PLUS*, I did not use either the User's Tool Kit or the Programmer's Tool Kit. However, I understand that the Forms Language of the User's Tool Kit is the basis of the screens

FIGURE 1.

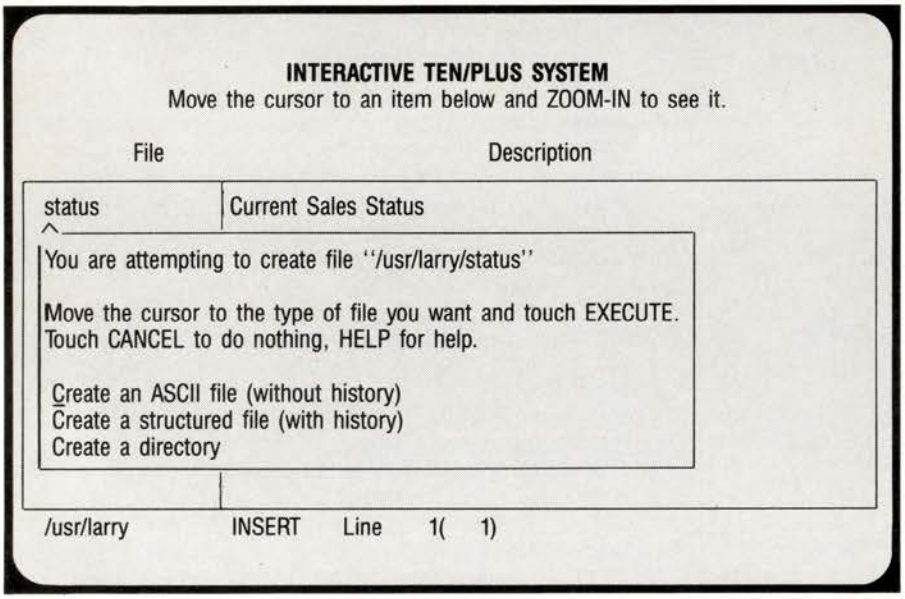
ZOOM-IN	Changes the display to a lower, or more detailed, level of a directory or form. For example, when in a directory to display a subdirectory or file listed in that directory, you would align the cursor on the appropriate entry and then invoke the ZOOM-IN command.
ZOOM-OUT	The reverse of ZOOM-IN, ZOOM-OUT changes the display to an outer, or more general, level of the directory, file, or form being viewed. By using ZOOM-IN and ZOOM-OUT, you easily can move through hierarchical file systems.
PICK-UP	Picks up an object — which can be a word, a line of text, a file listed in a directory, etc. — and puts it in a buffer so that it can be moved to another location.
PUT-DOWN	Replaces the object taken in PICK-UP into whatever location the cursor is currently pointing at. An object that is "picked up" and not "put down" is effectively deleted at the end of the editing session.
PICK-COPY	Operates like PICK-UP, except that only a copy of the object is "picked up," with the original object remaining in place.
PUT-COPY	Operates like PUT-DOWN, except that the object remains at the "top" of the buffer stack, so that subsequent copies also can be "put down."
INSERT	Creates a blank line or region; can be used to open up text blocks for the insertion of new material.
FORMAT	Reformats paragraphs, "cleaning up" margins after editing.
MENU	Presents the New Task Menu, a listing of general-purpose options that are applicable at all times, regardless of the current application.
LOCAL MENU	Presents choices that are applicable to the current application, such as Electronic Mail.

The basic *TEN/PLUS* commands are intuitive, "point-and-do" commands. The cursor is pointed at an object on the screen — whether data, text, or an entry in a directory — and the *TEN/PLUS* command is invoked. Many users will be able to perform all of their daily assignments using only the 10 basic *TEN/PLUS* commands.

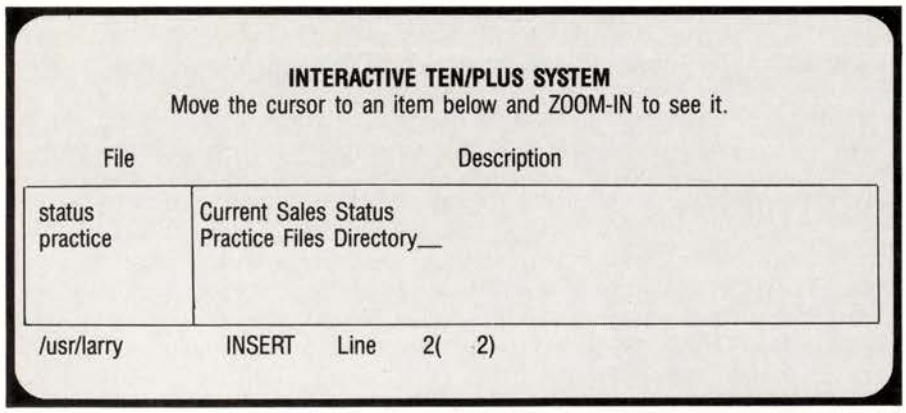
The 10 Basic* TEN/PLUS *Commands.



Screen 1. View of the* TEN/PLUS *editor.



Screen 2. Creating a directory under* TEN/PLUS. *The file name and description are typed on the directory display, and then a ZOOM-IN prompts the file creation menu.



Screen 3. A* TEN/PLUS *directory with two files created.

I was able to send messages via Usenet quickly and easily.

used in tailoring the User Profile. It appears to be a valuable tool and one that requires only very modest technical orientation to use well.

I did use some of the options, however, and they are outstanding. The INmail electronic mail system is superbly easy to use. The mailbox is displayed as a full-screen window, with fields indicating message subject, name of sender/recipient, and the date the message was sent. Electronic mail commands — send, reply, forward, delete, restore, file, and display new mail — can be invoked either from a local menu (see Screen 4) or by using *TEN/PLUS* commands. You may define subsidiary mailboxes in which to file messages. As an example, you might have a chron mailbox as a general file, a marketing mailbox, a technical mailbox, and a personal mailbox, each serving as a repository for a single type of message. INmail uses the same commands as the remainder of the *TEN/PLUS* package, and many text- or word-oriented users — salespeople, executives, secretaries, writers — seldom would be required to leave their mailbox.

Documents, including memos, letters, and reports, could be composed and edited in the mailbox and then either transmitted electronically or printed out on hardcopy and stored on-line.

When combined with *TEN/PLUS* INnet/FTP, the INmail option can provide communications to remote systems via a number of communications links. The networking links are transparent to the user; messages are sent and received in the standard INmail mailbox. (See

Screen 5.) I was able to send messages via Usenet quickly and easily.

Other options include a 3270 emulator, which provides access to IBM mainframes, and INterm, a multiterminal emulator that provides access to both UNIX and PC-DOS systems with *TEN/PLUS.*

UNIX Arrives In The Office?

After using *TEN/PLUS* extensively, including preparing the first draft of this article, I see it as being well-suited in the role of bringing UNIX to the office automation environment, or those environments that have a heavy mix of technical and non-technical users. It is easy to learn and easy to use. I learned the basics — enough to do most of my work — in about 20 minutes. It's a system that you can grow with, using the basics at first and adding the complexity piecemeal as needed.

The *TEN/PLUS* environment is not perfect. Some text processing operations seem clumsy under *TEN/PLUS* when compared to state-of-the-art word processors, although you ultimately can produce a document in whatever form you want. At times, the system seems slow in responding, a problem that seems to annoy technical users more than office workers. But, overall, the system deserves to be investigated by anyone looking for a front-end for UNIX-based systems.

TEN/PLUS is available on a number of UNIX ports, including ULTRIX, BSD 4.2, System V, PC/IX, AIX, IX/370 and Interactive's own IN/ix, as well as VMS. *—James Brunet is a technical writer from Los Angeles, California.*

TEN/PLUS
Interactive Systems Corporation
2401 Colorado Blvd., 3rd Floor
Santa Monica, CA 90404
(213) 453-8649
Price: Varies according to operating system and the application selected.

```
Subject:   Forwarded: First thoughts on the QuikSell account
To:        larry
  Cc:      brian

  Bcc:                                   Date: 18 Apr 1984 0957-PDT
From:      Janet Brown

I have extracted the juiciest parts here . . . let me know what you think.
^
          Electronic Mail
                                                  they want to accomplish with
Move the cursor to an item and touch EXECUTE.    soon with a couple of
Touch CANCEL to do nothing, HELP for help.       t down here my best
                                                  notes of my own on things we
  (1) Mail this message                          ls. Also, I have included
  (2) Reply to this message                      orrespondence from the
  (3) Forward this message                        a better idea of what has
  (4) Delete this message                        u have looked at all of this,
  (5) Restore deleted message                    questions you may have about
  (6) File this message in another mailbox       roach that will have a very
  (7) Show in-box and add new mail                if our proposals are very

  Size: 150 Lines     Sent by: janet at RALEIGH            Status:
/usr/larry/mbx/1                     INSERT      Line      6 (   150)
```

Screen 4. Display of* TEN/PLUS LOCAL MENU, *in this case, for Electronic Mail.

INTERACTIVE Electronic Mail

Subject	From/{To}	Date
Welcome to our newest brainstormer	john at BOULDER	17 Apr
Forwarded: First thoughts on the QuikSell account	janet at RALEIGH	18 Apr
Can you join us for dinner on Thursday?	{bobg at TULIP}	18 Apr
Forwarded: Did we miss the mark on Wharty?	janet at RALEIGH	18 Apr
Budget for the QuikSell project	{barb at FINANCE}	18 Apr
Benefits and Policies	john at BOULDER	18 Apr

/usr/larry/mbx   INSERT   Line   1(6)

Screen 5. Display of* TEN/PLUS *optional INmail mailbox.

Backup Issues And Considerations For The UNIX Environment

By Bob Saunders

Learning These Strategies Could Save You Time And Money.

For many UNIX system administrators, doing system backups is a seemingly laborious and unproductive task. After all, how often do users ask for files to be restored?

It has been said that losing files on a UNIX system is thought of almost as a by-product of working with UNIX. Due to the complex and often time-consuming task of restoring only a few files, rather than an entire filesystem, a level of UNIX competence is required, needless to say.

UNIX users who have used other operating systems find it difficult to accept the idea that destruction of a file might be permanent — not because files cannot be restored, but because so much time and effort is required that it is simpler to live without the data or recreate it as best you can.

With UNIX moving into the corporate and defense environments, this attitude is totally unacceptable, and must be changed.

To be reliable, backups need to be as automated as possible, and the restore interface must be as simple as possible. If users can initiate their own restores, system operator tasks can be reduced to simply mounting and unmounting media as required.

Why Bother To Backup?

You need to backup a system or part of a system only if it contains valuable data that cannot be recreated economically by any other means. Here are three good reasons for justifying the time and effort required to take effective system backups:

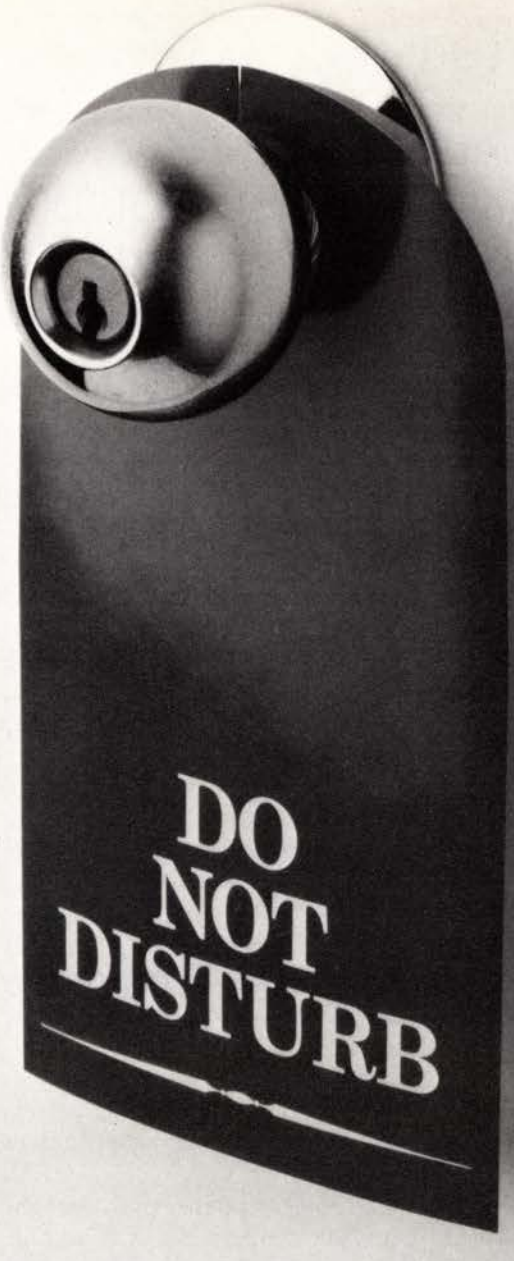
1. Computer hardware is more reliable today than it was 10 years ago. Dependence on this reliability improvement has led to complacency on the part of many system administrators. Head crashes still occur, which means that today you might loose 100 megabytes of data, while perhaps only several megabytes might have been lost with the technology of 10 years ago.
2. With more users and more data on a system, correspondingly more reliance is made on the system to provide results almost instantaneously. In a commercial situation, the inability to provide a prospective customer with the information he needs promptly might result in lost business.
3. A third very good reason to back up your system is to avoid a catastrophe such as a fire destroying the computer room. Media generated during backup operations can be rotated into offsite storage schemes. Although offsite storage administration can be a headache, it could save a company from ruin.

Basically, there are two types of backups: A full backup and an incremental (or partial) backup. A full backup ordinarily backs up all files and directories regardless of owner, permissions, size and placement within the filesystem. For the sake of simplicity, many system administrators choose to back up the entire system. Often it is easier to back up everything, rather than selectively choosing

VMS is the system of choice for most VAX™ users. It uses the VAX to the fullest, provides a reliable base for commercial and engineering activities, and has many available applications packages.

UNIX systems are increasing daily. UNIX tools have become standards for software development and UNIX applications are now common. From micros to workstations to supercomputers, most sizable installations will have at least one UNIX system.

How to get the best of both worlds? Buy an extra machine? Inconvenient, costly, one more thing to manage. Install a package that modifies VMS? Dangerous. Use a ''UNIX-like shell''? Not fully compatible. Replace VMS? Not likely.



UNITY® Operating System from HCR is the answer. An integrated package that runs under VAX/VMS, with comprehensive System V functions, no changes to VMS, complete UNIX programming tools including C and Fortran, transparent access between UNIX and VMS files, even ''uucp'' for linking to other UNIX machines. Our ''run-time'' option allows you to develop applications under UNITY and install binary copies on any VMS system. At last, you can have a compatible UNIX environment without disturbing VMS.

HCR provides full support and quality software products. Over ten years of UNIX experience has given us a world-wide reputation for developing systems software and advanced optimizing compilers. For more information, please call or write today.

The best of UNIX™ software without disturbing VMS.™

HCR

HCR Corporation

130 Bloor Street West
10th Floor
Toronto, Ontario
Canada M5S 1N5

Telephone (416) 922-1937
Telex 06-218072 HCR TOR
Fax (416) 922-8397

ENTER 528 ON READER CARD

Every version and derivative of UNIX provides a few utilities designed to back up the filesystems. All of them fall short of ideal backup tools and some are quite difficult to use.

those portions that *need* backing up. This can waste both backup media and system administrator time. For example, it isn't necessary to back up the "/ bin" directory, because it normally is possible to recover these files from the system software distribution media.

An incremental backup backs up files based on some selection criteria, the most common being modification time of the file (if the file has been changed within "n" days, select the file as a candidate for backing up, etc.). The major advantage of an incremental backup over a full backup is that, usually, much less data is written out to the backup media. A disadvantage of incremental backups is that a single incremental dump, in isolation, does not provide an accurate representation of the system. A number of incremental dumps is required. This can introduce an additional amount of administrative overhead in recording which backup media represents which part of the file systems.

The best strategy seems to be a combination of full and incremental backups. A full backup provides a reference point for future incremental backups. Then, subsequent incrementals back up only the data that changed between the full backup and the incremental backup.

How frequently backups should be performed depends on individual site requirements. Most sites choose a weekly cycle of five, six or seven backup operations. For example, a full backup is performed on Monday. Then, incremental backups follow on Tuesday through Friday, Saturday and possibly Sunday. On the following Monday, another full backup is performed. If system data changes frequently, once a day might not provide an adequate safety margin. Two incrementals per day might be the answer. Fairly static systems might get only one full backup and one incremental backup per week.

The 'Standard' Utilities

Every version and derivative of UNIX provides a few utilities designed to back up the filesystems. All of them fall short of ideal backup tools and some are quite difficult to use.

UNIX lacks any "real" tape-(diskette-) handling utilities. Most significant is the lack of removable media labeling. Without media label recognition embedded as part of the device driver software, labeling mechanisms have to be "invented" at the applications level.

The backup utilities available to the two UNIX communities will be considered here, the BSD 4.2 and System V offerings.

System V

System V offers four programs connected with system backups:

1. /etc/volcopy;
2. /etc/ff;
3. /etc/finc;
4. /etc/frec.

No data verification is performed by these programs. This might not show up as a problem until restore time.

"Volcopy(1M)" is used to copy (in or out) an entire filesystem. Essentially, the specified input device, which usually is the name of the filesystem being backed up, is copied to the specified output device. The copy is fast because no interpretation of selection is allowed. "Volcopy" differs from "cp" in that end-of-file on the output device is considered normal (end-of-tape) and causes a prompt for the next volume. The copy then resumes. Unfortunately, because no interpretation is performed, any filesystem errors (unallocated blocks, dup'ed blocks, etc.) also are copied faithfully. One file note: The input filesystem must be unmounted, or mounted read only. This is to avoid the possibility of a file being changed while it is being backed up.

The "ff(1M)" program performs the simple task of reading an input filesystem and essentially producing a table of contents. The table is composed of a filename and corresponding inode

FIGURE 1.

```
.          2
./lost+found          3
./sbmb.L28029         9
./SAVE 13
./e70476             21
./.SSLmacros         24
./sbmb.D28028        28
./fred  30
./sbmb.I00305        37
./stmp.L00304        41
./stmp.D00303        43
```

Example of "ff" output

Assemblers/Simulators/Compilers

Macro Assemblers/ Cross Assemblers for Series 3.0

Series 3 requires at least 256k of memory to run under MSDOS & CP/M86 and 54k TPA of memory to run CP/M80. Series 3 products have: Full Listing Control, Conditional Assembly & Built in Cross Reference. There is Unlimited Program Size, Number of Symbols and Macros. The Linkers output: Intel Hex, Extended Intel Hex, Tektronix Hex, and Motorola S19, S28, S37 formats.

Macro Assemblers/ Cross Assemblers for Series 4.0

Series 4 requires 512k of memory to run. These products have all the features of Series 3 plus: 32 Character Labels, Library and User Defined Sections. There is one Linker on Series 4 which outputs all the same formats as Series 3. Series 4 runs under MSDOS, UNIX, VMS and ULTRIX.

Simulator-Debuggers

The Simulator-Debuggers run with 256k of memory. The Simulator has 16 Breakpoints with optional Counter Field. The Symbol Table is limited only by the amount of memory. Buffers of 256 bytes may be opened for I/O ports. The Simulators are available for MSDOS, UNIX, VMS and ULTRIX.

C Compilers

The C Compilers support in-line assembly language and ROMable code, and includes the following: Macro Processor, full Floating Point support, complete Assembler, Linker, and Libraries.

	Macro Assemblers Cross Assemblers			Simulator-Debuggers		C Compilers	
	Series 3	Series 4					
Software Product	CP/M 80 CP/M 86 MSDOS	MSDOS AT&T UNIX	VMS Zeus ULTRIX	MSDOS AT&T UNIX	VMS Zeus ULTRIX	MSDOS AT&T UNIX	VMS Zeus ULTRIX
Super 8		199.50	995.00	149.50	499.50		
Z-8	199.50	199.50	995.00	149.50	499.50		
Z-80	199.50	199.50	995.00	149.50	499.50	500.00	2000.00
Z-280		299.50	1250.00				
Z-8000	299.50	299.50	1250.00				
1802	199.50						
6301	199.50	199.50	995.00	149.50	499.50		
64180	199.50	199.50	995.00			500.00	2000.00
6501	199.50	199.50	995.00				
6502	199.50	199.50	995.00				
65c02	199.50	199.50	995.00				
65c816		299.50	1250.00				
6800,2,8	199.50	199.50	995.00	149.50	499.50		
6801,3	199.50	199.50	995.00	149.50	499.50		
6804	199.50	199.50	995.00				
6805	199.50	199.50	995.00				
6809	199.50	199.50	995.00	149.50	499.50		
68c11	199.50	199.50	995.00				
68000,8,10	299.50	299.50	1250.00				
68020		399.50	1500.00			700.00	2500.00
8400/c00	199.50	199.50	995.00				
80452		199.50	995.00				
8044/51	199.50	199.50	995.00	149.50	499.50		
80515		199.50	995.00				
8080	199.50	199.50	995.00	149.50	499.50		
8085	199.50	199.50	995.00	149.50	499.50		
8086/88	99.50	99.50	1250.00				
8096	199.50	199.50	1250.00				
80186/286	199.50	199.50	1250.00				
83c351		199.50	995.00				
8748	199.50	199.50	995.00	149.50	499.50		
V20/30	199.50	199.50	1250.00				
Cops 400	199.50						
F8/3870	199.50						
NCR32	399.50						
NEC7500	199.50						
NSC800	199.50	199.50	995.00	149.50	499.50	500.00	2000.00
32000	399.50						

More formats available

ENTER 618 ON READER CARD

Name _____ (Please Print)

Company _____

Address _____

City _____ State _____ Zip _____

Phone _____

Product _____ Operating System _____

Series _____ Amount $ _____

Shipping $ _____ Total $ _____

Signature _____

Check ☐ COD (U.S. Only) ☐

MC/VISA # _____

Expiration Date _____

Educational discount available.

