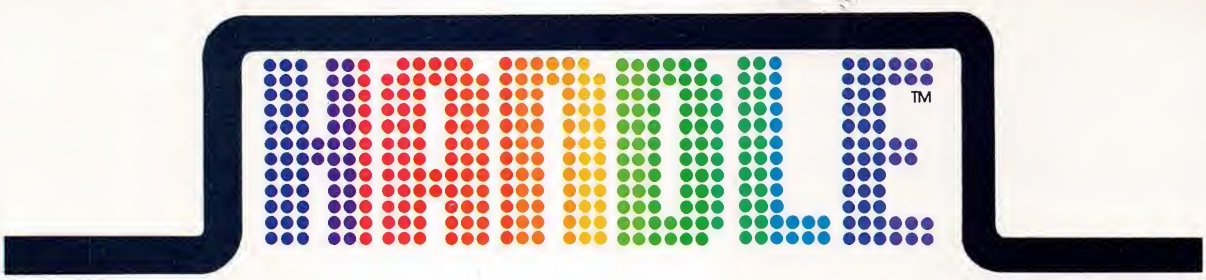
UNIX™ REVIEW

THE PUBLICATION FOR THE UNIX™ COMMUNITY

JULY 1985 $3.95

OFFICE AUTOMATION



Modular.
Integrated.
Now.

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

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

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

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

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

Handle Technologies, Inc.

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

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

UNIX V.2, FAST COBOL & PC-UNIX-HOST NETWORKING

THREE COMPLEX NEEDS ONE SIMPLE SOLUTION

When we designed the UNITE product line, we did it with the systems integrator in mind. For example, we knew if we packed the power of a minicomputer into a chassis the size of an IBM PC, gave it PC to UNIX to MAINFRAME communication and the ability to run COBOL programs under UNIX super fast, we'd really have something.

Well, we did it. It's called the UNITE 8i.

We think you'll find that it's the optimal, multi-user, multi-functional business computer for systems integration on the market.

True, the UNITE 8i is small and powerful. But like all CYB computers, it's also versatile, reliable and priced just right.

PERFORMANCE

The UNITE 8i utilizes a 10MHz MC68010 virtual processor to access up to 12 megabytes of RAM, including up to 4 megabytes of no-wait state memory. A second MC68010 (12.5 MHz) is a high-performance VPM I/O accelerator, which greatly increases the speed of the computer by off-loading the majority of the I/O functions from the main CPU.

The UNITE 8i's third source of power comes from its full implementation of UNIX System V.2 with demand paging, virtual memory, Berkeley 4.2 networking and Ethernet support.

PC NETWORKING

If you've designed your vertical solution to run on a multi-user system in a business environment, you've probably been faced over and over again with an installed base of PCs that your customer wishes to retain.

UNITE solves your problem with the friendliest DOS to UNIX interface available today.

Our UNITE networking software allows PCs to fully access your vertical package and still run DOS application software. Plus, it provides PC to IBM mainframe and DEC VAX communications.

VERSATILITY

If there's one thing systems integrators need, it's versatility. The UNITE 8i is only one of several supermicros from CYB. Each one can be configured with a variety of memory, disk, tape, language (C, BASIC, COBOL, FORTRAN-77, Pascal, LISP) and communications options.

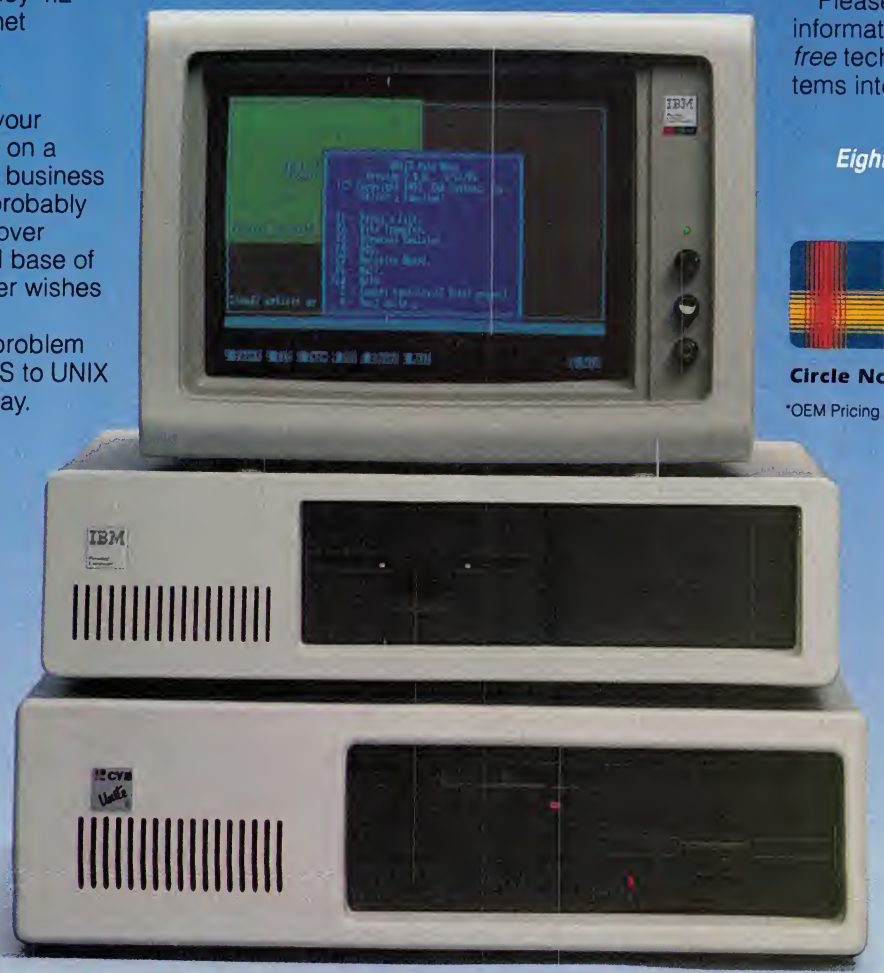
And if you have COBOL programs requiring fast execution we offer two alternatives. Run them under UNIX with PHILON's FAST/COBOL. Or choose the RM/COS operating system option. Either way, you can't lose.

RELIABILITY

CYB computers are rugged. We enclose only the highest quality materials and components in our sturdy, heavy-gauge aluminum cabinets.

After successful completion of rigorous pre-testing and burn-in, we back every system with a full warranty.

Please call or write for more information and ask about our *free* technical training for systems integrators.



Eight-user systems begin under $10,000*.

CYB SYSTEMS, INC
6448 HIGHWAY 290E
AUSTIN, TEXAS 78723
(512) 458-3224

Circle No. 256 on Inquiry Card

*OEM Pricing

CYB...Single Source Solution for Systems Integration

How we improved Structured Query Language.

Actually, we didn't change a thing.

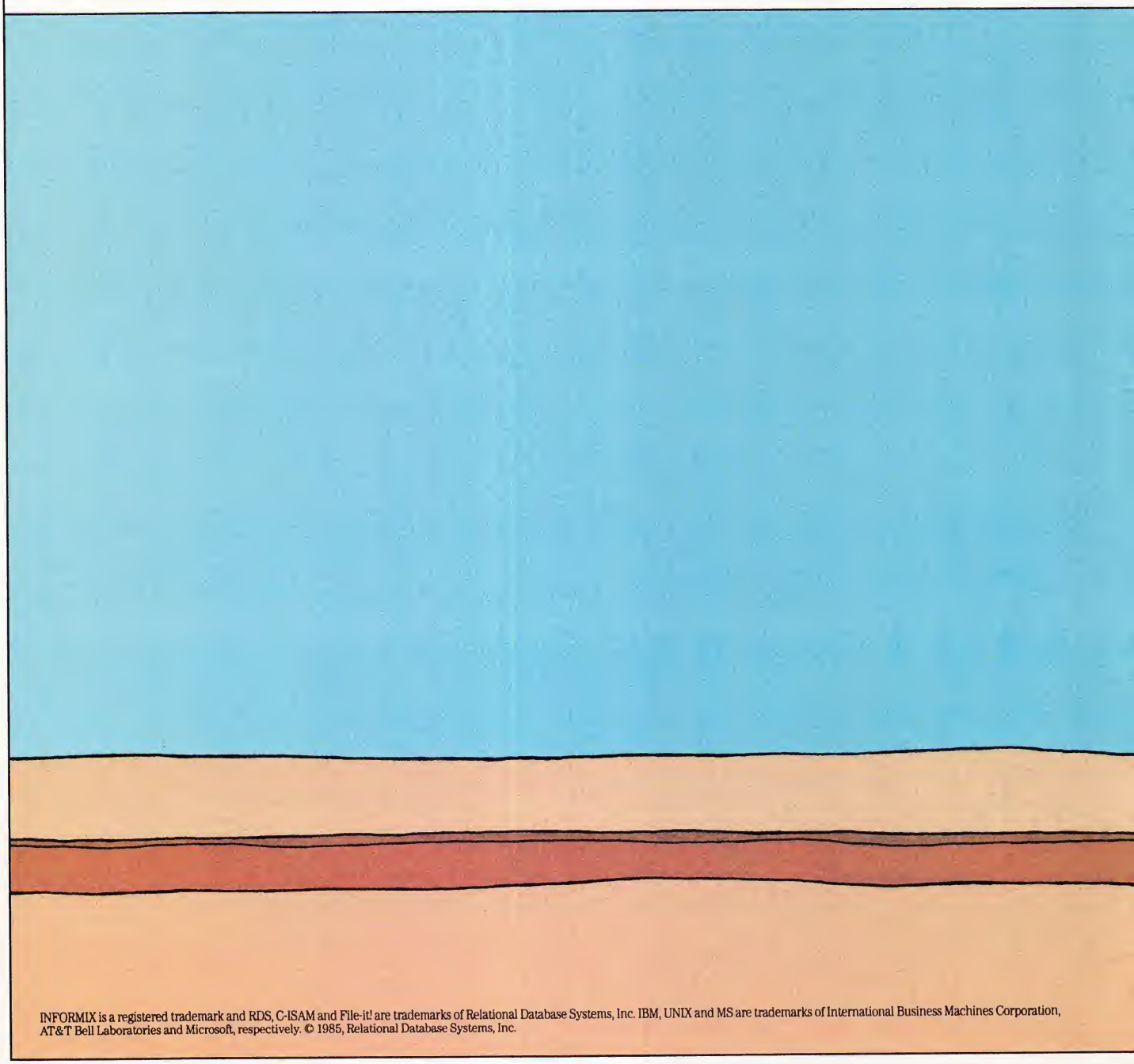
We just combined it with the best relational database management system.

Introducing INFORMIX®-SQL.

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

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

Like our embedded SQLs for C and COBOL. So you can easily link your programs with ours. File-it!™ our easy-to-use



file manager. And C-ISAM,™ the de facto standard ISAM for UNIX. It's built into all our products, but you can buy it separately.

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

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

So call us for a demo, a manual and a copy of our Independent Software Vendor Catalog. Software vendors be sure to ask about our new "Hooks" software integration program. Our number: 415/424-1300.

Or write RDS, 2471 East Bayshore Road, Palo Alto, CA 94303.

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

RDS™

RELATIONAL DATABASE SYSTEMS, INC.

UNIX™ REVIEW

THE PUBLICATION FOR THE UNIX COMMUNITY

Volume 3,
Number 7
July 1985

DEPARTMENTS:

6 **Viewpoint**

8 **The Monthly Report**
By Roger Strukhoff

18 **The Human Factor**
By Richard Morin

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

64 **Industry Insider**
By Mark G. Sobell

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

72 **C Advisor**
By Bill Tuthill

79 **The UNIX Glossary**
By Steve Rosenthal

83 **Recent Releases**

92 **Calendar**

100 **Advertiser's Index**

Cover art by Stephen G. Luker

FEATURES:

25 **UNIX IN THE OFFICE**

By Tanya Joyce

A look at what office automation entails and how UNIX plays a role.

OFFICE AUTOMATION

31 TOOLS FOR AUTOMATION

By Rob Warnock

Programming techniques suitable for uninitiated users. Includes sample scripts.

40 MAIL IN THE ELECTRONIC AGE

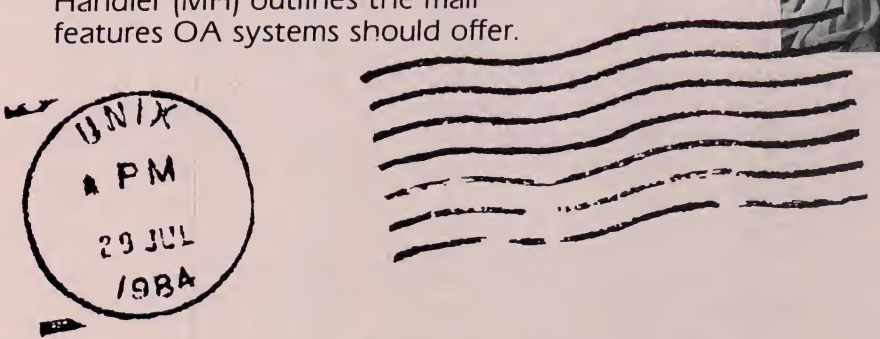
By Bruce Borden

The author of Rand Corporation's Mail Handler (MH) outlines the mail features OA systems should offer.

45 INTERVIEW WITH LARRY TESLER

By Dick Davies

An OA graybeard lays out office concepts from Xerox PARC and discusses the role UNIX might play.



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

UNIX REVIEW is sent free of charge to qualified individuals in the U.S. The publisher reserves the right to determine qualification. For those not qualified, there is an annual (12 issues) subscription charge of $28 in the U.S., $35 in Canada and $48 in all other countries (surface mail). Correspondence regarding editorial (press releases, product announcements) and circulation (subscriptions, fulfillment, change of address) should be sent to 500 Howard Street, San Francisco, CA 94105. Telephone 415/397-1881. Correspondence regarding dealer sales should be sent to 901 South 3rd Street, Renton, WA 98055. Telephone 206/271-9605.
Letters to UNIX REVIEW or its editors become the property of the magazine and are assumed intended for publication and may so be used. They should include the writer's full name, address and home telephone number. Letters may be edited for the purpose of clarity or space. Opinions expressed by the authors are not necessarily those of UNIX REVIEW.
UNIX is a trademark of Bell Laboratories, Inc. UNIX REVIEW is not affiliated with Bell Laboratories.

VIEWPOINT

The humane office

"Office automation" is a phrase many find disturbing—myself included. It conjures up visions of offices that aren't only paperless, but people-less. A scene influenced by "2001"—or is it "1984"?—comes to mind. Robots answer the telephones, VersaSecretaries stand at the ready, and an unblinking electronic eye alerts Big Brother to anything that moves.

Who knows? We may yet see that day. I hope not. But, if we do, it won't be in the foreseeable future. The office of today, tomorrow, and several tomorrows to come will continue to be run, operated, and staffed by real flesh-and-blood humans.

It's to these people that this issue is devoted. Happily, robotics are not yet a topic we need to focus on here. Rather, this issue studies the much larger body of software designed to enhance the office experience for humans.

This provides a better profile of office automation—one that shows computers at their best. Unlike people, office systems don't (usually) complain when drudge work weighs them down. Fatigue and boredom are not an issue. And, while not error-free, computer-produced documents, spreadsheets, graphics, and the like are measurably cleaner than comparable human output. Needless to say, the materials generated by computers also can be produced faster.

But what does all this spell for humans, apart from fewer jobs? It means better jobs, actually. With the necessary drudge work of the office consigned to computers, people are left free to think and create. The skills and qualities that are distinctively human are thus emphasized. Computers, after all, don't generate the ideas that fuel offices; they merely facilitate the work that takes place within them.

Office automation may have been discovered only recently by the press, but it's been a reality for some time. Think, if you will, of the typical office of as little as 10 years ago. A rather alien place, isn't it? Few of us, I think, would care to go back. I, for one, would sooner relinquish my car than my word processor. Use a typewriter? Me? Are you kidding?

This, perhaps, is not a new story. But a few chapters have been overlooked. The goal of this issue is to get at some of that.

Tanya Joyce gets the project rolling with a look at what "office automation" is, where it's heading, and what its connection to UNIX is. Defining the term is no small matter since it's first necessary to cut through the encrustments of buzz phraseology.

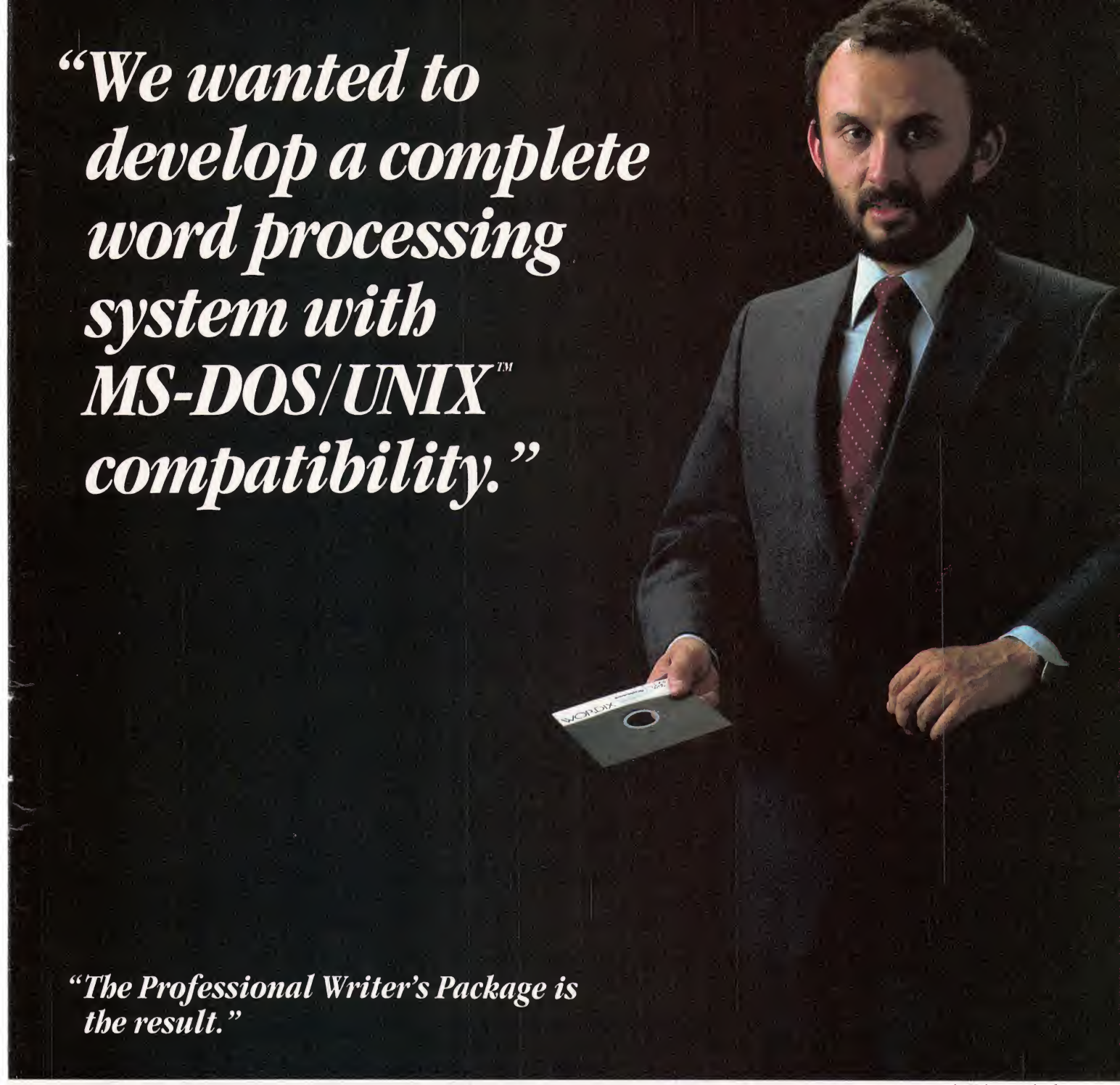
Home-spun office automation anchors the following piece contributed by Rob Warnock. The gaps between commercial office software are the target; a friendly tutorial on programming for the uninitiated office worker is the result.

Mail and its role in the office is explained by Bruce Borden, the gentleman who authored the Rand Corporation's Mail Handler (MH) software. Instructive comments about the mail features office software should offer form the heart of the article.

Final comments on the role of UNIX in office automation are offered by Apple's Larry Tesler, whose OA research reaches back to some of the field's earliest work done at Xerox PARC. Tesler's observations come by way of an interview conducted by Dick Davies.

Further contributions were made by reviewers John Mashey, Ted Dolotta, Eric Allman, Bill Freiboth, Steve Hendricks . . . and, of course, my trusty word processor.

Mark Compton



Iván Pérez-Méndez
President, Emerging Technology

"Drawing on years of word processing experience, we've designed a program with all of the features necessary for producing everything from simple letters and memos to complicated documents, like technical manuals and dramatic scripts.

"At the same time, we realized that you don't want to spend a lot of time learning a new program or thumbing through documentation. So the **Professional Writer's Package** comes with on-line help, on-line tutorials and a shell that takes you step by step through all phases of document preparation and file management.

"And the files created with the **Professional Writer's Package** are compatible with MS-DOS® files. Since the **Professional Writer's Package** has already been used successfully at thousands of installations, you get a proven product with proven capabilities.

"If you want to learn more about the **Professional Writer's Package**,™ call or write today."

EMERGING TECHNOLOGY 4760 Walnut Street, Boulder, Colorado 80301 303/447-9495, 800/782-4896

Circle No. 254 on Inquiry Card

THE MONTHLY REPORT

Off it goes, into the big brown dustheap

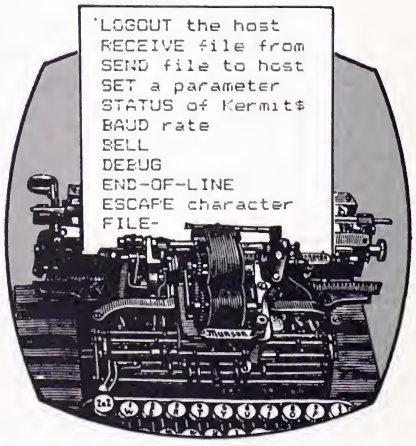
by Roger Strukhoff

The U. S. Air Force has put the brakes on a move to acquire huge numbers of UNIX systems. A request for information issued about a half year ago, known as AFCAC 251, indicated that the Air Force was thinking about purchasing 15,600 UNIX supermicros—an astounding number that would have represented more than $1 billion in sales.

An Air Force official involved in acquisitions said that bids responding to AFCAC 251 had numbered "in double figures", but that the Pentagon brass is now "telling us to slow down a bit." This official said the Pentagon is questioning a purchase of that size that consists of small computers only, and "is wondering if (smaller systems buys) were getting out of control."

A strategy panel will meet in the coming weeks to sort out exactly what is needed. "(High command) wants to be reassured that we are not abusing the program," the official said. After it is determined whether or not to proceed with large purchases of smaller systems, the need for UNIX will also be re-considered, he said. There are basically three options before the panel:

1) go ahead with AFCAC 251 as originally specified;
2) issue a new request for information, with as yet undetermined similarity to AFCAC 251;
3) abort the entire smaller system purchase.



The official said he expects the panel to arrive at a conclusion within the next three to six months. "If it goes to, say, nine months, then I think we'll abort," he said.

OUR POLICY IS NO POLICY

The portability of the UNIX operating system makes it an attractive option for many large-scale buys; indeed it seems to be a "natural" for military purchases. But there is "no set policy directed at UNIX (systems)", according to John Laine, director of information systems for the DoD. Laine, who works directly under Assistant Secretary of Defense John Latham, said "(UNIX-based systems) meet a lot of objectives" of the military. The portable nature of UNIX is chief among its attributes. "We don't want to get locked into systems that might not meet our needs down the road," Laine said.

"Interoperability" is another word thrown about these days. At root, the term means that the military, like the business world, is interested in systems that exhibit upward and downward compatibility, communicate readily, and make use of similar software—in a word, "interoperate." Laine split semantic hairs on this point. "You can get interoperability with other operating systems," he said. "(But) the transportability of UNIX is conducive to interoperability.

"The DoD does not mandate policy for purchase of computer systems," Laine asserted. "Individual branches (of the military) or certain departments may feel strongly about one system or another . . . but we can't standardize (on a specific system); that would be anti-competitive."

Laine also mentioned a key advantage of Berkeley 4.2, namely that it "complies with DoD standard protocol structure." It is here, in the area of protocols, that the DoD *does* mandate policy. "There are well-defined interfaces," Laine explained, adding that "though we won't say you

Tomorrow's Engineering Tool

Today

The C1200 computational system brings mainframe performance and features to the individual user. It provides the power and local control needed for today's most compute-intensive engineering and scientific applications, including modeling, analysis, and simulation.

A true UNIX 4.2 BSD implementation complements the register-intensive 32-bit advanced architecture providing 2.3 million Whetstones/second speed. Up to 24 Mbytes physical memory, 4 Gigabytes virtual memory, high performance disks and I/O subsystems enable the C1200 to deliver multi-user capability at a single user price.

The C1200 offers high resolution bit mapped color graphics and incorporates industry-standard compilers, networking, communications, and interconnection options to complement other systems from PC's to Supercomputers.

The C1200—Proof of Celerity's Dedication to Performance Technology.

Get in touch with tomorrow today. Contact Celerity Computing.

CELERITY COMPUTING

Corporate Headquarters 9692 Via Excelencia, San Diego, CA 92126, (619) 271-9940
Sales/Support Offices Framingham, MA/(617) 872-1552; Houston, TX/(713) 496-2128; Farmington Hills, MI/(313) 553-7833; Woodland Hills, CA/(818) 703-5141 and Santa Clara, CA/(408) 986-8466

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

Circle No. 237 on Inquiry Card

The $750 In Your

AT&T 3B2™

One select investment can put you on top of UNIX™ applications development. Today. No more waiting for the future.

That one investment is UX-Basic™. It increases your productivity and gets your programming done on time, and on budget.

One complete, easy to learn, professional programming language will run all your applications without change, on micros to mainframes.

UX-Basic gives you all the power of UNIX to develop your business and scientific applications. You get productivity tools to save you time and money including a full featured line editor, automatic syntax checking, on-line help, meaningful error messages, and real-time debugging, for fast program development and modification.

UX Software, Inc. has prepared for the IBM PC/AT™ and AT&T 3B2™ and AT&T UNIX PC™, a limited 1,000 copy edition of the UX-Basic Development System, exclusively for software developers and UNIX users, and code named the 9000 series.

UX-Basic Features Profile

- automatic file and record locking
- abbreviated command input
- automatic backup copy
- BCD arithmetic
- C-ISAM™ bundled in
- common and local variables
- date and time functions
- device and terminal independent
- European format standards
- formatted business printing
- long variable and label names
- matrix operations
- modular structured code
- multi-line functions and subroutines
- multi-line IF/THEN statements
- multi-tasking and multi-user
- OASIS™ Basic compatibility
- recursive routines
- re-entrant code
- sequential, direct and indexed sequential files
- shell access
- software protection options
- sixteen dimension arrays

IBM PC/AT™

Investment Future

The full UX-Basic Development System includes:

- UX-Basic: interpreter for creating source code.
- UX-Compile: the pseudo-code compiler.
- UX-Runtime: for executing compiled code.
- UX-Utilities: demonstration and maintenance programs.
- UX-Basic User Manual.

In addition to this, the 9000 series UX-Basic Development System edition provides extended service and support.

You will receive:

- A demonstration version of UX-Basic, including sample business programs, that you may duplicate and distribute.
- Direct product support from UX Software Inc., for 180 days.
- Free product updates for one full year.
- A subscription to /usr/uxsoft/notes, the insider's technical notebook for UX-Basic programming.

Order UX-Basic today. There is no more waiting for the future.

AT&T UNIX PC™

☑ I want to invest in my future.

Please rush me a 9000 series UX-Basic Development System, complete with sample business programs.

NAME _____

COMPANY _____

ADDRESS _____

_____ TELEPHONE _____

☐ AT&T 3B2 ☐ AT&T UNIX PC ☐ IBM PC/AT

PAYMENT ☐ AMEX ☐ MC ☐ VISA ☐ Check

To arrange for immediate delivery, call now (416) 964-6909.

UX Software, Inc.

10 St. Mary Street, Toronto, Canada M4Y 1P9

Telex: 065-24099 UXSOFT TOR

(416) 964-6909

Circle No. 260 on Inquiry Card

have to buy this system or that, we can say the system must have a specific interface."

UNIX EXPO LOSES IDENTITY

Computer Faire Inc.'s UNIX Systems Expo has apparently lost "the battle of show run", and will now be doing its old soldier impersonation at selected theaters. Following a surprisingly small turnout of exhibitors and delegates at the spring show in San Francisco, the UNIX Systems Expo fall show will be merged with the 8th Northeast Computer Faire, to be held September 26-28 in Boston. (The USE fall show had been scheduled for October 2-5.) There will be a separate "UNIX pavilion" at the NECF and a separate conference program, according to Computer Faire president David Sudkin, but one admission will grant access to the entire Faire.

The UNIX Systems Expo fall show will be merged with the 8th Northeast Computer Faire.

Sudkin expects a turnout of 20-25,000 at the NECF, down slightly from last year's attendance of 27,000. There will be a total of 600 booths, he said, including between 35 and 50 UNIX-oriented exhibitors. (Sudkin said there would have been 20-25 UNIX-oriented booths at the Computer Faire had USE gone on as scheduled.)

One reason for the lower projected attendance at this year's NECF is its location at the Bayside Convention Center, inconveniently located on the southern outskirts of Boston. The Faire was held at centrally-located Hynes Auditorium last year, but this facility is presently undergoing renovation.

It seems likely that the UNIX Systems Expo will fade away altogether after the Northeast Computer Faire, although Sudkin insists that NECF results will be reviewed before any decisions are made.

There may be other changes in store for Computer Faire, Inc., as well. The company was purchased by Prentice-Hall last year, and Prentice-Hall was in turn bought out by Gulf & Western. G&W then merged its publishing activities—Prentice-Hall, Simon & Schuster, and Esquire—into a single division of the company, where it does not see Computer Faire, Inc. fitting in. Rumor has it that the company can be bought for between $2 and $2.5 million.

