UNIX SYSTEM PORTABILITY

UNIX™/WORLD

Your Complete Guide to the Frontiers of the Unix System

VOL. I, NO. 6 $3 IN USA A TECH VALLEY PUBLICATION

REVIEWS: AT&T 3B2, LOCUS, PC BRIDGE

UNIX, C, AND APL

JOURNAL: DEVICE DRIVERS AND YOU

UNIFY FINISHES FIRST.

In one independent competition after another, UNIFY has proved itself the fastest UNIX data base management system. No wonder it's been selected by more computer manufacturers than all other UNIX data bases combined.

UNIFY speeds you through development and expedites program execution with some of the fastest and most powerful utilities of all, including:

Fully menu-driven design.

Built-in optimizers that select the fastest of four data access methods.

Industry standard IBM SQL query language, fully integrated to our own powerful report writer, for easy access by your end-users.

Ninety subroutines for advanced program development... the most complete package of its kind.

Horsepower for the long run. Unlike other data bases, UNIFY won't slow down under the weight of additional data or multiple users. It won't fall behind in the future either, because it's built with the power to support new features as they come on-board.

Judge for yourself. Send for our 300-page tutorial and 500-page reference manual—yours for only $95. Together they show you how to build virtually any application. To order, contact UNIFY, Dept.UW-11, 9570 S.W. Barbur Blvd., Portland, Oregon 97219, 503-245-6585.

UNIFY®

THE PREFERRED UNIX DBMS.

Networking Your System?

Change a 12 Month Design Cycle to 12 Days.

Time. Development time. Time to market.

If you're building your new computer system around a LAN, you're in a race against time.

Ethernet* gives you a fast start, but the protocol software and system integration efforts can take as much as a year — even longer.

But now you can shorten that design cycle. With EXOS™, from Excelan. EXOS technology gives you a complete LAN capability, hardware and software, that simply plugs into your computer system's backplane. Integration takes next to no time.

With EXOS, you can be up and running in as little as 12 days — even sooner.

EXOS, or EXcelan Open Systems, is a family of Ethernet front-end processor boards, standard high-level protocol software which runs on the front-end processor, and network application software which runs on the host computer.

EXOS technology includes front-end processors for all Multibus,* VME bus, Unibus* and Q-Bus based computers and software supporting the standard TCP/IP high-level protocol. And if your host system supports any variety of UNIX,* then half a dozen network applications are provided as well.

EXOS. It gives your computer system a complete LAN capability with the ease of plug-in integration. That can change your design cycle from 12 months to 12 days. And get your product to market before your competition.

EXCELAN

"Excellence in Local Network Technology"

2180 Fortune Drive
San Jose, California 95131
(408) 945-9526 TELEX 176610

Please circle Ad No. 34 on inquiry card.

You can receive an informative, in-depth analysis of leading-edge software industry trends and the UNIX* System Market with

the Yates Perspective


the Yates Perspective

VOLUME 2 NUMBER 3

APRIL 1984

UNIX Licensing Confusion Slows System V

Motorola completed its System V port, but hardware vendors may experience delays in obtaining software due to AT&T's licensing procedures.

Interview with Motorola's Bill Lowery

by Michael Dubrall

Inside This Issue

Featuring:

- **Competitive Strategies**
- **Market Analysis**
- **New Product Reviews**
- **Interviews**
- **Benchmarks**

"The Yates Perspective contains provocative and highly original research with no me-too analysis."

Otis Wilson, Manager of Software Sales and Marketing AT&T Technology Systems

"The Yates Perspective offers the best high-level overview of what's happening in the UNIX industry."

Karan Kauppila, President Handle Corporation

Published by Yates Ventures, *The Yates Perspective* is a monthly newsletter on the software market. With this SPECIAL OFFER you can subscribe to *The Yates Perspective* for a reduced rate of $349.00 per year. That's $101.00 off our regular subscription price.

*UNIX is a trademark of AT&T Bell Laboratories

☐ Yes! I want to subscribe to *The Yates Perspective* for a reduced rate of $349.00.