To order, call toll free in U.S. (including HI, PR and VI):

1 800 843-8144

In Colorado: (303) 369-5001

For more details, ask for a free brochure.

(Shipping is $20.00 per unit for overseas orders. Toll Free number does not apply to overseas. 2500AD pays COD charges.)

2500ADSOFTWARE INC

17200 East Ohio Drive
Aurora, Colorado 80017
(303) 369-5001

Offsite storage is a good way to avoid a catastrophe, at the price of an administrative overhead.

number, separated by a tab character (see Figure 1).

An inode is UNIX's way of representing files internally. Each file on a filesystem has a unique inode number. Inodes are allocated by UNIX dynamically when a file is created. Why "ff" is needed will become apparent later.

The "finc(1M)" program works much like volcopy, except that some limited file selection criteria can be given. Unlike volcopy, only a single output volume is possible. Again, the input filesystem should be mounted read only to ensure an accurate copy is recorded onto tape.

The "frec(1M)" program is the only file-level recovery program that can be used with volcopy- or finc-created media. As arguments, "frec" takes an inode number and searches the input media for a file with the matching inode number to extract. This is why the "ff" program is required. It is the only way to obtain a cross-listing of filenames and respective inode numbers.

A useful option to "frec" is that inodes to be recovered can be specified in an argument file. This can be useful for long lists of inodes that are to be recovered. It is sad, though, that "frec" is extremely cumbersome to use in its raw state. The "ff" program output cannot be used directly and quite a number of support "shell" programs are required to be able to use "frec" effectively.

Berkley "dump" and "restore" (versus System V's four) can be used to implement a fairly effective, though still not complete, backup strategy. The "dump" program performs in much the same manner as "volcopy", with a few improvements to boot.

First, "dump" automatically tracks its own use in a file called "/etc/dumpdates" (a previous version of the "dump" utility used "/etc/ddates", but this now is almost obsolete); "dumpdates" is kept in ASCII (another improvement over "ddates") so that an administrator can edit the log with an ordinary text editor.

The "restore" program will restore dumped files. A vast improvement of "frec" is that "restore" will allow an administrator to list the contents of a tape. When extracting a file, "restore" also places the file in its "correct" place in the filesystem, unlike "frec".

A very useful feature is the "restore" interactive mode. In interactive mode, an administrator is able to move around the files on the tape as though they actually were on disk. Files can be selected and placed on a list for extraction later using the command "extract".

Although "dump" and "volcopy" are improvements over their System V counterparts, they normally will be tools with which only an experienced UNIX guru would want to play.

'tar' And 'cpio'

These programs have been around for some time. Most systems have both; all systems have at least one, and they are by far the most universal method of software distribution. In their original forms, neither "tar" nor "cpio" could deal with more than one output volume. There are versions available that prompt for another volume to be mounted when the current volume becomes full. This is only a slight improvement, because the new volume is written from the exact point where the old volume failed. This might be halfway through a file and cause a problem if the old volume becomes damaged; the program cannot resync from the middle of a file so that volume (and any subsequent volumes) also will be "lost."

The "tar" program also has a number of other design flaws. It does not set an exit code, for example, making it difficult to detect if the tar failed within another program.

The "cpio" program has a huge security loophole. Protected files backed up by the system administrator can be un-"cpio"ed and ownership given to any user able to physically mount and read from the tape drive.

UNLESS YOUR DATA is of no value to you or to anybody else, you need to back it up regularly. Try to automate the process as far as possible; this encourages frequent use. A combination of full and incremental backups can provide a degree of data protection. The frequency of the actual backups must be determined by the value placed on that data and how often it changes.

The BSD backup utilities are superior to their System V counterparts, although both require a fair level of UNIX competence. No data verification is performed by either suite, so do-it-yourself efforts are required. Remember, a tape drop-out not detected during the write of data to the tape goes unnoticed until restore time. Input filesystems must not be allowed to change during the backup process or a corrupted version might be recorded.

Offsite storage is a good way to avoid a catastrophe, at the price of an administrative overhead.

Getting into the habit of providing effective backups for your users will save time and money in the long run.

—Bob Saunders is vice president of Customer Services at UNITECH Software in Vienna, Virginia.

CrystalWriter Plus

By P.P. Lau

A Multiuser Word Processing Program.

The most impressive characteristic of *CrystalWriter Plus* is the ease with which you can get started. An on-screen menu lets you create a new file or find an old one. (You also can get a file listing in any directory without exiting *CrystalWriter Plus*.) Function keys are displayed on the bottom of the screen. These displays stay with you throughout the creating and editing of a manuscript. They save a lot of memorizing and use of the manual.

Once into the program, you'll see how powerful it is. The word processing program has an "Object-based" formatting system. The formatting of documents is controlled by manuscript types (Syntactics calls these model documents) and "objects" within manuscript types. For someone who is used to embedded commands in formatting, this new program can be disconcerting. After using this system for almost one year, I still have a love-hate relationship with manuscript types and objects.

Manuscript Types And Objects

CrystalWriter Plus chooses to center manuscript formatting around manuscript types and objects. When creating a new file, you first must choose a manuscript type. Most line spacing functions then are predefined for that manuscript. Manuscript types for each file can be changed easily, so that the file is not confined to one format. Functions defined within manuscript types include vertical spacing (single, double, and triple space), right justification, and margins.

Objects within manuscript types further define spacing for such items as subparagraph, list, quotes, headers, and footers. Different objects can be called up and embedded when required within a file.

One difference between word processing and typing is that you can plan the layout of a document in word processing after the actual input. Nowhere is this more evident than in *CrystalWriter Plus*. Changing manuscript type is easy. You press Adjust-File, type in the new manuscript type, and press the Accept function key. To change the layout of a portion of a document, you must mark that portion of the document, Make-Object, and choose an object.

Manuscript types can take some getting used to. Once an effort is devoted to learning about different manuscript types and objects, it is easy to move around these predetermined formats. Frequently used manuscript types are: Plain, which is single-spaced unjustified text, and Report, which is single-spaced right-justified text. Right justification, however, leaves much to be desired. Often large gaps are left between words. Sometimes the first line of a block of indented text is impossible to line up on the left side. Other manuscript types, such as Pletter (for personal letter) or Bletter (for business letter) are superfluous.

Objects, however, are more useful. These include subparagraph (indented from left margin), quote (indented from both margins), and list (indented from left margin). The most useful are the various forms of header and footer. You can define header and footer for the entire manuscript, different ones for the odd and the even page, or ones appearing for one page only.

It is possible, of course, to modify these predetermined formats and save the modifications. A system administrator can define

A system administrator can define several manuscript types for other users on the system, so that most users do not need to learn the intricacies of manuscript types and objects.

several manuscript types for other users on the system, so that most users do not need to learn the intricacies of manuscript types and objects.

CrystalTypeset, a companion product of *CrystalWriter Plus*, does automatic page layout of documents such as letters, memos, proposals, reports and manuals. The user or system administrator uses manuscript types and objects to define standards for the appearance of different kinds of documents.

One major complaint that I have with manuscript types is that it is impossible to change top and bottom margins or paper length within a file. Once set, they are set for the entire manuscript. It is much easier to use embedded commands to change formats in the middle of a file.

Some useful formatting functions missing from *CrystalWriter Plus* include binding formats (different left and right margins on odd and even pages), whole page centering (for the title page or a one-page table), and newspaper-type columnar printing. [Editors note: Syntactics will offer parallel columns and newspaper-type columnar printing in a new version to be released this month.]

Moving Around The File

Moving around the file is easy. I mostly use scroll and home keys to move my cursor. The arrow keys on my terminals are sluggish (probably a function of the terminals). I can move to the top and bottom of the screen, top and bottom of the document, and the left and right edge of the screen with, at most, three key strokes. With the FIND PAGE command, I am able to move through a document page by page. The Find key allows me to find unique patterns in the document.

In order to set tab stops, enter into adjust-Tabs mode, at which point the cursor disappears from the text. A cursor along the ruler on the top edge of the screen appears.

Function Keys

The on-screen function key labels make *CrystalWriter Plus* one of the easiest word processing programs to use. There is no need to relabel keys with stickers or fumble with plastic overlays. The function keys allow you to perform many standard typing and editing functions, including copying and moving text, finding and replacing text, underlining and bolding, and saving files.

If the function keys are too cryptic for new users, there are two levels of on-screen help. Pressing Control-E gives you a two-line description of the current function. Press Control-E again, and a full-screen explanation appears. The Help function, however, appears to be an afterthought. Why not use one of the empty function keys instead of Control-E?

Some of the more useful functions include decimal tabs, find and replace, and footnotes. Footnotes can be created before the paragraph where the footnote reference appears, but are printed on the bottom of that page. Footnote numbers are incremented automatically.

CrystalWriter Plus also has two editing modes: insert and replace. The insert mode allows you to input new material while *CrystalWriter Plus* reformats the old material. Reformatting is done automatically when the file is saved, or it can be done manually when the Adjust function is used. The replace mode allows you to strike over old material. The function is accessed by pressing Insert or Ins-Char.

A problematic function is the delete. Once a block of material is marked, the delete key can erase the whole block without a trace. There is no way for you to recover from this mistake. A two-step delete process requiring you to confirm a delete will help. Another possibility is a de-delete function, with which you can recover a certain number of deleted lines. [*Editor's note:* The next version, to be released soon, will require you to confirm large deletions and will have an "undelete" function.]

Using some function keys on *CrystalWriter Plus* appears to be cumbersome at first. To center a line requires three keystrokes. However, after a minimal amount of use, keystroke combinations are memorized easily. I have my function keys set up on the numeric pads of my terminals, and function key combinations mostly are automatic. Macros give you a two-keystroke shortcut for any commonly used sequence.

Multiuser Environment

CrystalWriter Plus is designed for a multiuser environment. Moving documents to another user under a different login, however, requires exiting to UNIX and using the mail command. This puts too much burden on the office users. The same is true if a file has to be

> You can move in and out of the dictionary as often as needed, without having to save the file first.

erased. You must exit to UNIX to use the "rm" command. This presents a danger, especially in a multiuser environment. It is too easy to erase files because "rm" is unforgiving. This problem should be solved in the next release, according to Syntactics.

CrystalWriter Plus does not have any file management functions except for a file listing function. This is a problem because even in a small office, files quickly can accumulate to the hundreds. With a 14-character limit on file names, it is difficult to keep track of files. Editing multiple files is easy. The program allows you to open 12 files simultaneously. Although the program does not support windows, moving or copying text can be done by marking the text, calling up a second file, and using the Move or Copy function keys.

Printing

CrystalWriter Plus supports multiple printers. You can print one or more copies, or portions of a document. The program supports 10- or 12-pitch printing, but not proportional spacing. I have an HP LaserJet, and a Dataproducts dot matrix printer is attached. Printing to the LaserJet, however, can be adventurous, especially when headers and footers or forced page breaks are used. The program and the printer sometimes disagree on paper length, and unwanted page breaks appear. I also have had overlapping problems when printing two consecutive documents. The last page of one document can run into the first page of the next.

Chain printing several documents must be accessed through UNIX. This is an inconvenient and sometimes dangerous process. You should be able to chain-print documents without exiting to UNIX.

Dictionary And Mailmerge

A dictionary and a mailmerge function are indispensable in any professional word processing program. *CrystalWriter Plus* makes these two functions available.

The dictionary is great. It is an 80,000-word dictionary based on *Merriam Webster's Ninth New Collegiate Dictionary.* The dictionary can be accessed using the Find key. No program switching is required. You can move in and out of the dictionary as often as needed, without having to save the file first. Its versatility is the best I have used. You can check spelling for the entire document, only the part that is below the cursor, a single word, or only new additions to the document (since the last spelling check). The last option allows you to save time if the document has been checked before.

Once a misspelling is found, the program offers alternative spellings. You can use one of the suggested spellings, type in another spelling, ignore the suggestions, or add new words to the dictionary. New words are saved in a separate file and can be edited by the system administrator.

One missing function in the dictionary is a word count. This is important for writers, but probably is not useful in an office environment.

Medical and legal dictionaries also are offered as options. I have not used either product.

The mailmerge function offers most of the mailmerge functions in other word processing programs. Each mailing list can have up to 16,000 records, with a maximum of 255 characters and 40 fields per record. The next release will offer 2,000 characters and 90 fields. Maximum field length is 79 characters. Mailing lists can be created in *CrystalWriter Plus* or from an outside database program. *CrystalWriter Plus* even allows sorting of the list; the print menu lets you select a subset. Word processing programs are never intended as database programs, and I suggest creating the list in a database program rather than using *CrystalWriter Plus.*

Documentation

CrystalWriter Plus comes with a massive 300 + -page full-sized manual. While the writing is clear, it is imposing and difficult to handle. The company should consider separating the manual into a training section and a technical or reference section.

This is an excellent UNIX word processing program. It offers most of the features a multiuser word processing program should have. The over-reliance on UNIX functions and some problems with formatting detract from the program. While it's very easy to get started, learning to take advantage of the many features of this program can be more involving than many other programs. For anyone contemplating buying this program for an office, I would suggest allowing more than the usual time period for training. *—P.P Lau is a freelance writer based in LaVerne, California.*

CrystalWriter Plus
Syntactics Corporation
4655 Old Ironsides Dr., Suite 400
Santa Clara, California 95054
(408) 727-6400
Price: Varies with configuration; e.g., MS-DOS $495, MICROVAX II $1,795, Xenix $695

The *xtermcap* File

By Joseph Zitt, Naaman Reiz, Robert Segall and Oren Ben Kiki

Using Extended Terminal Capabilities.

Device-dependent functions traditionally have posed a problem for software developers. To use a terminal's full capabilities, programmers have had to include specific escape sequences. Therefore, the program could be run only on that single terminal. To include more terminals, developers either would be forced to limit the capabilities the program used, or would have to write cumbersome logic to control each different terminal.

UNIX-based software tries, whenever possible, to avoid specific references to devices. This, however, has had its cost: One of the most frequent criticisms of UNIX is that user interfaces often are drab and clumsy.

In the early 1970s, the developers of UNIX created the *termcap* file and a library of associated functions to handle some of the problems of terminal control. Using *termcap*, programmers could perform simple screen-handling functions (such as moving the cursor, clearing the screen, and underlining) through standard functions, independent of the terminal on which the program would be run.

SINCE THE CREATION of *termcap*, however, improvements in terminal capabilities have created a gap between the functions that are now standard on terminals and the properties that programmers could use. Color terminals have become commonplace, and most monochrome monitors can display text in a variety of intensities. Many terminals also can draw lines and boxes on the screen, and can hide a screen, then redisplay it after it has changed.

Termcap also has other built-in limitations. Since it was designed to work in octal, it cannot use any terminal codes that include the full eight bits (ASCII 128-255). It only allows two types of commands: boolean (indicating whether or not a capability exists) and codes (indicating the string that is sent to a terminal to perform a particular function). Commands cannot be defined in terms of other commands and cannot interact with each other. The size of *termcap* entries are limited and beyond the user's control, which allows very little room for possible expansion.

In developing *WWD/CHARM*, a C source application generator operating under UNIX, the programmers at WWD developed a new file, *xtermcap*, to handle the extended terminal capabilities. Using the *xtermcap* file, programmers can include additional terminal-independent functions in their software.

Capabilities of *xtermcap*'s exceed *termcap*'s limitations. The file and its associated libraries include, among their built-in capabilities, functions to draw lines and boxes on the screen, to change the displayed colors on color terminals, and to display different intensities and reversed video on monochrome terminals. The file also includes other necessary information about the terminal, such as whether it needs a blank space for each color change, and in which color the blank appears.

Remember how computers remembered? Mercury delay lines? Punched cards with 90 columns and round holes? Hand-wired magnetic cores? In case your memory needs refreshing, The Computer Museum would like to share its memories with you.

The Computer Museum Memory Poster
We have created a limited edition, 20″x32″ poster of the picture shown below. Printed in full-color, it includes an identification key to help you recall the memories you've forgotten. To get your poster, along with an information kit on museum membership, exhibits and activities, send a tax-deductible contribution of $25 or more to:
Memory Poster, The Computer Museum,
300 Congress Street, Museum Wharf,
Boston, MA 02210.
Please allow six weeks for delivery.

ENTER 620 ON READER CARD

YES! Please refresh my computer memories. A tax-deductible donation of $25 or more made payable to The Computer Museum is enclosed.

Name_____

Address_____

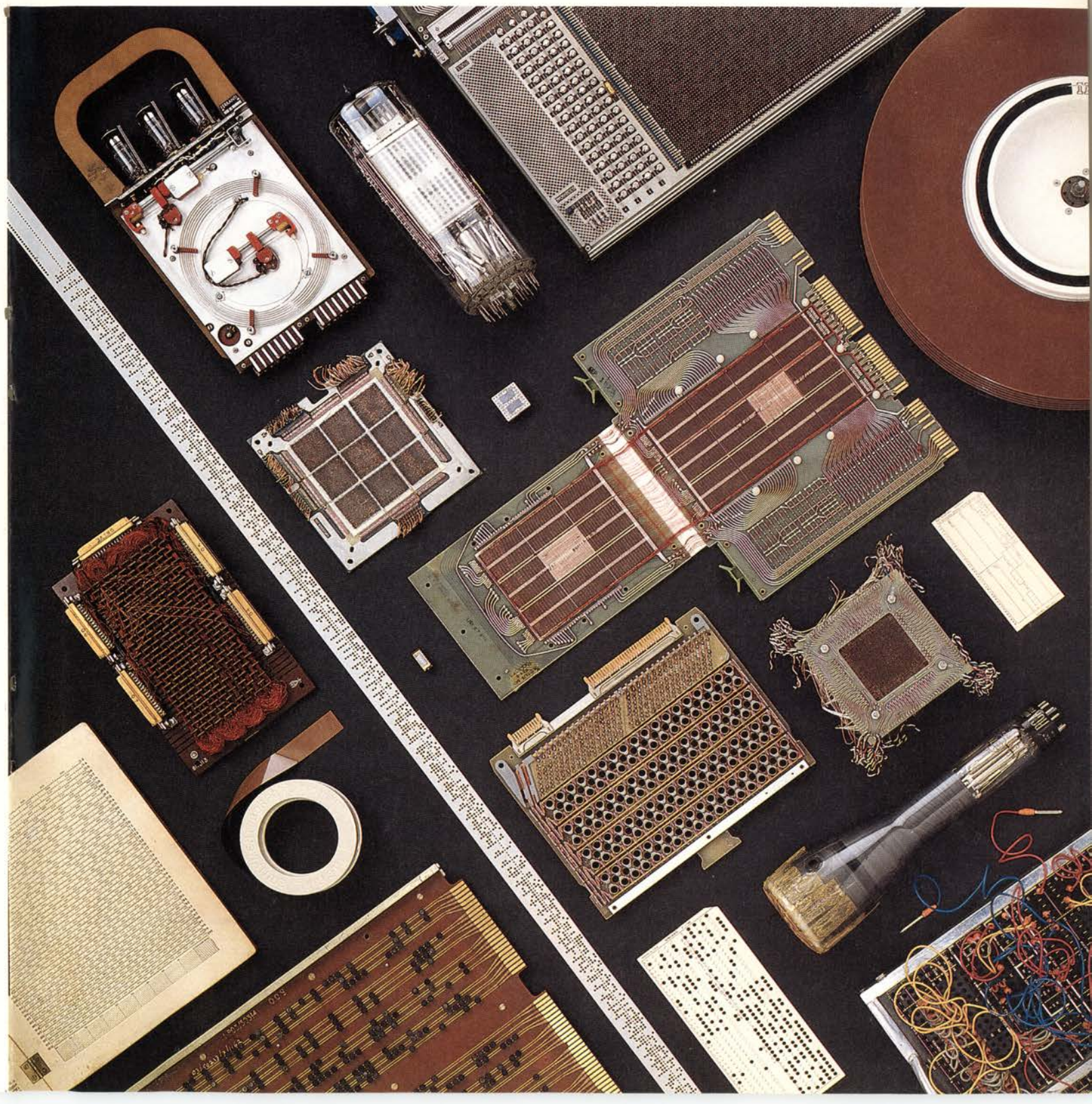
City_____ State____ Zip_____

The Computer Museum
Located on Museum Wharf
300 Congress Street, Boston, MA 02210
(617) 426-2800

E

COMPUTER MEMORIES FOR SALE

Special thanks to this publication, Scitex America Corp. (color separations), Grafik Communications, Ltd. (design), David Sharpe Studio (photography) and VM Software, Inc. (poster).