The Northeast will still have an autumn show devoted entirely

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

"UNIPLEX II"™

YOU'VE GOT IT!

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

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

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

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

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

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

OEM terms available. Mastercard and Visa accepted.

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

Write to: UniPress Software, 2025 Lincoln Hwy., Edison, NJ 08817 or call: **1-800-222-0550 (outside NJ) or 201-985-8000 (in NJ)**; Telex: 709418. Japanese Distributor: Softec, Telephone: 0480 (85) 6565. Swiss Distributor: Modulator SA, Telephone: (031) 59 22 22.

UniPress Software

Your Leading Source for UNIX™ Software.

to UNIX. UNIX Expo, sponsored by National Expositions Inc., is scheduled to be held September 18-20 in New York. Attendance at this show will be watched closely by industry observers who hope to find signs of how well UNIX will fare in the "real world."

HOOKS PROGRAM GETS UNDERWAY

There are about 15 new members of Relational Database Systems' ESQL/C "hooks club", but no new products have appeared quite yet. RDS announced the hooks program, in which participants receive a copy of Informix-ESQL/C, distributed partly in source code, at the Dallas UniForum Conference in January. There were about a dozen initial participants, but others came into the fold at about the time the software became available this spring.

"Standardization" is the magic word associated with the RDS hooks program. RDS has committed itself to IBM-developed SQL (Structured Query Language) as *the* tool for database retrieval and manipulation. With enough people participating in the hooks program, a large cross-section of UNIX software will theoretically become compatible with a common DBMS base. RDS claims this will lead to "true integration", since applications software (word processing, spreadsheets, graphics—the whole gamut) tied into the SQL hooks will be able to share a common data pool.

RDS Embedded SQL is comprised of a pre-processor, a back-end process called the "Database Agent", and a library of functions that communicate with the database agent. Companies participating in the hooks program receive source code for the library.

RDS is now mounting a major direct mail effort to enlist new members for its club. The effort is being directed not only at known UNIX software developers, but also at possible fellow travelers such as MS-DOS software companies who may want to migrate to multiuser UNIX applications, as well as a number of smaller developers who may have been shut out of the MS-DOS market.

THE STANDARDS CONUNDRUM

The quest for standards is urgent at all levels of the com-

TEXT EDITING

NEW RELEASE

UNIPRESS EMACS™

VERSION 2

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

Subject: Multi-window, full screen editor.

Multi-window, full screen editor provides extraordinary text editing. Several files can be edited simultaneously, giving far greater programming productivity than vi. The built-in MLISP programming language provides great extensibility to the editor.

New Features:

- *EMACS is now smaller and faster.*
- *Sun windows with fonts and mouse control are now provided.*
- *Extensive on-line help for all commands.*
- *Overstrike mode option to complement insert mode.*
- *New arithmetic functions and user definable variables.*
- *New manual set, both tutorial and MLISP guide.*
- *Better terminal support, including the option of not using unneeded terminal drivers.*
- *EMACS automatically uses terminal's function and arrow keys from termcap and now handles terminals which use xon/xoff control.*
- *More emulation – TOPS20 for compatibility with other EMACS versions, EDT and simple WordStar™ emulation.*

Features:

- *Multi-window, full screen editor for a wide range of UNIX, VMS™ and MS-DOS™ machines.*
- *"Shell windows" are supported, allowing command execution at anytime during an edit session.*
- *MLISP™ programming language offers extensibility for making custom editor commands! Keyboard and named macros, too.*
- *"Key bindings" give full freedom for defining keys.*
- *Programming aids for C, Pascal and MLISP: EMACS checks for balanced parenthesis and braces, automatically indents and reformats code as needed. C mode produces template of control flow, in three different C styles.*
- *Available for the VAX™ (UNIX and VMS), a wide range of 68000 machines, IBM-PC,™ Rainbow™ 100+, and many more.*

Price:

	Binary	Source
VAX/UNIX		$995
VAX/VMS	$2500	7000
68000/UNIX	395	995
MS-DOS	475	*

*Call for terms

For more information on these and other UNIX software products, call or write:
UniPress Software, Inc.,
2025 Lincoln Hwy.,
Edison, NJ 08817.
Telephone: (201) 985-8000.
Order Desk: (800) 222-0550.
(Outside NJ). Telex: 709418.
Japanese Distributor:
Softec 0480 (85) 6565.
European Distributor:
Modulator SA (031) 59 22 22.

OEM terms available.
Mastercard/Visa accepted.

UniPress Software

Your Leading Source for UNIX Software.

puter industry and the UNIX community is no exception. But are standards the panacea they have generally been assumed to be? In a driven effort to standardize, will the best solution, or even a good solution, evolve? Or, as with many human endeavors, will only the lowest common denominator result?

A standardization effort recently mentioned in this column was the merger of the /usr/group standards committee with the IEEE P1003 standards committee. Jim Isaak, director of product planning of Charles River Data Systems and chair of the P1003 committee, has elaborated on this effort. The first meeting of the P1003 was in February, the maiden voyage of the combined /usr/group-P1003 came in June, at the Portland Usenix show. This committee is attempting to establish a reference model for open systems architecture.

> An OSA model would try to draw together several individual models, so there's no gap to fill on the user's part.

Areas for which specifications need to be set include the bus, networking scheme, operating system, languages, graphics, data interchange method, data storage, DBMS approach, and—last but not least—the interface. Isaak explained that an OSA model would try "to draw together (several individual) models, so there's no gap to fill on the user's part."

It's likely that not everyone will be happy with the combined /usr/group-IEEE efforts. One software developer, Marc Rochkind, president of Boulder, CO-based Rochkind Software, has pointed out problems with a past /usr/group efforts to develop a UNIX system standard. "They addressed only Berkeley 4.1, for one thing," he said, "and (the more widely used) Berkeley 4.2 has diverged from System V far more than 4.1." Rochkind also noted that terminals and interprocess communications, "two of the critical issues," were not addressed.

"In a sense, things have gotten worse than they were two years ago," Rochkind stated. "People think there is a standard, so they go on to something else . . . meanwhile, software developers are no better off." Rochkind also feels any UNIX-based system standardization effort must develop an adequate record-locking facility, an area long regarded as a weakness of UNIX systems. Without such a specification, any standard would be "naive" and "inappropriate for DBMS", he said.

Rochkind does take a hopeful attitude toward the /usr/group-IEEE effort, though. "I certainly have nothing negative to say (about this new effort)," he said. "With the IEEE's experience in creating standards, I imagine they may be successful in this

area," he said.

The IEEE itself recognizes the problem of trying to create standards, stating that even though committee members "are presumed to have expert knowledge on the subject . . . it must be understood that an IEEE standard represents a consensus."

ORIENTAL BIZARRE

Most people are familiar with the frequently unwelcome sight of "boom boxes" on city streets, laying down a beat for all to hear—whether or not they care to hear it. Now no less a figure than Ma Bell herself has become annoyed by the ubiquitous noisemakers. It seems that Marantz, the large Japanese electronics company, was one of the first to offer portable music makers, introducing a system in Japan called the "UNIX" many years ago.

Perhaps it's appropriate that a term coined partly in jest should now be involved in what appears to be a rather comical situation.

The name was duly trademarked, and now Marantz feels it needs to have some Yankee dollars if it's to let AT&T use the term "UNIX" in Japan. Never mind that AT&T's and Marantz's development of the same term is completely coincidental and applies to non-competing products. AT&T, of course, has plenty of dollars, but doesn't wish to part with them to solve problems such as this. Perhaps it's appropriate that a term coined partly in jest should now be involved in what appears to be a rather comical situation, to non-AT&T folks, at least. Negotiations continue, according to AT&T.

Roger Strukhoff is the Associate Editor of UNIX REVIEW. ■

THE HUMAN FACTOR

Constraints, tools, and virtuality

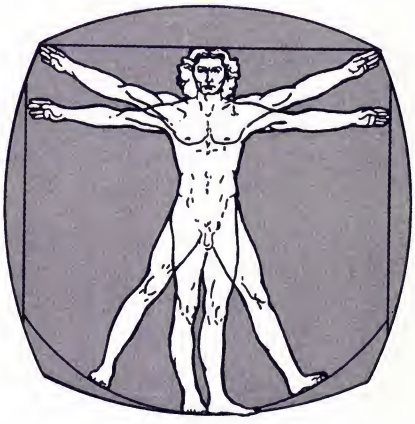
by Richard Morin

Simple tasks have a way of becoming more complicated. This is often the result of the discovery, or imposition, of constraints. Painting a room is easy; just drop a firecracker into a bucket of paint. A neat painting job, however, takes a bit more work. Fortunately, many constraints can be handled by using tools. One might even assert that any use of tools is an exercise in overcoming constraints.

Computers, as tools, allow us to circumvent limitations in our mental capacities. We have invented still more tools to help us use computers. Modern languages and operating systems allow a vast range of machine architecture constraints and complexities to be ignored.

UNIX, as an operating system, is almost fanatical on the subject. It hides details of implementation whenever possible. Give UNIX some data to be put into a file, and it cheerfully complies. It doesn't ask for the file's format, access method, or even its expected size.

As a multiprocessing system, UNIX allows users to act as if large numbers of computers were available to run processes. It then allows processes to ignore petty details like limitations on available amounts of memory, device peculiarities, and so forth. Each process is given its own environment and has occasional use of



the machine's time.

The UNIX process environment is rife with useful imaginary creatures. Most of these are entirely passive, responding only to commands from users or to instructions from other processes that act independently, reacting to the passage of time or to changes in the hardware or software environment. These daemons ("any of the secondary divinities residing between the gods and men") perform such tasks as coercing hardware devices into performing their duties in a reasonably cheerful manner.

Users are allowed, even encouraged, to populate their environments with any assortment of processes and files they can invent. They are assisted in this endeavor by a variety of free-floating aliases, variables, pipes, and pseudo-terminals. The implementation details of these handy, albeit imaginary resources are, of course, hidden from the user.

VIRTUALITY

This invisible provision of imaginary resources deserves a name. Let's call it "virtuality", generalizing a term already used to describe a range of features such as memory, file access, device metafiles, and so forth. It generally involves some sort of tradeoff, paying for a desired feature by sacrificing another less critical capability.

Floating point arithmetic is a good example of virtuality. Current computer systems pretend that they understand real numbers. Fortran compilers even call these numbers **REAL**s. The C language at least has the good sense to call them **float**s since, lacking infinite range and precision, they are not truly "real numbers".

Floating point numbers are, in fact, a kind of limited precision scientific notation. The range of a 32-bit floating point number is large, around 40 decimal places. Its precision, however, is limited to a little over six decimal places. Since a 32-bit integer is accurate to more than nine digits, a swap of precision for range has obviously been made.

This is a compromise with reality, but since it handles most of our computational needs, we accept it. The subterfuge can be safely ignored most of the time, since this is enough precision for most calculations. Still, truncation and roundoff errors do trap occasional unwary programmers.

Virtual memory is an even clearer case. The machine pretends that it has large amounts of imaginary memory. It keeps inactive pages of data on disk, and bails processes out of trouble when they reference unavailable memory locations. None of this is free, of course. There is, in fact, a considerable amount of overhead involved.

The system must allocate disk space for all the virtual memory it might *ever* wish to provide. It must also allocate substantial amounts of real memory for relocation data. Its time is occupied copying pages, restarting processes, and so forth. Finally, there are "gotchas" to be observed, as in floating point arithmetic.

The UNIX process environment is rife with useful imaginary creatures.

There is an implicit assumption that memory use will be limited to a few areas of memory at a given time. The system can serve those needs by keeping only active pages in memory. Consequently, programs with memory usage that bounces all over can easily bring a virtual memory machine to its knees.

Nevertheless, most programmers are willing—nay, eager—to embrace floating point arithmetic, virtual memory, and other such illusions. The costs are acceptable, and the benefits are enormous. Still, the frontiers of virtuality beckon us onward, and we must go.

VIRTUAL TIME

Response time is a constraint on any number of interesting applications. If a command takes too long, it won't get used—even if it does some really nice things. Since computers in fact spend a great deal of their time doing precisely nothing, a readily available, cheap resource is at hand. By finding ways to use this time productively, and saving the results for later, we can reduce the response time of nifty but difficult commands.

This is not a new idea; UNIX already does it in a variety of ways. The spelling checker searches a list of known words, trying to match the words found in a suspect document. In order to save the user's time, the **spellin** utility is run on new word lists to generate a hashed file. (See the "Programming Pearls" section of the May, 1985 *Communications of the ACM* for a fascinating description of the techniques employed.) The time taken to run **spellin**, and the storage used by the hashed file are obvious costs. Still, the user wants the **spell** command to run quickly, so the tradeoff is worthwhile.

> Response time is a constraint on any number of interesting applications.

Similarly, the **csh rehash** command spends time setting up a hash table for the directories in the user's path. This allows commands to be found quickly, despite the large number of files that might exist in the referenced directories. Perhaps we should extend the principle further.

The **find** command is extraordinarily useful, despite its rather painful syntax. It scans arbitrary portions of the file system for files that satisfy arbitrary criteria, which is very useful, but it's also very time consuming. Wouldn't it be nice to have a really fast **find** command?

Much of the data used by **find** is relatively static. Most files simply sit in the file system, not changing in the slightest over time. One could easily ask a daemon to build hash tables for interesting file characteristics. The daemon would, of course, have to be very obsequious, releasing control of the processor whenever a user desired it. This would leave the daemon with a great deal of usable time, since computer systems are typically idle a great deal. The daemon would also have to do its work in an incremental manner to allow for the continual changes that would take place.

The files developed by the daemon could then be used by a smarter **find** command, and searches could be performed much faster. The principle can be extended even further, however.

OFFICE AUTOMATION

A typical office generates and stores large numbers of letters and other documents. It is often necessary, or at least highly desir-

SERIX

ANNOUNCING SYSTEM V UNIX™

...puts your IBM Series/1® ahead of the pack!

SERIX is the high performance CMI version of AT&T's UNIX™ System V operating system with Berkeley 4.1 enhancements ported to the IBM Series/1 minicomputer.

SERIX transforms your Series/1 into an even more powerful, flexible, and convenient processor for general data processing, office automation, communications, and process control. Its advantages are outstanding:

Reduced software costs

Long term growth path
- Software is highly portable
- Provides access to a large, growing software base

More power from the Series/1
- Optimizing C compiler uses native code features
- All code reentrant
- Dynamic memory allocation without fixed partitions

Increased programmer productivity
- Large set of utilities
- Hierarchical file structure
- Pipes, forks, semaphores, and shared data segments

Other CMI Series/1 software
- RM/COBOL™
- UNIFY™ database management system
- ViewComp™ spreadsheet
- vi visual editor
- EDX™- to -SERIX™ conversion kit

CMI Corporation is a Master Value-added Remarketer of IBM Series/1 equipment. Leasing and other financial arrangements are available.
Contact us for further information.

Photographer - Michael Zagaris • UNIX is a trademark of Bell Laboratories • SERIX is a trademark of CMI Corporation • SERIX was developed exclusively for CMI by COSI. • IBM, Series/1, and EDX are trademarks of International Business Machines Corporation • UNIFY is a trademark of North American Technology, Inc. • RM/COBOL is a trademark of Ryan-McFarland Corporation • ViewComp is a trademark of Unicorp Software, Inc.

CMI

A Torchmark Company

CMI Corporation
SERIX Marketing
2600 Telegraph
Bloomfield Hills, MI 48303-2026
(313) 456-0000
TWX: 810-232-1667
Telex: 499-4100 ANS: CMI CORP. BDHS

Member CDLA Member ASCD

Circle No. 242 on Inquiry Card

able, to find the documents that relate to a given topic. A combination of **find** and **grep** can accomplish this goal by performing keyword searches. Unfortunately, the time involved can be prohibitive.

So we put a daemon to work indexing keywords from the entered files. We may wish to avoid indexing junk words like articles and prepositions. Rather than generate a keyword list, we could give the daemon a simple heuristic, such as: "Keep a list of the usage frequencies of every word found, and index only the least common words."

Better heuristics could certainly be found, but this would do for a start. With such indices in hand, users could do keyword searches in trivial amounts of time.

> **Since computers spend a great deal of their time doing precisely nothing, a readily available, cheap resource is at hand.**

One needn't limit this sort of thing to a single machine, of course. Increasing numbers of processors are becoming available on local networks. When a user requests a compute-intensive job, the machines on the network can bid for it, using their capabilities and current load levels as input. This allows the system to spread the processing load over a number of machines, keeping more of them busy while providing users with quicker service. Such a system is currently being used quite effectively in at least one networked UNIX environment.

IN SUMMARY

Constraints make life difficult, so we invent and use tools to overcome as many of them as possible. Remaining limitations can often be disguised to look like other, less objectionable ones. Floating point numbers disguise a lack of range as a lack of absolute precision. Virtual memory makes a lack of real memory look like a reduction in available disk storage and processor speed.

A lack of computing power (or time) is a common problem. If we can trade idle time for faster interactive response time, by precalculating assorted data or by sharing resources between processors, we win. By keeping our eyes open for ways to circumvent and/or disguise such constraints, we can often make our jobs and lives much easier.

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

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

Only Sperry can make the following four statements.

Our PC runs the XENIX™ system, as well as MS-DOS™.

Our 4 new microcomputers run the UNIX system.

Our new minicomputer runs the UNIX system.

Our Series 1100 mainframes run the UNIX system.

All of which means there is a great deal we can do for you.

For instance, our family of computers based on UNIX systems has incredible transportability for all your software.

And being able to accommodate from two to hundreds of users, it's impossible to outgrow our hardware.

Of course, this linking of all your computer systems can add measurably to your productivity.

And a fast way to find out more is to get a copy of our Sperry Information kit. For yours, or to arrange a demonstration at one of our Productivity Centers, call **1-800-547-8362.**

*UNIX is a trademark of AT&T Bell Laboratories
XENIX and MS-DOS are trademarks of Microsoft Corporation

SPERRY

Introducing an idea that makes obsolescence obsolete.

The UNIX* operating system from PC to mainframe.


MAÏLESSI

Escaping OFFICE DRUDGERY

UNIX office automation is here to stay

by Tanya M. Joyce

Has office automation arrived? Daily newspapers and trade periodicals ask this question regularly. Pictures of briefcase computers flying through the sky, drawings of clerical staffs at workstations extending into deep space, and friendly photographs of happy managers with clear, easy-to-read output accompany the headlines. If the amount of space devoted to the topic is used as an indicator, the conclusion must be that office automation *has* arrived.

Is UNIX an appropriate environment for office automation? A considerable body of evidence suggests that the system's multitasking, multiuser, and interactive capabilities offer a solid foundation on which to build the house of the "electronic office". All these capabilities reach back to the inception of UNIX.

Subsequent enhancements, including connectivity, access to centralized databases, and a growing selection of peripherals have added uprights to the house. Portability, standardization of the operating system, C, and an applications interface have followed. Now there is an emphasis on the development of applications packages and interfaces appropriate to a wide spectrum of users, both in terms of user sophistication and variety of applications.

BEHIND THE BUZZWORD

Before assessing UNIX as an office automation environment, it is first necessary to ask a fundamental question: what *is* "office automation"? The answer is not clear. The acronym, OA, has also been used to stand for "office applications" and "office administration". Why?

Automating anything implies an assembly line approach. There is a desire in some quarters to show that what we call an "automated" office is not in fact a dehumanizing place.

If an office is small, the presence of even one standalone computer may indicate to its proud owner that the office is automated. Concepts of system fragility may not even enter the picture. If the office is large, ease of interaction among a group of computer users in a dispersed system may be a touchstone of OA. Ease of producing sensible, business-oriented output may be another.

And what about the "paperless

Illustration by Heda Majlessi

office"? Experienced computer users tend to agree that this will not become a reality in the foreseeable future. Instead, a shift from repetitive activity, which computers are well suited to perform, to tasks that were considered too difficult or costly before the advent of office computers is likely. The *amount* of paper generated will decline but not disappear. What is *on* the paper will change dramatically. The use of computer graphics to summarize numeric data in visual form is one indication of this trend.

Almost all documentation accompanying business computer systems contains glowing accounts of how easy it is to use the product in question. Whether or not this is actually the case, ease of use seems to be an implicit criterion for good office systems.

The widest use of the term "office automation" seems, then, to suggest the presence of a computing facility that is easy for business professionals to use. "Ease of use" in the business community also implies a variety of features, some of which are frequently discussed, and some of which are implicit and thus best understood by system developers blessed with vision and an understanding of their customers.

There are four major types of tasks performed in an office automation environment: editing by way of a keyboard, rearranging data, "walking around" data structures, and computing. Keyboard editing includes "typing", backspacing, deleting lines, and making room for additional material within an existing document. This is the basis of what we call "word processing", though the emphasis is no longer on the word, but rather on the totality of a document.

Rearranging data means moving files and paragraphs within files, copying parts of files to other files, and so forth. The best systems allow this to be done by placing both the origin and destination in view on the screen without requiring the user to type in alphanumeric commands. Hard function keys and cursor keys are the means by which this is most effectively done.

Walking around data structures means moving through a system, learning where files reside, how to copy and move files effectively, and how to access data about processes in which users might be interested, such as printer queues and mail.

> The potential of graphics applications has yet to be realized.

Computing in an office environment may include the creation of code that can formulate calculations for a timesheet or payroll, or it might refer to shell scripts for tailoring a user's personal environment or devising a text formatting enhancement.

THE PIECES TO THE PUZZLE

The core of an office automation system consists of an editor, a mail system, and communications. The four tasks of office automation already mentioned are centered around uses of an editor. Fortunately, a number of good editors are beginning to appear in UNIX interfaces.

The test of a good editor is the number of people with different needs who can use it comfortably. Technical professionals are now finding work easier to do with many of the newer editors. New users, meanwhile, require less instruction from their colleagues and make more use of computers, reducing paperwork. Macros do not have to be changed or stripped out when, for example, a system programmer sends information to a technical publications department or when a project leader provides reports to an executive. A good OA system, however, does not preclude the use of several editors.

UNIX is a strong system for mail and communications. Both of these features have been part of UNIX for approximately 12 years. New user interfaces for mail functions contribute to UNIX strength in the marketplace. Communications capabilities have not been as easy to bring up to date as mail facilities, largely because the need to use new communications protocols and transport mechanisms has caused problems for companies that do not have a stable of UNIX gurus as part of their system development staffs. But the flexible nature of UNIX itself makes communications problems surmountable when the appropriate knowledge is at hand. The improvements that result are important since the potential for UNIX becoming a standard system is gaining strength.

A good office automation system allows any user to start off with a very small subset of available features and to grow (or not grow, as the user pleases) into areas of deeper knowledge. If perceived response time is fast, and if the system is reliable and secure, and if foolproofing has been built in, a business user may find that employing only a subset of available features makes for the most effective use of time. Additionally, though, a well designed office automation system allows easy access to UNIX itself for users who have

some experience and includes sophisticated applications, especially in areas such as document production and finance, as well as the ability to integrate color graphics with text and numeric data.

This leads to concerns for output. It is the opinion of several people who have longstanding experience with the development of office automation that the advent of inexpensive laser printers is more important to OA than are bitmap display terminals and user interfaces. In terms of reducing the amount of paper in the office and making computer usage functional for a wide variety of business users, this may well be true. Another important feature is the ability to produce appropriate output, which is to say the ability to direct output to whatever device is best suited to a need—line printer, character printer, laser printer, typesetter, or dot matrix printer (with one or more colors). Bitmap display terminals are optimal when output can also be directed to photographic paper, 35mm transparencies, and so forth. A smart OA system will also add another level of appropriateness: it will print in a style appropriate to the type of file specified. For example, a message may be printed with or without headers depending on whether it is printed as a mail message or as a memo.

The potential of graphics applications has yet to be realized. But about four years ago, an enterprising promotional videotape for an office automation system featuring computer graphics dramatically demonstrated their importance. Viewers saw how pages of hard-to-read numeric data could be effectively summarized in a few charts printed with a four-color ribbon.

sufficiently understood in the development of office automation systems. Of course, the picture needs to be good—one that is both easy to create and easy to comprehend.

OFFICE SOCIOLOGY

An overall concern for a good office automation system is that it display consistency of behavior. This means that user behavior should not have to change from application to application. Behavior that applies when creating a message, for instance, should be the same as the behavior that applies within an editor. Consistent behavior extends through several levels—on the screen, on the alphanumeric keyboard, and in the handling of hard and soft function keys. A user should not have to think differently within a variety of applications.

> Office automation cannot be forced on users.

The effects of office automation systems are varied. Office automation cannot be forced on users. In general, however, users tend to find some aspect of work that is made easier by use of a computing system. When first introduced, office automation can create chaos in an office. If fears persist, a system will take a long time to become a viable part of daily work. If an office automation system is already in place, however, new staff members will tend to use it readily.

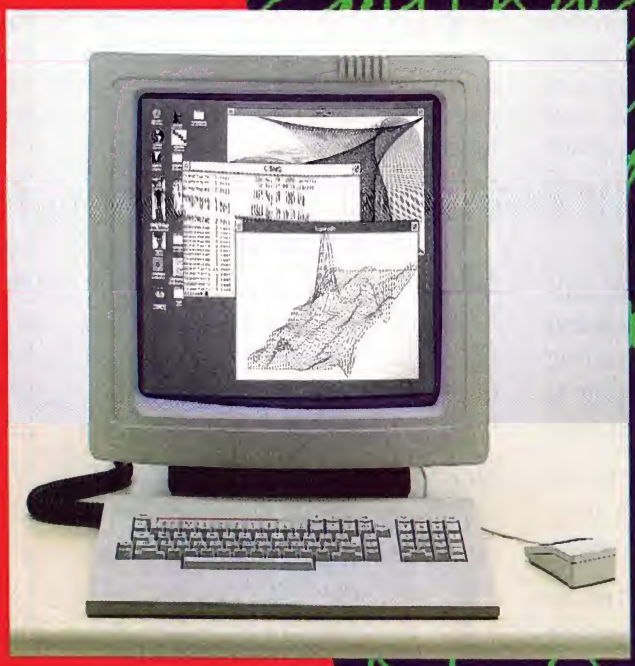
raised expectations of the workers themselves become apparent. New career directions for clerical workers and professionals up through middle management frequently result from access to office automation systems. Local customization of software intrigues many users who already have some office automation experience. Fundamental changes in how people work, moreover, tend to result from using electronic mail instead of memos. Viewpoints change as users come to take increased computing power for granted.

Faster communication also makes a difference sociologically. Electronic bulletin boards and short, well designed help facilities encourage people to learn quickly in a neutral environment. As one person put it, "My secretary can get mad at me, my computer doesn't." Systems that are geographically dispersed often encourage users to extend their abilities with mail and communications features. There tends to be more bandwidth in human contact as people establish for themselves new balances between time spent alone with their systems, whether at home or in the office, and the desire or need to communicate with colleagues face-to-face.

CAVEATS

Are there any problem areas in this rosy picture? There are several. The first one is small and will sort itself out in time. When business professionals rely on computers that are not part of a substantial network and don't come complete with experienced system administration personnel, unexpected downtime or component failure can cause major problems. This potential is a headache for today's small busi-

Optimum



Get more than you bargained for.

A powerful high-performance, VME based graphics workstation that's easy to upgrade to the MC68020.

Upgrades to the new Motorola 68020 are as easy as a single board swap in the Optimum V Series. And when you're ready to add on even more processing power, you have a wide range of sophisticated Integrated Solutions' products to choose from. Peripherals. Packaging alternatives. Performance features. Applications. All at low incremental cost.

High-performance graphics for the technical professional.

The Optimum V offers impressive features. Like a high-resolution (1280 x 1024), bit-mapped graphics display in monochrome or color. 32-bit VME architecture and AMD 29116 graphics display processor. Together with 11.2 MHz 68010, dual ported high-speed memory and UNIX 4.2 BSD, the Optimum V is designed to provide you, the technical professional, with state-of-the-art technical processing.

And Integrated Solutions makes processing even easier with a unique user interface. Multiple windows, icons and pop-up menus mean convenient access to programs, files and directories.

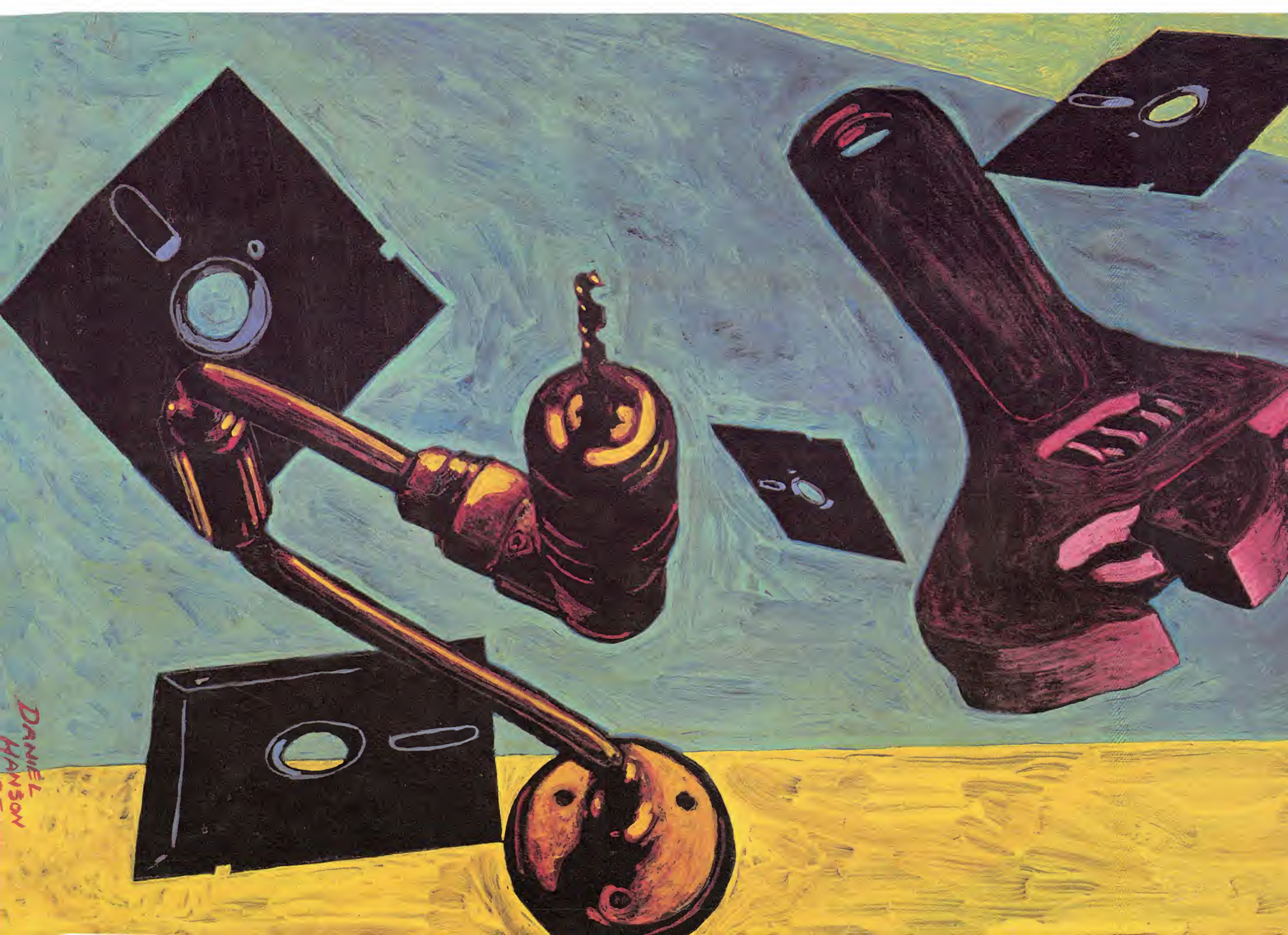
Support and more.