Name ____________________

Title ____________________

Company ____________________

Street ____________________ M/S ____________

City ____________ State ____________ Zip ____________

Phone ____________________

☐ My check or money order is enclosed.

☐ Please bill my company. My purchase order number is ____________.

☐ Please charge my (circle one) VISA/ Mastercard/American Express

Number ____________

Signature ____________

Exp. Date ____________

Address all correspondence to: Yates Ventures, 3350 West Bayshore Road, Palo Alto, CA 94303

Please circle Ad No. 52 on inquiry card.

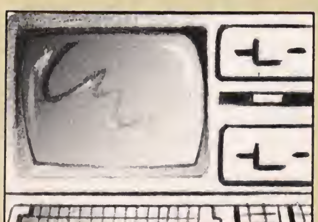
UNIX/WORLD

VOLUME 1, NUMBER 6

CONTENTS

A TECH VALLEY PUBLICATION

page 28



page 44

page 50

THEME

WIZARDS' WOES: THE PROBLEMS WITH PORTING — 28
by Vanessa Schnatmeier

Computer wizards and marketeers have hyped the Unix system's vaunted portability. But portability is a matter of definition. Our author takes an in-depth look at the issues.

MAYBE WE CAN STILL KEEP THE NAME — 38
by Gordon W. Waidhofer and Michael Sweeney

How do you define true Unix system portability, and, once achieved, do you really want it after all? Our two authors reach startling conclusions to those and other burning questions.

A SOFTWARE VENDOR'S SEARCH — 44
by Neal Nelson

Seeking refuge from the whims of Big Blue, a software vendor sets out for the promised land of Unix system portability. But does he find it? Stay tuned.

FEATURES

4.2BSD VERSUS SYSTEM V — 50
by Kathleen B. Wallace

Tales of incest, Oedipal complexes, and the tangled royal houses of Europe are but primers to the current 4.2BSD and System V debate.

UNIX, C, AND APL — 59
by David Saunders

The Unix system is a very-high-level operating system, so why are we still stuck on C? Our author argues for APL!

REVIEWS

THE AT&T 3B2/300 — 68
by Bruce Mackinlay

Our fearless reviewer takes a look at AT&T's vaunted supermicro standard bearer, and, with a few reservations, likes what he sees.

CONTENTS CONTINUED ▶

CONTENTS

CONTINUED

page 88

83

THE LOCUS DISTRIBUTED UNIX SYSTEM
by Jeffrey Copeland, Vanessa Schnatmeier, and Alan Winston

Locus Computing Corp.'s *Locus* distributed Unix system implementation proposes a transparent solution to the mainframe, mini, and micro link hassle.

88

PC BRIDGE
by Alan Winston

Want to run a spreadsheet, report generator, or color graphics package? The Locus PC Interface bridges the chasm between your MS-DOS-based PC and the Unix system world.

92

WRITING DEVICE DRIVERS
by Rik Farrow

That expensive new hard disk or fancy daisywheel printer you just bought won't run on your Unix system without a device driver. Here's an introductory look on how to write a device driver for your needs.

TRENDS

FROM THE PUBLISHER'S DESK 7
EDITOR'S CONSOLE 9
ON-LINE 13
mail 17
INSIDE EDGE 18
NEW PRODUCTS 100
NEWS FROM AT&T 109
NEWS FROM BERKELEY 110
sync 120