Capabilities of *xtermcap*'s exceed *termcap*'s limitations.

Since *xtermcap* is not limited to octal functions, it can include the full 8-bit character set. The user can establish a buffer size for reading the entry when the entry is read, allowing for a larger number of capabilities than *termcap*'s fixed buffer would hold.

Xtermcap also includes combinational commands, which can be defined in terms of other commands. Thus, for example, a command that displays green text on a black background could refer to another command that includes the specific terminal code for displaying green text.

The *xtermcap* information is kept in a different file from the *termcap* information. Thus, the *termcap* file can be kept unaltered. *Xtermcap* also can work in conjunction with *terminfo*, or other *termcap* variants used by other UNIX and XENIX systems.

Here is a brief overview of *xtermcap*'s structure and functions. A more complete description of *xtermcap* and its associated functions can be found in the *WWD/CHARM Technical Reference Manual*.

The Structure Of An *xtermcap* Entry

The *xtermcap* file is modeled after the standard UNIX *termcap* file, and uses a similar structure. Each *xtermcap* entry is preceded by a comment line, beginning with "#" to separate the entries. (There also must be two comment lines at the beginning of the file, and two at the end after the final entry, to allow for other definitions — such as plotters, printers, graphics terminals, and other peripherals — that are not discussed here.)

Each field in the entry ends in a colon (:). Several fields may appear together on a line. By convention, lines in the *xtermcap* file also begin with colons, except the entry name, which must begin with a vertical bar (|).

The first field in an entry contains the terminal's name or names. The names are separated by a "|". There is no limit on the number or type of names you can list for the terminal. The rest of the fields may be in any order.

Most *xtermcap* fields are in the following format:

command = value:

where *command* is a two-character string identifying the command, and *value* is the string or number that is sent to the terminal.

The characters in *value* are sent to the terminal as listed, with three exceptions (following C conventions):

\E sends an Esc to the terminal.
^ before a letter sends the control value of that letter. For example, **^A** sends the value of Control-A).
\: sends a colon (:) to the terminal. (This is necessary to distinguish it from the colon at the end of a command.)

For example, the field:

:hi = \E(:

in the *xtermcap* entry in Figure 1 indicates that the Esprit 3 terminal (or any color terminal) switches to high intensity display when it is sent **Esc (**.

Boolean commands (described below) do not use this format. Including the entry:

boolean command:

indicates that the command's value is true. Unless an entry specifies that a command is true, the default value is false.

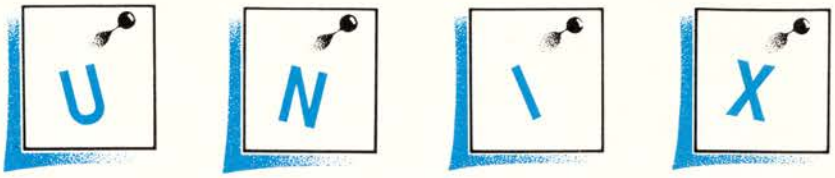
If other fields are not included, their values are assumed to be the same as those in the entry for a dumb terminal, which appears at the end of the *xtermcap* file.

Combination Commands

Several *xtermcap* commands work together in defining terminal capabilities. The **sc**, **co**, and **cc** commands,

FIGURE 1.

```
#
|esprit3:
:ic:cp=1:cs=2:
:sg=\E$:se=\E%:
:cr=I:dc=N:dl=A:dr=D:hb=K:bx=#:
:lc=L:uc=O:ul=B:ur=C:rc=M:vb=J:
:hi=\E(:ni=\E(:li=\E):is=\Eo:vs=\En:
:cc<green><black>=\EG0:
:cc<cyan><black>=\EG1:
:cc<yellow><black>=\EG2:
:cc<white><black>=\EG3:
:cc<black><green>=\EG4:
:cc<black><cyan>=\EG5:
:cc<black><yellow>=\EG6:
:cc<black><white>=\EG7:
:cc<black><black>=\EG8:
:cc<blue><black>=\EG9:
:cc<red><black>=\EG\::
:cc<magenta><black>=\EG;:
:cc<black><blue>=\EG=:
:cc<black><red>=\EG>:
:cc<black><magenta>=\EG?:
:sc<white><black>=0:
:sc<yellow><black>=1:
:sc<yellow><black>=2:
:sc<cyan><black>=3:
:sc<green><black>=4:
:sc<magenta><black>=5:
:sc<red><black>=6:
:sc<blue><black>=7:
:sc<black><white>=8:
:sc<black><yellow>=9:
:sc<black><yellow>=10:
:sc<black><cyan>=11:
:sc<black><green>=12:
:sc<black><magenta>=13:
:sc<black><red>=14:
:sc<black><blue>=15:
#
```



The Winning Combination

UNIX & UniForum & YOU

The UNIX* operating systems are acknowledged as the multi-user, multi-tasking computer operating systems of choice for the 1980's. And UniForum is the premier UNIX systems conference and trade show—the one event that UNIX systems buyers and sellers alike "can't afford to miss". YOU are the final important ingredient in this winning combination . . . so plan now to join us in Dallas • in February • for UniForum 1988.

UNIX Systems Users: It's not too early . . . to register for the UniForum 1988 exhibits (includes free UNIX Introductory Workshops). And to participate in the Show's complete conference and tutorial program. Call us at 800-323-5155 CST for information.

UNIX Systems Vendors: It's not too late . . . to exhibit at UniForum, your best 1988 UNIX systems marketing opportunity. You'll meet thousands of quality UNIX systems buyers, including end-users, dealer/distributors, OEMs, VARs, government, technical, research and education specialists.

UniForum™
The International Conference of UNIX Users

February 8 – 11, 1988
INFOMART • Dallas, Texas

Call Us TODAY At 800-323-5155 For Complete Details!

ENTER 557 ON READER CARD

Sponsored by /usr/group
The International Network of UNIX Users

UniForum 1988 • 2400 East Devon Avenue • Suite 205 • Des Plaines, Illinois 60018

UNIX is a registered trademark of AT&T • UniForum is a trademark of /usr/group

which *xtermcap* uses to change colors, contain references to the actual colors by name, getting further information from the other commands that refer to the same colors.

Xtermcap interprets user-specified numbers as color change commands, referring to specific combinations of colors. The *xtermcap* field **sc** (special colors) indicates the foreground and background colors that you want to display on the terminal when it is sent a given color number. (For monochrome terminals, color numbers are used to refer to the various intensities and reverses that may be available.)

The syntax for **sc** fields is:

sc<*color string 1*><*color string 2*>=*number:*

where *color string 1* is the name of the foreground color to be displayed, and *color string 2* is the name of the background color. (The arrows (< and >) are part of the actual command, not variable indicators.) Thus:

sc<**black**><**red**>=**16:**

indicates that the terminal interprets color code 16 by displaying black text on a red background.

Some terminals can show any combination of foreground and background colors, using different screen color codes, depending on whether a code refers to a color in the background or the foreground. Other terminals can display only a limited number of combinations, with a separate code for each combination. The *xtermcap* field **ic** indicates how a terminal interprets colors. If **ic** is true (the terminal "is combinational"), the terminal can display only limited combinations. If **ic** is false (the terminal is "non-combinational"), it can display any combination.

If **ic** is true:

cc<*color string1*><*color string2*>=*string3*

means that the terminal is sent *string3* to display the color *color string1* on a background of *color string2*. (The arrows (< and >) are, again, part of the actual command, not variable indicators.) The combination must be valid for this terminal.

For example, the Esprit 3 Color Terminal, whose *xtermcap* definition is shown in Figure 1, changes the display color for characters it displays when it receives **Esc G** followed by one of 16 characters. For that terminal:

cc<**black**><**magenta**>=**EG?:**

means that the terminal is sent the string **EG**? (where **E** means **Esc**) to display black text on a magenta background.

Non-Combinational Terminals

For a non-combinational terminal (which must receive separate codes to indicate foreground and background colors), the **cc** command indicates the strings in which you embed the color codes when you send it a color change command. The syntax for the **cc** command is:

cc=*string1*%**s***string2*%**s***string3:*

%s shows where the color change codes will be embedded (as in the **C sprintf** command). Any of the three codes may be blank; if the **cc** field is not set, the two strings simply are concatenated and sent to the terminal.

The **iw** field indicates the order in which the terminal is sent the color codes within the color change string. If **iw** is true, the background color precedes the foreground color. If **iw** is false, the foreground color comes first.

The **in** field indicates whether the terminal uses a different code to indicate a color in the background than it does to indicate the same color in the fore-

TABLE 1.

Value	Type	Description
sp	numeric	(Semi-place) The number of spaces the terminal uses to switch to line drawing mode.
cp	numeric	(Color place) The number of spaces the terminal uses to switch to a new color.
vp	numeric	(Visible place) The number of spaces the terminal uses to switch to invisibility.
ip	numeric	(Intensity place) The number of spaces the terminal uses to switch to other intensities.
cs	numeric	(Color side). The color in which the spaces the terminal uses in switching colors appear. Valid values are:
		1 The original color (on the right). 2 The color to which the print is changing (on the left). 0 Another color (usually the background color)

ground. If **in** is true, the codes for background and foreground are different. If **in** is false, the codes are the same.

(The **iw** and **in** fields are not meaningful for combinational terminals.)

The **co** field is interpreted differently for non-combinational terminals than for combinational terminals. Its syntax depends on the value of the **in** flag.

If **in** is true, the **co** field for a color contains the two codes for background and foreground display of the color, using the syntax:

co <*color*> = *foreground string: background string:*

(This is the only *xtermcap* field that contains a colon.)

If **in** is false, the **co** field contains the single code that displays that color, using the syntax:

co <*color*> = code string:

Using The Combination Commands

The **tgetstr( )** function, which *CHARM* runs when it defines the terminal functions it needs, combines these commands to create the color display functions.

Figure 1 shows the *xtermcap* entry for one terminal, the Esprit 3. For example, **tgetstr( )** would create the command to display color number 4 using these steps:

1. It checks to see if the terminal is combinational (if **ic** is true).
2. If it is, it then finds the **sc** command for color number 4. In this case, the command is:

sc <**green**> <**black**> = **4:**

3. It then finds the cc command to display green on black.

In the example, it is:

cc <**green**> <**black**> = \ **EG4:**

4. By combining the information in these entries, the program knows to

TABLE 2.

Value	Type	Description	
bx	string	Displays the Box character. Unlike other line drawing characters, this character does not need the characters to enter and exit line draw mode.	
sg	string	Enter line draw mode.	
se	string	Exit line draw mode.	
ul	string	Upper left corner	┌
uc	string	Upper connection	┬
ur	string	Upper right corner	┐
lc	string	Right connection	├
cr	string	Cross	┼
rc	string	Left connection	┤
dl	string	Down (lower) left	└
dc	string	Down (lower) connection	┴
dr	string	Down (lower) right	┘
vb	string	Vertical bar	│
hb	string	Horizontal bar	─

TABLE 3.

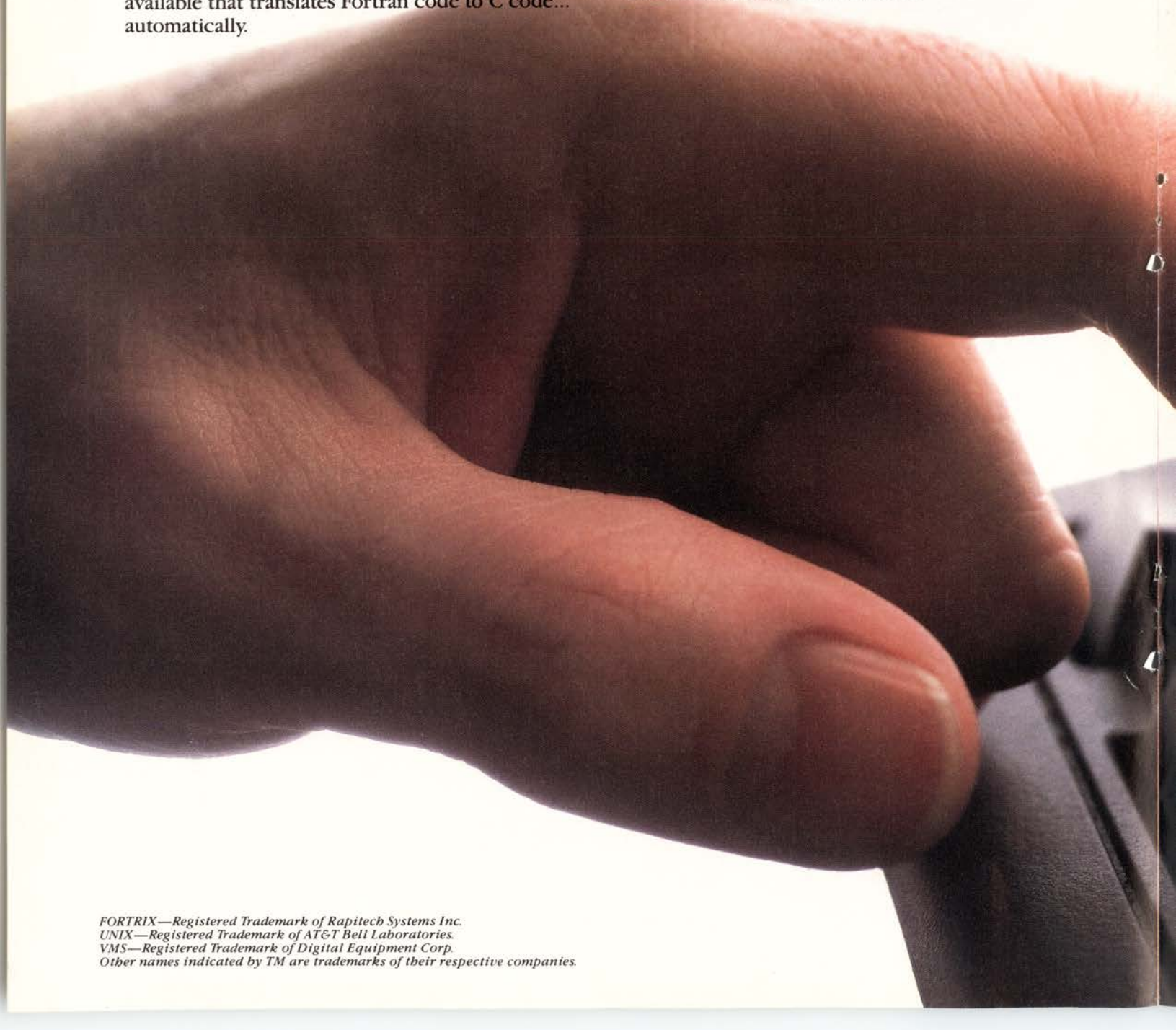
Value	Type	Description
hi	string	Change to high intensity
ni	string	Change to normal intensity
li	string	Change to low intensity

Now you can have to C conversion...at

Here's the word you've been waiting for: Hold on to your Fortran environment. Use it. Maintain it. Grow it. Protect it.

And then pour it into any new C language environment you desire...UNIX,™ VMS,™ anything. Because FORTRIX™-C is here. The only software available that translates Fortran code to C code... automatically.

It doesn't matter which Fortran dialect, style or sequence you have. FORTRIX will convert it all... and keep exactly the same format. Nothing is lost. Nothing is wasted. And because FORTRIX-C is a machine translation, you get all the cost advantages of extraordinary speed. That's 600 lines a minute. A full 50,000 lines in two weeks.



perfect Fortran the touch of a button.

What's more, FORTRIX is proven. Right now it's on site and working worldwide at over 100 locations... including IBM,™ AT&T,™ TRW,™ EXXON,™ Rockwell International, GE,™ Lockheed, Mitsubishi, Carnegie Mellon Univ., U.S. Army, U.S. Navy, Allied Bendix and many others.

All of this means you no longer have to face the limitations of Fortran code. Or endure two different and incompatible computer environments. Because with FORTRIX-C, you can keep your Fortran programs...and still move up to C.

The most important thing NOW is to prove it to yourself. If you'd like to know what FORTRIX can do for you, call our toll free 800 number. If it's more information you want, fill out this coupon and we'll send you technical details.

FORTRIX-C

The Last Word in Fortran to C Conversion.

ENTER 558 ON READER CARD

Rapitech Systems Inc.
Montebello Corporate Park Suffern, NY 10901
(914) 368-3000. Outside New York—1-800-FORTRIX. Telex: 509210.

DPU687

Name_____ Title_____
Company_____
Address_____
City_____ State_____ Zip_____
Phone_____
Hardware:_____
Lines:_____
O/S:_____
Language:_____

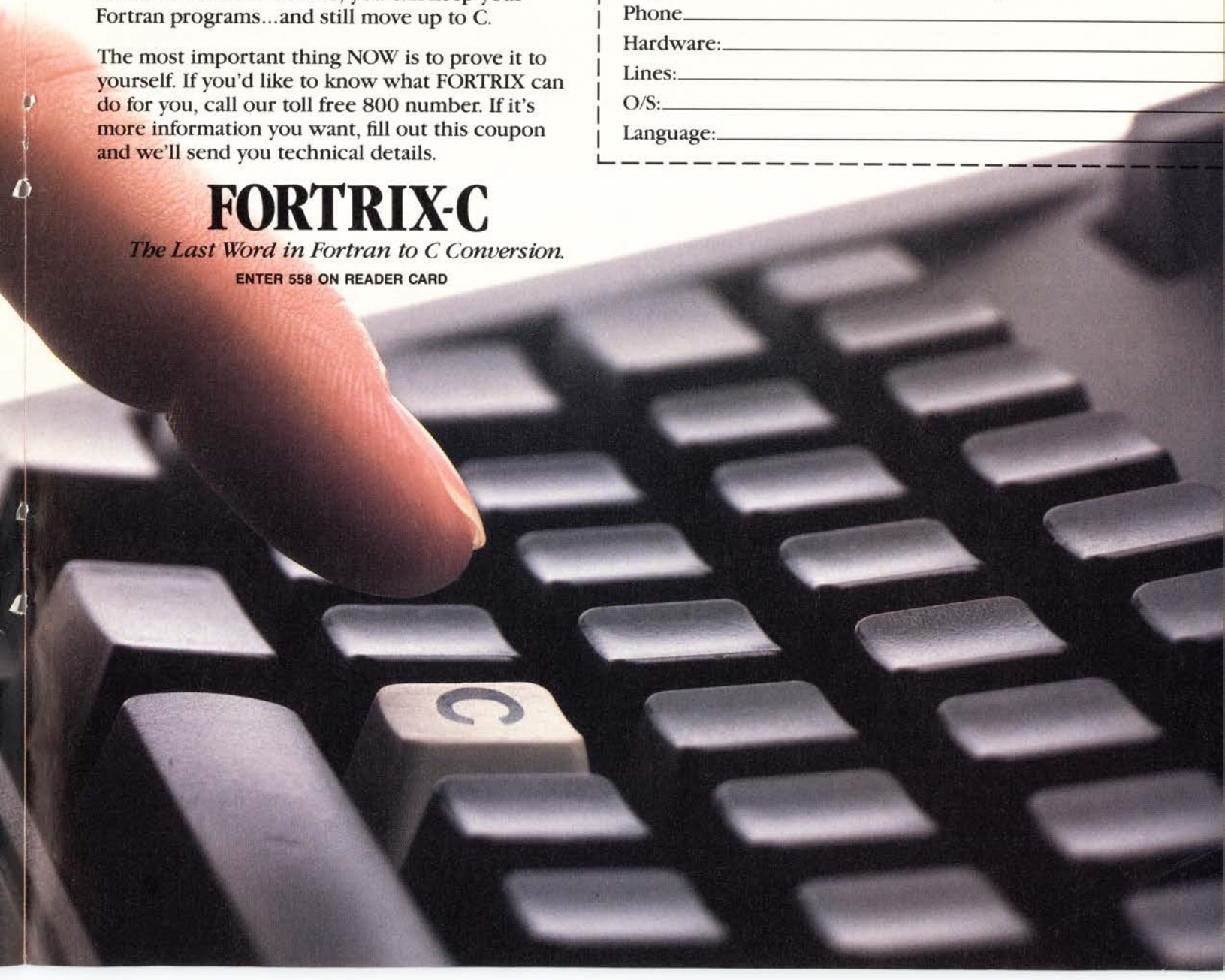


TABLE 4.