Integrated Solutions is behind you all the way with on-site hardware maintenance and local software support. Call us now for more information.

Integrated Solutions
An NBI Company
2240 Lundy Avenue
San Jose, CA 95131
800-538-8157, ext. 823
In California, 800-672-3470, ext. 823

Circle No. 241 on Inquiry Card

Call Integrated Solutions.


DANIEL
HANSON

TOOLS FOR AUTOMATION

A sudden proliferation of UNIX-based computers in office environments has been attended by the development of a wide variety of "user-friendly" software packages, capable of doing everything from postage calculations to checkbook balancing. Moreover, menu-driven shells shield users from direct contact with the Bourne shell or the Berkeley C shell.

To a large extent, this has been good. The software explosion has opened the world of UNIX-based systems to people who merely wish to *use* them, not necessarily *program* them. However, it has also created the perception that office/clerical users should do *no* programming whatsoever—even if doing so could make their lives easier. The expression "user friendly" far too often translates as: "The users are too stupid to know what they are doing, so we dare not let them near the operating system, much less the compiler."

Computer marketing personnel often forget that users are expected by their employers to be capable of handling the complexities of *manual* office procedures! To keep their jobs, they cannot be *too* dumb about what they are doing. However, this little detail is often glossed over, and users are treated as a lesser species.

To be sure, the vast majority of the office user's computing time is spent in word processors (or editors), spreadsheets, and electronic mail systems. Because these functions are largely environment-independent, many users have come to view various popular features as indispensable, and vendors thus have felt market pressure to offer elaborate features at competitive prices. No one would argue that users should ignore this by writing their own versions of "staple" applications. There are gaps between these applications, however.

An introduction to do-it-yourself office software

by Rob Warnock

Illustration by Daniel Hanson

PROGRAMMING IN THE OFFICE

The time spent working with a word processor, spreadsheet, or E-mail is one thing. But tasks that are specific to the type of office and to the individual's own work are quite another. The more specialized the task, the less likely it is that general-purpose software will be of any assistance.

To contend with this problem, several software packages sometimes are purchased to tackle a single task, with bits or pieces from one or another being applied to the task at hand. This expensive approach seldom succeeds, though, since products from various vendors rarely mesh well.

Rather than trying to use small pieces of general-purpose tools, it is sometimes better to create a small, special-purpose tool that does exactly what is required and no more—like walking a half-block through the park instead of taking a bus, a subway, and another bus to get around!

However naive a user might be at the beginning, and no matter how "user friendly" the system might claim to be, after a few months (sometimes *days*), the user will have usually dealt with at least a few "naked" shell commands.

While UNIX *potentially* is "user friendly", systems as delivered rarely are, except to programmers. This shouldn't be surprising since it's the small touches peculiar to each office—or individual—that serve to make a system truly "friendly". A mass-marketed system simply cannot accommodate each and every user—at least, not without some modification. A small amount of customization can make a large difference in ease-of-use and job satisfaction.

A LITTLE HELP FROM YOUR FRIENDS

Fortunately, clerical users in the UNIX environment often *can* create their own small tools, particularly ones designed to simplify common, repetitive tasks. The major barrier is lack of confidence.

Confidence comes from successful experience, but how does a typical office user get that initially? Basically, one learns simple programming in much the same way that any other office skill is mastered—from watching others in the same office at work on similar problems, and studying the solutions they devise.

A useful notion, borrowed from "copy center" terminology, is that of a *key operator*, or in this case, a *key user*. In each office, there are generally one or two users who are more inquisitive and adventuresome than the rest, and thus usually end up helping others with details of operation and troubleshooting. These people can be cultivated by an office system administrator or resident "UNIX wizard", if any, or by a salesman or installer (in the event of a more "turnkey" installation). Key users thus faciliate the sharing of "hints and kinks" and various time-saving techniques. In this way the "training" of programming happens naturally.

KEEP IT SIMPLE

Users who might do "office programming" should be exposed to the important concept of *abstraction*—that is, the substitution of a single task or command for a sequence of tasks. When executed, the single command will carry out the original sequence of tasks. In "office programming", most—but not necessarily all—locally-written commands will be shell scripts.

Simplification comes in two ways: first, the sheer volume of interaction (both typing and waiting) is reduced because only a single new command name needs to be typed; and second, the entire operation can be thought of as a single unit, allowing the user to ignore the execution details (once the command works properly, that is). Like a head cook overseeing a banquet, the user can concentrate on whole recipes (commands) instead of individual ingredients.

Another important concept is *parameterization*, which allows a single command to perform many variations of a task, as determined by the *parameters* or *arguments* provided at the time the command is executed.

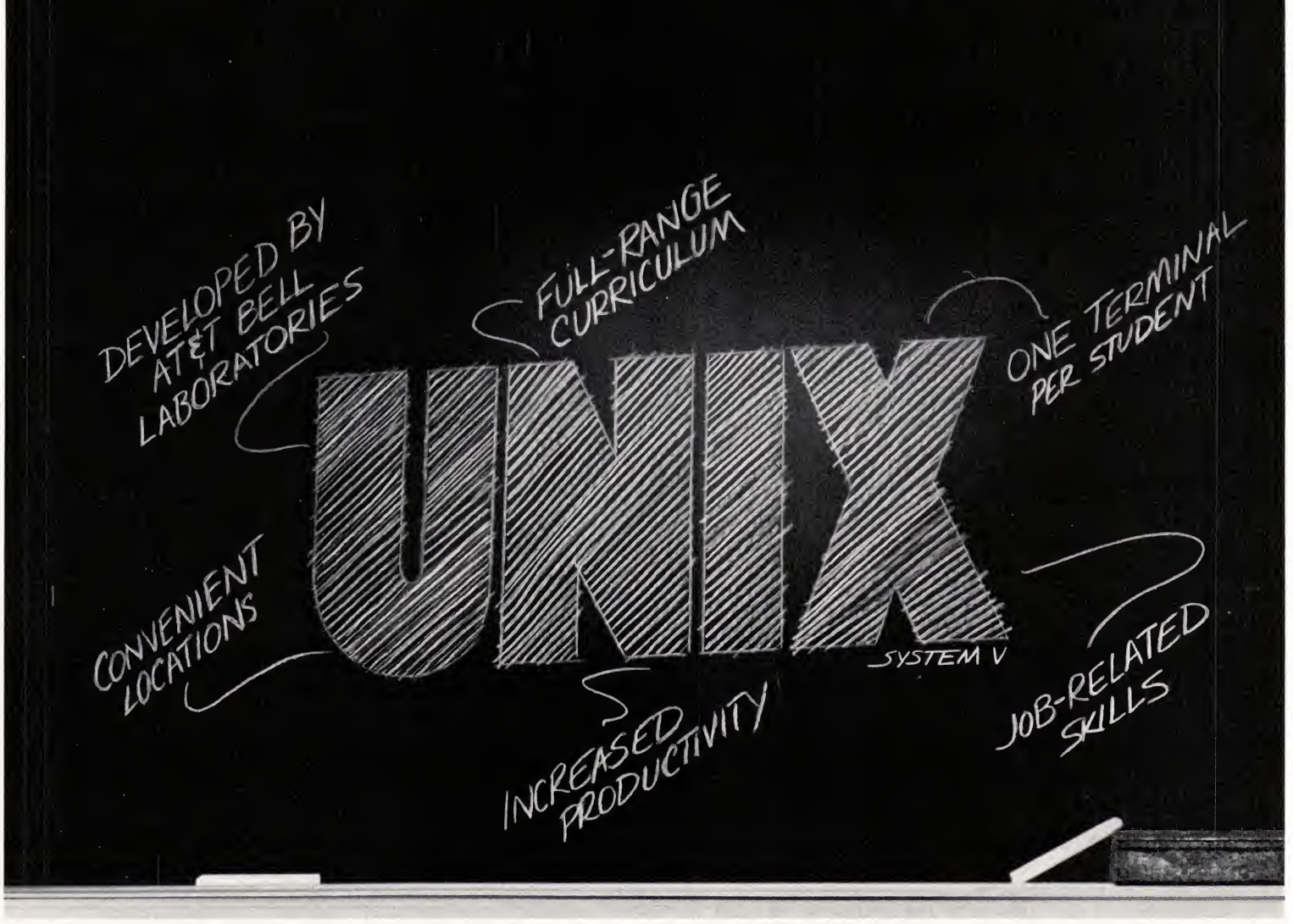
These two ideas—abstraction and parameterization—are the heart of modern programming discipline, but their use in the office must be taught by way of concrete example, rather than as lofty textbook theory.

SHOWING OFF

One of the greatest virtues of the UNIX operating system is the ease with which such examples can be created by using a shell. One simply creates a textfile for the command and stirs in the individual steps that must be executed (which are either commands that were provided with the system or are commands that have been defined locally). The textfile is then flagged as being executable (by means of the **chmod** program). That's all there is to it.

To illustrate some of these points, we'll look at a "tickler" file for reminders, a program for handling form letters, and a command to help write other commands.

In the interest of convenient presentation, let's assume that each user in these examples has a *bin* subdirectory in his or her home directory for private commands, that */usr/local/bin* is where system-wide local commands will be stored, and that *$HOME/bin* and */usr/local/bin* are the first two



HANDS-ON TRAINING THAT ISN'T SECONDHAND

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

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

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

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

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

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

So take your training from AT&T. And discover the power of UNIX System V—right from the source. **Call us today to reserve your seat or for a free catalog.**
1-800-221-1647, Ext. 53.

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

Name _____

Title _____

Company _____

Address _____

City _____ State _____ Zip _____

Call 1-800-221-1647, Ext. 53
or send coupon to:
AT&T Information Systems
P.O. Box 45038, Jacksonville, FL 32232-9974

AT&T
The right choice.

directories named in the **PATH** environment variable. Most of the utility commands used here are taken from Berkeley's 4.1BSD version of UNIX, especially the **echo -n ...** command for echoing without a newline. (If your system runs UNIX System V, use **echo "...\c"**.) Scripts are written using the Bourne shell (**sh**), rather than Berkeley **csh**. Also for simplicity of presentation, many error-checking and little "convenience" features have been omitted. Nevertheless, these tools should be useful when adapted to local environments.

TICKLED PINK

Suppose you want to maintain a "tickler" file of things you need to remember to do. The tickler file itself could simply be a textfile named *tickler* in your home directory. Let's assume it is. To see what's in the *tickler* file, you could type:

```
$ cat tickler
```

and the contents would be displayed. If your current directory wasn't the same as your home directory, you'd have to enter:

```
$ cat $HOME/tickler
```

which could get tiresome, especially if you did it often. To solve that, we can produce a command named **todo** to show the contents of *tickler*:

```
: 'todo - show contents of ticker file'
exec cat $HOME/tickler
```

```
: 'rmtick - remove items from tickler file'
# "rmtick <pattern>" will let you remove all items matching <pattern>
tmp="$HOME/.untick$$"
case "$1" in
"")     echo "usage: rmtick <pattern>"
        exit 1 ;;
esac
echo 'Items selected:'
grep "$@" $HOME/tickler             # see what matches
echo -n 'Delete these (yes)? ' ; read ans
case $ans in
".......q";"y";"ye";"yes";"")
        : o.k. ;;
*)      echo '"tickler" file not changed'
        exit
        ;;
esac
grep -v "$@" $HOME/tickler >$tmp # save everything EXCEPT matched lines
mv $tmp $HOME/tickler               # move the saved items back
```

Figure 1 — *The **rmtick** command for removing completed items from the "tickler" file.*

Each time you wanted to add something to the file, you could use your favorite editor program to edit *tickler*, but that might be slow. Some editors take a while to start up, and you'd have to type the full filename, the editing commands, and all the rest of it. If the items tend to be "one-liners", the command **tick** is probably more convenient:

```
: 'tick - add an item to the tickler file'
exec echo "$*" >>$HOME/tickler
```

So, to remind yourself of a doctor's appointment, you could just enter:

```
$ tick Appt. with Dr. Johnson Wednesday
$
$ (... other work ...)
$ todo
Pick up cleaning
Letter to V/P sales
Appt. with Dr. Johnson Wednesday
$
```

Like **todo**, **tick** should be placed in *$HOME/bin*, or if everyone uses it, in */usr/local/bin*.

What about removing items you've already completed? Again, a general-purpose text editor such as **ed** or **vi** could be used (and probably would be best if you were doing a massive cleanup or reorganization), but the **rmtick** command listed in Figure 1 can handle most common cases fairly well.

If you tend to have both dated and undated tickler items, you can leave the undated items in a file such as *tickler*, and put the dated items in a directory called, say, *tickdir*, with filenames of the form *yymmdd.title*. That is, if you have a seminar scheduled for June 22, you could make a file named *$HOME/tickdir/850622.seminar* to contain all the particulars of the seminar.

This style tends to be useful if your dated reminders originate as mail messages or as *netnews* items. From your mail or news reading program, you easily could save the message in *tickdir* with an appropriate name.

The following version of **todo** has been enhanced to show both kinds of items:

```
: 'todo - show contents of tickler file'
cat $HOME/tickler
echo "Today = `date6`" # Show "today" for reference
ls $HOME/tickdir
```

Other extensions are possible, depending on taste. The tickler entries could have priorities on them (with an initial letter, perhaps), and the **todo** command could sort them as they're displayed. (A reverse sort seems to work best. That way the high-priority items are left on the bottom of the screen if the top scrolls off.)

Stick this in your RAM.

All this common desktop clutter hides neatly away in your UNIX system's host computer. R Desk makes office communications far more efficient AND makes every user better organized, more productive.

With a keystroke, you summon your electronic desk calendar, your appointment book and your personal "to do" list. You can even set up meetings with others, automatically.

Or call up your Rolodex™—with as many as 8,000 names and addresses per user. Find the data you need and return at once to your program, or print out mailing labels!

There's a calculator built in. Plus an automatic telephone dialer that works with your modem. Even an electronic message pad, so phone calls are noted on terminals—instead of little pink slips of paper.

User IDs keep personal files private, while company lists (vendors, customers, etc.) can be open for everyone.

This powerful organizer runs on Motorola 68000 or Intel 80286 based systems. It comes on a standard 5¼" diskette. It's fully menu driven—but we'll include a how-to manual for diehards who insist on documentation.

Guaranteed to make you more productive in 30 days, or return for full refund!

R Desk™
BY R SYSTEMS, INC.

Circle No. 248 on Inquiry Card

Only $295!
Call toll free, (800) 527-7610. In Texas, call collect (214) 343-9188. Or send this coupon with payment today to:

R SYSTEMS INC.
11450 Pagemill Road, Dallas, Texas 75243

NAME _____

COMPANY _____

ADDRESS _____

CITY _____

STATE _____ ZIP _____

TELEPHONE _____

Send check or money order for $295. (Texas residents add sales tax.) Or charge to
___ VISA or ___ MasterCard

CARD # _____

EXP. DATE _____

SIGNATURE _____

Rolodex is a registered trademark of the Rolodex Corporation. UR785

ABOUT THOSE FORM LETTERS...

Many companies have a preferred inter-office memo style, so our next example illustrates one way this can be accommodated without a lot of typing. What's needed is a way to write a memo, store it in a meaningful place, make it look "right", and print it. We can do this with a **memo** command, which creates a template for a memo, which is then edited. The memo is printed using **nroff** or **troff**. See Figure 2 for an example. This is more complicated than the previous examples, but is a typical template-filling script. (If you use *netnews*, you may wish to look at the script in */usr/bin/postnews*, which performs a similar function in the creation of news items.)

```
'O: *** replace with real name of $who ***
'ROM: Rob Warnock
)ATE: 'date'
;UBJ: *** replace with full subject of $subj ***
'ILE: /u/rpw3/memo/$who/$d6.$subj

** replace with text ***

c: The Boss
   Distribution

ZZ fi exec ${EDITOR-vi} $who/$d6.$subj
 'memo - compose a memo and file it'
 "memo [ who [ subject ]]" creates/edits $HOME/memo/who/YYMMDD.subject,
 where YYMMDD is the output of the "date6" program, e.g. "850622.testing".

d $HOME/memo    # Move to the memo area
6='date6'       # Fetch YYMMDD
ase "$1" in     # See if "who" specified
")      echo -n 'To: ' # No, ask for it
        read who
        ::
)       who=$1
        shift
        ::
sac
ase "$1" in  # See if "subject" given
")      echo -n 'Subj: '
        read subj
        ::
)       subj=$1
        shift
        ::
sac
f [ ! -d $who ]         # Open a folder, if needed
hen     echo 'Making directory' $HOME/memo/$who
        mkdir $who
i
 If this is the first time we are editing this memo,
 create it with all the fields ready to be filled in.
f [ ! -s $who/$d6.$subj ]
hen     cat )$who/$d6.$subj ((ZZZ

                        $d6.$subj
```

gure 2 — *One possible command for creating a memo mplate.*

Entering the command **memo joe testing** will place you in your editor (default **vi**), with the following on your screen:

```
TO:   *** replace with real name of joe ***
FROM: Rob Warnock
DATE: 8 Tue Jun 4 19:36:57 PDT 1985
SUBJ: *** replace with real subject: testing ***
FILE: /u/rpw3/memo/joe/850604.testing

*** replace with text ***
    cc: The Boss
        Distribution
```

After editing, it might look like this:

```
TO:   Joseph Personage
FROM: Rob Warnock
DATE: Tue Jun 4 19:36:57 PDT 1985
SUBJ: Testing the ''memo'' facility
FILE: /u/rpw3/memo/joe/850604.testing

This is a test of the new ''memo'' command I copied
out of that UNIX Review magazine the other day.
What do you think?

cc: The Boss
    Distribution
```

And finally, after being sent through **nroff -ms**, we get:

```
TO:   Joseph Personage
FROM: Rob Warnock
DATE: Tue Jun 4 19:36:57 PDT 1985
SUBJ: Testing the ''memo'' facility
FILE: /u/rpw3/memo/joe/850604.testing

This is a test of the new ''memo'' command I copied out of that
UNIX Review magazine the other day. What do you think?

cc: The Boss
    Distribution
                        850604.testing
```

Instead of entering the **nroff** command directly, you could create a **pmemo** command (not shown), which is similar to **memo** in terms of computing the filename from the person and subject designated in the command line, but which offers a choice of formatting: you can choose to **nroff** your memo to the print spooler, or pipe it through **ul**, **cat -s**, and **more**.

The same approach can be used to generate templates for call reports, expense reports, monthly activity reports, or any other report that comes to mind.

INSTANT COMMANDS

If we're going to be making commands all the time, we need a command that helps us write them, called **mkcmd**:

```
: 'mkcmd - create/edit a shell script'
file="$1"
case "$file" in
"") echo 'usage: "mkcmd file" or "mkcmd path/file"'
      exit
      ::
esac
case X$file in
X*/*)     : o.k.          # Does the file name have a "/" ?
      ::
*)    file=$HOME/bin/$file # No, default to private copy
      ::
esac
${EDITOR-vi} $file  # Edit the command
exec chmod a+x $file      # Make the file executable
```

If the argument to **mkcmd** has a "/" in it, **mkcmd** uses the argument as given. If not, it prepends the user's private *bin* directory to the name. It then calls the editor to edit the file, and runs **chmod** to make it executable (so that the shell will regard it as a command and not just a textfile).

BRINGING IN A COUPLE OF KEYS

On many terminals, there are function keys that emit a sequence of characters when depressed. Many of these have the useful property of emitting sequences that just happen to correspond to legal UNIX filenames, plus a carriage-return character. On both Televideo and Fortune Systems terminals, for example, the function keys emit the sequence: <*CTRL-A*>, <*letter*>, <*Return*>. Other terminals emit similar sequences, or can be programmed to.

Since, in the case above, <*CTRL-A*> is an acceptable UNIX filename character, as are letters, you can view the sequence as being of the form: <*name*> <*Return*>. Consider what happens if you type, say, **mkcmd** <**f3**> (where <*f3*> is function key #3). The **mkcmd** command will see an argument (filename) of <*CTRL-A*><"*c*">, and you will edit a file of that name. When you exit, the file will be made executable, and from then on, it can be executed by simply striking function key #3.

You can completely customize your UNIX environment by creating one such function key command for each common procedure you perform. For instance, <**f1**> might clear the screen, <**f2**> might read your mail, <**f3**> might print any messages saved for printing, and <**f4**> might log you out.

On systems that permit the mounting of floppy disks as UNIX filesystems, such single-key scripts

```
: 'date6 - shell version'
# "date6" prints today's date as 6 numbers: "yymmdd"
# It is useful in constructing filenames inside shell scripts,
# by using the "`date6`" construct.
# Bugs: It will break in the year 2000.
set `date`
month=$2
day=$3
year=$6
case $day in
[0-9]) dd=0$day ;;    # Ensure two-digit days
*)     dd=$day ;;
esac
case $month in
Jan)    mm=01 ;;
Feb)    mm=02 ;;
Mar)    mm=03 ;;
Apr)    mm=04 ;;
May)    mm=05 ;;
Jun)    mm=06 ;;
Jul)    mm=07 ;;
Aug)    mm=08 ;;
Sep)    mm=09 ;;
Oct)    mm=10 ;;
Nov)    mm=11 ;;
Dec)    mm=12 ;;
esac
yy=`echo $year | tail +3c`
echo -n $yy$mm$dd
```

Figure 3 — *The script version of the* ***date6*** *command.*

```
/* date6 - C version
 * "date6" prints today's date as 6 numbers: "yymmdd"
 * It is useful in constructing filenames inside shell scripts,
 * by using the "`date6`" construct.
 * Bugs: Years 2000 and later will appear smaller than the 1900s.
 */
#include <sys/types.h>
#include <time.h>

extern struct tm *localtime();

main(){
        time_t now;
        struct tm *t;

        time(&now);
        t = localtime(&now);
        printf("%02d%02d%02d", t->tm_year % 100, t->tm_mon + 1, t->tm_mday);
        exit(0);
}
```

Figure 4 — *The C version of the* ***date6*** *command.*

can make painless the evoking of the various systems utilities needed for the mount operation. One key could mount the disk, check it for consistency, and give a directory listing. Another key could **sync** (if needed), list the directory, dismount the disk, and signal the user when it is safe to remove the floppy.

Combining **mkcmd** with such function keys permits even novice UNIX users to create labor-saving procedures for themselves. It is important that the first few encounters with "programming" be as simple as possible so that confidence can be instilled in users. This confidence can then expand to make the computer a truly valuable personal office automation tool (even for managers).

CONCLUSION

"Working with the computer" is often cited as a major source of frustration and dissatisfaction by clerical users. It is quite amazing, particularly to "real programmers", just how much difference the ability to perform small "hacks" can make in the attitude of users. Rather than feeling completely powerless, users who can develop these simple scripts learn that they can actually tailor the system to suit their individual work patterns. For them, the computer becomes a tool rather than a burden.

APPENDIX

Listed in Figures 3 and 4 are two versions of the **date6** program that was used several times during the course of the article. The first is a shell script; the second a C program. Since the shell script involves the execution of about four times as many processes as the C program, the latter is considerably faster. The shell script was used for about a year before any effort was expended to write the C program. This is as it should be. Shell scripts are very effective for trying out procedures, and—like the bus/subway/bus ride mentioned above—do manage to get you where you want to go. Still, after sufficient use has revealed that there are performance problems, you may consider re-writing some of your scripts in C.

Rob Warnock is an independent computer architecture consultant with nearly 20 years of experience in data communications hardware and real-time and time-sharing operating systems. He has worked for Digital Communications Associates, Xerox XTEN, and was the system architect responsible for the Fortune Systems 32:16. Mr. Warnock currently works and resides in Foster City, CA. ■

YOU CHOOSE:

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

MLINK™

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

Choose the best. Choose MLINK.

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

Please send me the following version(s) of MLINK:

_____ Altos 486/586/986 Xenix ($250)
_____ AT&T 3B2 Unix System V ($500)
_____ DEC PDP-11 SCO Xenix ($500)
_____ IBM PC/XT/AT or compatible ($250)
_____ Send me more information

MLINK is available for many systems.
Please call for current version information.

Introductory offer - we pay shipping!
Outside North America add $15.00 (U.S.)

___ Check enclosed ___ MasterCard ___ Visa
___ COD (Registered checks or money orders only)

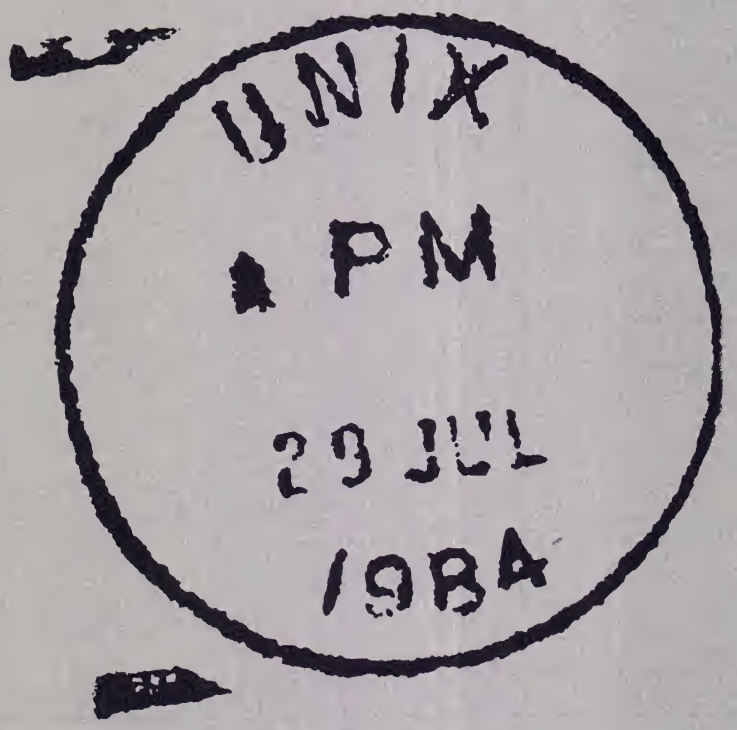
Card # _____ Exp. _____

Signature _____

Ship to:
Name _____

Address _____

City/State/Zip _____



MAIL IN THE ELECTRONIC AGE

The way it should be

by Bruce Borden

Electronic mail is not always well received, but when it is, the results can be amazing. Usually, the jumps in office efficiency are nothing short of miraculous, as traditional office games like paper memo shuffle, filing cabinet jam, and telephone tag come to be played less and less. There are, of course, other cases where office workers cannot be convinced of the virtues of electronic mail, forcing companies to either abandon their systems or relegate them to the engineering department, "where they belong".

History offers a very important lesson regarding the introduction of electronic mail. When the ARPANET was first installed in the early 1970s, it was designed for "resource sharing", not electronic mail transfer. The expectation was that the system would support distributed computing and data sharing. Hooks for electronic mail were added to the file transfer program/protocol only as an afterthought. Imagine everyone's surprise when mail traffic became the predominant use of the network. Many explanations were issued, ranging from comments on the quality of the system's tools to a lack of user knowledge about the capabilities of the network. But even now, 12 years later, electronic mail still represents a very large portion of the ARPANET's use.

To see why that's so, let's look at the following questions: Why use electronic mail? In what ways is it better than what it replaces? What distinguishes a *good* electronic mail system from a poor one? Why have some electronic mail systems failed?

THE WHYS AND WHEREFORES

The characteristics of electronic mail that best explain its usefulness include:

Distribution. A message may be delivered to more than one person—within a building or city, or between countries and, occasionally, companies. A question may be asked of one person or every member of a group simultaneously by using directed or broadcast queries, obviating the need to seek out replies on an individual basis.

Timeliness. An electronic message can be delivered quickly, available to recipients at their convenience. This is particularly useful when messages need to be sent between different time zones. An electronic message won't interrupt the recipient unless the recipient wishes it to.

Traceability and Ability to Reference. A phone call leaves no (readily accessible) traces, whereas electronic mail can be filed and forwarded. What's more, past messages can be reviewed, thus minimizing problems of redundancy, misinformation, and misdirection ("but I never said ...").

Thoughtfulness. Mail composition is not driven by the clock, as a phone conversation is. As a result, the sender can take time to frame a suitable message or response to a message, allowing information to be gathered offline rather than during the time one is online.

Environmental. There are no paper blizzards caused by electronic mail. An electronic message can be discarded cleanly—assuming the computer needs to be run anyway (perhaps a dubious assumption).