TRAINING

USER SPOTLIGHT 97
WIZARD'S GRABBAG 114
CALENDAR 116

/etc

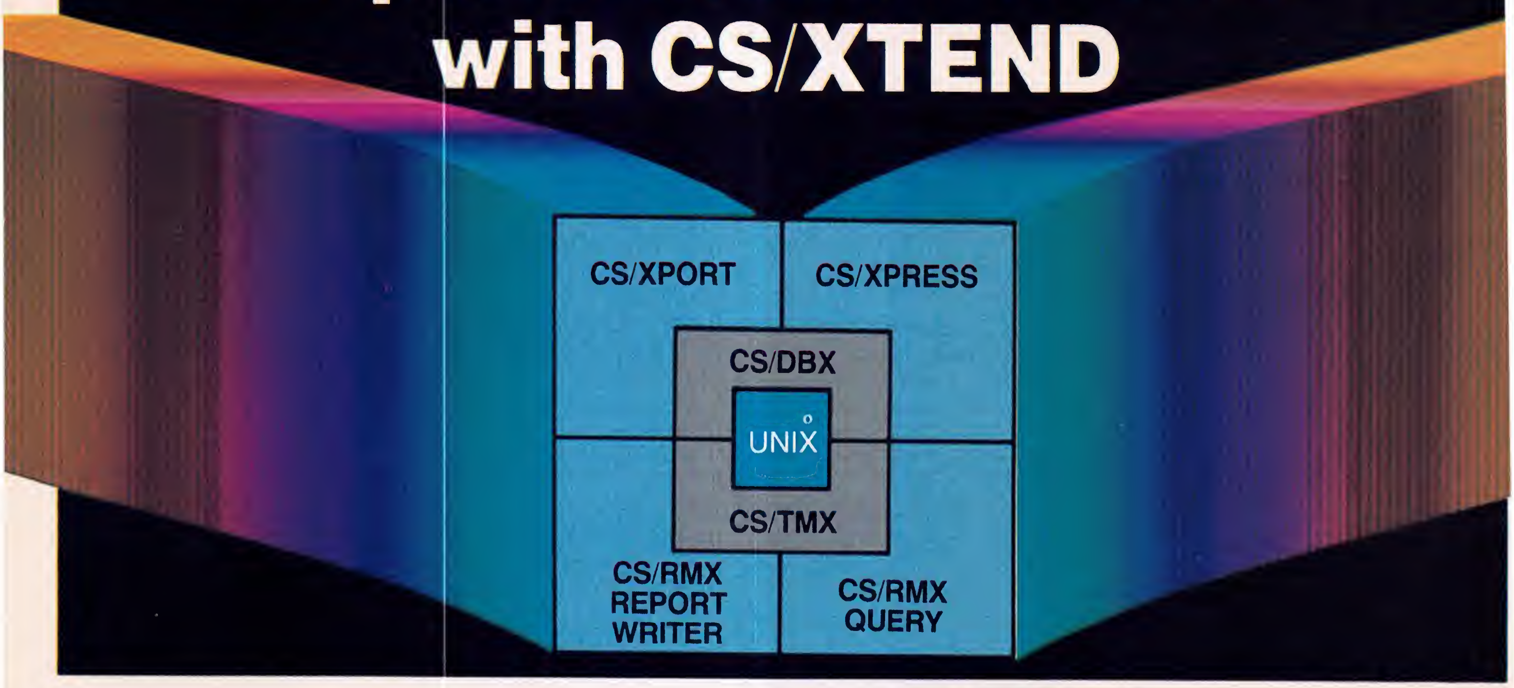
ADVERTISERS' INDEX 112
MARKETPLACE 99
CAREER OPPORTUNITIES 99

UNIX/WORLD (ISSN 0739-5922) is published monthly by Tech Valley Publishing, with offices at 444 Castro Street, 12th Floor, Mountain View, CA 94041, telephone (415) 964-0900. Third class permit #3481 Dallas, TX 75260. John M. Knapp, Publisher. RETURN POSTAGE GUARANTEED. Postmaster, send Form 3579 to UNIX/WORLD, 444 Castro Street, 12th Floor, Mountain View, CA 94041.

Subscriptions are $18 for one year within the USA and its possessions and Canada. Foreign subscriptions add $20 for surface mail. Remit in US funds drawn on a US bank. *Subscription questions and problems:* Contact Cheryl Hogan, Circulation Director. Printed in United States of America. Requests for reprints or bulk orders should be addressed to the Circulation Director.