Value	Type	Description
is	string	Invisible screen. (Blank the screen while changing the screen image.)
vi	string	Visible screen. (Redisplay the screen image hidden by a previous **is** command.)

send **Esc G4** to the terminal when it needs to display color number 4.

For a non-combinational terminal, it would proceed like this:

1. It first checks to see if the terminal is combinational. An ANSI-standard terminal, to use a second example, is not.
2. It then determines the string in which it will embed the proper color codes. For an ANSI-standard terminal, it is:

```
cc = \E[30m\E[40\E[3%sm\
  E[4%sm
```

where *%s* is the mask for the color indicators.

3. If **iw** is true for a non-combinational terminal, the background color appears in the string before the foreground color. In this case, it is not true; the foreground color comes first.
4. If **in** is true for a non-combinational terminal, the terminal uses one code to display a color in the foreground, and another to display it in the background. In this example, **in** is false; the same string is used in both cases.
5. It then determines the color combination to be used in displaying the color 4 by finding the appropriate **sc** command. In this case, it is:

```
sc<green><black>=4:
```

The color number 4 is interpreted (as for the Esprit 3) as green text on a black background.

6. The **co** commands for each color name show the codes that will replace the two %**ss** in the **cc** string to show the appropriate colors. In this case:

```
co<green>=2: and co<black>=0:
```

7. Thus, the program knows to send the following string to the ANSI-standard terminal to display color number 4:

```
Esc [30m Esc [40m Esc [32m
  Esc [40m
```

Spaces For Screen Changes

The rest of the *xtermcap* commands are either boolean or indicate strings to be sent to the terminal.

Many terminals reserve one or more spaces on the screen image for each color or intensity change. These spaces appear as if they are empty.

The fields in a terminal's *xtermcap* entry as shown in Table 1 indicate the number and color of spaces needed for these changes.

The *xtermcap* fields in Table 2 indicate the codes that the terminal uses in displaying line drawing characters on the screen. You can set the *xtermcap* entries to indicate other characters if you wish (curved corners, for example, or double lines).

The values in Table 3 show the codes that change the intensity of terminal displays.

If your terminal has only two intensities, set **hi** to the higher intensity, and both **ni** and **li** for the lower intensity.

Some terminals allow you to blank the screen as it is drawn, then redisplay the completed screen. You can use this feature through the *xtermcap* entries shown in Table 4.

A Sample *xtermcap* Entry

Figure 1 shows the *xtermcap* entry for the Esprit 3 terminal. The fields in the entry can be interpreted as follows. The first line:

```
|esprit3:
```

shows the name of the terminal. In the second line:

```
:ic:cp=1:cs=2:
```

ic: indicates that the field is combinational (it can display only certain combinations of foreground and background colors). The terminal needs one blank space in which to change colors (**cp=1:**); that space appears in the background color of the combination to which it has changed (**cs=2**). For example, if the screen display changes from green on black to black on red, the text will be separated by a red blank.

The third line:

```
:sg=\E$:se=\E%:
```

shows that the terminal switches into line draw mode when it is sent **Esc $**, and switches back to character display when it is sent **Esc %**.

The next two lines:

```
:cr=I:dc=N:dl=A:dr=D:hb=K:bx=#:
:lc=L:uc=O:ul=B:ur=C:rc=M:vb=J:
```

show the characters that correspond to the line drawing characters. When the terminal is in line drawing mode, it interprets **I** as a cross, **N** as a lower connection, **A** as a lower right corner, and so on. The fields in the sixth line:

```
:hi=\E(:ni=\E(:li=\E):is=\
  Eo:vs=\En:
```

show the codes to switch the terminal's intensity and visibility. **Esc (** switches

the terminal into normal intensity. (The same code is used for both high and normal intensity.) **Esc)** switches the terminal into lower intensity. **Esc o** hides the screen while it is being drawn. **Esc n** redisplays the screen.

Each of the next 15 lines contain a **cc** field, indicating the strings that display various combinations of colors on the screen. The Esprit 3 is, as the **ic** field indicated, combinational; it can combine display characters in any of seven colors on a black background or black characters on a background of any of those colors. Each color code for this terminal consists of **Esc G** plus another character. Thus:

```
:cc<green><black>=\EG0:
```

shows that the terminal switches to displaying green text on a black background when it is sent **Esc G0**.

The last group of lines, containing the **sc** fields, indicates the color combinations that the terminal displays when it is sent a numeric code. For example:

```
:sc<green><black>=4:
```

shows that the terminal interprets the command to display color number 4 by displaying green on black, using the appropriate code from the **cc** command (which, as shown above, is **Esc G0**).

The *xtermcap* and its function library can be expanded further to include other functions. For example, new commands might deal with opening, closing, and writing to windows, or switching screen pages.

With its additional definition types and expandable buffer size, *xtermcap* leaves room for you to use almost anything your terminal can do, giving UNIX systems a new (and increasingly necessary) freedom in designing efficient, pleasing, ergonomic user interfaces. *—Joseph Zitt is a technical writer at WWD, Brooklyn, New York. Naaman Reiz, Robert Segall and Oren Ben Kiki are the programmers at WWD responsible for designing* xtermcap.

Will The Real UNIX Please Stand Up?

By Rex Jaeschke

The Complicated Standards Issue.

First there was UNIX Version 6, closely followed by Version 7. The next "official" release was System III, then came System V Release 1, then Release 2 and finally, Release 3. Along the way, Ken Thompson spent a sabbatical at UC Berkeley and, as a result, UNIX was born on the DEC VAX 11/780 with the label UNIX 4.1 BSD. From that product came 4.2 BSD and finally, 4.3 BSD.

But others were not content to let Berkeley alone spoil AT&T's fun. The folks at Microsoft decided to have their own version based on Version 7 with some System III stuff. And, of course, they added their own goodies as well. They called it XENIX and they ported it to numerous machines, including some from Intel and DEC.

Subsequently, XENIX found its way into the hands of Santa Cruz Operations (SCO), which also made a version available for various OEMs while Microsoft concentrated on IBM.

Finally, XENIX System V.2 was released and Microsoft and SCO announced they were back together. But not completely, for Microsoft announced a deal with AT&T for an Intel-based 80386 version of XENIX which, for the very first time in UNIX history, would be sold as UNIX. (Until now, AT&T had not licensed the name UNIX to any other company.)

Further complicating things, AT&T was working closely with Intel on a "real" port — perhaps that was for the 80286 only.

Along with these projects, many more vendors purchased UNIX source licenses and began running this panacea of operating systems on just about every box you could think of. Interactive Systems helped IBM with PC/IX and VM/IX. Amdahl, Data General, DEC (with ULTRIX-11 and ULTRIX-32) and even Cray Research jumped on the bandwagon. And, of course, each vendor added "a little bit of value" under the guise of improving the product.

Now, for some companies, merely porting from an existing source code license was too easy, legally restricting or, for some other reason, not a good idea. So, they wrote their own equivalents. For some it was a matter of reverse engineering; for others it was a matter of providing UNIX functionality, but with an entirely different set of internals.

In any case, UNIX look-alikes blossomed. One in particular was IDRIS (and more recently CO-IDRIS) from Whitesmiths, Ltd. (a leading supplier of C, Pascal and UNIX tools to DEC and other environments). In fact, P.J. Plauger (president of Whitesmiths) and his band of programmers were pioneers in bringing commercial versions of C and UNIX to the world outside AT&T.

WHAT IS THE FUTURE of UNIX? How can it survive if it exists as dozens of different flavors? Well, for those of you hoping that either UNIX is dying or that your particular flavor will

become king, you might be in for a few surprises, not all of which are pleasant or will happen smoothly.

First we have AT&T's System V Interface Definition (SVID) which gives a recipe for remaining compatible with System V and future releases. This is AT&T's entry in the de facto standard stakes.

Then we have the IEEE P1003 Portable UNIX Standard, currently in trial use. This group has been meeting for two and a half years and has defined a common version of UNIX called POSIX (trademark of IEEE). P1003 has a formal liaison with the ANSI C Standards Committee X3J11, and the two groups have divided up the C function library among them — basically, anything operating system-related went to P1003, most other routines went to X3J11 and the remaining few were raffled off to the lowest bidder, thus ensuring that someone took responsibility for them.

Independent of these groups, X/OPEN was founded. Originally a consortium of European hardware and software vendors, X/OPEN now includes numerous U.S. companies in its ranks. These include DEC, Unisys and Hewlett-Packard. The goal of X/OPEN is more ambitious than the goals of other groups. X/OPEN not only wants a standard for an operating system interface, it also has included models for foreign language support, various language standards including C, FORTRAN, PASCAL and COBOL, and a database management system. X/OPEN members are intimately involved with SVID, P1003 and X3J11.

STANDARDS TAKE TIME and are inevitable with something as popular and diverse as UNIX. However, they involve many technical and political trade-offs and the situation is complicated further with the involvement of people from many different language and business cultures. The support for internationalization has been a much-debated topic for a year or so now and is close to being real. (POSIX and the ANSI C Standard likely will be the first popular standards to seriously address this area.)

Those of us intimately involved in the standards process are much more aware of the costs and benefits of standards-making and, generally, we have a considerable software investment to protect. For the most part, however, the real world of computing knows and/or cares little about standards except that it sees them as something evil that unnecessarily complicates its otherwise happy existence.

Having been a user, a vendor and now a consultant and writer, I can sympathize with all parties. However, it's clear to me that not only are these standards inevitable, they are absolutely necessary. It's not so much what a standard costs, but what a *lack* of a standard costs that matters.

DEC IS COMMITTED to both UNIX and C. Its representatives actively participate in both ANSI C and P1003 deliberations. DEC also was the first U.S. company to join X/OPEN. It plans to have implementations of ULTRIX-32 that conform to POSIX and X/OPEN and to have a C compiler that conforms to X3J11. Already its new VAX C V2.3 is being made available on ULTRIX.

In the last year, an ISO C language standards committee has been formed and it's quite likely that DEC will be represented on that, if it isn't already. And last, but by no means least, Jim Isaak, a prime mover behind the P1003 group, has joined DEC in a strategic planning capacity in its ULTRIX product group. *—Rex Jaeschke is the C Language editor of* DEC PROFESSIONAL *and a member of the ANSI C Standards Committee. He is a cofounder and the editor of* The C Journal, *the author of numerous books and works as a consultant and seminar leader. If you have questions or comments for Rex, write to him at 2051 Swans Neck Way, Reston, VA 22091.*

To obtain a copy of the *IEEE 1003.1 POSIX Trial Use Standard*, stock number SH10546 (book #967), contact:

IEEE Service Center
445 Hoes Lane
Piscataway, NJ 08854
(714) 821-8380
Price: $19.95

For a copy of the X/OPEN documentation set titled *X/OPEN Portability Guide* (Issue 2), Five-Volume Set, ISBN 0-444-70179-6, contact:
In North America:
Elsevier Science Publishers Co. Inc.
P.O. Box 1663
Grand Central Station
New York, NY 10163
In Europe:
Elsevier Science Publishers
Book Order Dept.
P.O. Box 211
1000 AE Amsterdam
The Netherlands
Price: $125

For information on the ANSI C X3J11 Standards Committee, contact:

X3 Secretariat CBEMA
311 First Street, NW
Suite 500
Washington, D.C. 20001
(202) 737-8888

To receive a copy of the X3J11 C Standard as issued for the public review period beginning November 7, 1986, contact:

Global Engineering Documents Inc.
Santa Ana, CA
(800) 854-7179
(714) 540-9870 ext. 245
Telex: globaldoc sna
Price: $65; includes the Standard and the Rationale document.

Copies of the *System V Interface Definition* are available from AT&T; (800) 828-UNIX.

Supporting The X Window System

By Vasudev K. Bhandarkar

Protocol Version 10 vs. Protocol Version 11.

In 1986 when DEC released its first UNIX workstation, the VAX-station/II GPX (see "ULTRIX 32w Software For The VAXstation II/GPX," *DEC PRO EXTRA—UNIX Edition*, May 1986), it was the first commercially available product based on the X Window System. Now the de facto standard in the workstation marketplace, Version 10-based X introduced color, a significant improvement over the earlier revision protocol, Version 6.

In January 1987, a large number of vendors worldwide pledged support for X. Today, X Version 10 runs on more pieces of hardware than any other window system. Engineers at MIT and DEC will soon release Version 11 of the X Window System.

What's new in Version 11 of the X Window System? How is it different from Version 10? When will it be available in the marketplace? As a preamble to answering these questions, let's take a look at X, its origins and history.

X IS A PORTABLE windowing system in UNIX that is network-transparent, runs on bit-mapped graphics displays and is available on a wide variety of hardware. Developed at MIT-Athena by Robert Scheifler and James Gettys, it is available on DEC, IBM and Hewlett-Packard workstations (see "The X Window System," *ACM Transactions on Graphics #63, Special Issue on User Interface Software*, 1986). X runs on all flavors of UNIX, as well as on MS-DOS, in particular on the PC AT. In addition, DEC has committed to provide X on its VMS-based workstations.

X supports an overlapping scheme of windows. Some windows lie on top of other windows, like sheets of paper on a desktop.

All windows form an inverted tree structure. Heading the window hierarchy is the root window that covers the entire screen. Its "offspring," children windows, are the clients of X. Every window, therefore, has a parent except the root window. Child windows can be larger in dimensions than the parent window, but output to a child is clipped by the parent's dimensions.

The X window server demultiplexes user input to various client programs running on both the local machine and remote machines on the network. It also multiplexes output requests from these programs and manages their display on the screen.

The X server runs on computers with either monochrome or color bitmap displays. Client programs communicate with the server via programming libraries, specifically C, Clu and Modula-2. The programming libraries "packetize" client requests into protocol requests and ship them to the server over the network or shared-memory connection. This protocol identifies the version of X. The most widely available and stable version of X available today is the Version 10 protocol, which has been enhanced and updated to Version 11 and server implementations are planned for the near future.

THE X SERVER RECEIVES and services connections from clients and clients send protocol requests to the server. The server looks at all requests, arriving over a byte-stream, and per-

Unlike X Version 10 where there was only one root window, Version 11 has an array of root windows . . .

forms the appropriate function. Because the server doesn't care about the origin of these requests (apart from some minimal initial client-host authentication), the requests could have arrived from either the local machine or the machine's network. A network-transparent architecture, it allows clients to be run from any host on the network, producing graphics on the workstation.

The X server has a very defined interface separating the device-independent and device-dependent portions. Therefore, hardware vendors can remove the generic (model frame-buffer) device-dependent module and plug in their own hardware-dependent module. Because most of the software above the device-independent interface is publicly available, applications will run on any machine that runs X (see Figure 1).

The X Window System can be customized. The window server provides all the hooks for programmers to write their own window (manipulation) managers. A window manager facilitates the movement, stacking and sizing of windows on the screen. To date, several window managers exist and window system users can run one or more of these window managers simultaneously. Furthermore, by creating different startup files, users can specify their own menus or bind different functions to combinations of buttons on the mouse and keys on the keyboard.

X Version 10 is a memory-less window system. The X window server does not support client applications in maintaining window contents. When several windows are put up on the display, some windows occlude other windows. When the windows on the top are moved about on the screen, the windows get unoccluded. Because X is memory-less, the client application must refresh the contents of the unoccluded portions.

Window damage also could occur when the window is resized by the client using the user-interface program, and no single refresh procedure exists. The client knows best how to perform the refresh in an efficient and correct fashion. Hence in X Version 10, the onus of refreshing window damage lies with the client application. Version 10 provides the client application with event reports containing information such as the size of the new window and the size of the damaged portion of the window.

In addition, Version 10 doesn't store state information, such as drawing primitives along with the attributes for each primitive. Client programs have to pass all the information about the output primitives with each request.

Why Do We Need Version 11?

When DEC's VAXstation II/GPX was announced, I visited a large aerospace company in the Northwestern United States that was evaluating a number of existing workstations. While admiring the basic philosophy and style of X, its engineers came up with a number of features that they wanted in X, features available in other workstations, including statefulness, extensibility, double-buffered output and a toolkit.

Jim Gettys, a co-developer of the X Window System offered a number of reasons for developing Version 11 of the X Protocol.

1. X soon would become a standard. When it was taken over by the Standard Committees, it would become difficult to change.
2. Vendors were introducing hardware that Version 10 couldn't run on 24-bit deep frame buffer devices.
3. X needed to support a number of different styles of window (manipulation) managers like the Andrew or MS Windows style.
4. Graphics capability needed to be enhanced and more versatile text output was needed.
5. Bugs in Version 10 needed to be fixed and overlooked features like Stipple Fill needed to be added.

Version 11 vs. Version 10

With Version 10, clients requested resources from the server and obtained resource IDs from it. The server kept track of all resources on a per client basis. In Version 11, when the client establishes the connection, it's given a range of resource IDs and then the client must keep track of resources. Performance improvement results because clients no longer make round trips to the server to get new resource identifiers.

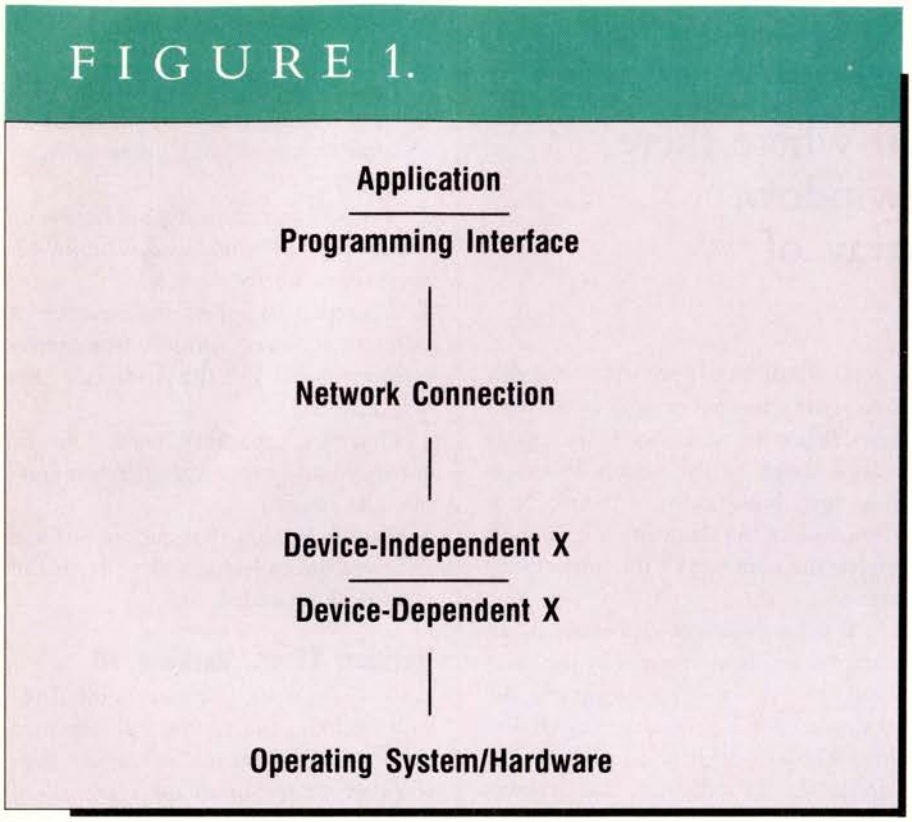
Unlike X Version 10 where there was only one root window, Version 11 has an array of root windows, one for each physical display serviced by the X window server.

No transparent windows are supported in Version 11. Instead, it provides windows with zero border widths and ones that share the background pixmaps of parents. Furthermore, there are input-only windows that behave like transparent windows.

In Version 10, only 16-bit quantities were allowed for pixel values and plane masks in the transport protocol. In Version 11, 32-bit deep frame buffers are supported.

The Version 11 design allows multiple virtual color maps associated with windows. Displays that have hardware

FIGURE 1.



support for additional color maps, say on a per window basis, could map the color map to a virtual color map, making it possible for windows to have truly independent color maps.

PARAMETERS THAT CHANGE infrequently need not be passed for every request. For example, specify line width only once before a series of line drawing requests are sent to the server. Like some existing standard libraries namely GKS and PHIGS, after you set the attributes, you specify on subsequent line drawing requests only the coordinates and the number of points to be drawn.

Thus, the cost of parameter checking is limited and the amount of parameter data transmitted over the network protocol is reduced. The new hardware that supports its own graphics operations keeps track of previous graphics operations and in essence graphics state. Also most color workstations have their own hardware. Thus, cheap graphics requests now are possible because less information can be sent to the X server to achieve the same functionality.

In Version 10, a basic request was 24 bytes, and only 16-bit pixel values and plane masks were required in them. If graphics requests were to remain stateless and Version 11 were to support 32 bits/pixel, the basic request almost certainly would have had to grow to at *least* 32 bytes. In Version 11, most requests can be fit into 16-byte requests, reducing the network overhead for most applications.