THE BAD OLD DAYS

Electronic mail primarily reduces phone usage and memorandum/snail-mail usage. It can also reduce hallway conversations, the time is takes to schedule meetings, and the number of meetings that have to take place.

The trimming of phone use is particularly important. Think of how often you've played telephone tag. I call you, you're out, so I leave a message. You return my call, I'm out, so you leave a message. I call you again, you're out, so I leave another message. You get the picture, no?

Happily, electronic mail does not suffer from this problem. Most implementations, though, suffer from another, perhaps more serious problem: the lack of a delivery notification system. Let's say I send a message, and I want to know if the intended recipient has received it. If I don't get a reply after a while, I might send another message (just to ensure the first didn't get lost), or I might place a telephone call. Both are equally wasteful of my time. Of course, there's another option: I might simply assume delivery, only to learn to my surprise later that the recipient was out of town and never saw the mail. There are very few electronic mail systems that solve this problem.

Developers would do well to

Illustration by J. Kell Davies

think of electronic mail as a cross between telephone conversations and memoranda exchange. Systems need to be as easy to use as telephones and provide output that's easier to file and refer to than memoranda. A good electronic mail system will be easier to use than a telephone for some things, but harder for others. Those systems that aren't much easier—and aren't a good deal more powerful—than memoranda are useless.

To make a phone call, you pick up the receiver, dial a number (or press a few buttons for a preprogrammed speed call), wait for an answer, ask for an extension, wait for an answer, and talk (full duplex). To send an electronic message, you call up the mail composition program, fill in a "To:" field (perhaps with a preprogrammed abbreviation), type the message (or tell it to your system by way of an integrated voice/data system), send the message, and wait for a reply (half duplex). When information is needed immediately (within a matter of minutes), a phone call or visit is the best means. When the information is highly interactive, and cannot be spread over days or weeks, voice contact again is called for. For all other communication—and by this I mean the lion's share of communications—electronic mail is the best solution.

To be effective, electronic mail must co-exist with the terminal/workstation human interface in much the same way the telephone does. That is, if a message arrives, users need to be able to choose the type and "volume" of notification (will the machine print a message, ring a bell, or change the color of an icon?). Users also need a way of dropping into the mail environment without affecting work in progress. While in the mail environment, they need access to filed mail messages, current messages, and preferably their current working environment. This allows them to easily get back to other work after reading the newly received mail, replying to it, filing it, or deleting it.

THE DREAM SYSTEM

How is a good electronic mail system distinguished from an average one—or a poor one? Take a look at the features.

User-selectable prompts. Users should be able to select the message header fields they want to be prompted for, as well as the ones they want to have filled in automatically. Some people, for instance, may resent being forced to specify a subject; others may have fields of their own creation they'd like to use.

Electronic mail causes no paper blizzards.

Header field addition. A mechanism that allows users to add header fields during message composition must be provided.

Independent "From", "Sender", and "Reply-to" fields. "From" specifies who the message is from, "Sender" specifies the person who actually sent the message, and "Reply-to" specifies a proper reply address, which may relate to either of the "From" or "Sender" addresses—or to a different address altogether.

Automatic fields. Some header fields should be provided automatically, such as the date of the message, the login ID of the sender (for authentication), and a unique ID (see *Conversations* below).

Message priorities. Properly used, these priorities should affect the disposition of the message at the receiver's end. A high priority message, for instance, might ring the bell of the receiver's terminal, while a low priority message might simply be placed in the receiver's mailbox without any immediate measures being taken to alert the recipient.

Delivery notification requests. When senders request delivery notification, the delivery receipts should be mailed to them at the time the original mail is actually added to each receiver's mailbox. Under a truly clever mail system, each user should have the ability to disable delivery reply messages from their own mailbox (perhaps on an individual sender basis). The reply receipts should either appear as normal mail to the sender, or be gathered together for review by the sender on demand.

Receipt requests. When senders file receipt requests, receipts should be automatically sent back whenever the original mail is actually read by each receiver. Again, there should be various options available on the receiver's end. Receivers should be able to turn off receipts, or have the mail receiving program query them whenever senders request a receipt. Another option would be to have the mail receiving program generate receipts as requested, but notify the receiver whenever they are generated. The receipts that are ultimately sent back to the sender should either appear as normal mail, or be gathered together for review by the sender on request.

Effectiveness. Senders should be able to specify the time within which a message should be read. Let's say I want to schedule a meeting for this Friday. I might specify that the message should be returned to me if it hasn't been

read by Friday morning (a copy might be left in the receiver's mailbox with an annotation indicating that it had been returned to sender). Let's further suppose that one of the receivers is going to be out of town for a week. The mail delivery system should be able to immediately "return" the message I sent to that person, since it will be out of date by the time he or she returns.

Conversations. This is complex. The basic idea is for a person to be able to initiate a conversation that can be referenced automatically. Let's say I send a message asking a yes/no question, and the reply comes back "yes" two weeks later. By then I likely will have forgotten what my original question was, so I will want the mail system to find it for me. This is usually done by identifying each message with a unique ID, and including the ID in the reply message.

Another situation that should be provided for is the multiuser conversation that has multiple branches. For example, I ask Joe and Bob what they think of an idea of mine. They each answer, and I continue dialogues with each of them individually. But, sometime later, I would like to be able to review the two conversations together as a sequential stream. You can imagine the complexity when there are as many as 20 or 30 people involved. It would be a wonderful thing if each participant in a conversation could choose between receiving only messages addressed to him or her, or receiving all the messages in the conversation.

Preview delivery. Managers should be able to have their mail screened by a secretary, just as they can with paper mail or memoranda. Messages marked confidential, or ones that come from some select list of individuals, should be delivered directly. Messages forwarded from the secretary, meanwhile, should be so annotated, and secretaries should be able to turn off this feature whenever they know they're going to be unavailable to process the mail.

Message reply. Mail packages should format reply messages on request, addressing them either to the sender only (or whoever is specified in the "Reply-to" address), or addressing them to all recipients. The reply should include a subject with the characters "Re:" prepended to the origi-

Continued to Page 88

Photos by Randy Becker

OF MICE AND MEN

An interview with Larry Tesler

The worst problems occur when people walk into their office and—without warning—find technology they never asked for.

There aren't many veterans of the computer industry who have tracked the history, promises, and prospects of office automation as closely as Apple Computer's Larry Tesler.

Following graduation from Stanford University in 1965 and a brief foray into the fledgling field of artificial intelligence, Tesler joined Xerox PARC in 1973. It was there that he encountered some of the earliest versions of "office augmentation" products. Arguing that OA systems needed friendlier interfaces in order to grow, Tesler helped pioneer a series of changes designed to make the machines look "more like familiar objects" to their users. As part of this effort, he studied computer training and employee transition problems in the workplace.

Tesler joined the Lisa development effort at Apple in 1980, and then moved to the Macintosh Group in 1984 to serve as Manager of Future Architectures. That's where Dick Davies quizzed him for UNIX REVIEW on the trends he sees in office automation and the ways in which he feels the UNIX operating system plays a role.

REVIEW: *How did you first become involved with office automation work?*

TESLER: I was in on the beginnings of what's now called "office automation" at Xerox PARC in the early '70s, but let me say that I don't really like the term "office automation" because it conveys the image of an office that runs itself automatically, like an automated factory. The image is one of people running around twisting knobs that help the office work. Offices will always be run by people, at least for the foreseeable future, so what we're actually automating is some of the busywork that people do.

REVIEW: *What surprises have there been?*

TESLER: Actually the main surprise for the people who originated the notion, like Doug Engelbart, has been that it has taken so long. He expected the office automation push to be something that would have already been achieved by now. Another surprise has been that, unlike the image people in the '50s had of a giant computer with a million terminals, we now have a lot of very cheap workstations that can be networked together. That has had an impact on how people view office automation. Still another surprise has stemmed from the expectation the original developers had that office automation would be such an important tool for people that they would be willing to spend a lot of time training, just like they spend a lot of time learning to type or learning to drive or learning to take shorthand. We haven't seen that, though.

I got involved in all this in the early 1970s—at just about the time it was becoming clear that in order to get people to accept this kind of technology in their office, it would have to be something that could be quickly learned. So, my focus from the start was to develop office automation tools that could be learned in minutes, hours, or days, instead of over the course of weeks and months.

REVIEW: *Like a good word processing system?*

TESLER: Right. We started by looking at the NLS system Engelbart's group had developed at SRI in the '60s.

REVIEW: *That early?*

TESLER: Yes, they invented the mouse in 1964 or '65. But it was a pretty clunky mouse, since NLS had a page-sized display. Some versions of NLS had a little bit of line-drawing graphics. But most had no graphics or graphics that

were very crude. One version, though, did have the ability to superimpose video on the screen along with the text, which was a very powerful capability that's only now beginning to be reintroduced. In many ways, there were features of that early system that are still not available to most people. The NLS system is now marketed by Tymshare as *Augment*, but it's gained little acceptance despite its power because it's relatively hard to learn.

For all that, the point is that even at that early date those developers were able to produce a full-page screen and a capability that's now called "outline processing", as we find in *ThinkTank* and *Framework*. They also had something which is called "Hypertext" now. That's a capability that you'll see coming out on personal computers very soon. Hypertext systems allow you to look at a document and see a cross-reference citation to another document. You can then point your mouse to something and say, "I want to see that," and the system will bring that document up onto the screen.

NLS was quite an amazing system. There were two reasons it wasn't accepted. The one that everybody understood was that the technology was very expensive. It cost $100,000 for one of these machines, but the price of it was expected to drop. The founders of PARC extrapolated the price of hardware and realized that in the '80s the hardware would become very cheap. They expected you'd be able to get something like an NLS System for something in the range of $1000 to a few thousand dollars. Their vision was that they would bring the price down on the system and then improve its user interface.

They weren't exactly sure, though, of how they were going to deal with the user interface. That's the part I got involved with—the development of a very fast user interface that could be learned quickly. In that effort, I started doing something which the SRI and PARC researchers had not done before—I started letting users test alternative interfaces while I observed them as a psychologist would.

I've used that methodology ever since. We used it in the design of the Lisa and the Macintosh at Apple. The idea was to sit people down and let them use the system. In the process of observing them, I noted why they were having problems. Up to that time, whenever people had problems, the advocates of difficult systems would just declare philosophically that, well, these things are just hard to learn, but because they're so powerful, people will just have to learn them.

REVIEW: *That also answers the question of why acceptance took longer than some of the original founders thought it would.*

TESLER: Right. It wasn't just price; it was the difficulty of learning. There was resistance to

the complexity so I developed a word processor which was actually the first word processor to run on Xerox's personal computer prototype, the Alto. The program was called Mini-Mouse, and a user could actually learn how to use it in five minutes. It wasn't very powerful. All it could do was insertions, deletions, movements of text, and searches. It offered fixed-pitch fonts but no proportional fonts. But when we took people right off the street who could type but had never seen a computer, they were able to use all the features of the system within five minutes.

Critics complained that Mini-Mouse wasn't very powerful—that it didn't have all the great features we needed. But it proved the point that simplicity was possible. The next problem was to figure out how to apply these principles to all the other things people need to do in offices besides typing little memos. That's what we spent the next seven years working on.

REVIEW: *What kind of productivity improvements could you measure?*

TESLER: There was no difference, of course, in the typing of a document—that always takes the same amount of time. But when a user went back to edit a document, it was twice as fast using this system as it was using standard systems like the Wang. It was also two to four times quicker to learn than any other system around, and for some systems the difference was even higher. So Mini-Mouse offered both faster editing and faster learning. It also proved that you could make systems that were easy to learn.

In fact, the best way to learn interactive systems is to have a tutor who gives you a brief demo, and then sits with you and coaches you as you use it. That seems to work best for all systems. The other way is to do a similar thing with a book, a tutorial—preferably an online tutorial.

We proved the point that simplicity was possible.

If we make the computer look like a familiar object to a person, like a typewriter or a telephone or a filing cabinet, then we've made it more intuitive and easy to learn. If it looks like a programmer's idea of a computer, with commands and statements, then this will only serve to confuse a person. On a typewriter, carriage returns go to the next line. On some computers, carriage returns terminate a command. That can really throw beginners. On a typewriter, the typing of letters puts text into your document. On most computers, typed letters can also be interpreted as the names of commands. People just aren't used to the idea that typing has two different functions. They're not used to typing commands. They're used to typing information.

We weren't the only ones to discover this. Companies like Wang put function keys on the keyboard. That was much clearer to people. The function keys were more like the ones people had used on typewriters for single spacing, double spacing, and backspacing, among other functions. But the trouble with function keys is that when you build 82 function keys on your keyboard and your program has 82 functions, you're in trouble when you want to expand by adding networks and other things because you've run out of functions. So we threw all the function keys off the keyboard, except for the ones we couldn't, like backspace and carriage return. In their place, we put menus on the screen. Having a mouse, people could point at menus. That was the basic idea we developed at PARC in the early '70s.

NLS had the right idea in using a mouse and a full-page display, but it was wrong in the way it used letter keys to specify commands and in the way it was generally based on a computer model. Another thing NLS had was something we call "modes", like "insert mode" or "move mode". One of the things I discovered in my studies was that modes really confuse people. The people designing systems never seemed to take it as a challenge that they should do something about. They simply assumed users would have to learn about modes.

I decided, though, that we could eliminate modes, or at least some of the really annoying ones like "insert mode" and "replace mode" and "overtype mode". They're just totally unnecessary.

REVIEW: *But modes are still widely used in some systems.*

TESLER: And those are the systems that are hardest to learn. Now, that brings up the question of the kinds of resistance older workers have shown.

REVIEW: *You mean the sociological impact?*

TESLER: Yes. When you try to teach people to use one of these programmer-oriented systems—the ones that completely defy the intuitive models people have of how their jobs work—there will often be resistance. If you give them something which works in a

way they're used to, there's going to be a lot less resistance. I think this has less to do with user interface and hardware, though, than it has to do with sociology—that is, with the way that technology is introduced into the office. Making user interfaces simpler and offering better training will lessen the time it takes to learn these systems, and it will also improve the response people have to them. But the question of whether these people will be willing to accept the system in the first place has much more to do with how management introduces it.

The extreme case is when the workers come to work one day and find that their typewriters are gone, with word processors set down in their place. Now, if on top of that, the word processors are hard to learn, you're certain to have trouble. But the worst problems occur when people walk into their office and—without warning—find technology they never asked for. If workers are involved in the selection and evaluation of equipment from the start, they'll be waiting for it when it arrives. But if, in the workers' minds, you associate layoffs with the arrival of new equipment, that's going to be a problem. Of course, the workers who stay will eventually get over that, but it becomes a tense situation. Companies that arrange for displaced workers to move to other parts of the company, or who introduce outside methods slowly and use simple attrition to take care of displaced workers have far fewer problems than those companies that abruptly shift over to computers. Suddenly in that situation, instead of 20 people and 20 typewriters, there are 10 word processors, 10 unemployed people, and 10 insecure people.

REVIEW: *And then the surviving employees get marched off to a training class.*

TESLER: Yes. Those are the people that display the most resistance. This has been the real problem people have had with office automation.

REVIEW: *Let's bring this discussion right up to the moment. Are there other problems with office automation that really stand out? A lot of things are happening in the field now. You've got the introduction of a whole new set of high-level power tools: graphics, communications, and electronic mail, to name a few. All bring symbolic changes. What are the problems you see this causing at this level, right now and in the next generation?*

TESLER: Well, at this point I don't think the main problem with these kinds of technologies lies in the introduction of them. I don't think people are as resistant as they once were. Most people want a computer. They know computers take care of a lot of drudge work.

REVIEW: *Electronic mail is showing up on more systems. What do you feel the importance of that will be?*

TESLER: First of all, a lot of the current electronic mail systems have terrible human factors. They are hard to learn and hard to use. What's more, people can't do what they want with them. Things like easy editing and easy filing and retrieval just aren't there. I don't think people appreciate what they could actually do if they had electronic mail that offered good human factors.

A lot of companies have to use more than one electronic mail system. They're communicating within their own building, to other parts of the company, or to remote sales offices or something, so they might be using two or three different mail systems. In addition to that, they're probably

also using a public network like CompuServe in order to access databases. These folks have a serious communications problem since they have to use, and learn, so many different systems.

REVIEW: *That's creating a class of specialists in many companies.*

TESLER: Right. So I think the answer that's coming is the advent of what are known as "front-end" electronic mail systems. An example is the Transend System for the IBM PC that offers a single user interface that can be used to access all of your communications systems. Under this sort of scheme, it becomes the job of something called a "protocol language" to figure out that when the worker says to delete a particular message, this means to do one thing on one system and quite another on a different sort of system. The user never has to know that. He just communicates his desires by way of icons and that sort of thing.

There is another side to the acceptance issue, which is overuse. It used to be that if you wanted to send a letter to 100 people, you had to type 100 letters. So you thought about it: "Is this really what I need to do?" Then, when we got word processors, you only had to type one letter and one mailing list, and then run Mail Merge. After that, you just had to make sure the printer didn't jam. Still, somebody had to stuff 100 letters into envelopes and put stamps on them. So you thought again, "Do I really want all 100 of these people to get this letter?"

With electronic mail, though, you write a letter and you say, "Send it to distribution list X", carriage return, and that's that. All of a sudden, 100 people receive a message. It doesn't take any extra effort on your part. Unfortunately, it costs these 100 people extra effort to look at the message and decide whether it's of any interest to them. So you end up with a junk mail problem. When it gets easy to distribute mail, it gets hard to filter it. I don't know how many places are running into this problem now, but it's inevitable. I know we ran into it at PARC years ago.

If we make the computer look like a familiar object to a person, like a typewriter or a telephone or a filing cabinet, then we've made it more intuitive and easy to learn.

Public networks already have rules of etiquette about how to answer somebody's message. Even though a single keystroke is all it takes to send a reply off to everybody else who received the original message, these rules say, "Don't do that. Nobody else wants to see your answer to the message." The default of the system itself should be to reply only to the person who sent the original message. Rules of etiquette aren't the way to solve the problem. The vendors should fix the software so that it doesn't make undesirable options easy to use. For example, we want to make it difficult for people to read other people's mail. We also want to make it difficult for people to steal confidential information. We don't want ease of use to get to the point where it's easy to steal other people's property. It should also be difficult for people to flood the network with junk mail, like responses to other people's questions. And it should be easy for you to answer questions that are put directly to you. For example, if you get mail from Joe, you shouldn't have to type his entire network address in order to answer his question.

REVIEW: *Let's pick up the UNIX angle for a moment since you're getting to the issue of what's difficult and what's easy, what's accepted and what's not. There are a lot of advantages to UNIX—portability, communications, multiuser capabilities, and so on. But what are the disadvantages?*

TESLER: I agree that portability, communications, and multiuser capabilities are the advantages. I think human factors is the weakest area.

REVIEW: *Are you referring to the length of learning time that's required and the fact that the system doesn't use icons?*

TESLER: There's also a reliability problem.

REVIEW: *Let's start with that.*

TESLER: Advantages and disadvantages tend to come together. Multitasking is one of the great advantages of UNIX. It's important in the workstation area, where a user might work on three different tasks in a five-minute period. For all that, the disadvantage of multitasking is that if the system crashes, then all the tasks crash. So when a shared resource like a database server, a file server, or a large shared computing engine crashes, everybody using that system has to stop working.

REVIEW: *Let's talk about the user interface.*

fo · rum, n. (pl. FORUMS)
1. A public meeting place for open discussion. 2. A medium (as a newspaper) of open discussion or expression of ideas. 3. A public meeting or lecture involving audience discussion. 4. A program involving discussion of a problem by several authorities.

*eForum designed by Marcus Watts, Copyright 1984, Network Technologies International, Inc. (NETI).

Electronic meetings continue the automation of knowledge transfer which started with electronic mail.

Electronic meetings are an extension of the communications revolution which started with electronic mail. It takes seconds to send a letter using electronic mail instead of days via regular mail. Certainly e-mail is a giant step in automating correspondence between two people.

eForum goes yet further to provide immediate communications automation. But for groups. It creates electronic meetings which allow attendees to participate in discussions using the dynamic ebb and flow of points, counterpoints, comments and conclusions just like in-person meetings.

From an economics point-of-view, eForum is the most cost effective method for bringing together the best minds in your company to meet on key issues—without the price of a single plane trip, the aggravation of schedule conflict or time-consuming delay.

eForum is a communications breakthrough product.

eForum lets you create electronic meetings with attendee lists as large as the company staff or as small as a three-person design team.

Not only can eForum handle hundreds of meetings for your company, but, at the same time, limits each participant to only attending meetings to which he belongs.

eForum, n. 1. Low cost electronic meeting system (as in needing no scheduling or travel to attend). v. 1. Automatically organizes, indexes, files and leaves a complete written record of entire meeting. 2. Allows adding more attendees than normal at no extra cost. 3. Gives plenty of time to think before responding. adj. 1. Keeps everyone up-to-date. 2. Doesn't let geographic or time zones determine who can attend the meeting.

The Electronic Meeting Manager

If you have ever attended a meeting, you know how to use eForum.

Simply attend eForum meetings any time convenient for you. Review new discussion materials. eForum keeps track of what you've seen. Enter your comments or new discussion points. Instantaneously, your ideas are available to every member of your eForum group regardless of geographic location. That's productivity.

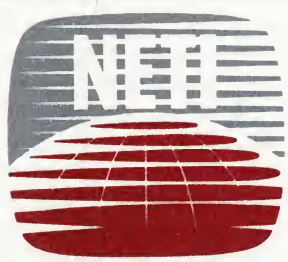
eForum has the flexibility to fit your communications needs.

- eForum 4000 - a national communications network available with a local phone call from most locations.
- eForum 2000 - UNIX™ based central host software for supermicro and minicomputers.
- eForum WS - software for the IBM PC and compatibles to interact with eForum central host software.

Call 1-800-638-4832 or in Michigan call (313) 994-4030 collect for information on:

- Automating your company's meetings by using General Electric Information Service, the world's largest communications network, to tie together your microcomputers and terminals.
- Creating your own meeting network for your department or company. Software, hardware and leasing available.
- Establishing OEM and VAR agreements to enhance the value of your software or hardware, with the communications power of eForum.

Circle No. 257 on Inquiry Card



Network Technologies International, Inc.
The Arbor Atrium Building
315 West Huron
Ann Arbor, Michigan 48103

™UNIX is a trademark of AT&T Bell Laboratories
™eForum is a trademark of Network Technologies International, Inc. (NETI)

TESLER: User friendliness is where UNIX completely falls down. An example is a UNIX mail system. Say you bring UNIX into an office and you tell the people there who have never used UNIX before, "You're going to have electronic mail. You don't have to learn all of UNIX. All you have to learn is electronic mail." Well, it's just not so. They'll have to learn how to log in. Hopefully, you'll already have some wizard in place who can set up their login files. Otherwise, they'll be completely lost, spending months getting their login files right. Hopefully, someone's already given these people a list that tells them about all the **mail** commands and what they do. But when they read that list, they won't have any idea of what it means. There are a lot of strange conventions they'll have to adjust to. Upper case means one thing, lower case means another. This strange, cryptic code puts users off right away.

But let's say these people go to a good training session where someone takes them through the system slowly and teaches them about the mailing system and all its subtleties. After a few days, they should be able to check their mail, file their messages, and answer their messages. But let's say a user is typing a message and needs to edit it. Now he needs to know the magic thing to do to get into an editor. Then he's got to learn how to use the editor. The editor has nothing to do with the mail system, but he's got to learn **ex, ed,** or **vi,** or whatever, just the same. Of course, the editor has a whole different command system than the mail system does, so some of the same commands mean different things, and many of the conventions are different.

So after your users have learned to deal with the shell, which has one command language, they've got to deal with the mail system, which has another command language, and then they've got to deal with an editor that has still another command language. Not only are the mnemonics different, but so are the conventions. It just doesn't have to be that way. A nice electronic mail system will, with a mouse, display your mail in windows. You can then edit it the same way you edit your memos, and after that you can go to a menu and indicate that you want to send a memo, and the system will give you a list of people you can select from by using a mouse. Then you click an "ok" button and you're done. There are variations on this, but there are a number of mail systems that operate in just this way. What you teach users is how to edit. If you make it easy enough, an editor can be learned in an hour—and it can be made into something that people won't forget.

But once people learn UNIX, they've made a big investment, and it becomes a challenge to increase their knowledge of the system. Once they have a stake in it, they resist the idea that other people might learn how to do something similar in an hour—even though it took them three months. These people then become UNIX "advocates".

REVIEW: *So there's a built-in resistance to simplicity?*

TESLER: Right. If someone gives these people an easy-to-learn system, it's going to be hard for them to learn. By then, they've got it completely wired into their brains that commands are to be typed on a keyboard. Their immediate reaction to a mouse is that it must be slower, it must have fewer features, or it must have something that's bad, because otherwise how could it be so easy? This suspicion results from a built-in resistance and it causes a problem. UNIX users generally say something like, "I don't need these features." That, in a sense, may be true, since they've already made an investment of time in learning to do things another way. The problem is, organizations are going to have new employees coming in, and these people might easily do something unintentionally—like delete a file or a directory full of files. There's no opportunity for confirmation and there's no way to get the file back. Bang! It's gone. And where did it go? That's why UNIX is an unsafe system for an office.

UNIX was designed for programmers—that is, full-time programmers who are experts and don't want to be bothered with extra confirmation keystrokes, because they're afraid they'll get slowed down. They know about regular backups and all the other things programmers do. But even then, programmers get themselves in trouble under UNIX and complain about it. But at least they're programmers and so have an understanding of how computers are.

REVIEW: *They're looking for speed and power, not ease of use.*

TESLER: Right. Often a programmer will delete a program accidentally, and then spend an hour trying to reconstruct it. He doesn't realize that the seconds he's saved by not having to hit a confirmation key or by not having to type an extra character is nothing compared to the time he's lost trying to find a backup tape and make all his changes over again. But programmers will be programmers. There's no way, though, that an office manager should accept that because there's no need for it.

REVIEW: *Is there a way to blend the best from UNIX and systems with friendlier interfaces?*

TESLER: One of the great things about UNIX is the ability to have shells like the Bourne shell and the C shell. One thing that's amazed me, though, is that so little work has been done to develop a portable window shell. Actually, they've worked on a window shell at Carnegie-Mellon under a contract with IBM. It runs on a Sun workstation. And Sun itself has a window shell that runs on UNIX. [AT&T's UNIX PC, designed by Convergent Technologies, also offers a window shell.] But I haven't figured out why there isn't a standard window shell that's portable and can have device-dependent drivers that make it work with any particular screen.

That would take care of one level of the problem. But you'd also have to go further. You'd have to provide a mouse, menus, icon-oriented text editors, icon-oriented mail systems, and everything else. What we need is a whole new applications software set. The reason I think it hasn't been done is that it would be such an enormous development job. Even given the current programming tools on UNIX, it would take years to develop all that software, and it would make UNIX even bigger than it is already, which is awfully large. You'd also have to decide if you were going to throw away the old shells and the old editors, and there would be a lot of resistance to that.

> One coming trend is that personal computers will be used as friendly front ends to software running on mainframes and servers.

In general, I like the UNIX operating system, but there are some features I wish it had. It doesn't lend itself well to Macintosh-style software that offers smooth mouse motion, it turns out. So that's a big problem you have to deal with. But if you were able to provide all the nice user interface features you need and if you were able to improve the operating system to support them and do all the other things you need to do, would you then still have UNIX? Maybe the thing to do is to start with UNIX and develop it into something that addresses these problems, or maybe the best thing is to start over from scratch.

REVIEW: *Can UNIX help propel office automation at all?*

TESLER: I think the important thing that UNIX offers office automation is not workstation software, but server software. Servers don't have a user interface, or if they do, only one person in each organization has to deal with starting it, scavenging the disk, and so on. It doesn't bother me that one trained person needs to be in the organization to deal with the UNIX shell. If the server runs UNIX, that's great, because then we can have electronic mail server software, database server software, file sharing server software, and print spooling server software—and all of it will be portable from server to server because it all runs on UNIX.

REVIEW: *Aren't you describing the direction Apple itself is taking to some extent with its automated officeware? Or with its server concept?*

TESLER: Oh, right, I see you know about the Lutzky-Baird system.

REVIEW: *Exactly. I was going to ask you about that.*

TESLER: Lutzky-Baird has a system that runs on a Zilog computer that works as a UNIX file server. It runs a file system and communicates over AppleTalk to a Macintosh running Macintosh-style applications, with no UNIX evident to the people who use the workstation. That's a very good example. I expect you'll see more of that kind of thing.

REVIEW: *Now let's take that and merge it into the question of where OA is going, irrespective of operating system.*

TESLER: There are two issues. One is greater market penetration, and the other is more augmentation for the people who already make use of office automation. Greater penetration is an issue that hinges on ease of use, marketing, sales, features, and other things. But let me turn my

focus to augmentation. It doesn't matter whether you're using office automation now. The question is: what will happen to your use of it later? As for the directions of the industry, consistent user interfaces are an obvious need I don't have to go into. Networking and electronic mail are also topics that are pretty well known. So let me talk about some things that are less well known.

One coming trend is that personal computers will be used as friendly front ends to software running on mainframes and servers. The Transend PC is an example. The Lutzky-Baird system is an example. There'll be lots more of that, with software running on a big system offering shared resources, and with a consistent user interface that allows users to interact with the system—whether it's in the next office or located across the country, or whether it's on the same brand of computer or a competitive make. Not only will this interface be friendly, but it will be integrated. You'll be able to take data that you bring down from one mainframe and integrate it with data that you've brought in from some public network. You'll then be able to integrate that with data you've produced yourself. You'll be able to analyze it locally or analyze it on the mainframe. One of the technological problems is how to get the data to flow readily. People are having a problem now with that. They need to download software from the mainframe to the PCs on their desks, which is wonderful, except that at 1200 baud, it can take a very long time.