Address all editorial correspondence to the editor at UNIX/WORLD, 444 Castro Street, 12th Floor, Mountain View, CA 94041. Unsolicited manuscripts can only be returned if accompanied by a self-addressed, stamped envelope. UNIX/WORLD is not responsible for lost or damaged manuscripts, photos, or artwork.

Extend the power and performance of UNIX™ with CS/XTEND



Cincom Systems, the industry leader in data base/data communication software systems, passes the power of its proven high performance software technologies for mainframe and minicomputers to the world of UNIX. CS/XTEND, our new fully integrated family of development and decision support components, makes it easy to create and implement even the most complex applications. And, our experience and expertise in developing quality software ensure that you get the high performance you need while using a minimum of machine resources.

CS/XTEND includes integrated solutions for professional system developers as well as productivity tools for non-technical information users:
CS/DBX—a high performance shared Data Base Executive with the power to handle multiple users concurrently accessing any number of files.
CS/XPORT—a Distributed System Interface that links computers so they can share data, providing the foundation for distributed data base networking.
CS/TMX—a powerful Terminal Management Executive that simplifies interactive programming by making programs independent of the terminal, and provides overall network management.
CS/RMX—an easy-to-use Retrievals Management Executive that provides relational access to the data base for interactive inquiries and report writing by both end-users and computer professionals.
CS/XPRESS—a dynamic Application Builder that lets non-programmers automatically generate, maintain and access custom-built files of information.

CS/XTEND is fully compatible with Cincom's TOTAL® Data Base Management System. That means you can implement applications to operate on different types of computers using identical data base techniques.

Find out today how CS/XTEND extends the power and performance of your UNIX system. Simply call or write, Cincom Ventures Division, 2300 Montana Avenue, Cincinnati, Ohio 45211.

800-543-3010 (In Ohio: 513-661-6000)

UNIX is a trademark of AT&T Bell Laboratories

Please circle Ad No. 5 on inquiry card.

Cincom Systems®

Excellence in Software Technology

FROM THE PUBLISHER'S DESK

It's time we stopped beating up on AT&T. They simply are not the helpless giant that so many magazine pages, including our own, proclaim them to be. Far from being Gulliver on the beach, they are beginning to look like the Frog Prince. Let's take a look at what they've accomplished since divestiture in January.

While the halls of Silicon Valley are resounding with the anguished cries of wounded venture capitalists, AT&T has brought out—in less than 10 months—a more comprehensive line of data processing equipment than anyone in the business (with the exception of IBM). And despite a fair measure of early scoffing from onlookers, the entire line, from PCs to mainframes, consists of credible machines at fair prices. Throw into the pot a few breakthrough products—in the form of office automation/PBX systems, a real-time Unix system, and a fully implemented fault-tolerant system—and they have done enough to spoil everyone's lunch: Tandem, DEC, and IBM. In the same time frame, HP has scrapped almost as many projects as AT&T has delivered, and Big Blue has released two products of moment: the abysmal PCjr and the PC/AT (and they still cannot decide whether they're serious about the Unix system or not).

And on top of its own true innovation, AT&T has had the humility to seek partners in technology that range from Convergent Technologies to Digital Research to Olivetti.

"Yeah, but can they market?" They have had short, punchy ads in *The Wall Street Journal* and on prime-time TV that explain the advantages of the Unix system for businesspeople. Very quietly they have established solid retail distribution of their PCs at Genra and Sears. And while no one was watching, they have established the world's largest direct sales force. (You can now order a PC, mini, or mainframe computer from the same customer rep who sells you your phones. (Sometime in the last few months that customer rep magically has become knowledgeable about your data processing needs and is quite anxious to discuss the advantages of leasing DP technology and putting it on your phone bill.) And they've proven that they finally understand the need for customer support.

Finally, corporate management has made aggressive moves to cut costs and reduce overhead. When the whole world world is watching, it can't be easy to freeze salaries and trim staff. Chrysler's Lee Iacocoa did it and is hailed as a genius in some quarters. To AT&T's credit, they made their bold moves before the bailout. And they managed their cuts at the same time they gracefully pulled off the largest, most integrated startup in the history of our industry.