The major line-drawing primitive in Version 10 was a command called XDraw, which drew a single polyline or wide line. In Version 11, this functionality has been split into separate polypoint, polysegment, polyline and polyarc primitives. The polyarc primitive draws simple circles and elliptical arcs.

Stipple Fill, overlooked in Version 10, puts a pattern as a mask bitmap on a region where drawing will occur and turns on only those bits. Version 10 only allowed fill with a pixmap (pattern as a source). Version 11 provides the Stipple Fill primitive.

Version 10 has no capability to allow drawing simultaneously with output. With such a facility, you could write to a hidden buffer and instantly update the screen. Version 11 permits graphics to any drawable (pixmap or window), but there's no guarantee that the window system can create arbitrary amounts of virtual memory for applications to write on. This gives the capability for running existing double buffered applications.

New Hooks For Window Managers

Version 10 of X was incapable of supporting "automatic" tiling like Andrew. Version 11 supports generalized window management facilities.

A window manager or an application to control placement of subwindows in the library packages can select Structure Redirect Control. Whenever there is an attempt to map, unmap, destroy, reposition, resize or alter the border of a subwindow, the selecting client is notified and the operation ignored. A window manager then performs the operation for the clients.

In Version 10, the metric information per font includes drawing direction, first and last character information, subscript and superscript information, underlining, strikeout, baseline, worst case character box and accelerator information. In Version 11, the per character information includes spacing as well as width information and character attribute flags. Sixteen-bit fonts also are available. You also can query font names using simple wildcard patterns.

Both source and mask fonts are available in Version 11. The existing intercharacter and space-padding text primitives are replaced with more general polytext operations, permitting additional padding and font changes

The extension mechanism allows X users to add facilities without conflicting with the core window system . . .

between characters. The existing interface is provided in the accompanying programming libraries.

In Version 11, you can mark a window as expensive to refresh, and some server implementations may attempt to restore occluded portions when they get uncovered and avoid sending exposure events to clients. Although most servers won't support this capability, it can be very difficult on color displays and have very significant performance impacts.

Version 11 also contains bit gravity. When a window is resized, its contents aren't necessarily discarded. You can request the server (though no guarantees are made) to relocate the previous contents to some region of the window. You also can default to the Version 10 behavior; i.e., discard window contents completely when the window is damaged.

Hooks allow the server to impose its own authentication schemes. The Version 11 server in the public domain doesn't do any host authentication per se.

The UNIX operating system doesn't provide any host authentication, so it must be added by local display servers or even by local sites. It's difficult to impose trusted node authentication. Instead, clients might have to provide a key and a password to identify themselves. This information must be built into the server before running on the display device.

Version 10 had a single input device, a three-button mouse, to control the pointer on the screen. Version 11 has been improved to allow different devices (tablet, touch screen, etc.) to control the pointer on the screen and support a variety of input devices. The server can attach different input devices to the pointer and get input from various input devices.

"Grabbing" input devices in Version 11 also is improved over Version 10. Arbitrary keyboard key grabbing now is allowed.

A mouse tablet or other device may control the pointer on the screen. The device setting is determined by the user. The X protocol doesn't define how or if the pointer "roams" between screens on a multiheaded display. In Version 11, the client can request hints about pointer motion, not possible in Version 10 where the server had to transmit every event to the client if the client requested it. With this new method, network traffic is greatly reduced, improving the performance of the X server.

The Version 10 X protocol is fixed. Requests coming across the network must be in a fixed range and there's no mechanism to re-engineer the server to perform additional functions. Version 11 has an extension mechanism. Certain pieces of hardware provide features that X doesn't do inherently, for example, anti-aliased lines.

The extension mechanism allows X users to add facilities without conflicting with the core window system and other extension packages. The facilities provided allow the querying of extension packages by name. If the extension exists, a major opcode is returned and a minor opcode can specify the desired function in the package.

As a result of this mechanism, manufacturers and user groups can add their own features tailored to their own hardware. Therefore, individual needs can be accommodated while the core protocol cannot and will not be changed.

Compatibility And Portability

Is Version 11 compatible with Version 10? No! Not only have new protocol requests been added, but even some of the semantics of existing protocol requests have been changed. Furthermore, the syntax of the programming interfaces has been changed so that applications must be rewritten to conform to the new programming interface.

One compatibility strategy suggested by Ram Rao of DEC's ULTRIX Engineering Group is to provide a nested Version 10 X server on top of the Version 11 X server, so developers can continue to use their old Version 10 applications while porting them to the Version 11 server.

Another is to provide a compatibility library that recognizes Version 10 protocol requests and translates them to Version 11. No concrete steps have been taken to implement either strategy.

The beta test for Version 11 began this month. All software and related documentation will be available from MIT-Athena. The proposed date for the final release and shipment will be September 14, 1987.

This article is based on Jim Gettys' initial commentary on the origins of the Version 11 X protocol.

Robert Scheifler of MIT's Laboratory of Computer Science was the author of the Version 11 X Protocol. Jim Gettys of MIT-Athena/DEC was one of the contributors. Other contributors include Dave Carver, Burns Fisher, Branko Gerovac, Phil Karlton, Scott McGregor, Ram Rao and David Winchell of DEC and David Rosenthal of Sun Microsystems. —*Vasudev K. Bhandarkar is an engineer in DEC's ULTRIX Engineering Group, Merrimack, New Hampshire.*

File Protection

By Matt Bishop

The Built-In Controls UNIX Provides.

UNIX file protection is simple and straightforward. However, many people are not aware of the richness of this powerful mechanism. In this article I'll discuss file protection in detail and describe some simple tricks to help you protect the information in your files.

Background

The UNIX file system is organized as a hierarchy of files. Think of a tree turned upside down, and replace branches with lines. Wherever lines join or end, put a node. If a node is *internal* to the tree, it corresponds to a UNIX *directory* file that contains a list of files in that directory. Otherwise, it corresponds to some other type of file. When the operating system needs to locate a file, it starts at the beginning of the given path and walks down the tree searching each successive directory until the file is found. For example, in Figure 1, to locate the file "/usr/mab/mbox", UNIX begins at the root directory "/" and looks for the file or directory "usr". It then searches "usr" for the file "mab". Finally, it searches that directory for the file "mbox". Note that the root directory "/", "usr", and "mab" all must be directories and must be searchable; otherwise, the system will fail to locate the file.

UNIX files are *owned* in the sense that a particular user creates them, and they are counted toward that user's disk space. The owner of a file is usually the person who creates it. Every user has a unique identification number, called the UID, and when a file is created, the UID of the creator is associated with the file. For example, if my UID is 213, and I create a file named *xxx*, the UID of *xxx* is 213. So, the person with the UID 213 (in this case, me) owns the file. Notice the way that last sentence was written. If my UID were changed one day, say to 625, the UID of file *xxx* also must be changed to 625. This can cause problems, because if the administrator changes someone's UID and forgets to change the UID of all his files, he no longer will own those files.

The group of a file is analogous to the owner of a file. UNIX users are divided into groups, and each group has an identification number called a GID. How the GID is assigned to the file varies from system to system. For example, 4.2 BSD UNIX assigns the file the GID of its parent directory, System V UNIX assigns the GID of its owner. But the important fact is that a file can have exactly one GID.

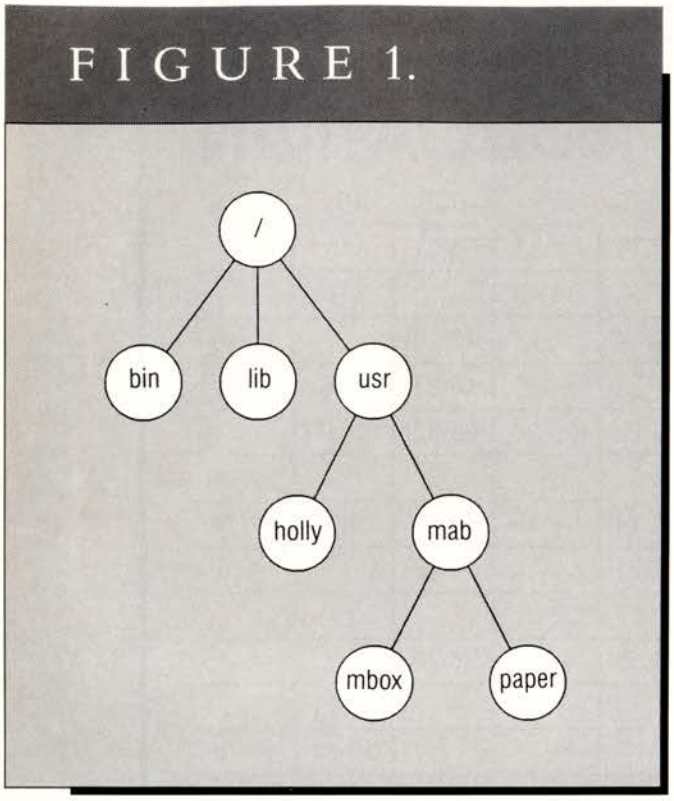
With these preliminaries fresh in our minds, we now can examine the way files are protected.

Permission Bits For Non-Directory Files

There are 12 permission bits in the protection word of a file; they customarily are divided into groups of three. Let's look at the lower three sets of three bits.

The first set of three bits encodes permissions for the file's owner, the second set permissions for the members of the file's group, and the third set permissions for everybody else. (See Figure 2.) When someone tries to access the file, UNIX first checks to see if that person is the file's owner and, if not, if that person is a member of the file's group. For

FIGURE 1.



UNIX file system structure.

example, suppose a file has permissions set so the owner can only read it, and anyone in the file's group can only write it. Even if the owner is a member of the file's group, he will be unable to write on the file. Figure 3 illustrates this procedure.

UNIX associates three modes with each file: read (the ability to display the contents of the file), write (the ability to change the contents of the file), and execute (the ability to run the program contained in the file). The first bit in each triplet of permission bits is set if read permission is granted, the second bit if write permission is granted, and the third bit if execute permission is granted.

Here are some examples. Suppose the lower nine bits of the permission word are 110110100. Split this into sets of three: 110 110 100. So, the owner of the file can read or write the file, the members of the group of the file can read or write the file, and anyone else can only read the file. As another example, suppose the file's permission bits were 111101101. Following the above procedure, the owner can read, write, and execute the file, and anyone else can read and execute the file (but not write it).

As you can gather from the above paragraph, writing permission bits in binary is painful and hard to read. Since permission bits are grouped into sets of three, and octal digits correspond to three binary digits each, permission bits are written as a string of octal digits. So, "110110100" normally would be written as "664", and "111101101" as "755".

> . . . the interpretation of the three modes is somewhat different for directory files.

We have not discussed the highest three bits of the permission bits. The first two are the **setuid** and **setgid** bits. Normally, when a program is executed, the UID and GID of that process are those of the person executing the program. However, if the **setuid** bit is set, the UID of the process will be that of the owner of the file. Similarly, if the **setgid** bit is set, the GID of the process will be that of the owner of the file.

For example, suppose a file owned by *mab* has the **setuid** bit set. User *holly* executes that file. The resulting process will have *mab*'s UID rather than that of *holly*. Had the **setuid** bit not been set, the resulting process would have had *holly*'s UID.

The third bit in this group is called the **sticky bit**, and on many systems can be set only by the superuser. It simply prevents the program text from leaving main memory once it has been placed there, and is used to keep heavily used programs in main memory. This speeds the program's startup time noticeably on many systems.

Permissions For A Directory File

Although the permission bits are called the same for both directory and non-directory files, the interpretation of the three modes is somewhat different for directory files. Reading a directory means being able to find out what files are in that directory. Writing a directory means being able to create new files, or delete existing ones, in that directory. Executing a directory means being able to search that directory to see if a named file is contained in that directory.

Let's look at an example. Suppose the directory "usr" contains three subdirectories called "bin", "mab", and "src". (See Figure 4.) If you can read "usr", saying:

```
ls usr
```

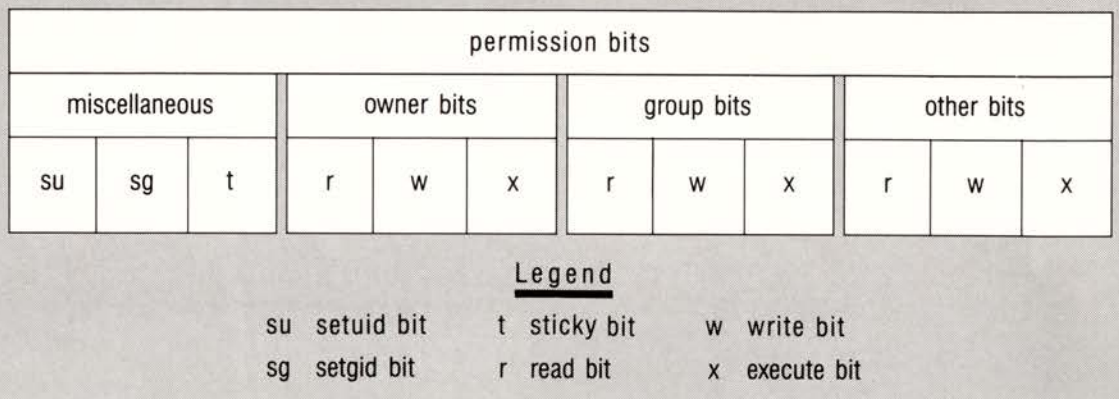
will print:

```
bin mab src
```

If you do not have execute permission on "usr", and you say:

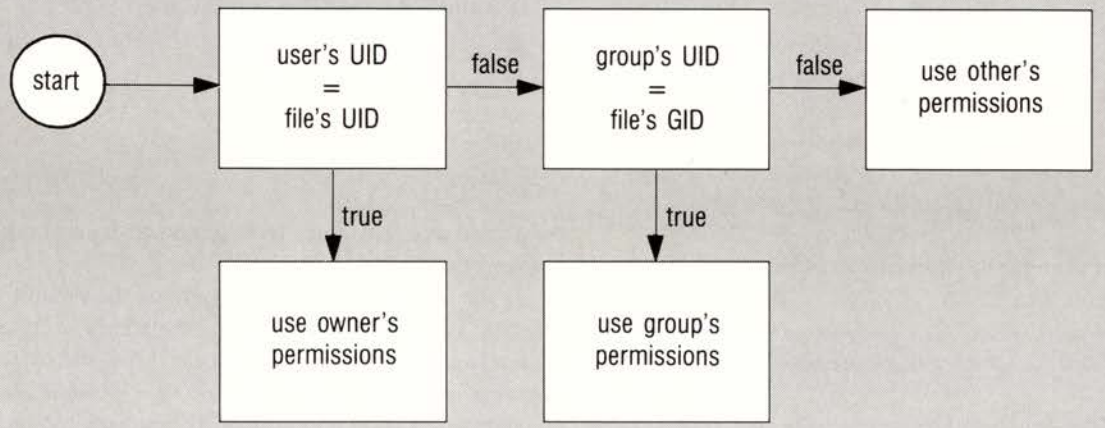
```
ls usr/bin
```

FIGURE 2.



Permission bits.

FIGURE 3.



How user permissions are determined for a file.

you will get the error message "usr/bin not found." (In fact, if you don't have read permission on "usr", and you type:

```
ls usr
```

you will get:

```
usr/bin not found
usr/mab not found
usr/src not found
```

because "ls" cannot obtain any information about the three subdirectories other than their names!)

The *chmod*(1) Command

How do you change the permission mode of a file? The command *chmod*(1) exists for this purpose. Its general format is:

```
chmod permission_mode filenames . . .
```

The *permission_mode* can be in one of two forms. The first

one is the permission mode you want to assign; it *must* be in octal. For example:

```
chmod 4755 rogue
```

makes the file *rogue* readable and executable by everyone on the system, writeable only by the owner, and **setuid** to the owner of that file. As another example:

```
chmod 644 rogue.help
```

makes *rogue.help* readable by everyone on the system and writeable only by the owner.

The second form exists only on some systems; look in the manual to see if it will work on yours. The form is:

```
chmod who op permission filename . . .
```

Who is one of **u** (for "user's permission"), **g** (for "group's permission"), **o** (for "others' permission"), or **a** (for "all users' permission"). *Op* indicates how rights are to be assigned; + means the permissions are to be added to those already there, - means the permissions are to be taken away from those already there, and = means the permissions are to replace those already there. *Permission* is the code letter for the desired permission; they are **r** (for "read"), **w** (for "write"), **x** (for "execute"), **s** (for "set"; if used with **g** it means "setgid", and if used with **u**, means "setuid"), and **t** (for "sticky").

Let's suppose, in the example with *rogue*, the file originally was mode 664 (readable by all users, but writeable only by the owner and members of the group). To change it to mode 4755, you could use any of the previous commands:

```
chmod a+x,u+s,g-w rogue.help
```

(which changes the permission bits as indicated; it leaves all other bits alone), or:

```
chmod u=srwx,g=rx,o=rx rogue.hints
```

(which sets all permission bits to the indicated state). If the mode of *rogue.help* were 700, to make it mode 644 you could use the command:

```
chmod u-x,g+r,o+r rogue.help
```

The *umask*

When you create a file from a C program, you have to specify the permission mode of the file. For example, you might say:

```
creat("xxx",0666)
```

to make the file readable and writeable by all users. However, before the file protection mode is set, the given permission is changed, by bitwise *and*ing it with the bitwise negation of some variable called the *umask*. If, for example, your *umask* were 022, the bitwise negation of that is 0755, so the file *xxx* would be created with permission mode 644 (that is, you can read or write the file, but everyone else can only read it). Some convenient values of the *umask* are:

```
077 only the owner has any permissions
022 only the owner can write the file
002 only the owner and group members can write the file.
```

The *umask* can be set by using the command:

```
umask nnn
```

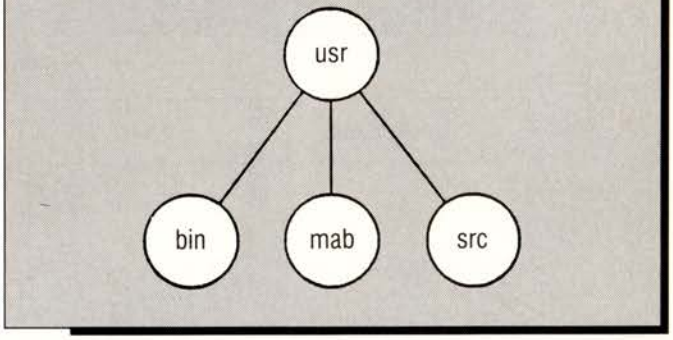
(which sets it to *nnn*) from the shell level, or by using the system call *umask*.

Be aware that on many systems the default value of the *umask* is 0, which means that the mode you specify when creating a file is the mode of a file. Unfortunately, some programs create files in modes 666 or 777, assuming the user's *umask* will cancel any unwanted permissions. So be sure you set your *umask* to a sane value!

In this article, I've covered the most important aspects of the built-in controls UNIX provides for file protection. These mechanisms are simple yet powerful. They provide a consistent framework for controlling who can access a file, as well as the ability to prevent dangerous errors in protection when creating files. With these mechanisms, you can improve the protection of files containing data that you wish to keep secret.

—Matt Bishop is a research scientist at the Research Institute for Advanced Computer Science (RIACS), NASA Ames Research Center, Moffett Field, California.

FIGURE 4.



Sample file system.

From DEC PROFESSIONAL

FREE BULLETIN BOARD SERVICE FOR OUR READERS

WHO: **Our on-line bulletin board is free to all subscribers.**
WHERE: **From your terminal to ours . . . just log in.**
WHAT: **It's called ARIS (Automated Reader Information Service). Dial in and "talk" to staff, advertisers, other readers.**

CHECK IT OUT!

- Expert help for computer problems.
- Talk to the editors about articles and issues.
- Help colleagues find computer solutions.
- Find out about new products and used equipment.
- Download programs from our library (Free!).

HOW: 3 Easy Steps . . .

1. Set your VAX terminal to 7 bits, 1 stop, no or space parity. (300, 1200, or 2400 BAUD available)
2. Dial (215) 542-9458.
3. You will be asked to enter your subscriber number (it's on your mailing label) and your name.

That's all it takes to access our new message center where you'll find 26 Special Interest Groups (SIGS) listed. One of them is sure to interest you.

GET ON-LINE! DIAL IN NOW!

NEW MESSAGE CENTER
26 SPECIAL INTEREST GROUPS AVAILABLE!

UNIX For Office Automation

By Harry J. Foxwell

The ACS's Effective, Centralized OA System.

When I began working with the UNIX operating system, and my programmer colleagues asked me to describe UNIX in 25 words or less, I usually would answer "It's a programmer's dream, and an end user's nightmare." Inherent in that evaluation was a chauvinistic assumption that the computing world was divided into two groups: programmers who understood and users who didn't (and probably couldn't). How could mere users comprehend the intricacies of "grep," "awk," I/O redirection, pipes, background processes, shells, path names, aliases, and profiles?