REVIEW: *It does.*

TESLER: There are a lot of things being done about that. One approach is background downloading. That's what is done in TopView, for example. Another way to do it, which is an approach we're more interested in at Apple, is to do intelligent downloading. If you think about it, your personal computer has a lot less memory than your mainframe. So why bring down a lot of data from the mainframe? Probably because you want to look at it. But you can't look at a lot of data at the same time. Your screen is only a few inches by a few inches. You're either plotting a few numbers or looking at a table, or something like that. It can't be more than, say, a few thousand characters. At 1200 baud, that takes only a few seconds. So why are people taking minutes and hours to download files? Because they're bringing up stuff they don't need to look at right now.

The important thing that UNIX offers office automation is not workstation software, but server software.

Intelligent downloading is a matter of having the mainframe or the server hold onto the data until it's actually needed. And then, even at 1200 baud (which will hopefully go up to 4800 baud soon), you can get that data over. Of course, using a local area network, you can get things at much faster rates. On Appletalk, for instance, you can obtain data at a quarter of a million baud, so the problem is a lot less aggravated, but it still exists.

REVIEW: *What's being done to improve that situation? Selective menus, for example?*

TESLER: No. Let's say I have a large mailing list on the server. One thing I could do is transfer the whole thing over to my workstation and then scroll through it. The other thing I could do is transfer the first 20 addresses over to my workstation and just display those. As I start to scroll and get to the point where I need some more addresses, the workstation could then send a message automatically to the server asking for more; the user doesn't have to do anything—he just keeps on scrolling. This is really a software function. It results from software being smart enough to ask for the data when it needs it.

As another example, let's say you want to produce a chart. Instead of bringing all the data over and crunching it down to some small amount on the PC, you can get the server to crunch the data for you and then send only those numbers you're going to plot. Servers and mainframes are great at computing numbers and dealing with large amounts of data. Personal computers are great at plotting numbers. So why not let each system do what it's best at? Of course, the other way to solve the problem is to make personal computers bigger and faster, and that is happening. Five years from now, my comments here might sound ridiculous, because we may have trillion-byte disk drives on every personal computer by then.

REVIEW: *But they will be expensive.*

TESLER: Well, costs are coming down. But in any event, at some point these comments may seem to be ridiculous. For the moment, though, the issue is intelligent downloading versus background downloading. Both have value. Our future research at Apple is involved more in intelligent downloading. There's an analog

in printing. Instead of shipping bitmap images over networks to print on our laser printer, we send an encoded representation of the printed page, called a PostScript specification, which can be hundreds of times smaller than the number of bits needed to represent the document. This is an example of intelligent uploading. You can encode a document in such a way that very little information has to travel. As a result, communications take less time and place less strain on the network.

This accomplishes two things. We're cutting down the load on the network, which means you can either use a cheaper network or you can have more people on the network doing more things with it. We're also cutting down the users' waiting time. But if they just do background downloading, then they can't work on their data until it has all arrived. ing, they can start work on it right away. You're taking maximum advantage of the UNIX server's multitasking, as well as maximum advantage of the intelligent workstation's dedication to the user.

REVIEW: *Another feature that might bring more people into OA is voice.*

TESLER: I think voice input to computers is important, but not for office automation. I think it's important for home computing. Let's say you're at home reading a newspaper. Wouldn't it be nice to say, "Turn off the light and turn off the coffee pot," so that you could remain seated? When you're at home, there are very few people so there's not usually a lot of ambient noise, unless the television is blaring. If the television is blaring, voice is useless in the home too—at least until we get much better signal processing. If the best way to communicate with it is to talk to it. But in an office you tend to be close enough to your computer to touch it. If you're not, you'll probably be close to other people's computers. So if you talk to your computer, all those other computers will hear you too. And there's a lot of other noise in many offices.

If you have a private office, voice makes more sense. I can see voice being used by the executive who has his own office, doesn't like to type, is working on a plan for something, and has both hands occupied, with papers all over his desk. This fellow can say, "Get me the last quarter's financial results," without having to move his hands over to the keyboard. That's where voice will have a role in business. But when I think of office automation, I think of rooms full of people with terminals everywhere. I just don't see how voice can be a great help there. You'd have to have a little headphone. Also, people tire of talking all day.

REVIEW: *You'd certainly have cross-talk and you'd have to deal with a certain shyness that comes with not getting a verbal response.*

TESLER: Yes. I think it's a bit weird to talk to computers, personally.

REVIEW: *So voice becomes a luxury or specialty item?*

TESLER: I think it will be a special form of input. Eventually, voice recognition may be at the stage where you can have eight people talking together in a room because you'll have a microphone that's capable of sorting out all the voices. Before that happens, though, signal processing will have to reach a point where voices can be separated from each other. Our ears can do that—so why not


HEWLETT
PACKARD

Almost priceless.

Hewlett-Packard presents the first 32-bit UNIX™ system under $5,000.

Introducing the Integral Personal Computer.

It's the newest member of the HP 9000 family of HP-UX workstations. And it delivers the kind of power and flexibility the name implies.

With a 16/32-bit MC68000. A graphics co-processor. And 800KB of standard memory, expandable to 5.8MB.

Its UNIX kernel (HP-UX/RO) is built into ROM. Which means the Integral PC can run most applications *without* an expensive hard disc. Which means more people can afford to run your UNIX software than ever before. But not only can they afford to run it. They can learn to *use* it in minutes. Because the Integral PC is one UNIX system that's easy to use. Its see-and-select user interface, built-in window manager and optional mouse mean even novices can tap the power of your software. No one has to learn a bunch of cryptic commands or go through a lot of expensive training. There's even an entry-level, self-paced tutor disc with every machine.

What's more, the Integral PC goes where you need it. The complete system—including our revolutionary ThinkJet printer—packs into a single 25-pound package that takes up less than a cubic foot.

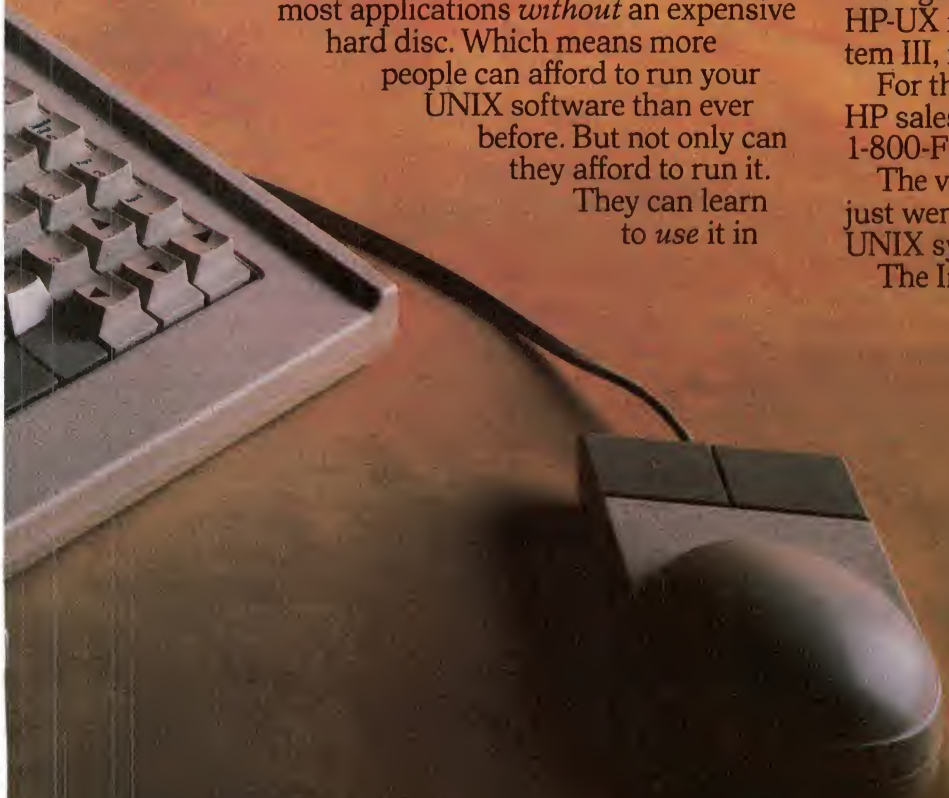
So get started porting your software now. HP-UX is an enhanced version of UNIX System III, including Berkeley 4.2 BSD features.

For the name of the authorized HP dealer or HP sales office nearest you, call toll-free 1-800-FOR-HPPC.

The value of your investment in UNIX software just went up. Because the price of a complete UNIX system just came down.

The Integral Personal Computer. Just $4,995.*

HEWLETT PACKARD



RULES OF THE GAME

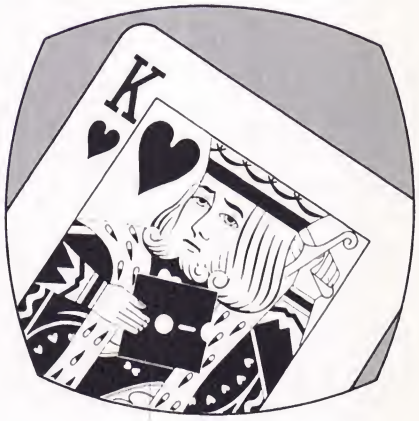
The phantom market

by Glenn Groenewold

Recently I participated in a panel put on in a major commercial center teeming with lawyers. The topic was legal applications for UNIX software. Although it might have been expected that at least some local attorneys would take advantage of the painless opportunity to acquire information concerning legal software, not one of the attendees was a practicing lawyer wishing to be educated about computers.

Not surprising, really. A short while before, a major bar association in the same metropolitan area undertook sponsorship of what was to be a two-day program intended to acquaint its members with computer fundamentals and specific programs that could be used in their practices. The cost of the presentation was nominal and the location convenient. But the affair was "postponed" and never rescheduled, because almost nobody signed up.

The aloofness that the nation's lawyers have demonstrated toward computer technology has been frustrating for many in the industry, because on the surface, attorneys appear to constitute a huge domestic market for multitasking computer systems. The United States has over a half-million practicing attorneys—more than any other country in the world—and law schools continue to grind them out in staggering numbers. Moreover, law practice seems a natural for computer applications, since it involves many repetitive tasks and requires that enormous amounts of data be monitored. So why have lawyers been so resistant to this new technology?



Nobody has satisfactorily answered this question, and I suspect that the actual explanation must be extremely complex. But I have some thoughts on the subject that I'm happy to share. After all, some of them may also apply to other pockets of resistance to computers.

TERMINAL SEXISM?

Take a good look at your computer terminal. Try to visualize it as someone who'd never seen one might. Aside from the screen, which after all doesn't look all that different from the familiar TV set, what do you see?

A keyboard, of course. And what does the keyboard remind you of? A *typewriter*, no? Now, in a traditional office setting (and law offices are *very* traditional), who is it that ordinarily uses typewriters?

Right.

Of our learned professions, the law has been the most male-dominated. As recently as 30 years ago, females simply didn't go to law school. Period. (Oh, there was one enrolled in my school, but she was the only child of one of the state's most prominent lawyers, and who else was there to carry on the family tradition?)

Although this situation has changed dramatically in recent years, old traditions don't fade gracefully. And until recently, the tradition in law offices has been that only secretaries were to use typewriters.

Actually, most lawyers *can* type—it's not easy to get through law school without acquiring the skill—but even today many don't wish to admit it. I've known female attorneys who were especially adamant about this, fearing that proficiency in typing would result in a loss in status.

If this strikes you as far-fetched, how else can you explain the dogged reluctance of attorneys, male or female, to have a

4.2BSD

FROM NOW ON, CONSIDER IT SUPPORTED

When it comes to Unix™ systems, "standard" isn't always good enough.

Experts agree that the most powerful and most technically advanced **Unix** system is the Berkeley version. That's why 4.2BSD from Berkeley is the operating system of choice for software development, networking, engineering, CAD/CAM and demanding scientific applications. Other **Unix** systems don't have the features advanced users require.

But 4BSD was developed at a university, so it has never had real-world support. User assistance, bug fixes, updates and enhancements have not been provided.

Now that's changed.
MT XINU, the 4BSD specialist, supplies:

- Fully supported 4.2BSD-based binary licenses (MORE/bsd) for VAX™ computers.
- 4.2BSD source support and source updates for current 4.2BSD source licensees.
- Enhanced 4.2BSD-based source software for new sites, with or without redistribution rights.
- Full support for a wide variety of DEC™ and non-DEC peripherals.
- Assistance for OEM's and hardware manufacturers developing 4.2BSD-based products.

MT XINU personnel have been involved with 4BSD development from the beginning. Now we are producing 4BSD performance enhancements, advanced networking, other **Unix** system extensions, and support for new peripherals and architectures. As a service, we distribute 4BSD bug reports and proposed bug fixes to the community. Our years of experience can speed and improve your 4BSD implementations.

4.2BSD. It's always been better than just "standard." Now, with MT XINU, consider it supported.

"We know UNIX™ Backwards and Forwards"

MT XINU™

UNIX™ SUPPORT FROM BERKELEY

Circle No. 247 on Inquiry Card

739 Allston Way, Berkeley, CA 94710 ■ 415/644-0146 ■ ucbvax!mtxinu!mtxinu

MORE/bsd and MT XINU are trademarks of Mt Xinu Inc., DEC and VAX are trademarks of Digital Equipment Corp., UNIX is a trademark of Bell Laboratories.

The Firebreathers continue on the cutting edge of high performance computers.

The most powerful line of computer systems made. Gould PowerNodes™ and CONCEPT/32s.® Any way you slice it they beat VAX.™ Not just by a little. By a lot. Whether you're comparing power, price or availability, we've got just the edge you need to ax the VAX.

For starters, our mainframe PN9000 and CONCEPT 32/97 are up to twice as fast as the VAX 8600. And even though the mid-range PN6000 and CONCEPT 32/67 are 30-50% smaller than the VAX 11/780, they're still up to three times more powerful. Which means they'll run your applications faster, much faster, than VAX.

More power for a slice of the price.

Despite their superior power, Gould computers cost quite a bit less than the equivalent DEC computers. Just take a look at the chart on your right.

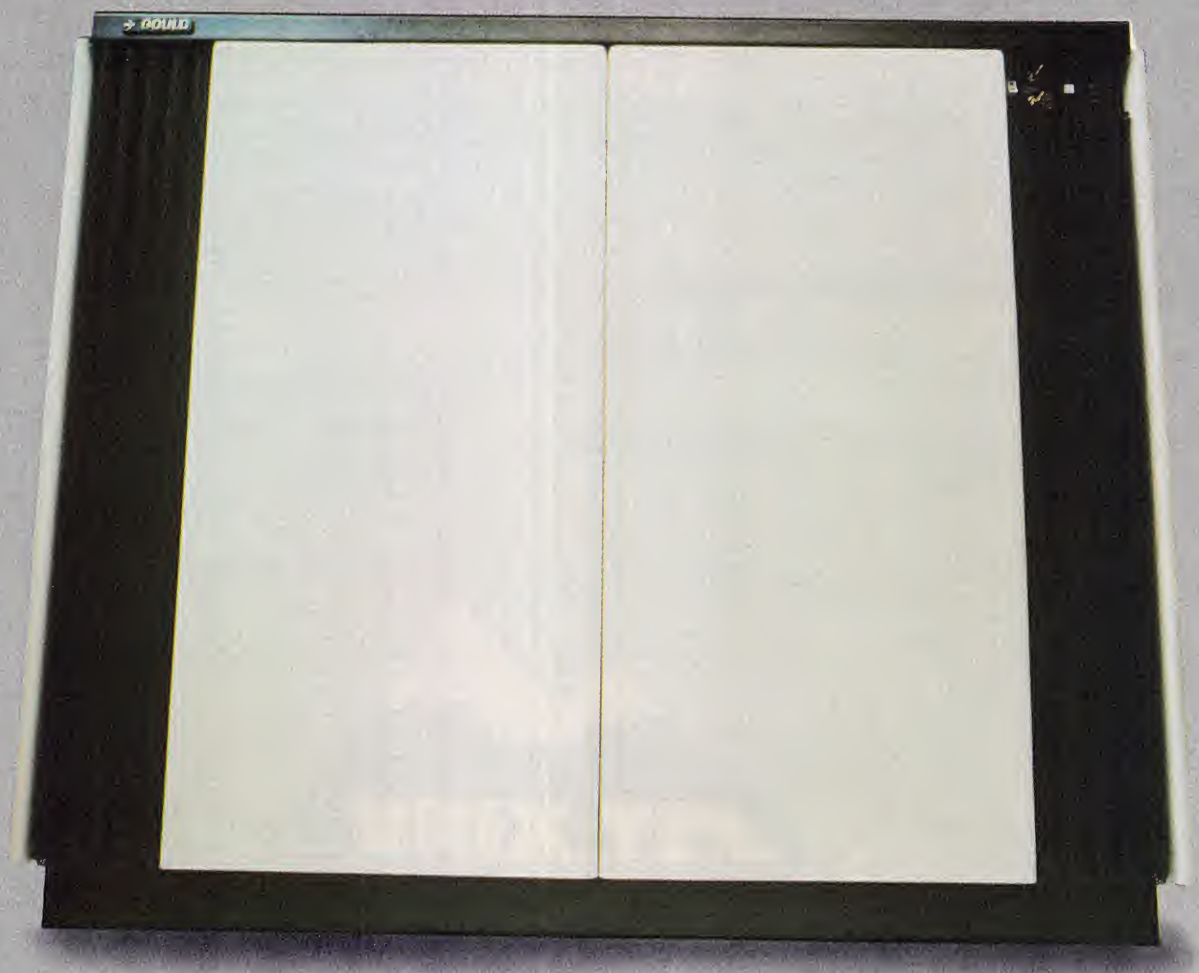
Our mid-range models cost 40% less than the VAX 11/780. Our mainframes cost about 30% less than the new VAX 8600.

In fact, all Gould computers come with a price that won't come to blows with your budget. The bottom line is more power for less money. You don't need a computer to figure out what an advantage that is.

Operating environments that are a cut above the rest.

There's also a choice of system software to consider. Gould offers a unique implementation of the UNIX* operating system—UTX/32.™ It's the first system to combine the UNIX standards of System V with Berkeley BSD 4.2. Which also makes it the only system that allows you to access virtually

Only Gould computers have a big enough edge to ax the VAX.



MODEL	$(000)	MIPS/$
VAX 8600	575	7.6
PowerNode PN9000	400	11.7
CONCEPT 32/97	400	11.7
VAX 11/780	235	4.8
PowerNode PN6030	140	12.1
CONCEPT 32/6730	140	12.1

any command you want whenever you want.

Or if your application demands a real-time environment, Gould's MPX/32™ operating system offers performance that's unmatched in the industry. And that puts Gould far above the rest.

Delivery that's right on the mark.

Unlike the VAX 8600, that has up to a 12 month wait for delivery, when you order either a Gould PowerNode or a CONCEPT/32 system they will be shipped within 90 days ARO. And that's a schedule we'll go to any length to meet. Because we know how annoying it can be to wait months, and years, for something you've just agreed to pay a healthy sum of money for. Especially when the only thing you can do while waiting is watch your work pile up.

You can also be sure with Gould you're getting a computer that's backed by years of experience. It's the kind of experience we used to develop the first 32-bit real-time computer. The kind of experience that has the competition running to keep up.

If you need more information or just have a few questions, give us a call at 1-800-327-9716.

See for yourself why VAX no longer cuts it. Go with a Gould computer and ax the VAX.

GOULD
Electronics

terminal in their own office? The fact is most lawyers haven't even accepted typewriters into those sanctuaries as yet. And the few that have usually either hide them or make certain that the machine is an anemic portable, manifestly unsuited for turning out finished documents.

In actuality, many law firms *have* admitted computers—usually in the form of word processors—into their premises. But nearly always, this technology is utilized only by secretaries and clerical employees. Typically, the lawyers maintain that they don't know anything about these devices, and, furthermore, that *they don't need to.*

It may seem strange that a bunch of people who usually take the position that they know more than most anyone else about virtually everything in the world should abdicate all knowledge of computer technology to employees they regard as lower in status, but that's exactly what attorneys, as a group, have tended to do. And this constitutes a major obstacle for vendors and consultants seeking to bring lawyers into the computer age.

WHAT, ME LOOK UNDER THE HOOD?

Practicing attorneys do have another, more valid, excuse for affecting a lack of interest in computers. Their minds have to be among the most overburdened on earth. The volume of technical material with which a lawyer must be familiar is staggering. And the load becomes heavier with each passing year. Legislatures and administrative agencies constantly crank out laws and regulations, to the point where legal publications can't keep up. Courts issue more and more published opinions every year, and lawyers have to keep abreast of this flood of material.

Other demands on a lawyer's time have also become brutal. The number of court appearances required for a given case has multiplied, and because dockets are crowded, it's become common for long hours to be spent in court waiting for cases to be called. A burgeoning number of forms must be filled out, lengthy depositions must be attended, and anxious clients must be counseled.

Law practice seems a natural for computer applications.

What's more, detailed records must be kept. Lawyers have to worry about the IRS like everyone else, and increasingly they have to be concerned about protecting themselves against possible malpractice suits.

It's sometimes difficult for people whose lives revolve around computers to accept the idea that there might be intelligent people who don't have the available time or intellectual energy to interest themselves in this marvelous new technology. But many lawyers are genuinely in this position. They don't want to have to learn *anything* about computers, because they know that a little knowledge can be a disquieting thing, and they've got plenty to worry about already.

For that matter, how many of the people you know would be driving automobiles now if today's cars made the same demands on operators as those of 60 or 70 years ago? How many of us could cope successfully with repairing engines ourselves, patching innertubes, and navigating unmarked roads?

Yet, in many ways, computers today present challenges similar to those posed by old-time cars. You have to know a good deal in order to be able to use one, and things do go wrong. Few medium-sized law firms have the financial resources to keep full-time systems administrators who can keep their computers operating on the payroll, and delays of even a few hours can be fatal in law practice.

Consider this in light of the fact that no one has accused UNIX of being particularly user friendly. Even a completely shielded system doesn't approach the user ease of the modern automobile, where you need only turn the key and step on the gas.

SELLING COMPUTERS TO LAWYERS

For all the difficulties it presents, the legal market for computer technology is a tantalizing one. For some time, customized systems for law firms have been designed by consultants and vendors. Recently, some hardware manufacturers have been developing integrated software packages tailored to specific vertical markets such as the legal field. The legal market was the first one selected by Fortune Systems for its vertical marketing programs, under which it is contemplated that the company's dealers will market recommended software provided by other suppliers.

One problem is that all lawyers do not have the same requirements. For instance, some firms routinely are involved in complex litigation, while others rarely appear in court. The former will find a litigation support system, such as that offered by Liticom, to be of assistance in keeping track of the enormous amount of data generated in the course of a complicated lawsuit, while the latter prob-

ably would have little use for this special application.

This suggests that while many law firms will be able to get along nicely with packaged legal software, there will always be a niche in the legal market for the independent consultant and for custom-designed systems.

OTHER PROBLEM AREAS

Whatever the approach used to equip law offices with multiuser, multitasking computer systems, certain problems seem to be inescapable.

The first, of course, is that it will be necessary to convince individual attorneys that they each *need* and *want* terminals in their offices. One of the ways to do this is to present a software package that effectively shields users from the terrors of the basic operating system, but preserves the utility and versatility inherent in UNIX. For the novice user, this means a foolproof menu that nevertheless provides sufficient options for users to realize the fullest benefits of the system.

But a successful solution to this first problem may carry the seeds of a different malaise. As the lawyer becomes familiar with the system, he or she may chafe under the limitations imposed by the menu, which can begin to appear simple-minded and unnecessarily time consuming. Therefore, it may be desirable for the system to permit a knowledgeable user to circumvent the restrictions that are created by the interface.

Another problem is engendered by inevitable changes in personnel. At all levels within an office organization, employees will depart for greener pastures, or to produce babies, or to take 'round-the-world cruises. Months of training and experience depart with them. In the past, law office skills have been fairly well standardized, so that new employees were readily interchanged with those they replaced. But there's not a great deal of standardization in those multiuser computer systems gradually taking hold in law office environments, even among the ones based on similar operating systems.

Finally, there is the problem of reliability. The law is one of the few areas in our society in which there are deadlines that are absolute. If a particular document *must* be on file at the court clerk's office by 5 pm on a given day, it is *not* good enough to have it there when the office opens the next morning. The possible import of a system crash at 3 pm should be apparent.

Despite the many difficulties, the likelihood is that lawyers in time will accept the assistance of multiuser, multitasking computer systems, mainly because they have no choice if they are to survive in the evolving legal environment. But it's also likely that the process of capitulation will not be an entirely graceful one. Anyone trying to crack the legal market would be wise to keep this in mind.

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

INDUSTRY INSIDER

A sleeping giant stirs

by Mark G. Sobell

Digital Equipment Corporation (DEC) may not have stunned anybody with its recent release of the MicroVAX II, but it would be a mistake to take the announcement lightly. Based on a proprietary microprocessor dubbed the "VAX-on-a-chip", DEC claims the new machine has 90 percent of the horsepower offered by a VAX 11/780. What makes this particularly interesting is that the new machine is priced between $20,000 and $44,000, a savings of at least $100,000 over a full-sized VAX 11/780.

Along with the MicroVAX II, DEC released its VAXstation II, a high-performance graphics workstation. The VAXstation II is a MicroVAX II with a QVSS graphics engine that drives a 19-inch monochrome monitor featuring a 1024 x 864 pixel resolution.

The general reaction to the new products has been something on the order of, "It's a DEC product, so it will be reliable, it will be supported, and—as a result—it will be accepted." It's not quite that clear, though. While looking at DEC's new offerings, it is important to separate the graphics workstation from the timeshared computer as they address two distinct markets.

DEC's traditional market has been the timeshared computer market and there is little doubt that its MicroVAX II will do well in

this area, though it will get a run for its money from the new 68020-based products starting to appear (the Altos 3068, for example). DEC's reputation will have to overcome the price differential between its MicroVAX II and some of the new supermicros on the market, but then DEC has always been more expensive than its competitors.

The graphics workstation market is a relatively new one for DEC and there are some well-established players already in the field. DEC's first entry was the VAXstation I. With the VAXstation II, DEC will be trying to catch up with Sun and Apollo.

WHAT'S INSIDE?

The heart of the two new products is the MicroVAX II 78032 chip. It is a 32-bit microprocessor implemented in a 68-pin surface mounted package featuring 32-bit internal and external data paths, instruction prefetch, and on-chip demand-paged virtual memory management. Included are sixteen 32-bit general purpose registers and 31 interrupt levels.

The MicroVAX 78132 floating point unit works in conjunction with the 78032; it handles floating point data types and accelerates integer multiply and divide functions.

Both of the MicroVAX II products are based on the Q-bus, which is time-multiplexed. Although a multiplexed bus requires fewer lines and thus reduces costs, it also means that DEC's new machines are significantly slower than they would be had they been equipped with either the UNIBUS or MASSBUS used by the full-sized VAXen. Use of the Q-bus means that other Q-bus peripherals will be available for service on the MicroVAX II, but it also means that larger disk drives and many of the other peripherals that work with the larger VAXen will not work directly with the MicroVAX.

DEC'S FUTURE

Analysts are saying that the future of DEC rides on these products. For the past four years, DEC profits have shrunk, partly because DEC never got its foot in

the microcomputer door. Now DEC is playing a dangerous game. A win depends on the acceptance of its new MicroVAX chip-based products in offices that are traditional strongholds for companies like IBM and Wang. Whether it wins or loses this battle, DEC sales of its smaller VAX products (11/725, 11/730, and 11/750) are sure to suffer—who, after all, is going to buy a less powerful machine for more money? Of course, DEC still has the high end of its product line to rely on: the VAX 11/780, the VAX 11/785, and the VAX 8600.

Despite the fact that DEC has a lot at stake, it surprisingly has not restructured pricing at the lower end of its VAX line. For now, the big VAXen still support a faster bus than the MicroVAX, they are physically equipped to handle more I/O connectors than the MicroVAX, and they can work in VAX clusters (tightly-knit groups of VAXen). But it will not be long before these features come to the MicroVAX. The real question is: what will happen to the rest of DEC's product line?

COMPATIBILITY

DEC claims that its entire VAX line, including the MicroVAX II, is object code compatible, which is to say you should be able to compile code on a 11/780 and run it on a MicroVAX II and vice versa. This compatibility will surely be a boon in the VMS (DEC's proprietary operating system) marketplace where there already is much software developed for the larger VAXen. It will also be important in the UNIX marketplace because DEC is supplying its UNIX entry, Ultrix, on the MicroVAX II.

Reports from beta test sites indicate that the MicroVAX II is all DEC says it is. Richard Newton, a professor of Electrical Engineering and Computer Science at the University of California at Berkeley (a MicroVAX II beta test site) said that he has been moving *compiled* (linked binary) code from a full-sized VAX 11/780 running 4.2BSD to the MicroVAX II running Ultrix. He has not found a case where the code needs to be re-compiled or re-linked, and what's more, he said that the code can be compiled and linked on the MicroVAX and moved to the VAX 780 without any problems.

Newton also actually improved on DEC's claims for the processing speed of the MicroVAX II. He did note, however, that his MicroVAX II has 9 MB of RAM, which is definitely a factor when running large programs. "The worst case we could develop showed the MicroVAX II running 14 percent slower than our full-sized VAX—and that took a lot of effort on our part," he said. "Most of our tests between the two machines came out a wash; sometimes the MicroVAX II won." The university's SPICE II electronic circuit simulation program, for one, ran faster under Ultrix on the MicroVAX than it did on a VAX 780 under 4.2.

DEC's reputation will have to overcome the price differential between its MicroVAX II and some of the new supermicros on the market.

Something else interesting apparently happened when Newton's department was testing Fortran 77 programs. "The compiled Fortran programs we moved from the VAX 780 to the MicroVAX II ran 6 percent slower on the MicroVAX II," he said. "But when we recompiled the code on the MicroVAX II, the programs ran 2 percent faster on the MicroVAX than on the 780." DEC attributes the performance increase to ongoing bug fixes and improvements in the f77 compiler.

ANOTHER PERSPECTIVE

I also spoke with Bill Joy, historically one of the most important forces behind Berkeley UNIX and currently Vice President of R&D at Sun Microsystems. Joy stated, "The MicroVAX II is DEC's first major price/performance improvement since the VAX 750. It's especially significant for VMS users. However, that does not necessarily mean it's the best choice for a UNIX user." After pointing out that DEC does not support a diskless node VAXstation—a very cost effective type of node in light of today's RAM prices—and noting that DEC's idea of a color graphics system is a kludge (a separate Tektronics color monitor attached to a monochrome VAXstation), Joy questioned the wisdom of using the slower, time multiplexed Q-bus to support a powerful 32-bit processor.