It may be time to stop snickering and start worrying.

John M. Knapp
President/Publisher

UNIX/WORLD

STAFF

Publisher
John M. Knapp
Editor Emeritus
Dr. Rebecca Thomas
Executive Editor
Philip J. Gill
Art Director
Dale Bates
Editor at Large
Lauren Weinstein
Copy Editor
Andrew Ould
Editorial Production Coordinator
Monica E. Berg
Production Manager
Gerald A. McEwan
Production Assistant
James McEwan
Staff Photographer
Ralph Cooksey-Talbott
Circulation Director
Cheryl Hogan
Circulation Consultant
John Klingel
Circulation Assistant
Beth Doolin
Assistant to the Publisher
Nancy Thompson
Cover Photograph
Ralph Cooksey-Talbott

CONTRIBUTING EDITORS

Dr. Brian Boyle, Gnostics Concepts: Market Research; **Steve Bourne,** Silicon Graphics: Shells; **Gene Dronek,** Aim Technology: System Optimization/Benchmarking; **Dr. Bill King,** UC Davis: Writer's Workbench; **Bob Marsh,** Plexus Computers: Supermicros; **P.J. Plauger,** Whitesmiths: Work-alikes; **Omri Serlin,** ITOM International: Market Analysis; **Deborah K. Scherer,** Mt. Xinu: Software Tools; **Lauren Weinstein,** Vortex Technology: Networking/Communications.

ADVERTISING SALES

Advertising Production Coordinator
Candy Isaac

New England, New York, New Jersey, Pennsylvania, and Eastern Canada: Charles E. Lynch, Nancy Wood, Merrie Lynch, 61 Adams Street, Braintree, MA 02184, (617) 848-9306. **Midwest:** John J. Geier, 444 Castro Street, 12th Floor, Mountain View, CA 94041, (415) 964-0900. **Denver:** R.H. "Bud" Foster, 14 Inverness Drive East, Bldg. A, # 232, Englewood, CO 80112, (303) 790-8181. **Dallas:** John White, Donald Moeller, 5801 Marvin D. Love Freeway, Suite 303, Dallas, TX 75237, (214) 941-4461. **Los Angeles:** Gordon T. Sutton, Carole Ann Philips, 524 W. Commonwealth, Suite G, Fullerton, CA 92632, (714) 525-6686. **Miami:** Ray Rickles, Ray Rickles and Company, 3783 Pinetree Drive, Miami Beach, FL 33140, (305) 532-7301. **Atlanta:** Scott Rickles, Ray Rickles and Company, 699 Carriage Drive, N.E., Atlanta, GA 30328, (404) 252-4567. **San Francisco:** Morton McDonald, Ginger McDonald, Ian McDonald, 265 Bayridge Office Plaza, 5801 Christie Avenue, Emeryville, CA 94608, (415) 653-2122/967-6400. **Oregon:** Frank M. Eaton, 510 South First, P.O. Box 696, Hillsboro, OR 97123, (503) 640-2011.

TECH VALLEY PUBLISHING

President
John M. Knapp
Office Manager
Verlene J. Perry
Receivables
Constance C. Bolak

EDITOR'S CONSOLE

Portability. That word alone in large part has helped propel our vital and explosive segment of the computer industry—the Unix systems marketplace—to its current prominence.

But what does that word mean for vendors and users, and what are its implications? Is it just another bandwagon buzzword coined by clever computer marketeers for their own self aggrandizement? Our industry already is littered with too many self-serving expressions, like *user-friendly, windowing, powerful,* and a dozen or more others.

Or is portability a true technological breakthrough that will revolutionize the computer industry?

The answer to these questions, I'm afraid, is that the famed portability of the Unix system and its variants is more a myth culled from the incantations of our Unix system wizards and marketeers than a reality.

The Unix system marketplace has not reached the level of standardization (and some might add, mediocrity) that has been achieved in the PC arena, where a user can walk from IBM PC clone to IBM PC clone with the same software, insert the diskette, and watch it work each and every time.