For years UNIX has enjoyed a somewhat deserved reputation as an unfriendly, if not hostile, operating system. The litany of criticism is well known to both UNIX supporters and detractors: It was hard to learn and use, there was little application software, its commands were cryptic with unintelligible error messages, it performed poorly with I/O-intensive applications, and it did not command a high level of vendor support. But UNIX has survived and flourished in nearly all areas of modern computing, including software development, engineering, manufacturing and that toughest of end-user environments, office automation. In spite of the time-worn complaints against UNIX, most of which are no longer valid, the bottom line is, to paraphrase a recent comment by *InfoPro*'s David Fiedler, whether users are getting their work done.

The American Chemical Society (ACS), a large professional association headquartered in Washington, D.C., is getting its work done with ULTRIX. The Washington-based staff of nearly 400 workers uses two VAX 785s running ULTRIX for much of their business and office computing. The society's Chemical Abstracts Services division in Columbus, Ohio, employing more than 1,200 publishing, chemistry and data processing professionals, uses 14 VAX 785s, all running ULTRIX.

USING BASIC UNIX TOOLS in addition to commercial application packages such as *20/20* for spreadsheet work, *UNIFY* for database systems, and Interactive System Corporation's *INed* and *INmail* for word processing and electronic mail, the ACS has built an effective, centralized, office automation system.

The flexibility of UNIX, the portability of its programs, and its natural ability to network with other computers have allowed ACS to tailor an "end user computing environment" for many of its departments and individual staff members.

The two Washington-based VAX systems are linked to each other and to the IBM mainframes and VAX computers in Columbus. A number of smaller, special-purpose systems, such as an NCR Tower 32 running UNIX System V in one department, also are part of the network. DEC VT220 terminals are the primary VAX workstations, although there are several dozen IBM PCs and compatibles used for local processing. The PCs run VT100 terminal emulation software when their users need to communicate with the VAX systems.

The staff of the ACS is typical of many large businesses with a mixture of clerical and professional personnel, most of whom are not trained in data processing. But the majority of the system users have learned the rudimentary UNIX commands, and some even have mas-

tered grep and awk. They have done this without a friendly interface; their view of UNIX is their shell command line.

Although menus and window interfaces would have allowed beginners to achieve basic productivity in a short time, the command line approach proved to be more easily expandable by the user and didn't get in the way of innovative or expert users. A tremendous effort in user education was required and is continuing. Although some users excelled easily while others still are struggling, user productivity is high.

One of the most often used features of the system is electronic mail. Employees use it for routine business and personal communication, and for file transfers among the various systems in Washington and Columbus. Although no formal cost/benefit analysis ever has been performed on the system, one manager estimated the society was saving $250,000 per year in secretarial services that would have been required to support the level of staff communication provided by electronic mail.

The ease with which various software and hardware features can be added to UNIX systems is an asset and a liability. It's easy to add terminals and printers of any brand, to the point where the centralized processors become overburdened. At that point comes the difficult choice of expanding the central resource or replicating smaller, distributed systems. Current UNIX technology makes either choice acceptable in terms of cost and performance.

BUILDING A MULTIVENDOR SYSTEM requires a core of UNIX and data processing experts to make everything work and to fix things when they don't. I once counted nine different vendors potentially involved with a user running *UNIFY* on the VAX through a PC workstation. No one vendor could support all the hardware and software required for that connection, so the ACS has a data processing department of programmers and system engineers to help users develop their applications and to keep the system growing and running.

Perhaps the most serious problem with large UNIX systems like the one at ACS is the growing number and size of disk files. A short-sighted solution is to keep adding more disk capacity, an approach made more attractive recently by the decline in the cost per megabyte of high-performance/high-capacity disk drives.

What UNIX needs is tools for managing and archiving files. Like pack rats, users will store obsolete files like in intricate mazes of subdirectories and then forget about them. Until software is available to help users delete unused files and to help system administrators move infrequently accessed files offline, UNIX system users will face a constant struggle against the limits of disk storage. ACS has not found a commercial package yet that will perform this function.

Most PC users run MS-DOS as their only operating system. Like many companies, however, ACS is experimenting with PC workstations that run some form of UNIX. Some of the society's IBM PC XTs and ATs run Microsoft's XENIX or IBM's PC/IX; some users have AT&T 6300 Plus PCs running UNIX System V. Although each of these versions of PC UNIX has some utility, especially in maintaining a consistent operating system across varied machines, there are several problems with "desktop UNIX workstations."

UNIX is essentially a multiuser operating system. As such, its users log in, it has tools for user-to-user communication (MAIL and WHO, for example), and its files have owners with access privileges. UNIX also has a superuser administrator whose job it is to configure the system, make backups, and trouble-shoot problems.

Most UNIX systems are busy running background and scheduled processes, even when no one is logged in. All of this is completely foreign to MS-DOS users, who generally have complete authority over their systems. PC users don't worry about file ownership and permissions, can customize their systems as they see fit, and can turn their PCs on and off without worrying that other users or processes will be affected.

Desktop UNIX in its current form, therefore, is not the solution to providing UNIX productivity tools to PC users, nor are the bridge programs designed to run DOS applications as subprocesses under UNIX; such programs perform poorly on PC processors like the 8088 and 80286. In one experiment at ACS, an MS-DOS-based terminal emulator failed to run at more than 300 baud when called as a DOS task under UNIX. Proponents of such an approach will have to wait for more powerful PC processors, such as the 80386.

Since MS-DOS already is established on PCs, a better approach is to take advantage of DOS's UNIX-like capabilities, such as I/O redirection and pipes. Most PC users don't even know about these features. Add a few UNIX tools for manipulating text, (like the UNIX *MKS Toolkit* from Mortice Kern Systems, or *UNIX Tools for DOS from Dr. Dobbs*), and the integration of UNIX into the office environment will be complete. —*Harry J. Foxwell is manager of systems engineering for the American Chemical Society's Office & Information Systems department.*

American Chemical Society
1155 16th St., N.W.
Washington, DC 20036
(202) 872-4600

Interactive Systems Corporation
2401 Colorado Ave., 3rd Floor
Santa Monica, CA 90404
(213) 453-9649

Mortice Kern Systems Inc.
43 Bridgeport Road, East
Waterloo, ON Canada N2J 2J4
(519) 884-2251

Application And Database Performance

By Richard B. Curtis, Ph.D.

Accepting UNIX As A Business Operating Environment.

In the office of the '80s, productivity enhancement is of paramount importance to business computing. Office productivity often can be directly related to the performance of the computer system that services office needs. This performance issue has become a driving force behind the entrance of new technologies of software and hardware that address speed, flexibility and power.

The push for cheaper computing, and the technological breakthroughs that have followed, enabled the emergence of application developers who could provide business software for mid-range departmental machines. However, many users are coming to the mid-range systems from the less user-hostile PC environment expecting to have the same sort of intimate interactions and speedy response. Almost inevitably they are disappointed.

Transaction flow is the most critical measure of a system's performance, and response time is the most critical factor in user friendliness. (People have been more willing to put up with a system that is arcane and fast than one that is congenial and slow.)

As these more sophisticated users vie for access to rapidly expanding corporate databases of information, they perceive the computer system's performance begin to drop.

This article will review some of the causes of performance degradation common in UNIX and propose some strategic solutions. This will not be a primer on tuning a UNIX system.

THE EFFORT TO IMPROVE performance is an exercise in identifying and eliminating bottlenecks. Potential bottlenecks for computers that service multiple users include:

- Single processors
- Software architecture
- Data redundancy
- Disk I/O

Eliminating each bottleneck results in a finely tuned system that delivers maximum performance in spite of system load. This solution is described in the sections that follow.

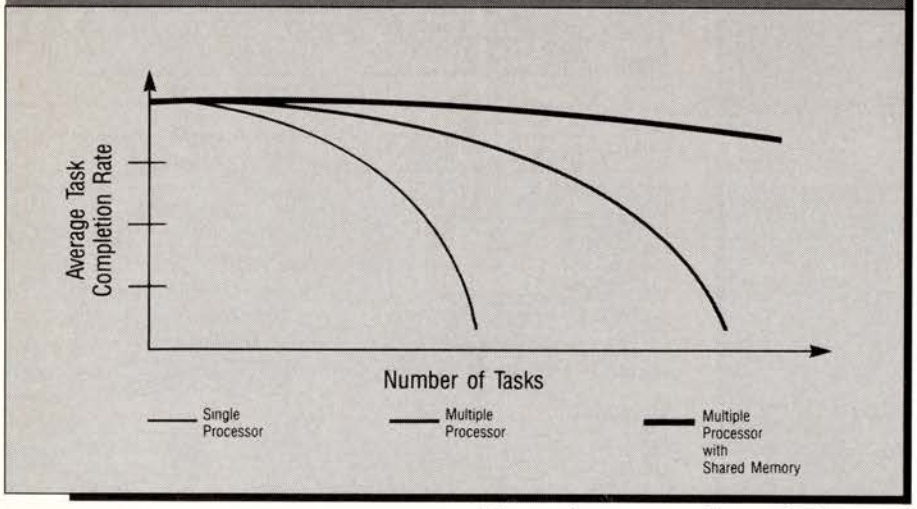
Single Processors

Sitting glumly at their terminals as the cursor crawls from field to field, the users on a heavily loaded system watch with ambivalence as another terminal is hooked up. It's good news that there is another person to share the workload, bad news that there is another competitor for the system's resources.

As the processor's utilization approaches or exceeds 100 percent, the system stumbles to its knees and the users become disgruntled. The more CPU-intensive the processes, the more frequently drops in performance occur and the longer they last. The more concurrent processes the single CPU or "uniprocessor" architectures are expected to support, the faster and more severe the degradation in performance.

When the machine becomes loaded, it panics and starts "thrashing" as the system spends more time figuring out what to do than

FIGURE 1.



Expected average task completion rates.

actually doing anything. The results of thrashing are long delays and an almost sullen work force.

Commercial computer systems that take advantage of multiple processor architectures to share the load of tasks being performed are now available. These systems with two to 20 processors can provide better response time than a single processor, because many user processes do not compete for the same processor resource.

Multiple processor machines don't make programs run faster, but they can provide better response time for that despairing group at the terminals, and they can delay the dreaded moment when the system is loaded to the thrashing point. These machines achieve better response by dispersing the workload across the available processors instead of funneling it through one processor.

Figure 1 illustrates a comparison of average task completion rate for uniprocessor machines versus multiple processor machines given a multiuser workload. As the number of tasks increases, the time to process each task changes relatively slowly.

On the other hand, uniprocessor machines show marked lengthening of task completion times when the number of concurrent tasks increases.

Software Process Architecture

With their new multiprocessor machine, the users are optimistically realizing some response time improvement, but the dramatic difference they were hoping for has not materialized. Other factors can congest the work flow, limiting the number of users a system successfully can serve and the speed of that service.

The architecture of the software being run is a determinant in how efficiently the load can be spread over the multiple processors.

In organizations with many users sharing one system to conduct their business, most of the processes on that system are database applications. These applications have two distinct categories of tasks.

The first is data manipulation, whereby the database management system interactively will perform searches, updates, additions and deletions of records. The second category is user interaction, during which the applications generate reports, perform calculations, format data and print portions of this data.

Figure 2 depicts the two-process architecture consisting of an SQL database-server process for each corresponding requestor process. The database-server accepts statements, such as SELECT, INSERT, UPDATE and DELETE, and performs the interactions with a main memory buffer pool and the disk to retrieve the desired data or perform the desired update. The requestor processes handle all user interface activities including formatting reports and processing interpreted or compiled application code.

Many database architectures consist of one database-server process shared by many requestor processes. This creates a situation where a requestor is waiting in a jostling queue of requestor processes for the results to a database statement.

The database-server has become the bottleneck, because there is only one server process. Some database systems actually have the requestor and server functions together in one large process, which makes the load even more difficult to balance across the multiple CPUs.

The two-piece architecture can be implemented on multiple processor systems to enhance performance by providing a database-server process for each requestor process. This means that the database-server does not become the bottleneck by overloading one processor.

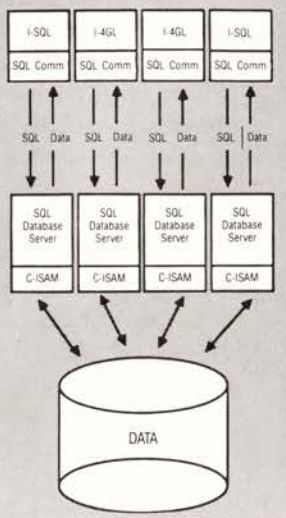
The algorithm of the multiple processor computer allocates the workload and "scatters" both requestor and database-servers among the available processors. With such a distribution, the database-servers can process multiple SQL statements, not just concurrently, but actually simultaneously.

Figure 3 shows an example of one server process per computer versus one server process per requestor. This example illustrates the bottleneck that occurs with one server process being accessed by multiple requestors.

Data Redundancy — Buffer Pool Contention

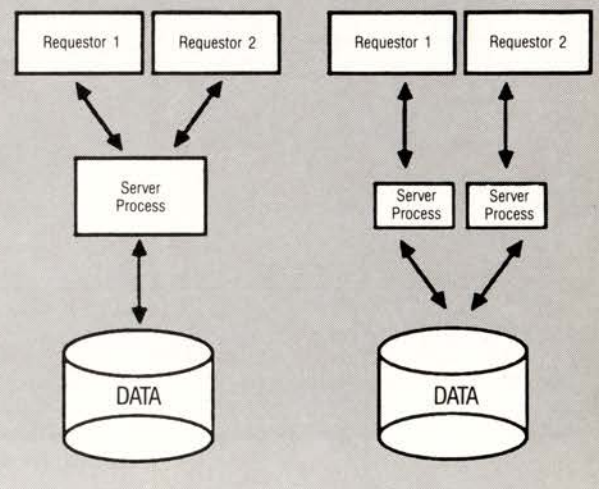
Using a split-architecture DBMS on their multiprocessor machine, the users

FIGURE 2.



The two-process architecture.

FIGURE 3.



One server process vs. two.

are guardedly approaching cheerfulness. Since the CPU utilization bottleneck has been removed, a great many users can search and update the same database at the same time. Still, the throughput is not quite what was expected.

The next most likely bottleneck concerns the management of buffer pools kept in memory by each database-server process.

Examining the use of memory for a typical multiuser application provides us with a memory map similar to Diagram A in Figure 4. Notice that memory is allocated for the operating system, database code and user data required by the application.

In UNIX, program code is reentrant and can be shared in memory. Since all server processes are the same program, only one code segment needs to be loaded into memory. Each invocation of the database-server process, however, receives its own private data segment. As Figure 4 shows, 150 KB of data are required for all users in this example.

(This 512-KB total memory space with three users on a uniprocessor machine is shown only for purposes of illustration. Actual multiprocessor machines would have more memory and more users. Also, the treatment of the operating system varies.)

Most databases have some records that are far more active and shared by more users than the remainder of the records. Data being accessed by one user is frequently duplicated in another user's buffer pool, wasting valuable memory resources.

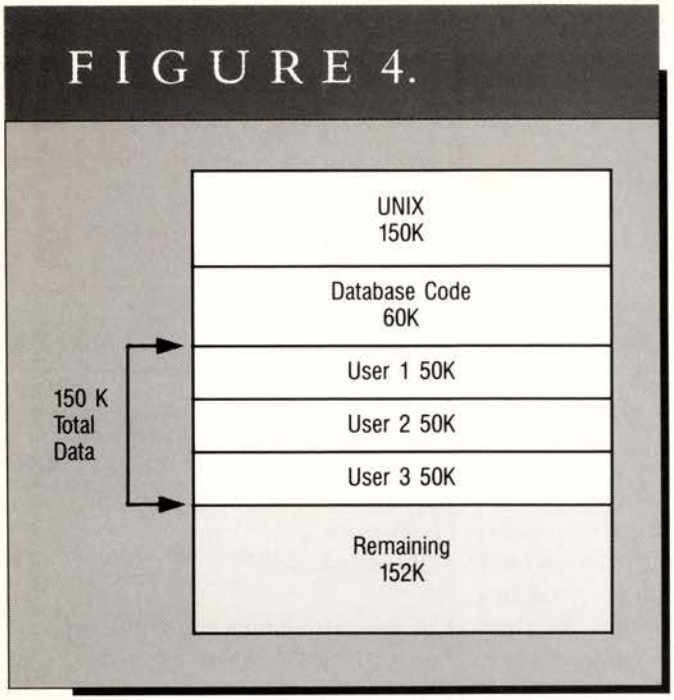
After the processor speed bottleneck is removed, buffer management becomes the point of contention. One process never knows if its buffers are still accurate, so it must always check and frequently "refresh" itself from the new version of the data on the disk. The solution is to share the buffer pools. This can be done using the shared memory facilities of UNIX System V.

Using System V shared memory, a server process searches for data and reads several pages of indexes and records into buffers in the shared data segment. The next time any process requires a search, it achieves an increase in efficiency by using the information in the shared buffers rather than reading it from disk anew.

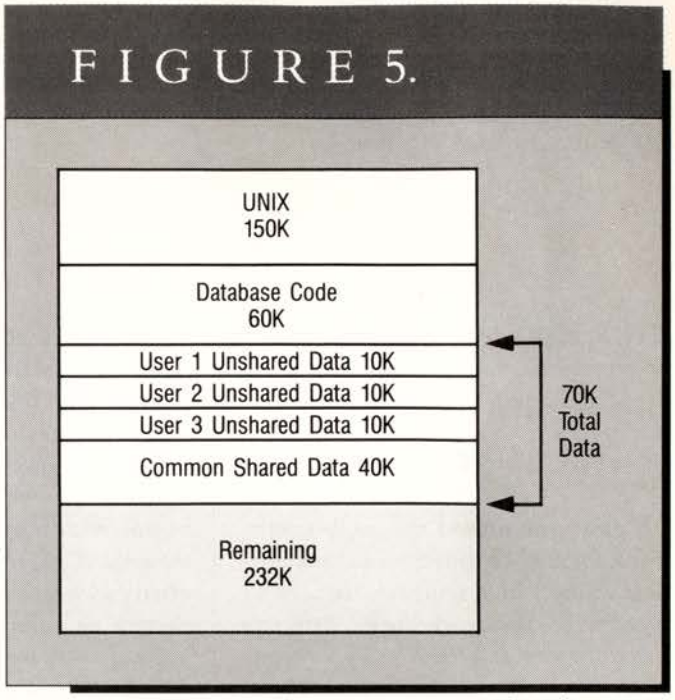
Figure 5 illustrates the reduction in memory requirements when shared memory is used. Using the same application, each user now shares a common pool of data and may require a small unshared data segment for local variables and such, unique to that user's process. The total private data in memory is reduced to 70 KB, a significant savings over non-shared memory. The memory freed up now can be configured for use in the common shared buffer pool.

Common data buffer pooling not only eliminates the need to refresh data buffers, but also consolidates many small buffer pools into one large buffer pool. This large buffer pool then can contain more disk data, reducing the number of overall disk accesses.

Furthermore, now there is no refresh penalty, which, without shared memory, encouraged small buffer pools. With shared memory, the larger the buffer pool the better. When very large buffer pools are used, "RAM disk" speeds can be approached. This invites using additional main memory to increase speed. As the buffer pool fills with frequently accessed data, users of these



Multiple data segments.



Shared memory.

records may not be accessing the disk at all.

The work force is now verging on contentment as it churns through the work. Occasionally, at times of heavy use on large databases, the system still slows and the lurking dread continues that, should the operating system crash, large pieces of data will be lost and confusion will reign.

Unless the buffer pool is larger than the database (which is not likely), disk I/O, the fourth limiting factor on performance, is the next bottleneck. Reducing the frequency of disk access should produce favorable response times even when more users are added.

UNIX constructs several I/O-related barriers that are appropriate for the operating system's general functions, but which inhibit a database from achieving optimal performance.

- After a write statement, UNIX will buffer data and write it later.
- UNIX has a fixed block size (usually 512K) which cannot be controlled by an application that chooses to use UNIX files.
- When it does get around to writing to disk, UNIX puts data wherever there is space, which can result in a single file being broken up and spread around on several non-adjacent tracks.

A very fast, high-volume DBMS must take more control over disk layout and I/O. A "database operating system" like this could include advanced internal and disk data structures and tuneable parameters that can adjust the size of shared memory and the unit of disk I/O. A database like this also would be adaptable to the hardware, taking advantage of the strengths of each machine.

When there are many gigabytes of data on the machine and large numbers of users, the database engine knows the best way to arrange the data on the disk and in the buffer pool to optimize performance. It can control the I/O and thus arrange data on the disk in ways that optimize its retrieval.

It is interesting to ponder the consequences of implementing these solutions. Consider the good response time, even with hundreds of users, that can be achieved from a machine with 16 CPUs (or more) and 32 MB of RAM (or more) in a shared buffer pool managed by a specialized database server back-end. If there is any normal locality to the data use, for example, 10 percent of the data being accessed 90 percent of the time, such a machine should be able to handle transaction processing applications currently only handled by mainframes.