During our talk, Joy stressed that DEC is pushing VMS on its new products rather than Ultrix, and wondered aloud when DEC would demonstrate its commitment to UNIX. "When will DEC bring up its full range of quality VAX software under Ultrix?" he asked. "When will it support a network file system? And why did it choose a proprietary tape format for its quarter-inch drive

instead of using the ANSI standard?" Joy later changed tracks, though, and went on to praise DEC for choosing Ethernet, "unlike IBM, which is still using relatively obscure networking technology."

Joy summed it up this way: "People are stuck on the notion that all they need is a (VAX) 780, but supermini performance has progressed beyond that point. DEC claims that the MicroVAX II is two years ahead, but if you examine the price/performance of today's microprocessor-based systems, such as those based on the 68020, I don't think its claim will hold up. Unlike VMS users, UNIX users have a wide choice of micro engines. In order for DEC to make a strong showing in the UNIX market, it will have to substantiate its claim that it is ahead of its time."

It should be noted that Sun expects to release 68020-based workstations in the not-too-distant future.

LASER DISK

Another interesting offering unveiled by DEC during the course of its MicroVAX II announcement was a compact optical disk (CD) reader. The reader uses read-only disks that are slightly smaller than 5 inches in diameter but store 600 MB of data—which is the equivalent of two complete sets of encyclopedias! The CD reader is priced at about $2200.

DEC also announced it would be going into the disk production business to assist its customers in moving data onto compact disks. Prices in quantities above 1000 will be in the neighborhood of $30. Look for other manufacturers to go into the CD production business shortly.

CONCLUSION

DEC has established a strong presence in the workstation realm with its MicroVAX series. The MicroVAX II will certainly be successful in replacing low-end VAXen. The question now is: how much market share can DEC take away from major office computer manufacturers?

DEC's success in the graphics workstation arena will depend on whatever other products it comes out with in its VAXstation line. Offering a color display as an integral part of its workstation would be an important start. Powerful graphics software, both at the applications level and at the system level would round the product out nicely. The series' support of the GKS (Graphics Kernel System) standard is good, but right now there's not a lot of GKS-based software around.

A bright star on the horizon: Interleaf has reportedly rewritten its word processing/graphics/typesetting system to run under VMS on the VAXstation. Theoretically, it should also run under Ultrix.

BRIEFS

What lurks on the competitive front? Gould has agreed to use its international sales organization to distribute Sun systems worldwide, and Sun has also announced that Hyundai will have exclusive distribution rights for Sun products in Korea. Hyundai is the largest industrial conglomerate in Korea with 1984 revenues of $10 billion. As a bonus, Sun is anticipating that Hyundai will use quite a few workstations internally.

Denmark selected Cromemco for its Preferred Product Status listing in the category of supermicros. Cromemco was the only United States manufacturer selected for the list. Among the other manufacturers on the list are Danish Data Electronics and Olivetti.

NCR has announced the Mini-Tower, a 68010-based system running System V. The $7000 system will come with a half megabyte of memory, a 25 MB hard disk, a 1 MB floppy disk, and eight serial ports.

From the rumor mill come whispers of a 68020-based Macintosh to be sold by Apple for under $10,000. Hmm ... a nice, friendly environment for UNIX, no?

Finally, AT&T is in the news again with the announcement of a letter of agreement between its Technology Systems division and Honeywell calling for negotiations for the purchase of many of the assets held by Honeywell's Synertek subsidiary. The assets under discussion include Honeywell's metal oxide semiconductor (MOS) design and manufacturing facilities in Santa Cruz, Bangkok, Singapore, and Munich. John Nemecek, executive vice president of AT&T Technology Systems Group said, "with the possibility of acquiring this added capacity for producing high-technology chips, AT&T will underscore its intent to be a major player in the worldwide component business."

If you have an item appropriate for this column, you can contact Mr. Sobell at 333 Cobalt Way, Suite 106, Sunnyvale, CA 94086.

Mark G. Sobell is the author if "A Practical Guide to the UNIX System" (Benjamin/Cummings, 1984) and "A Practical Guide to UNIX System V" (Benjamin/Cummings, May 1985). He has been working with UNIX for over five years and specializes in documentation consulting and ***troff*** *typesetting. Mr. Sobell also writes, lectures, and offers classes in Advanced Shell Programming and* ***troff*** *macro development.* ■

DEVIL'S ADVOCATE

Keeping track

by Stan Kelly-Bootle

When I worked for what is now the largest computer company in the world, IB-something-or-other, way back in the '50s, I not only knew every available computer model, I also knew, off the cuff, where they were installed, what kinds of jobs they were running, and which cards had an "X" in column 80. Indeed, I was on object-code terms with all known programs and their perpetrators, often with a degree of intimacy too hot not to cool down.

Even today, I still try to keep abreast, but my omniscience is not what it used to be. Yes, I hear your howls of disbelief, but facts are facts. It is certainly true that Knuth and I, pooling our knowledge-bases, as it were, would be able to answer all worthwhile questions (our respective lacunae happen to be non-intersecting). But, left on my own, I have been known to turn the other hedge.

When the Univac reached double figures I was forced to commit my private Users' List to the spare pages at the back of my diary so that I could refresh my memory just before ACM dinner parties. An up-to-the-minute knowledge of "who-got-what" was considered a major social asset, beating a black belt in Mah Jong or a working acquaintance with the Earl Scruggs five-string banjo. Of especial merit was the ability to

drop casually or counter-drop bitingly some obscure snippet of computer gossip ("Groupe Drouot has gone Bull, would you believe?" "Really? When I dined with Jacques last night, they weren't even on the old Short List." "How bizarre! I had breakfast with Jacques' wife this morning . . . she's the real decision-maker, you know . . .")

My *pièce de résistance*, guaranteed to empty any bar, was a snappy overview of the fifty Z22s installed by Konrad Zuse.

Then the pace quickened with thousands of 1400s and hundreds of 9000s (this column refuses to bore you with precise numbers), forcing all but the bravest to abandon the notion of a comprehensive user list. I, of course, perservered right through the deluge of DEC 10s and DEC 10 look-alikes, gamely filling several notebooks a day, and beginning to wonder, "My God, is it all worthwhile?"

Incidentally, some of you may be surprised to learn that the parasitic art of building look-alikes and act-alikes predates Compaq; in fact, exploiting the success of one's competitor predates even Data General, Honeywell, and AlphaMicro. An advertisement in the *London Times*, dated May 15, 1851, proclaimed:

> **At Last! Comcog Unveiled!**
> **Analytic Engines for Affordable Disbursements!**
> **A Truly Wondrous Break Through!**
> **All of Mr. Babbage's Most Popular Sequences Will Run As Are!**
> **Stretch Your Differencing Pennies!**
> **Comcog Might Well Be the LAST Analytic Engine You Will Ever Need to Purchase!**
> **How Do We Spell Compatibility? C-O-M-C-O-G!**
> **(Of Late Dilligently Demonstrated at the Palace of St. James' Before Crowned Monarchs and Learned Skeptiks!)**

Maintaining my User List became a full-time horror during the Micro Boom of the '70s,

Try listening to UNIX

That's what we do in UNIX REVIEW
We listen, and look, and probe
It's surprising how much there is that's fresh and new.
Just waiting to be shared
It's often that way with things that are simply elegant.
Like a shell. Like UNIX

UNIX™ REVIEW

THE MAGAZINE FOR THE UNIX COMMUNITY

whereby 20 years of architectural bungling were suddenly concentrated into a single, mass-reproducible chip . . . and the Sorcerer's Apprentice, known as Intel, could not find the right incantation to stem the flood.

If pressed, I would now have to admit that I have no idea who is using the TRS-80 serial number #198563, nor could I cite the attic in which the Coleco Adam #6543 is stored. Because of the gray market I have lost track of several Sinclair ZX81s; worse still, people are slotting all manner of boards into their PCs, changing operating systems, keyboards, drives and monitors without advising me or IBM. There must be hundreds of so-called PC users on my list who are essentially covert Macintosh users, but too ashamed to admit it. Look, if you want a Mac, buy a Mac, and stop screwing up my User List!

The parasitic art of building look-alikes and act-alikes predates Compaq.

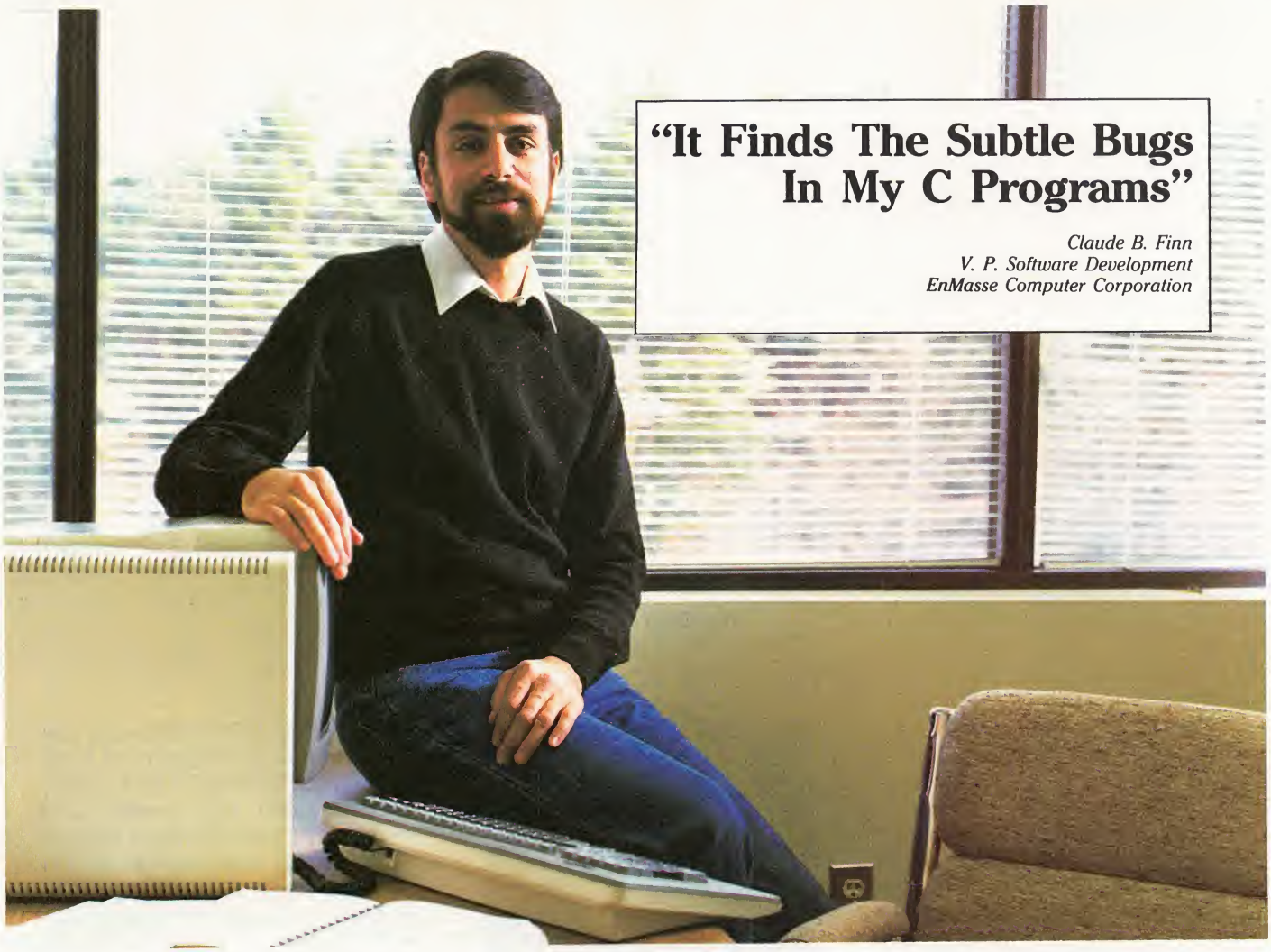
Likewise, I am reliably informed by no less a source than *Recent Studies at a Well-Known West Coast University* that some of you are also tampering with your Microwave Ovens! My TMS1000 and PPS4 lists, tedious enough to maintain in the best of times, are being unnecessarily distorted by undercover chip-switching. It is a grotesque misuse of the Motorola MC68020 and the System V/68 OS! Your Cup'O'Noodles will *not* warm up any quicker with a 32-bit timer! And, is it not reckless overkill to use:

```
$ grep steak oven
```

to check the contents of your TV dinner?

I am planning to publish my Universal Comprehensive User's List, warts and all, with a daily supplement. It will not be cheap, but remember, it's more than just a tabulation of models and owners. We'll show you exact configurations and all the applications (including filenames, file contents, and the source code). The disassembled ROM listings and *Current Password* sections alone will quickly repay your modest investment. Founder subscribers will receive, absolutely FREE, a copy of my new book, *Winchester Technology: Are We Just Scratching the Surface?*, plus a set of matching steak knives, a bamboo endive steamer, and a rare poster showing the UNIX REVIEW Editorial Review Board hard at work.

Liverpool-born Stan Kelly-Bootle has been computing, on and off, at most levels since the pioneering EDSAC I days in the early 1950s at Cambridge University. After graduating from there in Pure Mathematics, he gained the world's first post-graduate diploma in Computer Science. He has authored "The Devil's DP Dictionary" and co-authored "Lern Yerself Scouse" and "The MC68000 Software Primer". ■



"It Finds The Subtle Bugs In My C Programs"

Claude B. Finn
V. P. Software Development
EnMasse Computer Corporation

The SAFE C™ Family Can Literally Cut Software Development Time In Half. For UNIX™ and VAX/VMS.™

"Evasive bugs that use to eat up days — I'm finding them in minutes. Stray pointers, errant array indexes, parameter mismatches, misuse of string functions...I'm using Safe C automatic error detection to find them all."

Claude Finn is one of the many C programmers who have discovered that the Safe C family of software development tools dramatically enhances programmer productivity and improves software reliability and portability. Most Safe C customers have recouped their investment in these tools within the first month of active use. And with the security of Safe C their programmers are sleeping a lot easier!

The Safe C Family includes the Runtime Analyzer, Dynamic Profiler, Standalone Interpreter, English to C Translator and C to English Translator. They may be purchased separately or as a group.

CCA Uniworks, Inc.
Productivity Tools for Programmers
20 William Street · Wellesley, MA 02181
Circle No. 279 on Inquiry Card

For more information or to place an order call our customer representatives at

800-222-0214

In MA (617) 235-2600
or mail this request form today.

Please send me information on:

- ☐ The Safe C Development Tool Family
- ☐ AI Development Tools
- ☐ CCA EMACS Unix/VMS Editor Environment
- ☐ Your complete line of state-of-the-art programming tools
- ☐ Please send license forms

Name _____
Title _____
Company _____
Address _____
City, State, Zip _____
Phone ( ) _____

UR785

CCA UNIWORKS, INC.
20 William Street · Wellesley, MA 02181

A Crowntek Company

Unix is a trademark of Bell Laboratories, VAX and VMS is a trademark of Digital Equipment Corporation, MS/PC-DOS is a trademark of Microsoft, Safe C is a trademark of Catalytix Corporation, CCA EMACS is a trademark of CCA Uniworks, Inc.

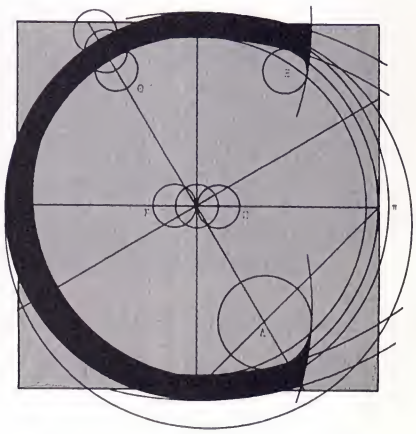
C ADVISOR

How Xenix stacks up

by Bill Tuthill

According to marketing hype, Microsoft's Xenix makes up 80 percent of the UNIX micro market. This figure is somewhat deceptive, though, because some companies (like Fortune Systems) license through Microsoft, but nevertheless deliver their own version of UNIX. Moreover, the figure relates to the number of machines sold rather than to the number of people using them. As a result, it fails to account for the many UNIX minis and mainframes that may serve as many users as would 50 micros. All the same, there can be no doubt that Xenix makes up a substantial portion of the UNIX market.

Simply because of the sheer number of Xenix systems, applications written for the office automation market are likely to encounter Xenix sooner or later. The best-selling UNIX micro, Tandy's TRS-80 Model 16, and its replacement, the Model 6000, both run Xenix. The Altos line of computers also runs Xenix, and Santa Cruz Operation supports Xenix on IBM's PC-XT and its various look-alikes. What's more, IBM has put its stamp of approval on Xenix as one of the operating system options for the PC/AT. Even AT&T itself offers Xenix, on its PC 6300. The office automation market, unlike the scientific/



engineering market, leans toward small machines from large suppliers.

True to its reputation, Microsoft has slowly built Xenix into a quality product. The first Xenix ports were not very good, but with the passage of time, they've gotten better and better. Most people now run Xenix version 3.0, although version 5.0, with System V compatibility, will be available in late summer (there will be no version 4.0). Today, Xenix 3.0 is better than certain System V ports on the market in terms of speed, reliability, and completeness. Many Berkeley enhancements not available on System V are already included in the Microsoft system.

Xenix could best be described as a UNIX system based on Version 7 that offers Berkeley enhancements and System III compatibility. Microsoft had added **termcap**, **vi**, and **curses** long before AT&T put them on System V. Microsoft also grafted on the System III terminal driver, described in this column last July, and System III compatibility libraries. The C environment on Xenix is largely compatible with both Version 7 and System III (which is much like System V).

Xenix exemplifies what is good about UNIX. It runs on a wide variety of microprocessors, including the Intel 8088, 8086, and 80286; the Motorola 68000, 68008, and 68010; and the Zilog Z8000. The user interface is remarkably similar across these processors. Microsoft devised a single format for executable files, called *x.out*, which is much better suited to the rigors of microprocessors than the *coff* format on System V. When you compile a program, however, the executable is still placed in the *a.out* file.

HARDWARE CONSIDERATIONS

Since Xenix runs on several different microprocessors, C programmers must account for architectural differences. The Zilog chip and all the Intel chips have 16-bit integers, while the Moto-

Mankind searched the world over for the multiuser operating system of the future.

Then IBM® chose XENIX® for the PC AT. And the future was *now*.

THE SANTA CRUZ OPERATION PRESENTS

XENIX NOW!

AN SCO PRODUCTION IN EXCLUSIVE ASSOCIATION WITH MICROSOFT CORPORATION
THE MULTIUSER, MULTITASKING PC BLOCKBUSTER "XENIX NOW!"
STARRING VISUAL SHELL • MULTISCREEN™ • MICNET • THE BERKELEY ENHANCEMENTS
AND INTRODUCING C-MERGE AS THE MS-DOS DEVELOPMENT ENVIRONMENT
FEATURING WORLD FAMOUS SCO TRAINING AND SUPPORT FOR DEALERS • END USERS • ISVs • OEMs
AND AN INTERNATIONAL CAST OF HUNDREDS OF XENIX APPLICATIONS
INCLUDING LYRIX™ AS THE UNIX/XENIX WORD PROCESSING SYSTEM
PRODUCED AND DIRECTED BY THE SANTA CRUZ OPERATION
SCREENPLAY ADAPTED BY THE SANTA CRUZ OPERATION FROM ORIGINAL STORIES BY MICROSOFT AND AT&T
IN BREATHTAKING SELECTABLE COLOR
NOMINATED FOR ★ BEST DOCUMENTATION! ★ BEST SUPPORT! ★ BEST TRAINING!
★ BEST ELECTRONIC MAIL AND NETWORKING! ★ MOST APPLICATIONS!
★ MOST COMPLETE UNIX SYSTEM!

SCO
THE SANTA CRUZ OPERATION

RELEASED FOR MOST POPULAR PERSONAL COMPUTERS.
APPLICATIONS ALSO AVAILABLE: LYRIX, MULTIPLAN®, INFORMIX®,
LEVEL II COBOL™, 3270 MAINFRAME COMMUNICATIONS.

(408) 425-7222
TWX: 910-598-4510 SCO SACZ

M MULTIUSER OPERATION SUGGESTED
XENIX WILL TURN YOUR PC INTO A REAL COMPUTER

The Santa Cruz Operation, Inc., 500 Chestnut Street, P.O. Box 1900, Santa Cruz, CA 95061 (408) 425-7222

Circle No. 243 on Inquiry Card

rola chips, as implemented by Microsoft, have 32-bit integers. The Zilog and Motorola chips have forward byte ordering, while the Intel chips have backward byte ordering, like the PDP-11. All Xenix C compilers generate code that performs software sign extension.

The Intel 8088 and 8086, and the Zilog Z8000 impose a severe memory restriction on software—processes are limited to no more than two memory segments of 64K. Motorola 68000 chips have such a large address space that programs will probably encounter memory board capacity limitations before the chip runs out of address space. That's good since Xenix does not have virtual memory, as Berkeley UNIX does.

The Intel 80286 has three addressing modes: the small model, with less than 64K of code and data; the medium model, with more than 64K of code and less than 64K of data; and the large model, with more than 64K of both code and data. Using the large model is somewhat like running a 68000. Xenix supports all three addressing models; you compile a program according to whichever one suits your needs. Of course, some processors (like the 8086) can't support all three addressing models.

SOFTWARE CONSIDERATIONS

Many users and programmers who try Xenix say, "it's just like UNIX". This is sometimes meant as a disparaging remark, implying that Microsoft has done little to improve UNIX. But the fact of the matter is the reverse is true. Users of disparate UNIX systems recognize their favorite features in Xenix because Microsoft delivers a product that is highly compatible with all the older versions of UNIX. When release 5.0 comes out, Xenix will also be compatible with System V. Unfortunately, Xenix will probably never be compatible with 4.2BSD. The only vendors I can think of who deliver systems that are compatible with more flavors of UNIX are Pyramid and CCI, both of whom offer not only System V compatibility, but 4.2BSD compatibility as well. [It should be noted, though, that Altos now offers a "Booster Pack" implementation of the Berkeley Fast File System

as an option for all its systems with at least a megabyte of RAM.]

Unlike System V from AT&T, Xenix comes with a full C development environment, including **make**, **lint**, **SCCS**, **lex**, and **yacc**. Under System V.2, these tools are available only as an extra-cost option. There are no surprises in this environment; everything works the way you would expect. Xenix systems based on the 68000 and the 80286 seem the most solid, and thus the best-suited for program development.

Both Xenix 3.0 and AT&T's System V represent significant improvements over the standard System III release because of concerted efforts on the part of both AT&T and Microsoft to clean up System III weaknesses.

The C support libraries on Xenix appear to be compiled straight from System III. This means that useful routines such as **getopt()** for command-line parsing, **bsearch()** for binary searches, **lsearch()** for linear searches, **strtok()** to find a token in a string, and **regex()** for matching regular expressions, are available. It also means that C programs on Xenix should use **strchr()** instead of **index()** and **strrchr()** instead of **rindex()**—the old routines from Version 7.

From Berkeley UNIX are included the **termlib** routines discussed in this column last August, and the **curses** library, presented last November. Programmers who can't live without Berkeley enhancements will find **vi**, the C shell, **more**, and more.

Microsoft has added several unique features not found on many other UNIX systems. Xenix 3.0 provides a set of system calls to allow user processes to share data space. Another set of system calls permits unrelated processes to communicate with each other using semaphores. Both shared memory and semaphores are similar in function, but not design, to the IPC facilities available on System V. (System V provides a unique set of message queue facilities.) Xenix provides both file and record locking; System V has neither, and 4.2BSD has only file locking. A system call named **lock** can be used to lock a process in memory in order to guarantee good response time. A similar function in System V named **plock** lets users lock and unlock text or memory segments. An-

IMPLEMENTATION VARIATIONS

Xenix	Version 7	System III	System V	4.2BSD	Description
access	access	access	access	access	
acct	acct	acct	acct	acct	
alarm	alarm	alarm	alarm	alarm(3)	
sbrk	sbrk	sbrk	sbrk	sbrk	
chdir	chdir	chdir	chdir	chdir	
chmod	chmod	chmod	chmod	chmod	
chown	chown	chown	chown	chown	
chroot	chroot	chroot	chroot	chroot	
chsize	-	-	-	truncate	
close	close	close	close	close	
creat	creat	creat	creat	creat	
creatsem	-	-	-	-	create instance of binary semaphore
dup	dup	dup	dup	dup	
dup2	dup2	-	-	dup2	
exec*	exec*	exec*	exec*	exec*	
exit	exit	exit	exit	exit	
fcntl	-	fcntl	fcntl	fcntl	
fork	fork	fork	fork	fork	
fstat	fstat	fstat	fstat	fstat	
ftime	ftime	-	-	gettimeofday	
getpid	getpid	getpid	getpid	getpid	
getpgrp	-	getpgrp	getpgrp	getpgrp	
getppid	-	getppid	getppid	getppid	
getuid	getuid	getuid	getuid	getuid	
geteuid	geteuid	geteuid	geteuid	geteuid	
getgid	getgid	getgid	getgid	getgid	
getegid	getegid	getegid	getegid	getegid	
ioctl	ioctl	ioctl	ioctl	ioctl	
kill	kill	kill	kill	kill	
link	link	link	link	link	
lock	lock	-	-	-	lock a process in memory
locking	-	-	-	flock	file and record locking
lseek	lseek	lseek	lseek	lseek	
mknod	mknod	mknod	mknod	mknod	
mount	mount	mount	mount	mount	
-	-	-	msgctl	-	message queue access control maintenance
-	-	-	msgget	-	message queue creation
-	-	-	msgrcv	-	message queue retrieval
-	-	-	msgsnd	-	message queue update
nap	-	-	-	-	sleep for short time
nice	nice	nice	nice	nice(3)	
open	open	open	open	open	
opensem	-	-	-	-	open a semaphore
pause	pause	pause	pause	pause	
pipe	pipe	pipe	pipe	pipe	
-	-	-	plock	-	lock text or memory segments
profil	profil	profil	profil	profil	
ptrace	ptrace	ptrace	ptrace	ptrace	

IMPLEMENTATION VARIATIONS

Xenix	Version 7	System III	System V	4.2BSD	Description
rdchk	-	-	-	-	check if there is data to be read
read	read	read	read	read	
sdget	-	-	-	-	attach to a shared data region
sdfree	-	-	-	-	release a shared data region
sdenter	-	-	-	-	enter a shared data region
sdleave	-	-	-	-	leave a shared data region
sdwaitv	-	-	-	-	synchronize use of shared data
-	-	-	semctl	-	semaphore access control maintenance
-	-	-	semget	-	semaphore creation
-	-	-	semop	-	semaphore maintenance
setpgrp	-	setpgrp	setpgrp	setpgrp	
setuid	setuid	setuid	setuid	setuid(3)	
setgid	setgid	setgid	setgid	setgid(3)	
-	-	-	shmat	-	attach a shared memory segment
-	-	-	shmctl	-	maintain shared memory data structure
-	-	-	shmdt	-	remove shared memory segment
-	-	-	shmget	-	create shared memory data structure
shutdn	-	-	-	reboot	
signal	signal	signal	signal	signal(3)	
sigsem	-	-	-	-	signal process waiting on semaphore
stat	stat	stat	stat	stat	
stime	stime	-	-	settimeofday	
sync	sync	sync	sync	sync	
time	time	time	time	time(3)	
times	times	times	times	times(3)	
ulimit	-	ulimit	ulimit	ulimit(3)	
umask	umask	umask	umask	umask	
umount	umount	umount	umount	umount	
uname	-	uname	uname	gethostid	
unlink	unlink	unlink	unlink	unlink	
ustat	-	ustat	ustat	-	get file system statistics
utime	utime	utime	utime	utime(3)	
wait	wait	wait	wait	wait	
waitsem	-	-	-	-	wait on a semaphore
write	write	write	write	write	

Figure 1 — *How Xenix system calls compare with those of Version 7, System III, System V, and 4.2BSD.*

other Xenix system call, **chsize**, truncates files to a certain size.

The table in Figure 1 shows all the system calls available under Xenix, compared to related system calls on Version 7, System III, System V, and 4.2BSD. Note that Xenix is most closely related to Version 7—its kernel is V7-based, with library routines taken from System III and Berkeley UNIX. Xenix also keeps track of time the same way Version 7 does. Some system calls from System III have been added, such as **ulimit** and **ustat**.

My prediction is that Xenix 5.0 will retain as many enhanced V7 features as possible from release 3.0—despite its commitment to be System V compatible. Those Xenix enhancements that don't inhibit compatibility will be kept. After all, why throw away utilities whose mettle already has been proven?

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

UNIX™ & TODAY

The Magical Mixture

Are you dizzy with software application backlogs? You need a straightforward cure—**TODAY**, the **UNIX** fourth-generation language that is breaking time zones in software development.

TODAY runs under **UNIX** to provide application portability across a wide range of machines and with a range of databases. Your application developers will enjoy a COMPLETE application building environment.

Features:

- Non-procedural specification with powerful recursive logic and Decision Tables
- Synonyms, Menus, Prompts, Helps and Defaults for streamlined definitions
- Tailoring control allowing multi versions of a base application
- Screen Painter
- A Report Generator which includes a Painter
- Push-button Self-documentation
- Audit Trails
- Source-code security through run-time only configurations
- Developed Applications instantly portable across **UNIX**-based machines

Visit our booth—No. 1312—at INTECH 85, Moscone Convention Center, San Francisco, CA, Aug. 27-29.