Don't get me wrong. It's not that I am saying that the Unix system is not portable. In fact, it is highly so, but in a far more sophisticated way that is largely removed from the average, naive computer user. In our zeal to promote the Unix system, we may be guilty of having misled and misguided many.

Instead, I believe that the concept of true Unix system portability across processors must be judged by comparison to other operating systems. In short, the Unix system is simply more portable than its predecessors by a large degree.

For both the user and the vendor, then, the Unix system does indeed provide a greater degree of insulation against the whims and fancies of computer architects and the turbulent fortunes of the computer industry. And that's more than either has been offered before.

Moreover, as we move toward standards in the Unix system community, more and more of the portability issues will fade away, and the unfulfilled promise of the Unix system will come ever closer to reality.

The reason I've gotten up on this soapbox to put the Unix system portability myth in its proper prospective is this. It is not that I don't want the Unix system to succeed. Far from it. There are many reasons why the Unix system can and will succeed—its multitasking, multi-user capabilities, for instance.

However, all summer long the news from the computer industry in general and from California's legendary Silicon Valley in particular has been bad and getting bleaker. Layoffs, Chapter 11 filings, depressed earnings. And that long hot summer threatens to fade into an even longer, cold winter of discontent.

It is time now to begin worrying about our image, ladies and gentlemen. Not just our image in the industry, but our image in the nation at large. The time to overpromise our ability to deliver has passed. The market has become unforgiving, and so too may many users. We risk a backlash.

So, as I have said before, this magazine will depict a realistic and honest portrait of the benefits and capabilities of the Unix system. To that end, we have zeroed in on the issues surrounding Unix system portability this month.

Freelance writer Vanessa Schnatmeier leads off with a discussion of portability issues and performance implications that users should be aware of.

All that is portable may not be gold, however. Volker-Lehman Systems managers Gordon W. Waidhofer and Michael Sweeney tackle the problem of defining portability and discuss why ultimate portability may not be the promised land we all think it might be.