AS USERS SEEK TO increase the performance of their software/hardware systems, the previously described bottlenecks to efficient multiuser performance must be considered. In this consideration, the performance-oriented solution involves multiple processor machines using software that takes advantage of those machines' capabilities.

Multiple database-server processes allow the CPU demand to be spread across several CPUs, and sharing, rather than fragmenting, the data buffers eliminates concurrency control problems that can slow the system down.

Upgrading to System V and using a dedicated back-end to manage database disk I/O will result in a system designed for the expanding needs of the user, with the flexibility to change as the work load changes. *—Richard B. Curtis, Ph.D., is product manager at Informix Software, Menlo Park, California.*

BOOKS

UNIX Shell Programming
Stephen G. Kochan and Patrick H. Wood
Hayden Book Company 1985
422 pages
Paperback, $24.95

I highly recommend this well-written book for anyone interested in shell programming, in particular, the AT&T System V (Bourne) shell. Effective organization and clear, concise style make this book appropriate for both the "occasional" and "dedicated" shell programmer. The authors assume the reader is familiar with UNIX system fundamentals, but a quick review of the basics is provided in case the reader has forgotten some details.

The authors' philosophy is to teach by example; in fact, the book's main strength is its generous use of examples. The discussions on regular expressions and single and double quotes, for instance, were greatly enhanced by the use of effective examples.

Several shell programs are developed throughout the book. As the book progresses, the reader sees these programs gain sophistication. Every line of each program is thoroughly explained. The authors' technique of using the same programs in various stages of development provides excellent continuity. Alternative approaches to the same task frequently are shown so the reader can see the relative merits of various methods.

The book is logically organized with the "building block" material in the earlier chapters. For example, Chapter 3 is dedicated to an explanation of what the shell is, what it does, and how it interacts with other system programs. The book is arranged with longer or more difficult chapters alternating with shorter or easier ones. The chapters are well organized — each begins with a synopsis of the subject material. Chapters and main sections often end with a table summarizing important or complex commands; these tables are very useful for quick reference. An especially noteworthy organizational tool is Appendix A, "Shell Summary," which not only summarizes every shell command, but discusses variables, special parameters, parameter and file name substitution, quoting, I/O redirection, and functions. Examples of each feature or command are included. Appendix A effectively can be used as a standalone reference tool, and usually eliminates the need to search through the book.

A set of exercises is included at the end of each chapter. A typical exercise requires the reader to write a shell program or enhance one of the programs that was developed in the chapter. These exercises do a good job of reinforcing the material presented in the chapter; they typically are realistic and challenging problems. My only criticism of this book is that no answers are provided for the exercises. Although the problems usually have more than one solution, the inclusion of even one answer would be helpful to the reader.

Aside from the minor omission mentioned above, this is an excellent book for those interested in shell programming. The authors effectively cover this material in a very readable fashion. The many examples, good organization, and outstanding command summary (Appendix A) make this book well worth a $24.95 investment.

Exploring the UNIX System
Stephen G. Kochan and Patrick H. Wood
Hayden Book Company 1984
371 pages
Paperback, $20.95

If you're looking for in-depth explanations of UNIX operating system concepts, this is not the book for you. If you're looking for an excellent overview and basic understanding of the important components and features of UNIX, however, you've come to the right place. This is a well-written, well-organized book describing the UNIX operating system, in particular, UNIX System V, Release 2. The authors assume the reader has no prior knowledge of UNIX or any other operating system; accordingly, Chapter 2 is dedicated to explaining what an operating system is and does. Those more familiar with UNIX need not turn away — this book is so broad in scope that even "experienced" users are likely to pick up a few tidbits.

One of the book's main strengths is in providing excellent background information. The introduction (Chapter 1) contains one of the best explanations I've seen for the popularity of the UNIX operating system; Chapter 2, as mentioned above, explains what an operating system is; and Chapter 3 is dedicated to the file system, a fundamental part of UNIX.

This book achieves the authors' stated goals: getting the reader started, giving a good overview of what's available, and teaching the UNIX philosophy

that "small is beautiful." This philosophy applies not only to the physical size of the operating system (UNIX was designed to be small and memory efficient), but to the design of UNIX commands. UNIX has a vast assortment of commands that perform small, well-defined functions; it also has the tools required to combine these commands to perform more sophisticated functions.

Another strong point of the book is the use of tables, diagrams and examples. Commands often are summarized in a table at the end of a chapter or a section. A particularly useful table in the introduction associates topics in which readers may be interested, with recommended chapters. The chapter entitled, "Screen Editing With vi" makes especially good use of diagrams for illustrating what a file looks like before and after issuing vi commands. Finally, the authors use plenty of examples where necessary. The chapter on shell programming, for instance, would not be nearly as readable without its many well-constructed examples.

The diversity of the topics covered in this book make it worthwhile to list them. The following is a brief synopsis of each chapter: Chapter 1 contains much background information and states the goals of the book; Chapter 2 explains what an operating system is and does; Chapter 3 describes the UNIX file system; Chapter 4 covers basic commands; Chapter 5 describes more powerful capabilities such as file name substitution, I/O redirection, pipes, and command files; Chapter 6 covers the basics of shell programming; Chapter 7 explains how to use vi, the full-screen editor; Chapter 8, entitled "UNIX in the Office," discusses "talking" with other users, electronic mail, calendar commands, calculator programs, and word processing; Chapter 9 illustrates why UNIX is so well-suited for program development; Chapter 10 gives a brief introduction to UNIX system security with a discussion of passwords, file protections, the *su* and *newgrp* commands, and file encryption; Chapter 11 discusses the use of a modem, uucp, cu, and networking; and Chapter 12 is a long, informative chapter on such system administration issues as system startup and shutdown, file system maintenance, system accounting, and setting up uucp. Finally, the appendices contain a list of books and papers recommended if you need more information, a comprehensive overview of often used commands, and a complete command summary.

As the list above indicates, the authors of this book have sacrificed some detail in order to cover a broad range of topics. This makes the book particularly well-suited to beginning users and managers who need a good overview of UNIX.

—Reviewed by Lori A. Snyder

MARKETPLACE

Informix Provides DBMS Products

Bringing a new generation of development tools to users of DOS networks, Informix Software, Inc. offers a DBMS package based on Standard Query Language (SQL), for DOS-based LANs.

The enhanced product capabilities of INFORMIX-SQL, INFORMIX-ESQL/C and C-ISAM enable developers to create database applications that enhance the capabilities of LANs and are portable to micro-, mini- and mainframe computers.

Available for DOS networks that support the MS-NET standard operating system interface to MS-DOS 3.1, the Informix products minimize traffic over the LAN by permitting the user to transfer only pertinent files or portions of a file.

INFORMIX-SQL, INFORMIX-ESQL/C and C-ISAM are available for licensing in configurations ranging from four- to 32-node networks.

To learn more, contact Informix Software, Inc., 4100 Bohannon Drive, Menlo Park, CA 94025; (415) 322-4100, Telex: 361834.

Enter 705 on reader card

Simpact BSC Products Available For DEC

Simpact Associates Inc. has announced a family of binary synchronous communications (BSC) protocol interfaces for DEC computers. These products provide multiple, high-performance, link-level BSC connections that are well suited to applications in the news service, banking, securities trading and insurance industries.

The family consists of four products, implementing the 2780/3780, 3270, HASP and financial market data "feeds" protocols. Each product includes the protocol software, a multiport front-end Intelligent Communications Processor (ICP), a distribution panel kit and user documentation. The protocol software runs on the ICP, thereby offloading the time-consuming, interrupt-intensive communications tasks from the host computer. Versions of the products are available for DEC's VAXBI, Q-bus and UNIBUS computers. Driver support is provided for the VMS, ULTRIX or RSX operating systems.

Prices range from $5,300 to $10,500, depending on the bus and operating system selected. Availability in 30 to 60 days.

For more information, contact Simpact Associates Inc., 9210 Sky Park Court, San Diego, CA 92123; (619) 565-1865.

Enter 718 on reader card

Unipress Offers vi-PLUS

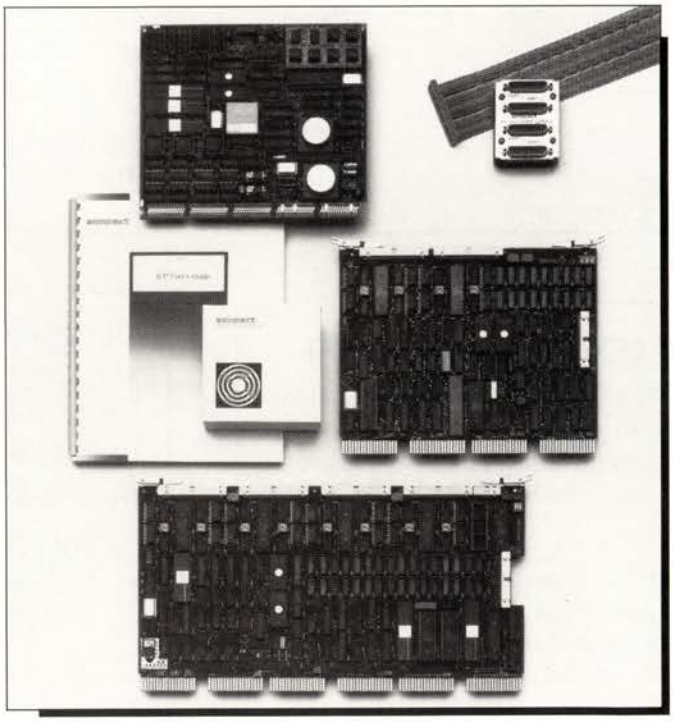
UniPress Software, Inc. has released a new UNIX-oriented text editor called vi-PLUS. The UNIX operating system is provided with the "vi" editor; vi-PLUS is a greatly improved vi.

vi-PLUS is completely vi compatible, but contains improvements such as multiple windows, interactive interface to the UNIX operating system, and extensibility through macros and the MLisp extension language. Standard vi has not adapted these facilities.

vi-PLUS has multiple windows that allow the user to edit and view several files or portions of the same file simultaneously. Files can be regular text, program output, help text, or any other data. vi-PLUS also takes advantage of windowing systems built in to many hardware/software environments, including Xwindows and Sun windows. Mouses on these systems also are supported.

vi-PLUS' interface with the UNIX operating system allows the user to run "shell windows" without exiting the program. vi-PLUS provides the user the ability to customize the editor through macros and the built-in MLisp programming language. Many extension packages are included with vi-PLUS.

vi-PLUS is available for a wide range



Simpact's BSC communication protocol interfaces for the VAXBI bus, Q-bus and UNIBUS.

of computer systems running UNIX, XENIX, ULTRIX and other UNIX derivatives. vi-PLUS is priced at $645 for small PC-style workstations (IBM-PC AT/RT, Sun), $1,590 for large (VAX 750/780, Pyramid) and $3,995 for superminicomputer (VAX 785, 86xx, Gould) systems. Existing UniPress EMACS licensees can inquire about special pricing for vi-PLUS.

UniPress Software, Inc. is located at 2025 Lincoln Highway, Edison, NJ 08817; (201) 985-8000. Telex: 709418.

Enter 702 on reader card

Quality Software Announces New Products

Quality Software Products has made several announcements in recent months.

The company's most recent release is Q-Calc V3.3, a spreadsheet program that contains most of the features found in the top-selling DOS spreadsheets, plus the ability to filter spreadsheet data through arbitrary subprocesses written in any language.

Quality Software also announced Q-Calc Standard, a version of Q-Calc adhering to the standard set by Lotus 1-2-3 release 2.0. Q-Calc Standard is a 1-2-3 work-alike. Spreadsheets may be moved at will among DOS and UNIX machines, or accessed in a network environment. The advanced spreadsheet user will find all of the features of 1-2-3 release 2.0, including the full macro language, the database commands and all of the graph types, coupled to the identical user interface.

In addition, Q-Calc Standard retains all of its UNIX-based features such as shell access, data filtering through other programs and terminal independence via the termcap/terminfo database.

Q-Calc Standard initially will be available for the VAX, Pyramid, Sun, PC-AT under Xenix and System V and the AT&T 3B series of computers.

A new release of the MasterPlan project management package, V1.1, also was announced. Some of the major enhancements are the ability to incorporate the Work Breakdown Structure for a project, with two levels provided for grouping activities and resources, individual resources may have their own calendars, and a resource Gantt screen showing the commitments of each resource over time.

To learn more, contact Quality Software Products, 348 S. Clark Drive, Beverly Hills, CA 90211; (213) 659-1560; TELEX: 494-7422QSP.

Enter 700 on reader card

COMBOARD Lets DEC, IBM Communicate

Software Results Corporation has announced the COMBOARD UNIX/SNA Q-bus for DEC's ULTRIX Marketing Group in New Hampshire.

COMBOARD UNIX/SNA Q-bus is a complete hardware/software interconnect that allows DEC computers to communicate with IBM and other central mainframes.

COMBOARD works as a front-end processor handling all character translation (EBCDIC-ASCII), compression and decompression, transmission error recovery and other protocol operations. The system supports 3278 (LU.2) ter-

minal, 3777 RJE (LU.1) file transfer and 3287 (LU.1 and LU.3) printer emulation software.

COMBOARD features significant savings of computing capacity by handling the interconnect workload rather than the DEC host. The system also can operate at speeds up to 56K. COMBOARD can provide lower overhead, higher throughput and economical operations.

The COMBOARD UNIX/SNA Q-bus is priced from $8,700 to $11,300.

For more information, contact Ernest DeRose at (800) SRC-DATA or (614) 267-2203.

Enter 701 on reader card

FIGARO Supports Graphics Output Devices

Template Graphics Software and Tech-Source Inc. have announced support of FIGARO, the only conformant implementation of the Programmer's Hierarchical Interactive Graphics Standard (PHIGS), with the Tech-Source GDS-3800 VMEbus Graphics Controller. This product combination provides high-level PHIGS graphics support for both system integrators and OEMs.

FIGARO operates on VAX/VMS, IBM CMS and MVS, Apollo/Aegis and many UNIX-like environments. FIGARO also supports more than 175 graphics output devices ranging from high-resolution displays to laser printers and pen plotters.

For more information, contact Stu Reile at Tech-Source Inc., 2955 Xenium Lane, Minneapolis, MN 55441; (612) 559-5716; or Martin Plaehn at Template Graphics Software, 9645 Scranton Road, San Diego, CA 92121; (619) 457-5359.

Enter 703 on reader card

General Robotics Offers UNIX System V.3

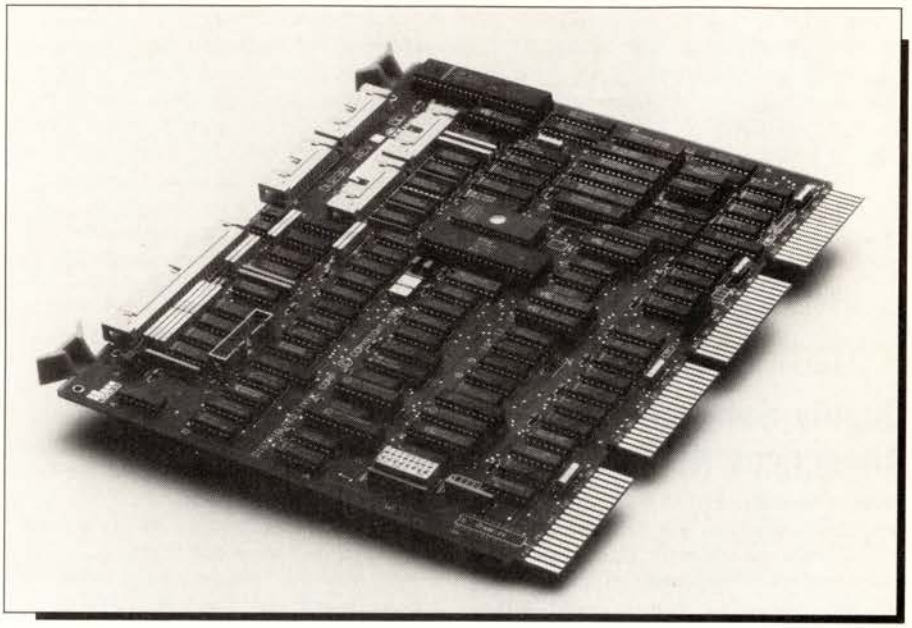
General Robotics Corporation's (GRC) PYTHON family of 32-bit DEC Q-bus-compatible UNIX engines now support System V.3. The GRC offering is its version of National Semiconductor's authenticated port for AT&T's new UNIX release.

One- to two-user V.3 will be standard on all PYTHON, PYTHONS/JR and SUPER PYTHON CPUs. Multiuser licenses will begin at $650. A full networking option will begin at $350. License upgrades from V.2 to V.3 also will be available.

System V.3 offers enhancements to release 2 and provides extended networking features capable of producing "seamless" local system clusters and remote networks.

To learn more, contact General Robotics Corporation, 23 S. Main Street, Hartford, WI 53027; (414) 673-6800; TELEX: 6713838.

Enter 704 on reader card



DILOG's new DU236 UNIBUS SMD controller offers configuration and throughput enhancements.

DILOG Announces New Disk Controller

A new quad height SMD disk drive controller has been introduced by Distributed Logic Corporation (DILOG) that provides significant throughput and configuration enhancements for users of DEC's VAX 8650, VAX 11/730 through 11/785 and PDP-11 UNIBUS computers.

The DU236 disk controller permits the interfacing of up to four SMD/HSMD class drives with transfer rates up to 2.5 MB per second to any VAX UNIBUS computer, and each of the drives also may be mapped as multiple logical units. It is fully compatible with the Mass Storage Control Protocol (MSCP) drivers contained in UNIX, RT-11, RSX-11M+, RSTS/E, VMS and MUMPS operating systems.

The DU236 also supports MSCP-compatible bad block handling. Unique DILOG firmware enhancements provide user benefits that not only significantly improve disk transfer throughput, but simplify subsystem configuration.

The new DU236 SMD/HSMD disk drive controller is priced at $2,950. Volume discounts are available.

To find out more, contact DILOG, 1555 S. Sinclair Street, P.O. Box 6270, Anaheim, CA 92806; (714) 937-5700; TELEX: 6836051.

Enter 717 on reader card

Emerging Technology Has New Software

Emerging Technology Consultants today announced the availability of its EDIX and Professional Writer's Package software products for the VAX series of computers running under the UNIX family of operating systems.

EDIX is a full-screen text editor designed for programmers, writers and other professionals who require power and flexibility. EDIX includes multiple windows and buffers, adjustable screen and window sizes, subprocess execution with standard output capture, configurable keyboards and online help and tutorials.

The Professional Writer's Package is a complete word processing and document development system. This package offers all of the capabilities needed to produce the most complex documents, including textbooks, technical manuals, business plans and reports, newsletters, dramatic scripts and legal documents. The Professional Writer's Package also

includes a 66,000-word spelling checker and an automatic index generator.

To learn more, contact Emerging Technology, 4760 Walnut Street, Boulder, CO 80301-4538; (303) 447-9495; TELEX: 705 128.

Enter 707 on reader card

Alliant Has Support For NCS On The FX/Series

Alliant Computer Systems Corporation announced support for the Network Computing System (NCS) on its FX/Series computers.

NCS, developed by Apollo Computer Inc., is a commercially available set of distributed computing products for developing and running applications programs across networks of multi-vendor computers.

With NCS, users can distribute modules, or parts of a single application program, to the type of computer — workstation, traditional minicomputer, minisupercomputer or supercomputer — best suited to each module's task.

NCS allows users to improve productivity by off-loading computationally intense tasks to Alliant's high-performance processors.

Alliant systems support a number of industry and de facto standards including Ethernet with TCP/IP, the Network File System (NFS) and FORTRAN-77 with VAX/VMS extensions.

To find out more, contact Alliant Computer Systems Corporation, One Monarch Drive, Littleton, MA 01460; (617) 486-4950.

Enter 706 on reader card

Syntax, Masscomp Reach Third-Party Agreement

Syntax and Masscomp have announced a third-party marketing agreement for Syntax's newest LAN server, SMBserver. The Syntax product implements Microsoft's Server Message Block (SMB) specifications and operates on Masscomp's line of minicomputers. Syntax's SMBserver works in conjunction with Excelan's Ethernet controllers for the Masscomp and PC computers and is completely compatible with Microsoft's MS-NET and IBM's PC-NET LAN software.

The SMBserver is priced at $4,000.

For more information, contact Syntax, 6642 S. 193rd Place, Suite N107, Kent, WA 98032; (206) 251-8438.

Enter 708 on reader card

New Products Available For UNIX 4.3 BSD

Advanced Computer Communications (ACC) has introduced three X.25 network communications products for DEC computers running the UNIX 4.3 BSD operating system. They are the ACP 629 and ACP 6250 for VAXs and the ACP 5250 for the MicroVAX II.