Call **TODAY** —(408) 727-4464

Or write: **bbj Computer Services Inc.**
2946 Scott Boulevard
Santa Clara, CA 95054

Circle No. 234 on Inquiry Card

THE UNIX GLOSSARY

Office talk

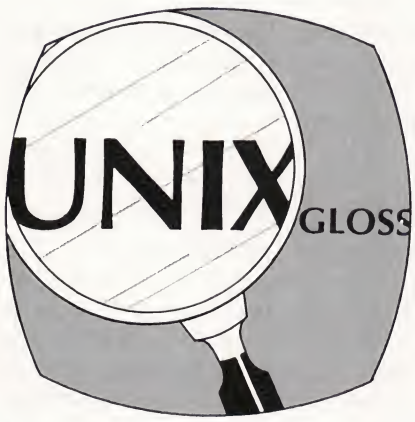
by Steve Rosenthal

Note: only those aspects of these terms that pertain to office automation are included in this listing.

administrative workstation—a term used in connection with multistation office computer systems to refer to a system intended for use by a secretary or other support person. The complement to this unit is the *executive workstation* used by a manager or "professional". Most UNIX systems make all system resources available at each terminal, so the difference between these two workstations in a UNIX context is more likely to be one of styling and configuration. See also *executive workstation* and *professional workstation*.

big five—as applied to software, the "big five" refers to spreadsheets, word processing, database managers, graphics, and communications. These five applications are the ones that are most commonly found in office and professional use, and hence are included in most integrated packages.

cluster—a group of terminals or small systems that share common resources, particularly mass storage. They also often share



fast printers. Multiuser UNIX systems are often treated as clusters of independent workstations rather than as single systems with multiple access points.

COM—an abbreviation for Computer Output Microfilm. This refers to the process whereby a computer sends records directly to microfilm without first printing a full-sized image on paper or film. Several UNIX text preparation programs have been adapted for COM output.

cut and paste—the edit operation by which material is removed from a document and then repositioned in another location or another document. It is analogous to cutting apart a paper copy of a document and pasting it back together in a different order. Often, the term specifically applies to efforts accomplished visually on a screen using a mouse or other pointing device. Similar operations performed with commands from a keyboard are referred to as "text movement" operations.

dedicated—a computer system that has been configured specifically for a particular use, such as word processing. UNIX systems tend to allow multifaceted use, even if they are ultimately utilized to run a single application.

disk server—a shared disk storage unit that provides storage for multiple systems, but leaves to each system the task of managing its own storage. Disk servers, or more sophisticated file servers, are often used in networks of small-scale UNIX systems.

editor—a program intended for the creation and revision of textfiles. Editor programs are usually simpler than full word processing systems since they generally lack extensive formatting commands. A UNIX system normally includes one or more program-oriented editors, but for office applications

GLOSSARY

most users prefer full-featured word processing systems.

E-mail—a common shortened form of "electronic mail" used to refer to message store-and-forward systems that service multiuser computers, local area networks, or wide-area networks that offer information utilities. UNIX has a built-in E-mail facility (accessed by using the **mail** command), but several firms offer proprietary alternatives that are more powerful, more flexible, and/or easier to use.

environment—the combination of the operating system, graphics extensions, desktop managers, network programs, and so on that a program runs under. Although quite workable for scientific and systems use, most developers find that the UNIX environment needs extensions or overlays if it's to provide for flexible office use.

executive workstation—a single-user terminal or desktop computer intended for use by people who get assistance with their clerical chores from a support person (who uses an *administrative workstation*). Executive workstations may have fancier cases than their administrative counterparts, but they typically feature smaller keyboards, and often have no printer (or have printers that only are suited to quick rough drafts). See also *administrative workstation* and *professional workstation*.

file server—on a network of connected computers, "file server" refers to a network unit that provides disk file storage and retrieval. A file server might be a computer with a hard disk, or a special disk unit designed for this purpose. Unlike a disk server, which responds to physical positioning commands but leaves disk allocation and management up to the connected systems, a file server provides services and responds to requests specified in terms of logical files.

forms package—a database or database-like program with a user interface that imitates paper forms. Many businesses have learned that a transition to computers can be made easier if existing forms and procedures are retained. Forms processors are still rare under UNIX, due in part to the lack of a standard graphic environment in which form images can be drawn.

icon—a symbolic picture meant to convey possible actions the user can take or status information the user can respond to. Icons are often used to form pictorial menus that provide a series of choices from which the user may choose. When combined with desktop mice pointing devices (for cursor control), icons are increasingly popular as components of visual front ends for UNIX as well as other operating systems.

integrated—said of programs that share data and have a common operator interface. Integrated programs also can be invoked from one another without the need to first go back to the system or shell level. Because UNIX is difficult for novice users, many new UNIX office software packages that offer complete user environments are beginning to surface.

OA—the common abbreviation for office automation, the application of computerized and other automatic techniques to office work.

OCR—an abbreviation for optical character recognition, the process by which a computer can read printed or lettered material. OCR units had been expected to form an important link in the automated office, but the tenden-

cy to replace typewriters with computers has meant that the reading of typed documents into a computer system has grown less important. UNIX has no special provisions for OCR, leaving any support for it to special drivers and application programs.

power user—a common industry term for users of desktop systems who are knowledgeable and demanding about performance in particular applications, but who may or may not have any broad or deep understanding of systems or computers in general. Increasingly, the term "power user" also has come to imply someone with the ability to requisition or purchase sophisticated desktop systems.

professional workstation—a terminal or desktop system intended for use by people who primarily do their own work, including the final transfer of results to some suitable output form. This contrasts with *administrative workstations*, designed primarily for those who do clerical support work, and *executive workstations*, designed for people who have their clerical work done for them. Most UNIX workstations fall into the professional workstation category. See also *administrative workstation* and *executive workstation*.

QWERTY—a term for the most popular arrangement of keys on English-language keyboards (QWERTY being the first six letters in the upper row of alphabetic characters). Many UNIX programs assume the presence of a QWERTY keyboard, but some systems can take advantage of hardware or software that allow them to switch to more efficient arrangements.

raster printing—the type of printing used by laser printers and many other new print technologies. In raster printing, the output device sweeps over the entire page, creating an array of printed dots. Under this scheme, the system or device translates the output that has been requested into an appropriate series of dots—a process that is relatively resource-intensive and slow (it usually becomes the main job of the printer controller that manufacturers add to basic raster print engines). Most UNIX print utilities assume output from a character printer or a dot matrix printer, so special drivers are generally needed to use them with raster-based laser printers and typesetters.

resource sharing—the ability to provide access to disk memory, printers, communications lines, or other useful subsystems within a cluster, network, or multiuser system. UNIX systems are generally designed to facilitate sharing, using such features as the ability to spool output to printers and utilize path commands for accessing disks.

secretarial shift—the normal shift arrangement that changes the meaning of characters only while the shift key is held down (unless the shift is locked). This is in contrast to the shift-in, shift-out pattern used in data communication, where a separate shift character leaves a shift in effect until it's reversed. The term "secretarial shift" is also used to

indicate a shift to the upper legends on a keyboard, including the characters listed above the numbers on a keyboard and the capital versions of alphabetics.

server—a station on a network that handles special chores, such as disk storage, printing, or communications. A dedicated server handles only its special chore, while a shared workstation allows both shared and local use of its resources. UNIX networks often use disk or file servers to provide extra storage because of the large number of utilities and large amount of reference material it takes to make up a full UNIX environment.

spelling checker—a program that checks a textfile to ensure that all its words are included in an online dictionary of correct spellings. Most UNIX systems provide the **spell** utility for this function, but most word processing packages have their own spelling checkers that include more complete dictionaries and are better integrated with the word processing systems they attend.

spelling correction—a program or feature that suggests the most likely word or series of words to substitute for strings found in a textfile that do not match any word in the online spelling dictionary. Although this feature is not included in the standard UNIX **spell** program, spelling correction is now a common feature or option for other word processing packages.

turnaround document—a document created by a computer system in a form suited for automatic entry at a later time, usually via OCR (optical character recognition). Some office forms and industrial tags are now designed in this fashion.

voice mail—the capacity for recording and delivering stored voice messages through a centralized system, typically one that has been built into a telephone system. Most newer systems store the messages in digital form and can also accept supplementary commands entered with tone telephones.

work group—a collection of people who work on common projects and who might be expected to share a large number of files and programs on a computer system. UNIX includes facilities for defining such groups.

workstation—a fancy word for a computer or terminal best suited for use by one person at a time. UNIX can now be used to drive affordable workstations, as a bewildering range of products now appearing on the market testifies.

WYSIWYG—an abbreviation for "what you see is what you get", a common way of characterizing word processors, typesetters, and other systems that produce as output a copy of what you see on the screen.

Comments, questions, corrections? Please send them to Rosenthal's UNIX Glossary, Box 9291, Berkeley, CA 94709.

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

RECENT RELEASES

XENIX PAYROLL

MCBA, Inc. has developed a payroll package, written in RM/Cobol, that runs on the Altos 986 under Xenix and combines the company's accounts payable, accounts receivable, and general ledger programs into a modular Cobol accounting system.

Employees can be accessed either by number or partial name. The program automatically generates a standard payroll for full time and part time employees. It allows numerous taxable or non-taxable earnings and deductions, keeps tax tables, processes manually written checks and adjustments, and allows manual override.

The price of MCBA's RM/Cobol payroll package ranges from $900 to $1500. Reseller licenses are available.

MCBA, Inc., 2441 Honolulu Ave., Montrose, CA 91020, 818/957-2900.

Circle No. 277 on Inquiry Card

DATAMEDIA ADDS COMMUNICATIONS AND STORAGE PROCESSORS

The XCP UNIX communications processor and XSP UNIX storage processor are available from Datamedia for its 932 supermicrocomputers. The two new processors are both configured with a 10 MHz 68000 CPU and half a megabyte of shared system memory.

The XCP offloads communications tasks from the 932's CPU. Two RS422 serial lines, each supported with local DMA, provide either asynchronous or bit and byte-synchronous communications. The processor also supports four RS232C asynchronous ports, as well as one serial and one Centronics parallel printer interface. It allows data transfer rates of up to 256 kilobits per second.

The XSP offloads file management functions from the 932's CPU. It supports dual DMA ports, one to the disk bus, the other to the system bus. It also supports command chaining and performs disk caching and overlapping seeks for up to four disk drives.

The XCP and XSP processors are each priced at $3250.

Datamedia Corp., 491 Amherst St., Nashua, NH 03063, 603/886-1570.

Circle No. 276 on Inquiry Card

TCAT IS NO DOG

The Test Coverage Analysis Tool for the C language (TCAT/C) from Software Research Associates (SRA) employs a feedback loop that tells programmers both what has and what hasn't been tested. Its aim is to discover errors early, with a resulting savings in code development costs.

TCAT/C has a C language instrumenter, a special "runtime" program to collect test coverage data and write tracefiles, and a coverage analyzer to produce the coverage numbers. It also includes an interactive runtime package, a set of utilities for manipulation of tracefiles, and a tracefile archive.

TCAT/C is available for VAX systems running UNIX and VMS. A multiuser single-CPU license is $2500; a multiuser, multi-CPU license is $9000. SRA says it can port TCAT to virtually any operating system on a special order basis.

SRA, 580 Market St., San Francisco, CA 94104, 415/957-1441.

Circle No. 275 on Inquiry Card

SMALL LISP ARTICULATES FILTERS, NETS, FRONT-ENDS

The UniLISP kernel requires less than 32K of program space, allowing it to be used concurrently to build interpretive filters, networks, and natural language front-ends. UniLISP works under **vi** and other editors, can link and unlink files and pipes, offers concurrent communications between kernels, has **sort** and **reverse** extensions, and is portable to a wide variety of hardware. It is available for the DEC Pro-300 Series, IBM PC/AT, and Intel 310.

Contact Grant Ricketts, 7623 Leviston St., El Cerrito, CA 94530, 415/527-1438.

Circle No. 274 on Inquiry Card

STRETCH PERFORMANCE WITH THE RACK

Ann Arbor Terminals is offering a 19-inch rack panel accessory to its XL terminals. The option is part of the company's efforts to build terminals that stand up to heavy industrial environments. The XL units can operate in ambient temperatures between 10° and 40° C (50° and 104° F).

For receive-only applications,

Dataproducts 8250 color dot matrix printer

these terminals can be bought *sans* keyboard. Display sizes can vary between 18 to 30 lines of 80 characters, with five modes. There is also an invisible cursor mode.

For process control applications, the company's GXL and GXL+ terminals provide 768-by-600 dot resolution.

All terminals have 15-inch screens, with a choice of white, amber and green phosphorescent display.

Prices start at $1145; the rack panel is an extra $75.

Ann Arbor Terminals, Inc., 6175 Jackson Rd., Ann Arbor, MI 48103, 313/663-8000.

Circle No. 273 on Inquiry Card

SYSTEM FOR FOUR IS $15,995

Micro Craft Corp. has announced a four-user 68000-based UNIX system with a 50 MB hard disk for $15,995. The Dimension 68000 system has a megabyte of RAM and two 400K drives. It runs on the UniPlus+ operating system, UniSoft's version of System V. The system comes with C, 68000 assembler, and Snobol; **ed, sed,** and **vi**; and **nroff, troff, tbl,** and **eqn**. There are four serial ports, six expansion slots, a disk controller for as many as eight drives, an RS232 port, and a graphics port.

The system is also offered in other, more modest configurations: with no terminals and a 20 MB disk ($9995); no terminals and a 50 MB disk ($12,895); and four terminals with a 20 MB disk ($14,995).

Micro Craft Corp., 4747 Irving Blvd., Suite 214, Dallas, TX 75247, 214/630-2562.

Circle No. 272 on Inquiry Card

COLOR PRINTER IS QUICK CHANGE ARTIST

The Dataproducts 8250 color dot matrix printer has what the company calls an integrated paper handling system that allows setup change from continuous forms to automatic sheet feed in 30 seconds. A combination of continuous forms, cut sheets, envelopes, or labels can be stored. A 16-character LCD tells the user what is going on.

The 8250 operates in draft mode at 400 characters per second. A maximum 132-character line can be printed, and graphics are realized at a resolution of 240 by 240 dots per inch.

The retail price of the 8250 is $2650. OEM pricing is $1900 in quantities of 100.

Dataproducts Corp., 6200 Canoga Ave., Woodland Hills, CA 91365, 213/887-8451.

Circle No. 271 on Inquiry Card

POWERFUL IMAGERY

The Talus Image Management System converts an IBM PC/AT into an image processor. Written in Lattice C for UNIX portability, the system is positioned as an office automation product, and is initially being marketed to OEMs and VARs.

Images of documents and diagrams, forms, news clippings, or notes are scanned at a rate of 11 seconds per page, and digitized at 200 by 200 pixels per inch. Images can be sized, edited, manipulated, printed, communicated, and printed. The system performs image compression, reconstruction and scaling through hardware rather than software.

The complete system includes the T20/20 PC Image Scanner ($4995); T20/22 Image Compression Board ($1295); and TalScan Image Management Software ($595).

Talus Corp., 985 University Ave., Los Gatos, CA 95030, 408/354-5322.

Circle No. 270 on Inquiry Card

MICROPROCESSOR TOOLS RUN ON VAX/ULTRIXen

Boston Systems Office (BSO) has made available its family of microprocessor development tools for DEC VAXen running the UNIX-based Ultrix operating system. BSO's compilers, debuggers, and assemblers support more

than 40 microprocessors, including the Intel 80186, 8087, and 8051, the Motorola 68000/010 and 6809, and the Zilog Z80 and Z8000.

The tools also operate on Digital's VMS, UNIX, and RSX systems. Prices for the Ultrix tools start at $4000.

BSO, 128 Technology Center, Waltham, MA 02254-9164, 617/894-7800.

Circle No. 269 on Inquiry Card

FLOATING POINT ADDED TO CCI LINE

Computer Consoles, Inc. (CCI) has added a floating point/math accelerator to its Power6 line of central processors. Execution rates are said to be on the order of 7500 Whetstone KIPS for single precision. The Power6/32 central processor with floating point/math accelerator runs on both Berkeley 4.2 and System V.

The basic Power6/32 CPU, with 4 MB of memory, tape and disk controller, terminal, battery backup, asynchronous controller and UNIX license costs $146,000. The floating point/math accelerator option is $16,500.

CCI, 97 Humboldt St., Rochester, NY 14609, 716/482-5000.

Circle No. 268 on Inquiry Card

LITTLE TOWER OF POWER

NCR has added a relatively low-cost version of its Tower system, the Minitower, measuring 24 inches high, 5 inches wide, and 25½ inches deep. The MiniTower is based on the Motorola 68010 and runs UNIX System V. It comes with half a megabyte of memory, a 25 MB hard disk, a 1 MB floppy drive, eight RS232 ports, and UNIX software; it has complete software compatibility with other NCR Tower systems.

This system is designed for use as either a standalone entry level system or as part of a large distributed network. It carries a base price of $6995. A UNIX extension module costs another $755.

NCR Corp., Dayton, OH 45479, 513/445-2075.

Circle No. 267 on Inquiry Card

UNIX VIABLE ON LAN

ViaNetix has developed a UNIX version of its ViaNet local area network (LAN). There are several configurations available, allowing DOS and UNIX-based systems to co-exist in a network. With a virtual terminal option, UNIX-based systems can act as a file server to DOS-based micros.

ViaNet acts as a bridge between the operating systems and several popular LAN protocols. It supports Ethernets, IEEE 802.4 token bus and 802.5 token ring implementations, as well as some proprietary LANs. Microprocessors supported include the Intel 808X, 80188, and 80X86, and Motorola's 680X0 chips. It uses a hierarchical, tree-structured directory system. Its DOS implementation appears as a superset of the tree structure, with each system a subdirectory of the network. The name of the DOS network is "Z:", reflecting a syntax similar to that with which DOS users are acquainted. In the UNIX implementation, ViaNet appears as a directory, called "net", beneath the UNIX root directory.

ViaNetix, Inc., 5766 Central Ave., Boulder, CO 80301, 303/440-0700.

Circle No. 264 on Inquiry Card

INTERACTIVE E-MAIL

Network Technologies International, a subsidiary of NETI Technologies, has developed eForum, an interactive electronic conferencing system. Users are supplied with communications software, and access eForum through General Electric Information Services Company (Geisco). The software automatically documents the time, date, and sender of messages. The eForum command set allows scanning and display of information in conference sessions, and a menu-driven word processor provides text manipulation capabilities.

Network Technologies International, Inc., The Arbor Atrium Building, 315 W. Huron, Ann Arbor, MI 48103, 313/994-4030.

Circle No. 266 on Inquiry Card

MOTOROLA MARKETS PHILON COMPILERS

Philon's Fast/Compilers are now available from Motorola for its Four-Phase Series 2000 and 6000 systems. The compilers are of modular design, based on an intermediate code called Phi-Code. The three compilers sold through Motorola include: Fast/C ($950), Fast/Basic-C ($600), and Fast/Basic-M ($2,200).

Philon, 641 Ave. of the Americas, New York, NY 10011, 212/807-0303.

Circle No. 231 on Inquiry Card

UNIX IN THE OFFICE

Continued from Page 27

distributed and dispersed systems come down in price, improve in reliability, and offer good interfaces that allow system administration activities to be performed by non-technical staff members.

There's a second problem that is more far reaching and much harder to solve. On a tour of a company with an exceptionally good office automation system in which each staff member had a workstation, the only typewriter to be found was in the shipping department. Taking note of that, the justifiably proud executive giving the tour said as an aside, "Our company would grind to a halt if the electronic mail system crashed."

A crash of this kind could be the result of anything from a thunderstorm to the failure of an air conditioning unit to a user error the system could not compensate for. This fragility points to human factors considerations that providers of office systems would do well to address. The potential implications are as serious as the burning of the library at Alexandria or the devastation of English monastic libraries in the fifteenth century. But, barring a catastrophe of this kind, the future of OA looks good and is already a topic of discussion and debate.

The same executive who commented on his electronic mail system also observed two other problems that have yet to be addressed. First, a CRT screen is humorless: one has to be very careful in the wording of messages that are sent over an electronic mail link because—absent a face, vocal intonation, and body English—what was meant as a joke may well be received as in insult. There is, therefore, a new, developing symbology:

:-) means "I'm kidding"
:-(means "I'm sad"
:-[means "I'm angry"

Another problem this executive noted is that the ease with which people with good electronic mail systems can use mailing lists makes it extremely easy for users to send *much* mail to *many* recipients. Hence, electronic mail has a great potential for causing a megafold multiplication of junk mail.

> Electronic mail has a great potential for causing a megafold multiplication of junk mail.

LOOKING AHEAD

Articles in current periodicals speculate on new developments in OA. One key issue is whether or not a universal workstation will become a reality. A look at history shows it will. When this author was growing up, the 78-rpm turntable was the default in most homes. But then some teenagers in the neighborhood acquired new, smaller 45-rpm record players. For reasons of both size and speed, 78s could not be played on a 45 turntable and vice versa. But a breakthrough finally occurred—a turntable that accommodated 78s, 45s, 33s, and even the short-lived 16-rpm records became available. A similar development will happen in the workstation realm when software developers allow their imaginations to make use of the features included in current hardware. Advances in firmware already indicate this trend.

Whether or not there is a future for lap-sized computers is another issue hotly debated in current periodicals. These products *do* have a future—as yet another evolution in the music industry shows. A few years ago, the "ghetto blaster" was the status symbol of portable stereos, whether on city streets, at the beach, or in the mountains. The miniaturized, pocket-sized player with high-quality earphones is a comparable status symbol now. A necessary feature of viable lap-sized computers is a provision for plugging them into larger systems when they aren't being "ported" about.

Whether we call it "office automation", "office applications", "office administration", or something else, the use of computers in the business community is here to stay. Time will show that this is a positive development for human life. In a television interview a few years ago, Grace Murray Hopper was asked if she thought computers would dehumanize humanity. She replied, with a characteristic sparkle in her eyes, "I remember when they said that about the telephone."

Tanya M. Joyce is a technical editor with over 10 years of experience with the software we now consider part of office automation. Her company, Leading Edge Studios, specializes in documentation management, technical writing, and visual arts. Comments and questions may be addressed to: Leading Edge Studios, P. O. Box 884311, San Francisco, CA 94188. ■

Background material for this article was presented at the 1985 UniForum Conference by T. A. Dolotta, Senior Vice President for Technology, Interactive Systems Corporation; and John R. Mashey, Manager of Operating Systems, MIPS Computer Systems.

UNIX™ SYSTEM V. SOFTWARE THAT STACKS UP.

High-quality application software written for **UNIX** System V is available now. And new business programs are continually being developed.

That means increased market opportunities for VARs and ISVs. And end users can be certain their investment in a **UNIX** System V-based computing system is a smart, sound one.

Quantity and quality

The market for products that are based on **UNIX** System V has grown significantly over the last two years. There are now hundreds of packages written for **UNIX** System V. And this growth will continue.

Quality applications—general business packages and development tools—are now available for the AT&T **UNIX** PC. And AT&T is putting together even more hardworking, industry-specific software to run on AT&T 3B2 and 3B5 Computers.

With so many packages available, VARs can offer more turn key systems based on **UNIX** System V to an ever-growing customer base. ISVs can sell software to even more customers. And end users can invest in a **UNIX** System confident there is plenty of quality software available.

Software you can bank on

UNIX System V has gained acceptance as a powerful, versatile computing standard. More hardware vendors like NCR, Altos, Motorola, Perkin-Elmer and Sperry are joining AT&T in offering products based on **UNIX** System V.

And we've introduced the System V Interface Definition. Software written under the Interface Definition can run on current and future releases of AT&T's **UNIX** System V, as well as various System V derivatives offered by AT&T licensees. So there will be an even larger, more comprehensive base of portable software from which to choose.

Our comprehensive **UNIX** System V Software Catalog lists a full range of packages that run under **UNIX** System V. For end users it's a reference guide to the programs available. And for developers it's a smart way to ensure packages will have even greater exposure to the growing **UNIX** System V market.

To learn more about **UNIX** System V market opportunities, order the **UNIX** System V Software Catalog—at $19.95 plus tax. Call 1-800-432-6600 and ask for Operator 397.

UNIX System V. From AT&T. Consider it standard.

AT&T
The right choice.

ELECTRONIC MAIL

Continued from Page 43

nal subject (unless it already starts with "Re:"). The reply should also include a field that effectively reads "In-Reply-To: your message of such and such a date, message-id: xxx".

Message forwarding. Mail packages should provide an easy method for forwarding one or more messages to a new list of recipients (presumably people who are not on the original distribution list).

Distribution lists. Mail packages should provide both private and global distribution lists.

Automatic forwarding. Mail sent to one address should also be easy to forward to another address at the recipient's request. This can be useful when tracking a user who has changed systems or accounts, or when delivering mail addressed to generic accounts, like "bugs" or "system".

Delivery to a program. Users should be able to "receive" each mail message by invoking a program, which is to say that users should be able to specify a program for the mail delivery program to call whenever messages are received. Such a capability facilitates the implementation of news and message systems that don't require modifications to the mail system itself.

Hierarchical indexed message folders. Users should be able to file messages into one or more folders that can be tagged in some fashion for later reference. This calls for treating messages as structured database objects.

Formatted message display. Users should be able to tailor the formatting of message displays. Among the things they should be able to specify are the order and position on the screen of each field, the fields that are not to be displayed, the place where extra, unspecified fields should be displayed, the procedure to follow when dealing with long lines, and so forth.

Listing for easy scanning. The mail system should provide a message compact display containing one or two-line descriptions of both newly received messages and pre-existing ones. The format of this display should be user-selectable, using a facility similar to **printf**.

Network interface. The mail system should interface with multiple heterogeneous networks and delivery mechanisms.

Shared folders. A mechanism should exist to allow projects or groups of users to share working folders, preferably by way of an automatic delivery system using a generic mail alias.

At present, I know of no mail system, under UNIX or any other operating system, that supports *all* of these features. The two most common mail systems under UNIX, usually referred to as Bell mail and Berkeley mail, provide only a few of these features. The MH (Mail Handler) system from The Rand Corporation, supports many more—but not all.

TROUBLE SPOTS

There are many reasons why electronic mail systems fail to gain general acceptance and, by extension, fail to deliver large gains in office efficiency.

The most common flaw is

limited access. For electronic mail to be useful, *everyone* must have direct access to it. If you want to send a message to 10 people, you will end up sending paper copies even if only one of the 10 doesn't have access to electronic mail, and your time savings thus will have been eroded. Access does not necessarily mean a terminal on each desk, but that is certainly the ideal situation. Anything short of this is an obstacle to the efficient use of electronic mail.

The second most common reason that electronic mail systems fail to take hold is a lack of system reliability. If mail doesn't always get delivered in a timely fashion (within a few minutes if it's local, "by tomorrow" if it's long distance), or if the system the mail package runs on is often down, users will not use electronic mail.

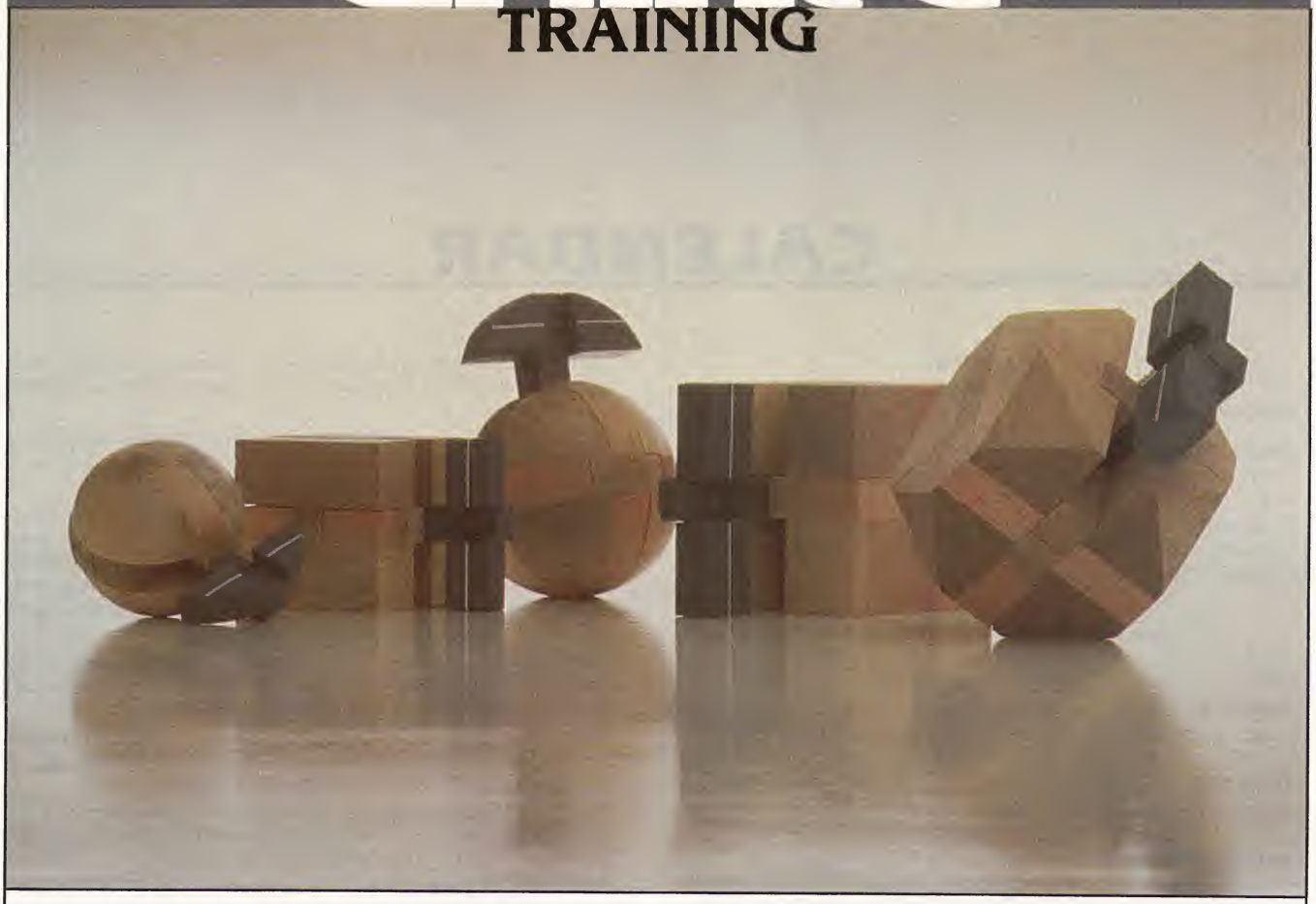
After these two primary failures, there are a number of annoyances that—when taken together—might lead to the failure or inefficient use of electronic mail. If, for instance, the mail package is not easy to use, or has a less than satisfactory system for delivery notification, it may not be used. There must be some easy mechanism for finding the electronic addresses of users, whether they're local or long distance. The imposition of rules about what may and may not be discussed via electronic mail is another factor that may inhibit the system's use. The monitoring of mail (which is *very* common) can be another chilling factor that will lead users to rely on telephones they can trust to be private.