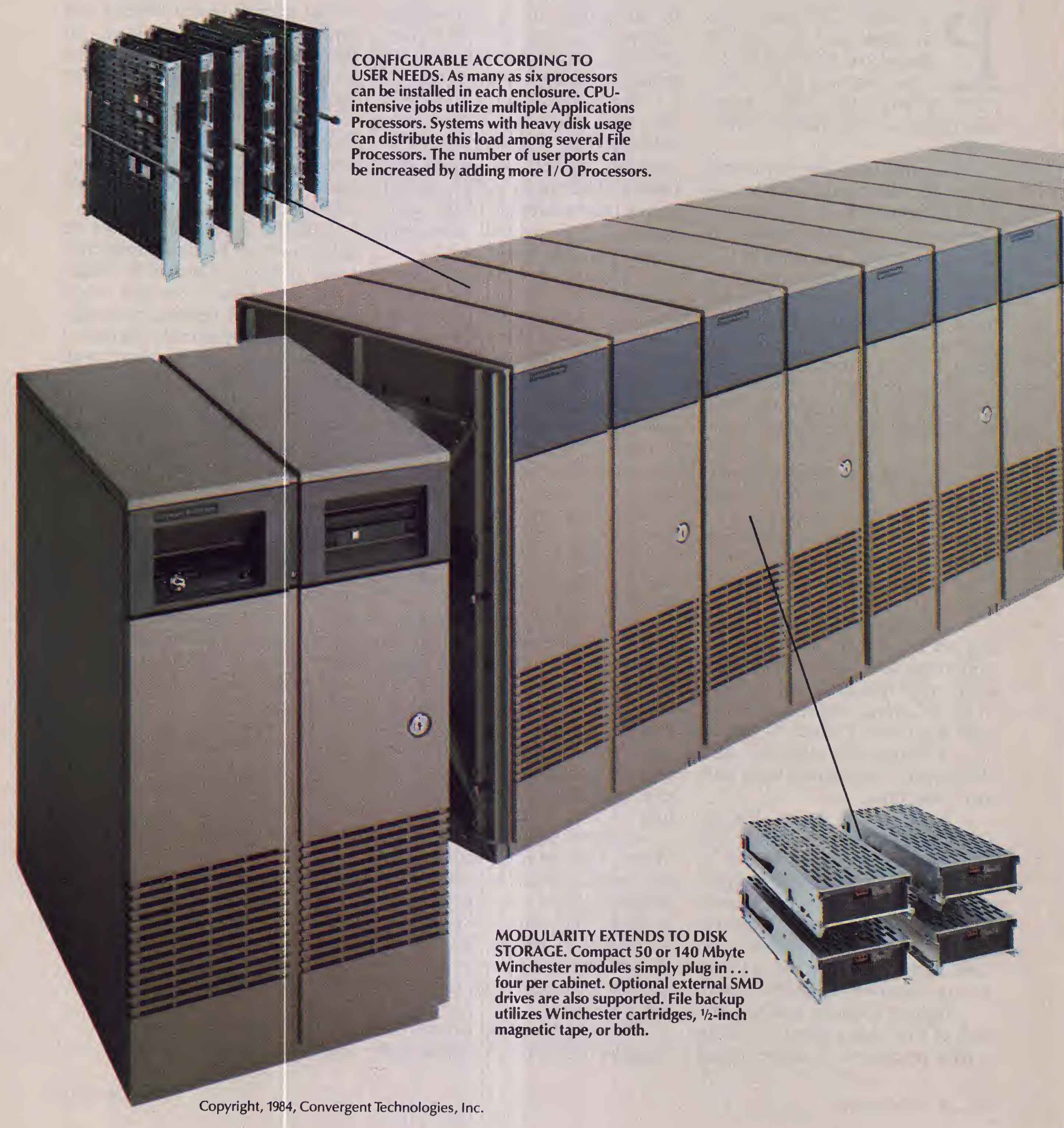
Then, software and systems reseller Neal Nelson tells his story of how and why he came to the Unix system after years of living under the fleeting whims of computer giants.

Along with these and other feature articles this month, we present a review of AT&T's vaunted 3B2 supermicrocomputer. It's considered the best price/performance computer AT&T has offered so far, and from all reports will bring smiles to Unix system users and headaches to the competition.

One final note: Dr. Rebecca Thomas, our Editor Emeritus, has taken a vacation from her monthly *C Tutorial* journal articles. At this time, she would like to request that interested readers submit useful software tool programs to her for possible publication in upcoming issues. Requirements for these entries are outlined in Vol. 1, No. 4. Dr. Thomas can be reached by mailing your entries to our main editorial office. □

Philip J. Gill
Executive Editor

IT WILL GROW ON YOU.



CONFIGURABLE ACCORDING TO USER NEEDS. As many as six processors can be installed in each enclosure. CPU-intensive jobs utilize multiple Applications Processors. Systems with heavy disk usage can distribute this load among several File Processors. The number of user ports can be increased by adding more I/O Processors.

MODULARITY EXTENDS TO DISK STORAGE. Compact 50 or 140 Mbyte Winchester modules simply plug in . . . four per cabinet. Optional external SMD drives are also supported. File backup utilizes Winchester cartridges, ½-inch magnetic tape, or both.

MegaFrame™. Now OEMs can offer a high-performance UNIX™-based system that can't run out of performance.

OEMS can now deal cost-effectively with the problems encountered when user applications produce computing demands that outstrip the capabilities of conventional systems.

Convergent Technologies' MegaFrame is a revolutionary new UNIX-based super-minicomputer—so innovative in its architecture that it represents the ultimate in multiuser systems design. It grows exponentially from a system offering minicomputer-level performance to an enormously powerful engine serving as many as 128 users with 36 parallel processors, 24 megabytes of RAM and gigabytes of disk storage.

No other system can match the MegaFrame's potential for field expansion. It enables manufacturers and systems builders to keep pace with today's requirements for more and more computing services . . . but *not* at the cost of discarding hardware or performing expensive CPU upgrades.