The new 4.3 BSD version of UNIX was released by the University of California, Berkeley, last year to replace its UNIX 4.2 BSD. Since the ACP device drivers are compatible with both UNIX 4.2 and 4.3 versions, 4.2 users can connect to X.25 networks without concern about compatibility if they later upgrade to 4.3.

The ACP 629, 6250 and 5250 are single-board, front-end communication

processors that perform on-board X.25 protocol processing in firmware. The user will benefit from enhanced system performance, elimination of most network overhead in the CPU and more application software space.

Price for the ACP 629 is $6,490; the ACP 6250, $6,490; and the ACP 5250, $4,200.

Learn more by contacting Advanced Computer Communications, 720 Santa Barbara St., Santa Barbara, CA 93101; (805) 963-9431; TWX: 910 334-4907.

Enter 712 on reader card

JLCS Releases DENT-11 V4.1

J.L. Computer Systems (JLCS) has released version 4.1 of its DENT-11 system. DENT-11 is an accounting and patient management system designed for the medium to large dental practice or clinic.

The new features include support for four insurance plans per patient; full terminal support for VT200 (7-bit or 8-bit codes), VT100 and PCs; correspondence with more than 3,000 form letters, speed mailers or post cards; automatic daily backup of all data; flexible and powerful report writer; insurance forms printed immediately for the patient; interface for popular word processors; virtual terminal and file transfer packages for communications with PCs or other computers and reports displayed on the user's display spooled to one of four printers (per user) or held in a file for later printing.

DENT-11 is developed for the Micro PDP-11 computer family and uses the features of S&H Computer Systems Inc.'s, TSX-Plus V.6 multiuser operating system with process windowing and Digital Information Systems Corporation's DBL V.4 language. DENT-11 is also available for the MS-DOS operating system.

JLCS system prices range from $26,000 to more than $100,000.

For more information, contact J.L. Computer Systems, 3401 Lancaster Avenue, Wilmington, DE 19805; (302) 998-8030.

Enter 713 on reader card

DECnet CommUnity Expands

National Semiconductor Corporation and Celerity Computing have been granted licenses to port and distribute CommUnity, Technology Concepts' flagship software product that allows non-DEC computers to become full nodes in a DECnet network.

CommUnity is the only implementation of Digital Network Architecture (DNA) for non-DEC systems. The software allows the users of a host computer system to access the resources of a network of CommUnity and Phase IV DECnet systems with up to 1023 nodes per area and up to 63 areas per network.

As an Ethernet-based end-node implementation of DNS, CommUnity provides task-to-task communications, network virtual terminal, remote file access, remote file transfer and network management.

Celerity's C1200, C1230 and C1260 provide compute power of up to 20 million instructions per second and can support up to 256 interactive users with system prices ranging from $38,000 to $94,000.

For more information, contact Technology Concepts Inc., 40 Tall Pine Drive, Sudbury, MA 01776; (617) 443-7311; FAX: (617) 443-7310.

Enter 714 on reader card

NCA Markets STAFFWARE

NCA has acquired the exclusive worldwide rights to market all DEC hardware-compatible versions of FCMC Incorporated's STAFFWARE work group productivity software.

STAFFWARE lets users create electronic documents that can be distributed automatically according to predefined rules. Its user-friendly features allow non-technical personnel to automate much of the administrative routine associated with the processing of forms used every day.

STAFFWARE is presently available for UNIX systems as well as IBM and compatible PCs. The VMS version is under development.

To learn more, contact NCA Corporation, 3250 Jay Street, Santa Clara, CA 95054; (408) 986-1800; TWX: 910-339-9258; FAX: (408) 496-1099.

Enter 709 on reader card

Telexpress Releases Teleterm Series 7

Telexpress Inc. has released Series 7 of Teleterm. Series 7 under MS-DOS is a terminal emulator, customization product and communications system solution.

Some new features include: phone manager, which supports auto-dial and login as well as modem definition capability; expanded unattended operation mode for automated communications sessions; CKermit file transfer protocol now has been added to those previously supported; Teleterm proprietary, XModem and ASCII unblocked.

The program has proved to be popular as a workstation into UNIX/XENIX and VAX systems as well as mainframes, minis and other micros.

Series 7 is priced at $195 per personal computer license.

For more information, call Telexpress Inc. at (609) 877-4900 or write Telexpress at P.O. Box 217, Willingboro, NJ 08046.

Enter 710 on reader card

Edge Computer Acquires Toltec

Edge Computer Corporation has acquired 51 percent of Toltec Computers Inc. Toltec will operate as a subsidiary to distribute Edge Computers with applications using the PICK operating system.

Edge manufactures and markets high-end (DEC 8650 to 8800 class) 680X0 compatible engines, servers and multiuser UNIX systems through OEMs and system integrators in commercial and technical markets. Toltec markets SYMETRIX, an operating environment that integrates PICK with UNIX on the Edge system. Toltec markets through dealers and value-added resellers.

For more information, contact Edge Computer, 7273 East Butherus, Scottsdale, AZ 85260; (602) 951-2020.

Enter 711 on reader card

Medical Manager Released On UNIX

Systems Plus Inc. has released version 6.0 of The Medical Manager physicians' practice management software that will support systems running the UNIX and XENIX operating systems.

The package provides comprehensive features that automate single and multiple practitioner offices from the moment a patient calls for an appointment to electronic insurance billing and patient recall.

The most impressive new feature offered as an option with version 6.0 is The Custom Report Generator. Personalized Programming developed The Custom Report Generator for Medical Manager to augment the more than 100 reports that are provided as standard features with the package.

The Custom Report Generator provides users with the power to create simple one-line reports as well as detailed multilevel reports from information contained in Medical Manager database files. Once a report has been created, it can be

integrated into a custom menu as an option for future selection by the user. It can be purchased for use with all formats of Medical Manager.

For more information, contact Systems Plus Inc., 500 Clyde Avenue, Mountain View, CA 94043; (415) 969-7047.

Enter 715 on reader card

Keyboard Remapping Available With V2.2

Walker Richer & Quinn has announced that remapping and support of the IBM-enhanced keyboard is available with version 2.2 of Reflection 2 terminal emulation software. Reflection 2 links IBM compatible PCs with DEC minicomputers.

Version 2.2 comes with sample mappings for simulating the VT220 keyboard on IBM PC-, XT-, or AT-enhanced keyboards, the HP Vectra and Key Tronic's KB-5151 keyboards. With a program for generating keyboard map files, users can create their own definitions to use with favorite applications or different hosts.

Reflection 2 provides accurate emulation of VT220, VT102 and VT52 terminals on IBM-compatible personal computers. Features include background processing, Walker Richer & Quinn's proprietary command language and error-checking file transfer protocols including KERMIT, XMODEM and the firm's own VAXlink and UNIXlink host programs.

Reflection's optional PLUS feature backs up and restores PC files using the host computer's disk storage.

Reflection 2 users can update to include the new features for $75, or at no charge if within their 90-day warranty period. Reflection 2 is priced at $199. The PLUS feature is an additional $50.

For more information, contact Walker Richer & Quinn Inc. at Eastlake Avenue E., Seattle, WA 98102; (206) 324-0350.

Enter 716 on reader card

Whitesmiths Releases XA8 Cross Assemblers

The XA8 series of cross assemblers for 8-, 16- and 32-bit microprocessors is now available from Whitesmiths Ltd.

The XA8 series provides the standard features expected in modern assemblers including full listing facility, macros, repeats, relocatable and absolute assembly, conditional assembly and separation of a program into functions.

In addition, XA8 provides several enhanced features including a large name list with a capacity of more than 32,000 symbols and the capability of dividing the name list between memory and disk, with user control over the number of symbols resident in memory. The optional Programming Support Package, XSUP, includes linkers and other utilities required to use an XA8 cross assembler to full advantage.

The cross assemblers are available hosted on the MicroVAX (VMS/UNIX), PDP-11 (UNIX/RSX11M+), VAX-11 (VMS/UNIX), XENIX/86, 68K (UNIX/IDRIS) and MS-DOS.

The licensing fees for each cross

assembler range from $400 to $3,600, depending on host, target and other options.

For further information, contact Whitesmiths Sales Department. The toll-free sales number is (800) 225-1030. Inside Massachusetts, call (617) 692-7800. Whitesmiths Ltd. is located at 59 Power Road, Westford, MA 01886; TELEX: 750246 SOFTWARE CNCM.

Enter 720 on reader card

WorldWide Data Offers New Products

Source One International has announced the formation of a new venture, WorldWide Data. The company will develop and market targeted software products to address growing government, corporate and VAR sectors.

Initially, WorldWide Data is offering two product lines: WWD/CHARM, a C-source generator for UNIX-based VAX/VMS multiuser computers; and FormsGen, an easy-to-use system for creating and managing electronically generated forms on IBM, Wang and DEC equipment.

For more information, contact WorldWide Data, 1333 60th Street, Brooklyn, NY 11219; (718) 438-2807.

Enter 721 on reader card

802.3 Gateway Available For DevelNet

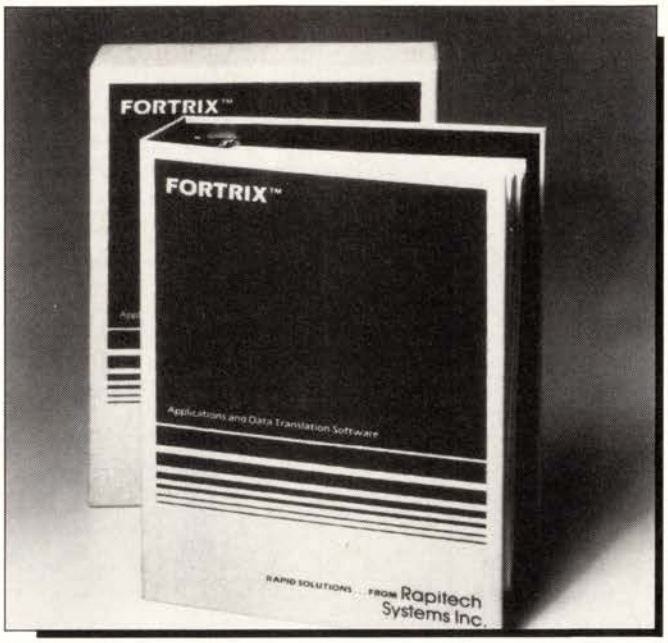
Develcon Inc. has announced the availability of an integral 802.3 gateway card for its Networking Data PBX (NDPBS) offering, DevelNet. This new resource permits asynchronous DevelNet users access to 802.3 LANs operating under TCP/IP and provides an efficient method for interfacing LANs handling file transfers and for NDPBXs engaged in interactive, asynchronous data exchange.

Designated the Model 7182/TCP, this integral product is a single card that can be installed in any DevelNet node to provide a true, direct gateway to an 802.3 LAN for up to 64 virtual circuits. The TCP/IP standard is defined completely throughout all protocol layers, with no proprietary layers, which makes it fully compatible with any LAN operating under the TCP/IP protocol.

The 7182/TCP is priced at $5,000.

For more information, contact Develcon Inc., Suite E, 6701 Sierra Court, Dublin, CA 94568; (415) 829-6200.

Enter 722 on reader card



Rapitech Systems Inc.'s Conversionware version 4.2 speeds FORTRAN to C translation.

Conversionware Speeds FORTRAN To C Translation

A newly released, enhanced version of Rapitech Systems Inc.'s FORTRIX-C Conversionware, which automatically translates FORTRAN programs and files to C language, has been introduced by the company as version 4.2.

The program runs on VAX/VMS, UNIX, MS- DOS and other systems, and on IBM, DEC, AT&T, etc.

It converts 50,000 lines of FORTRAN into C code in approximately two weeks, while manual conversion requires more than one man-year of programming time.

FORTRIX-C version 4.2 has extended and amplified several features and improved some programming limitations. The parser has been made more powerful to recognize even unlikely configurations allowed by the standard. Multiscripted variables now may be passed to subroutines in any of the configurations supported by FORTRAN 77. External and intrinsic statements now are supported fully by FORTRIX-C version 4.2 as well as all of the standard generic intrinsic functions.

The latest version also allows for the handling of formatted and unformatted I/O to conform more closely to the exact definition of the FORTRAN 77 standard. Preconnected scratch files now are supported. Function and subroutine names can be passed as arguments and complex logical groupings involving multiple conjunctions now are fully translated.

Further information is available from Rapitech Systems Inc., Montebello Corporate Park, Suffern, NY 10901; (914) 368-3000; outside NY call (800) FORTRIX.

Enter 723 on reader card

Chromatics Unveils 3-D Graphics System

Chromatics Inc. recently unveiled a high-speed, high-resolution color graphics system in 2-D and 3-D configurations.

Both CX-II 2-D and 3-D models offer 1536 X 1152 display resolution on a 60-Hz, non-interlaced monitor, driven by a 13-board, high-performance graphics engine.

The 2-D configuration of CX-II includes 4 MB of display-list memory, eight planes of double-buffered memory and text overlay and a Sun- and UNIX-based GKS software library.

The 3-D system provides a shading processor and CX3D microcode and firmware with a 2-MB display-list memory, an eight-plane, double-buffer memory and text overlay. It runs VAX/VMS-based CX3D, C and FORTRAN.

The CX-II 2-D and 3-D models are priced at $39,995.

To learn more, contact Chromatics Inc., 2558 Mountain Industrial Blvd., Tucker, GA 30084.

Enter 719 on reader card ■

advertisers index

SALES OFFICES

Professional Press, Inc.
921 Bethlehem Pike
Spring House, PA 19477
(215) 542-7008

VICE PRESIDENT Helen Marbach
NATIONAL SALES MANAGER Jeffrey Berman
ADVERTISING MANAGER Connie Mahon

REGIONAL SALES MANAGERS

MID-ATLANTIC Connie Mahon
INTERNATIONAL Helen B. Marbach
MIDWEST, SOUTHEAST Peter Senft
CLASSIFIED ADS & USED EQUIPMENT Mary Browarek
MARKETING SERVICES Dan Manieri, Spring House, PA
Patricia McCauley, Lexington, MA

NEW ENGLAND
Cynthia Davis
Professional Press, Inc.
5 Militia Drive, Suite 106
Lexington, MA 02173 • (617) 861-1994

NORTHERN CALIFORNIA, OREGON and WASHINGTON
A. G. Germano
Professional Press, Inc.
715 El Camino Real, Suite 206
San Bruno, CA 94066 • (415) 873-3368

SOUTHERN CALIFORNIA (San Diego area)
Kathy Buckley-Miller
Professional Press, Inc.
2365 Seaside Street
San Diego, CA 92107 • (619) 224-9045

SOUTHERN CALIFORNIA and SOUTHWEST
Terry Buckley, Greg Cruse
The Buckley Companies
881 Dover Drive
Newport Beach, CA 92663 • (714) 722-1242

PUBLISHER

Carl Marbach

You Can't Marry Your Cousin

Inbreeding leads to a degeneration of the species. Human beings get away from this by usually not marrying anyone as close as a cousin, certainly not a sister or brother. An injection from a fresh gene pool is necessary to keep us spiraling up instead of down.

The UNIX world is in danger of shutting itself off from the world. UNIX software suppliers hire UNIX gurus from schools that embrace UNIX. Commercial UNIX shops are populated by people who grew up on UNIX. That's on the inside.

On the outside, UNIX suppliers advertise to the UNIX world. UNIX database products are sold to UNIX users. UNIX word processing is promoted to UNIX shops. UNIX EXPO and UNIFORUM are attended by UNIX suppliers selling to the UNIX community.

Who is telling the rest of the world about UNIX? Who is bringing in new blood to write programs, design systems or improve old ones?

If you are waiting for AT&T to sell UNIX to the world, just remember that it hasn't been able to sell computers, so who is going to believe its operating system? Relying on others to tell your tale is not only wrong, it won't work. You have to do it yourself. The famous 10th century teacher Hillel observed, "If I am not for myself, who will be for me? And if I am only for myself, what am I? If not now, when?"

For UNIX, now is the time. Branch out, bring in new people, tell the world about what you are doing and how it fits what they need to do with computers. Hardware is getting cheaper, but we need systems to solve problems, not just iron. If UNIX is an answer, you have to shout it out and let people know. This magazine goes to a large segment of DEC computer people who have or are planning to have UNIX. It is a start, but it needs your help to make it grow into the voice it needs to be.

Help us now. Give a copy to a friend, write an article, tell an advertiser that he should support your attempt to bring the word to the community.

Another famous saying I remember is, "If you will it, it is no legend." UNIX needs all of us to continue to grow and prosper, but it won't happen if someone else has to do it.

THE HIGH TECH RDBMS.

Purity of Purpose
Purity of Design
Discover it.

There are many general purpose Relational Database Management Systems competent at solving "traditional" database problems.

Only one of them however, was designed with the adaptability to properly address the new and expanding range of "High Tech" applications being made possible by continuing advances in hardware technology. Empress.*

This purity of design is why growing numbers of developers of such applications as CAD, Voice Messaging, Photogrammetry and Simulator Design (to name but a few) are selecting Empress, attracted by its adaptability and surefootedness.

Empress alone provides them with the control and flexibility they need while offering the acknowledged productivity benefits of a true relational DBMS, including such expected features as SQL and complete multi-user functionality, as well as a powerful fourth generation application development tool, M-Builder.

In addition, Empress, which currently runs on most UNIX*-based systems as well as VMS* and DOS*, can operate in distributed mode now on several of the most popular networks such as NFS*, DecNet*, and the Apollo Ring.

Not only can any type of data be stored (in effect you can create your own data types), but you can implement your own customized operators to act on your data. It's that flexible.

Call today and discover why Empress is the software tool of choice for High Tech applications.

Empress
The High Tech RDBMS

Rhodnius
Rhodnius Incorporated
250 Bloor Street East, Toronto, Ontario, Canada M4W 1E6
Tel: (416) 922-1743

*UNIX, VMS and DecNet, NFS, and Empress, are trademarks of AT&T, Bell Labs, Digital Equipment Corporation, Sun Microsystems, Inc. and Rhodnius Incorporated respectively.

ENTER 619 ON READER CARD

TAKE YOUR VAX WHERE NO VAX HAS GONE BEFORE

SCO Professional™
1-2-3 WORKALIKE

1-2-3 FUNCTIONALITY FOR THE VAX!

Now VAX™ users have access to SCO Professional™—the ULTRIX™-based workalike of Lotus® 1-2-3®—that offers a user interface and functionality identical to 1-2-3 Release 1A! It even reads and writes directly to existing DOS .wks files, and has extended functionality, such as the larger spreadsheet size and sparse matrix memory management of 1-2-3 Release 2, plus file locking data protection for the multiuser environment.

Microsoft C
XENIX/DOS CROSS DEV

XENIX/DOS CROSS-DEVELOPMENT ON THE VAX!

The same great XENIX®/DOS development environment of SCO XENIX System V for PCs is now available for unprecedented performance with ULTRIX on the VAX! Microsoft® C for the VAX includes the complete C compiler, MASM macro assembler, DOS and XENIX linkers and related tools, plus DOS and XENIX libraries and include files. Object code is generated for both 8086 and 80286 processors, allowing creation of programs compatible with either XENIX or DOS environments—including use with Microsoft Windows.

SCO Lyrix™
WORD PROCESSING

FULL-FEATURED WORD PROCESSING FOR THE VAX!

Add to all of this an advanced, user-configurable word processing system—the new SCO Lyrix® 5.0 —that allows direct menu access to ULTRIX utilities and commands, as well as other application programs! Intuitive and easy to learn for new users, SCO Lyrix 5.0 has features for advanced users such as spelling checker/corrector, automatic footnoting, section/paragraph numbering, widow/orphan line control, and table of contents generation. Use SCO Lyrix 5.0 on your PCs as well as your VAX, and train all your users on the same word processing system.

HOT-LINE SUPPORT
DOCUMENTATION
USER TRAINING
NEXT EXIT

SCO's multiuser traveler's services are unrivaled, and start with SoftCare™ Support, including a toll-free hotline! You can even enjoy 30 days of Introductory SoftCare with the purchase of each product –on us! SCO's excellent documentation is considered a standard for multiuser road guides. And a wide variety of training courses is available for all SCO products.

SCO SALES · INFO
1-800-626-UNIX

Call us today to order any of these and other exciting SCO products such as Multiplan® and Informix® for both PCs and larger systems. VARs, ISVs and other qualified customers can "mix and match" for cumulative discounts across the entire SCO product line. We look forward to booking your next UNIX® System tour.

All SCO applications are also available for your PCs running XENIX System V.

SCO
THE SANTA CRUZ OPERATION

(800) 626-8649
(408) 425-7222 · FAX: (408) 458-4227
TWX: 910-598-4510 SCO SACZ
uucp: ...decvax!microsoft!sco!info

ENTER 608 ON READER CARD