3Com Corporation (a company that offers an Ethernet product) attempted from its inception to ensure that access to its electronic mail system would be as open as possible by including everyone's system ID on the company phone list. Enough terminals were installed to allow everyone reasonable access to the system (which initially was only a small UNIX box). Each new employee was given an account, a packet of literature on using the system, and an orientation on using electronic mail. Paper memoranda was kept to a minimum, and extended discussions on any number of topics were common.

Electronic mail is central to effective office automation. Before buying a new system, place effective electronic mail facilities high on your checklist of desirable features. Purchase reliable hardware and networking technology that can grow along with your needs. And, whatever you do, make sure you don't adopt policies that restrict mail usage. Instead, encourage the use of electronic mail by giving *everyone* access.

Prior to taking his current post as Engineering Manager at Silicon Graphics, Inc., Bruce Borden developed the IP/TCP package for Excelan's Ethernet board and helped found 3Com Corporation. He moved to 3Com from Rand Corporation where, as director of the company's Information Systems Laboratory, he authored the MH (Mail Handler) system. His first experience with UNIX came 11 years ago when he set up Version 4 on Harvard University's undergraduate computing system. Mr. Borden currently directs development work on Silicon Graphic's high performance color workstation. ■

UNIX™ & 'C' TRAINING



MANY UNIX-BASED SYSTEMS ONE UNIX TRAINING COMPANY

The Computer Technology Group provides the UNIX training solution. Training to fit the complexities of your UNIX-based system.

Three factors make the Computer Technology Group the experts in UNIX and 'C' language training:

- Experience, through training thousands of students worldwide in live seminars, with thousands more using our video training at their own locations.
- Extensive Curricula Supporting All UNIX Versions, creating a client base of manufacturers, software developers and end users.
- Quality of Instruction, with instructors and course developers who are experts in teaching UNIX and 'C', as well as in designing and implementing a variety of UNIX-based systems.

ONE UNIX TRAINING COMPANY MULTIPLE DELIVERY SYSTEMS

Whether you're training two, 200, 2000...you can select the most efficient and economical training solution for your unique environment:

- Public Seminars offered in major cities throughout the world.
- On-Site Seminars for training customzied to your system and to specific groups within your organization.
- Video-Based Training for consistent training that is always available at your location.
- Interactive Videodisc Training, which dynamically tailors courses to the individual—from novice to expert programmer.

ASK FOR OUR 48-PAGE COURSE CATALOG, WHICH PROVIDES:

- Comprehensive course outlines
- Course prerequisites
- Curriculum recommendation for multiple audiences
- Guidelines for cost-effective training media selection
- Current seminar schedule

CALL (800) 323-UNIX or
(312) 987-4082 in Illinois

™ UNIX is a trademark of Bell Laboratories.

COMPUTER TECHNOLOGY GROUP

Telemedia, Inc.

310 S. Michigan Ave., Chicago, IL 60604

Circle No. 252 on Inquiry Card

CALENDAR

EVENTS

SEPTEMBER

September 18-20 National Expositions Inc., New York. "UNIX EXPO". Contact: Don Berey, 14 W. 40th St., New York, NY 10018. 212/391-9111.

September 26-28 8th Northeast Computer Faire, Boston. Contact: Computer Faire, Inc., 181 Wells Ave., Newton, MA 02159. 617/965-8350.

TRAINING

JULY

July 1-2 Computer Technology Group, London: "Shell Programming". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, IL 60604. 800/323-UNIX.

July 3-5 Computer Technology Group, London: "Using Advanced UNIX Commands". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, IL 60604. 800/323-UNIX.

July 8-9 Interactive Systems Corp., Santa Monica, CA: "Advanced Commands for Programmers". Contact: Claire Donahue, 2401 Colorado Ave., 3rd floor, Santa Monica, CA 90404. 213/453-8649.

July 8-12 Computer Technology Group, San Francisco: "C Language Programming". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, IL 60604. 800/323-UNIX.

July 8-12 Computer Technology Group, London: "UNIX Internals". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, IL 60604. 800/323-UNIX.

July 8-12 Information Technology Development Corporation, Cincinnati: "INFORMIX Relational Data Base". Contact: ITDC, 9952 Pebbleknoll Dr., Cincinnati, OH 45247. 513/741-8968.

July 8-12 Computer Technology Group, Dallas: "C Language Programming". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, IL 60604. 800/323-UNIX.

July 9-11 Computer Technology Group, Washington, D.C.: "UNIX Administration". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, IL 60604. 800/323-UNIX.

July 9-11 Computer Technology Group, New York: "UNIX Administration". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, IL 60604. 800/323-UNIX.

July 9-12 Integrated Computer Systems, Los Angeles: "UNIX: A Comprehensive Introduction". Contact: Integrated Computer Systems, 6305 Arizona Pl., P.O. Box 45405, Los Angeles, CA 90045. 213/417-8888.

July 10 Bunker Ramo Information Systems, Trumbull, CT: "UNIX Marketing". Contact: Bunker Ramo, Trumbull Industrial Park, Trumbull, CT 06611. 203/386-2223.

July 10-12 Interactive Systems Corp., Santa Monica, CA: "UNIX Architecture: A Conceptual Overview". Contact: Claire Donahue, 2401 Colorado Ave., 3rd floor, Santa Monica, CA 90404. 213/453-8649.

July 15-16 Computer Technology Group, Washington, D.C.: "Advanced C Programming Workshop". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, IL 60604. 800/323-UNIX.

July 15-16 Computer Technology Group, San Francisco: "Shell Programming". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, IL 60604. 800/323-UNIX.

July 15-16 Computer Technology Group, New York: "Advanced C Programming Workshop". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, IL 60604. 800/323-UNIX.

July 15-16 Computer Technology Group, Dallas: "Shell Programming". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, IL 60604. 800/323-UNIX.

July 15-16 Interactive Systems Corp., Santa Monica, CA: "Using Ten/Plus". Contact: Claire Donahue, 2401 Colorado Ave., 3rd floor, Santa Monica, CA 90404. 213/453-8649.

July 15-17 Computer Technology Group, London: "UNIX Administration". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, IL 60604. 800/323-UNIX.

July 15-19 Bunker Ramo Information Systems, Trumbull, CT: "Introduction to UNIX". Contact: Bunker Ramo, Trumbull Industrial Park, Trumbull, CT 06611. 203/386-2223.

July 15-19 Zilog Education and Training Department, Campbell, CA: "UNIX Users Seminar". Contact: Kay Ferrell, 1315 Dell Ave., Campbell, CA 95008. 408/370-8000.

July 15-19 Information Technology Development Corporation, Cincinnati: "UNIX for End Users". Contact: ITDC, 9952 Pebbleknoll Dr., Cincinnati, OH 45247. 513/741-8968.

July 16-19 Integrated Computer Systems, Los Angeles: "Programming in C: A Hands-on Workshop". Contact: Integrated Computer Systems, 6305 Arizona Pl., P.O. Box 45405, Los Angeles, CA 90045. 213/417-8888.

July 17-19 Interactive Systems Corp., Santa Monica, CA: "Customizing Ten/Plus". Contact: Claire Donahue, 2401 Colorado Ave., 3rd floor, Santa Monica, CA 90404. 213/453-8649.

July 17-19 Computer Technology Group, San Francisco: "Using Advanced UNIX Commands". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, IL 60604. 800/323-UNIX.

July 17-19 Computer Technology Group, Washington, D.C.: "Advanced C Programming Under UNIX". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, IL 60604. 800/323-UNIX.

July 17-19 Computer Technology Group, Dallas: "Using Advanced UNIX Commands". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, IL 60604. 800/323-UNIX.

July 17-19 Computer Technology Group, New York: "Advanced C Programming Under UNIX". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, IL 60604. 800/323-UNIX.

July 17-19 Digital Equipment Corp., Boston: "Comprehensive Overview of the UNIX Operating System". Contact: Digital Education Resources, 12 Crosby Drive, Bedford, MA 01730. 617/276-4949.

July 22-23 Computer Technology Group, London: "Advanced C Programming Workshop". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, IL 60604. 800/323-UNIX.

THE FIRST INDUSTRIAL STRENGTH SUPERMICRO

The Universe 2400 is the first 32-bit supermicro computer built for demanding environments.

RUGGED VME PACKAGING

The tower model Universe 2400 has a steel enclosure, tough plastic outer panels, and a steel pedestal. VME-standard PC boards with screw-in mounts and gas-tight connectors help handle high temperature, vibration, and shock.

THE 32-BIT PATH TO 68020

With a 32-bit, 4KB cache and a 12.5MHz 68000 microprocessor, Universe 2400 runs with no wait states at 1.25 MIPS. And its 40MB, 32-bit VMEbus supports full 32-bit data transfers to 32-bit memories and disk channels, and provides a clean path to the 68020.

ISO/OSI/MAP STANDARD NETWORKING

Our 7-layer ISO-standard UniverseNet™ ties Universe systems into multi-vendor networks including Ethernet (802.3) and the General Motors MAP network (802.4), making the 2400 an ideal Computer-Integrated Manufacturing (CIM) platform.

UNIX SYSTEM V AND REAL TIME

UN/System V (derived from UNIX System V under AT&T license) gives you application compatibility, plus powerful real-time extensions, contiguous files, shared data, and record/file locking.

UNDER $6,000, QUANTITY 100

That includes 12-slot VME chassis, 512KB 32-bit RAM, 4 serial ports, 68000-based SCSI/SASI channel controller, 20MB Winchester, and floppy, in tower, rack, wall, or table mount models. For details, contact Charles River Data Systems, 983 Concord St., Framingham, MA 01701, (617) 626-1000.

CHARLES RIVER DATA SYSTEMS

UniverseNet is a trademark of Charles River Data Systems. Ethernet is a trademark of Xerox. UNIX is a trademark of AT&T.

Circle No. 232 on Inquiry Card

July 22-23 Productivity Products International, Orlando, FL: "The Concepts of Object-Oriented Programming". Contact: Barbara Dunn, Productivity Products Int'l, 27 Glen Rd., Sandy Hook, CT 06482. 203/426-1875.

July 22-24 Interactive Systems Corp., Santa Monica, CA: "UNIX Fundamentals". Contact: Claire Donahue, 2401 Colorado Ave., 3rd floor, Santa Monica, CA 90404. 213/453-8649.

July 22-24 Zilog Education and Training Department, Campbell, CA: "UNIX System Administrator". Contact: Kay Ferrell, 1315 Dell Ave., Campbell, CA 95008. 408/370-8000.

July 22-26 Bunker Ramo Information Systems, Trumbull, CT: "Advanced UNIX". Contact: Bunker Ramo, Trumbull Industrial Park, Trumbull, CT 06611. 203/386-2223.

July 22-26 Computer Technology Group, Washington, D.C.: "Berkeley Fundamentals and 'csh' Shell". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, IL 60604. 800/323-UNIX.

July 22-26 Computer Technology Group, San Francisco: "UNIX Internals". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, IL 60604. 800/323-UNIX.

July 22-26 Computer Technology Group, Dallas: "UNIX Internals". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, IL 60604. 800/323-UNIX.

July 22-26 Computer Technology Group, New York: "Berkeley Fundamentals and 'csh' Shell". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, IL 60604. 800/323-UNIX.

July 22-August 2 Information Technology Development Corporation, Cincinnati: "C Programming Language". Contact: ITDC, 9952 Pebbleknoll Dr., Cincinnati, OH 45247. 513/741-8968.

July 22-August 10 Rocky Mountain Institute of Software Engineering, Colorado Springs and Aspen, CO: "1985 Summer Tutorial Program". Contact: RMISE, P.O. Box 3521, Boulder, CO 80303. 303/499-4782.

July 23-26 Integrated Computer Systems, Washington, D.C.: "UNIX: A Comprehensive Introduction". Contact: Integrated Computer Systems, 6305 Arizona Pl., P.O. Box 45405, Los Angeles, CA 90045. 213/417-8888.

July 24-26 Computer Technology Group, London: "Advanced C Programming Under UNIX". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, IL 60604. 800/323-UNIX.

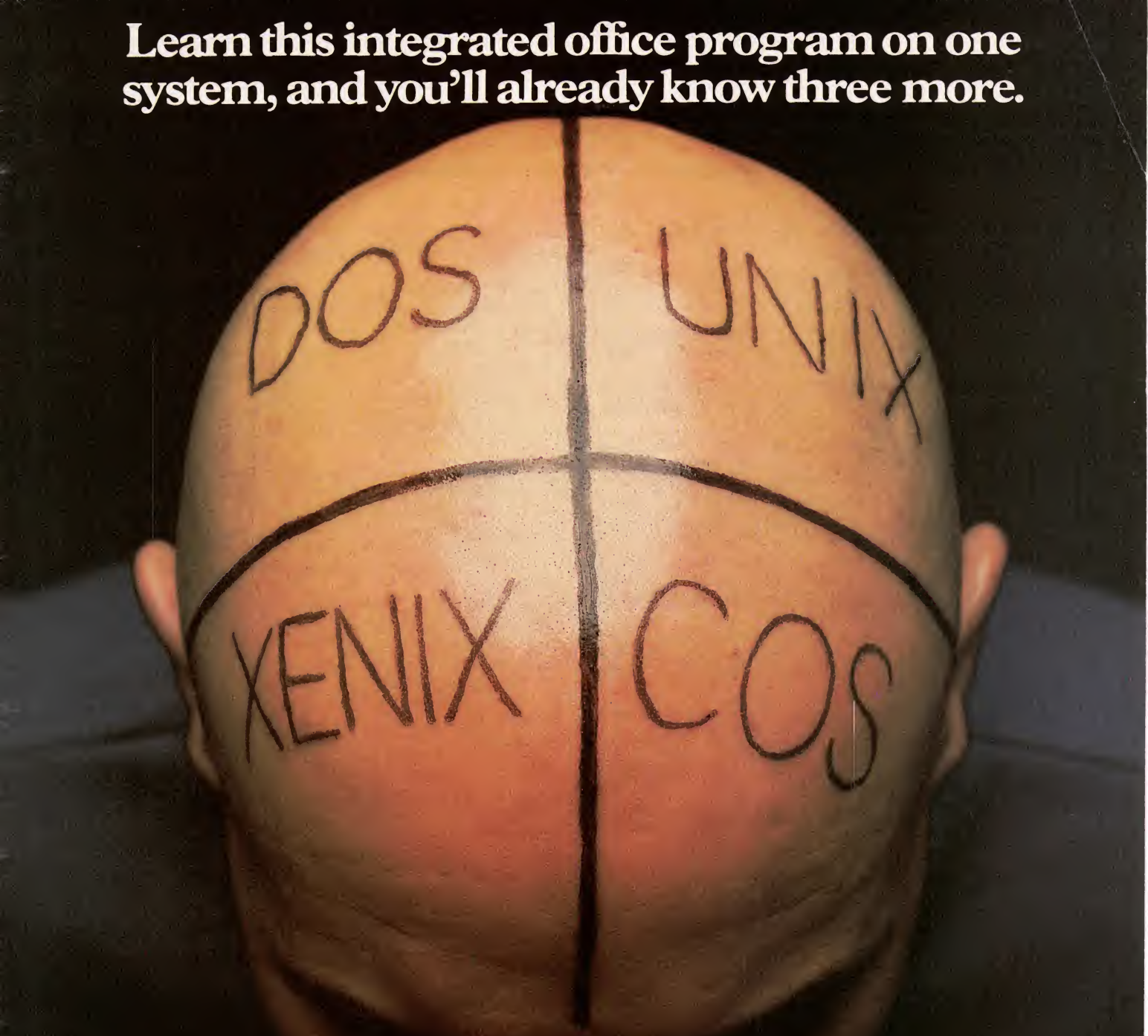
July 25-26 Interactive Systems Corp., Santa Monica, CA: "Using the Shell". Contact: Claire Donahue, 2401 Colorado Ave., 3rd floor, Santa Monica, CA 90404. 213/453-8649.

July 29-30 Interactive Systems Corp., Santa Monica, CA: "System Administrator's Overview". Contact: Claire Donahue, 2401 Colorado Ave., 3rd floor, Santa Monica, CA 90404. 213/453-8649.

July 29-August 1 Zilog Education and Training Department, Campbell, CA: "C Programming". Contact: Kay Ferrell, 1315 Dell Ave., Campbell, CA 95008. 408/370-8000.

July 29-August 2 Computer Technology Group, London: "Berkeley Fundamentals and 'csh' Shell". Contact: Computer

Learn this integrated office program on one system, and you'll already know three more.



R Office ties a powerful word processor together with the file management, table spreadsheet, report generation and desktop management chores common to nearly every modern office.

Incredibly, the UNIX version is identical to the DOS version. And the RM/COS. And the XENIX.

Now you can switch from a stand-alone PC to multi-user computer systems without missing a beat. You take all the functions, all the commands, all your learning with you.

Yates Laboratories gave R Office highest scores in its recent Competitive Edge report, praising its "straightforward and efficient user interface which products designed by committee typically lack."

It's the only office automation program many companies need to own. (And the only one their suppliers need to support.)

We designed R Office to make the most out of very little memory, too. After the first 320K bytes of RAM for installation, each additional terminal requires only 32K. (Compare that with other, so-called multi-user software.)

R Office is already helping small companies and major corporations improve their office productivity. And unlike hopeful imitators, R Office is available, today, to help yours.

Write for information to R Systems, Inc., 11450 Pagemill Road, Dallas, Texas. Or phone toll-free, (800) 527-7610. In Texas, call collect, (214) 343-9188.

R Office™
BY R SYSTEMS, INC.

UNIX AND COS ARE TRADEMARKS OF AT&T LABS AND RYAN-McFARLAND CORP, RESPECTIVELY. XENIX IS A REGISTERED TRADEMARK OF MICROSOFT, INC.

Circle No. 236 on Inquiry Card

Technology Group, 310 S. Michigan Ave., Chicago, IL 60604. 800/323-UNIX.

July 30 Computer Technology Group, Boston: "UNIX Overview". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, IL 60604. 800/323-UNIX.

July 30 Computer Technology Group, Washington, D.C.: "UNIX Overview". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, IL 60604. 800/323-UNIX.

July 30-31 Bunker Ramo Information Systems, Trumbull, CT: "UNIX/C Applications". Contact: Bunker Ramo, Trumbull Industrial Park, Trumbull, CT 06611. 203/386-2223.

July 30-August 1 Computer Technology Group, Dallas: "UNIX Administration". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, IL 60604. 800/323-UNIX.

July 30-August 1 Computer Technology Group, San Francisco: "UNIX Administration". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, IL 60604. 800/323-UNIX.

July 30-August 2 Integrated Computer Systems, Washington, D.C.: "Programming in C: A Hands-on Workshop". Contact: Integrated Computer Systems, 6305 Arizona Pl., P.O. Box 45405, Los Angeles, CA 90045. 213/417-8888.

July 31-August 2 Interactive Systems Corp., Santa Monica, CA: "Interactive Networking Tools". Contact: Claire Donahue, 2401 Colorado Ave., 3rd floor, Santa Monica, CA 90404. 213/453-8649.

July 31-August 2 Computer Technology Group, London: "UNIX Fundamentals for Non-Programmers". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, IL 60604. 800/323-UNIX.

July 31-August 2 Computer Technology Group, Washington, D.C.: "UNIX Fundamentals for Non-Programmers". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, IL 60604. 800/323-UNIX.

AUGUST

August 5-6 Computer Technology Group, San Francisco: "Advanced C Programming Workshop". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, IL 60604. 800/323-UNIX.

August 5-6 Computer Technology Group, Dallas: "Advanced C Programming Workshop". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, IL 60604. 800/323-UNIX.

August 5-6 Interactive Systems Corp., Santa Monica, CA: "Advanced Commands for Programmers". Contact: Claire Donahue, 2401 Colorado Ave., 3rd floor, Santa Monica, CA 90404. 213/453-8649.

August 5-7 Computer Technology Group, Boston: "UNIX Fundamentals for Programmers". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, IL 60604. 800/323-UNIX.

August 5-7 Computer Technology Group, Washington, D.C.: "UNIX Fundamentals for Programmers". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, IL 60604. 800/323-UNIX.

August 5-9 Information Technology Development Corporation, Cincinnati: "UNIX for End Users". Contact: ITDC, 9952 Pebbleknoll Dr., Cincinnati, OH 45247. 513/741-8968.

August 6 Computer Technology Group, London: "UNIX Overview". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, IL 60604. 800/323-UNIX.

August 7-9 Interactive Systems Corp., Santa Monica, CA: "UNIX Architecture: A Conceptual Overview". Contact: Claire Donahue, 2401 Colorado Ave., 3rd floor, Santa Monica, CA 90404. 213/453-8649.

August 7-9 Computer Technology Group, Dallas: "Advanced C Programming Under UNIX". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, IL 60604. 800/323-UNIX.

August 7-9 Computer Technology Group, San Francisco: "Advanced C Programming Under UNIX". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, IL 60604. 800/323-UNIX.

August 7-9 Computer Technology Group, London: "Unix Fundamentals for Non-Programmers". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, IL 60604. 800/323-UNIX.

August 7-9 Digital Equipment Corp., Houston: "Comprehensive Overview of the UNIX Operating System". Contact: Digital Education Resources, 12 Crosby Drive, Bedford, MA 01730. 617/276-4949.

August 8-9 Computer Technology Group, Boston: "Shell as a Command Language". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, IL 60604. 800/323-UNIX.

August 8-9 Computer Technology Group, Washington, D.C.: "Shell as a Command Language". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, IL 60604. 800/323-UNIX.

August 12-14 Computer Technology Group, London: "UNIX Fundamentals for Non-Programmers". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, IL 60604. 800/323-UNIX.

Please send announcements about training or events of interest to: UNIX Review Calendar, 500 Howard Street, San Francisco, CA 94105. Include the sponsor, date, location of event, address of contact, and relevant background information.

UNIXEXPO™

For one week in September

New York City becomes the heart of the UNIX universe

Join the thousands of your colleagues who will seek answers to meet their business needs…and come away with a full understanding of UNIX solutions and applications.

Take advantage of the best UNIX has to offer:

- An exhibition featuring over 200 of the leading suppliers of UNIX based hardware, software and services.
- A tutorial program designed and developed by AT&T —the most respected source for the UNIX System.
- A conference examining the advantages of UNIX solutions in the business environment.

Plan now to attend and profit from THE PROVEN UNIX MARKETPLACE.

For all the details contact: UNIX EXPO 14 West 40th Street, New York, N.Y. 10018 Telephone: 212-391-9111. TELEX: 135401 DIMCOMM.

ADVERTISER'S INDEX

Adax Inc. 66
AT&T Information Systems . . . . . 33 ,87
B.A.S.I.S. 16
bbj Computer Services, Inc. 78
Bell Technologies 43
Bunker Ramo Information Systems 70
Ceegan 80
Celerity Computing 9
Century Software 20
Charles River Data Systems 93
CMI Corp. 21
Cogitate 82
Computer Cognition 94
Computer Technology Group 91
Concentric Associates 99
Corporate Micro Systems 39
CYB Systems 1
DSD Corp. 14
Dynacomp 63
Emerging Technology, Inc. 7
Franz, Inc. 82
GIC . . . . . Cover III
Gould 60,61
Handle Technologies . . . . . Cover II
Hewlett-Packard Co. 56,57
Image Network 43
Information Technology Development Corp. . . 12
Integrated Solutions 28,29
Interactive Systems Corp. 89
Marc Software 74,75
Microware Systems . . . . . 37,Cover IV
MIPS Software 94
Mt. XINU 59
NETI . . . . . Centerspread
Patel Consultants 81
Perennial Software 90
Quality Software Products 96
Quelo, Inc. 85
R Systems, Inc. 35,95
Relational Database Systems 2,3,67
Santa Cruz Operation 73
Scientific Placement 81
Sperry Corp. 23,97
Textware 22
Unipress Software 13,15,17,19
University of Toronto 88
Uniworks 71
UNIX Expo 98
UNIX Review 69
UX Software 10,11
Z-Systems 90

COMING UP IN AUGUST

Database Management

- **A Look at Generic Database Issues**
- **The UNIX Connection**
- **Back-end Database Machine Strategies**
- **Transactional Processing Under UNIX**
- **Product Comparisons**

NOW THAT YOUR SALESMAN HAS GONE.

Realize the full power — the power that is UNIX™.

GIC continues to assist some of the largest companies in Europe and America, to do just that!

We are one of the most experienced training and consulting firms in the UNIX world

We are your single source for a complete UNIX service.

Call (513) 421-UNIX for more information.

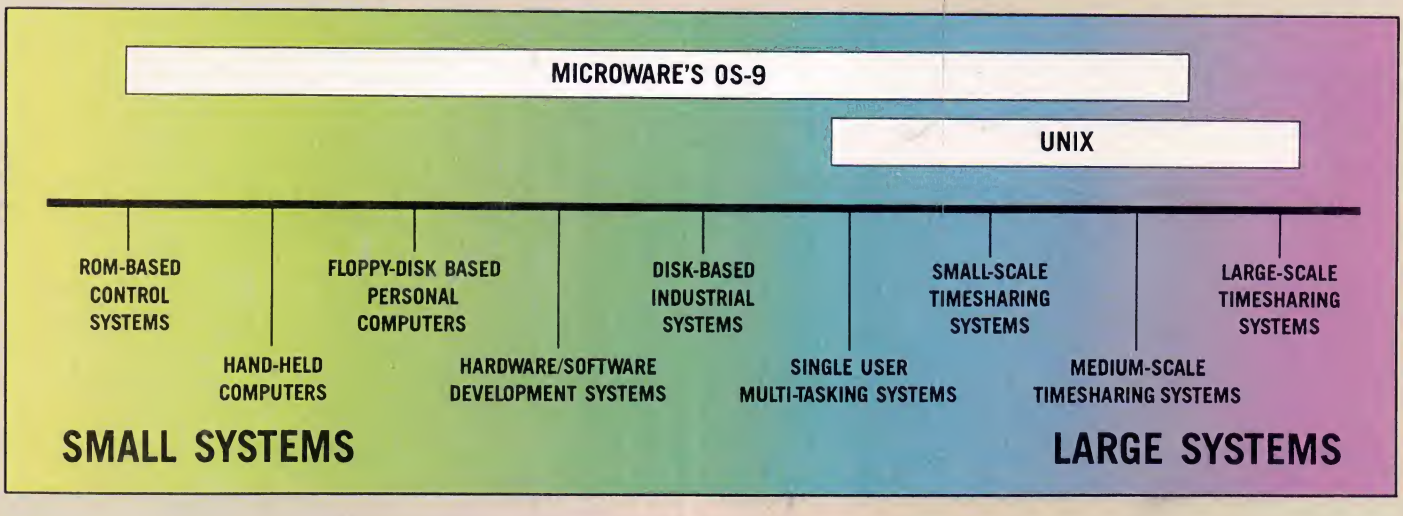
gic

the sensible approach

P.O. Box 1368
Cincinnati, Ohio 45201

Circle No. 233 on Inquiry Card

Only Microware's OS-9 Operating System Covers the Entire 68000 Spectrum



Is complicated software and expensive hardware keeping you back from Unix? Look into OS-9, the operating system from Microware that gives 68000 systems a Unix-style environment with much less overhead and complexity.

OS-9 is versatile, inexpensive, and delivers outstanding performance on any size system. The OS-9 executive is much smaller and far more efficient than Unix because it's written in fast, compact assembly language, making it ideal for critical real-time applications. OS-9 can run on a broad range of 8 to 32 bit systems based on the 68000 or 6809 family MPUs from ROM-based industrial controllers up to large multiuser systems.

OS-9'S OUTSTANDING C COMPILER IS YOUR BRIDGE TO UNIX

Microware's C compiler technology is another OS-9 advantage. The compiler produces extremely fast, compact, and ROMable code. You can easily develop and port system or application software back and forth to standard Unix systems. Cross-compiler versions for VAX and PDP-11 make coordinated Unix/OS-9 software development a pleasure.

SUPPORT FOR MODULAR SOFTWARE — AN OS-9 EXCLUSIVE

Comprehensive support for modular software puts OS-9 a generation ahead of other operating systems. It multiplies programmer productivity and memory efficiency. Application software can be built from individually testable software modules including standard "library" modules. The modular structure lets you customize and reconfigure OS-9 for specific hardware easily and quickly.

A SYSTEM WITH A PROVEN TRACK RECORD

Once an underground classic, OS-9 is now a solid hit. Since 1980 OS-9 has been ported to over a hundred 6809 and 68000 systems under license to some of the biggest names in the business. OS-9 has been imbedded in numerous consumer, industrial, and OEM products, and is supported by many independent software suppliers.

Key OS-9 Features At A Glance

- Compact (16K) ROMable executive written in assembly language
- User "shell" and complete utility set written in C
- C-source code level compatibility with Unix
- Full Multitasking/multiuser capabilities
- Modular design - extremely easy to adapt, modify, or expand
- Unix-type tree structured file system
- Rugged "crash-proof" file structure with record locking
- Works well with floppy disk or ROM-based systems
- Uses hardware or software memory management
- High performance C, Pascal, Basic and Cobol compilers

microware® OS-9™

MICROWARE SYSTEMS CORPORATION
1866 NW 114th Street
Des Moines, Iowa 50322
Phone 515-224-1929
Telex 910-520-2535

Microware Japan, Ltd
3-8-9 Baraki, Ichikawa City
Chiba 272-01, Japan
Phone 0473(28)4493
Telex 299-3122

 See us at NCC '85, Booth #1444

Circle No. 255 on Inquiry Card