MegaFrame's architectural breakthrough. Dependence on traditional single-CPU shared-logic architecture is the root of systems bottlenecks.

Convergent's response: a novel system utilizing *multiple* specialized processors to distribute workloads for optimum performance—even if user needs are unpredictable or subject to rapid change.

MegaFrame's virtual memory Applications Processors each have a 32-bit CPU, up to 4 Mbytes of RAM and run a demand-paged version of UNIX System V. Up to 16 of them can operate in parallel.

The File Processors effectively function as back-end machines providing DBMS, ISAM and other disk-related services. Up to six File Processors each with four disks can operate in parallel.

Terminal and Cluster Processors can also be added—the latter serving front-end communications needs. They off-load communications from the other processors by running protocols such as SNA and X25 networks.

MegaFrame's daisy-chained cabinets offer total expansion potential of up to 36 slots. OEMs configure the system needed for specific applications simply by adding the correct number/combination of processors.

Flexibility in applications development. Inclusion of one or more Applications Processors allows running UNIX System V. All standard UNIX tools are provided, along with COBOL, FORTRAN-77, BASIC interpreter and compiler, plus Pascal.

The "least-cost solution" to serving a wide range of UNIX-systems needs, MegaFrame has won acceptance from OEMs in the U.S. and abroad. The uniqueness of its modular design, its versatility in providing upgrade-path options and its price/performance advantages give it market-share potential of outstanding dimensions.

The system that will grow on you starts at a very attractive price: about $20,000 for a system that effectively supports 16 users. Send now for a comprehensive Information Package including reprints of magazine articles. It explains how MegaFrame's growth potential can impact favorably on your plans for growth in the UNIX market.

Convergent Technologies, Data Systems Division, 3055 Patrick Henry Drive, Santa Clara, CA 95050. Phone: 408/980-0850. Telex: 176-825.

MiniFrame™: the entry-level multiuser UNIX system.

Starting at under $5,000 for a single-user system, Convergent's MiniFrame offers outstanding capabilities for small to medium sized organizations running large UNIX-based applications. Utilizing an MC68010 microprocessor operating at 10Mhz, with no wait states, it provides impressive CPU speed—comparable to VAX™-11/750 running the AIM™ Benchmark. MiniFrame features virtual memory management, with demand-paged implementation of UNIX System V. It runs as many as eight terminals, with up to 50 Mbytes of integral mass storage. MiniFrame and MegaFrame are object-code compatible, allowing OEMs to offer a *complete family* of systems unrivaled in price/performance characteristics.

Convergent Technologies

Where great ideas come together

Please circle Ad No. 9 on inquiry card.

FORTUNE'S FORTUNES AND OTHER NOTES

BY PHILIP GILL

While the news from that "erstwhile darling" of the microcomputer industry, Fortune Systems Corp., has been mostly good and getting better all summer long, it's still a little too early to pronounce the turnaround complete.

On the plus side, Fortune has gone through great pains this summer to shed its tainted image. For example, gone are Homer Dunn and Dave Vandenberg, two co-founders who were two remaining vestiges of the discredited Freidman regime. The official line is that the two resigned.

However, just shortly after those two gentleman left, Fortune laid off 30 to 40 other employees. These moves were taken ostensibly to reduce headquarters overhead and apparently did not affect the field sales operations. But the action was seen by some as a very necessary step needed to purge the firm of its old image.

The news of the layoffs and resignations was just about cold when Fortune disclosed that after several highly publicized quarterly losses, the firm again rejoined the world of the living corporations by turning a tidy little bit of profit ($39,000, to be exact) for the second quarter ended June 30. Total sales were up 69 percent to boot —to $20.3 million from revenues of $12 million for the same period the year before.

Then, in August, Fortune disclosed that it was in merger/takeover negotiations with North Star Computers of San Leandro, Calif. North Star is an IBM PC-compatible microcomputer maker, and the action is a natural one for cash-rich (though profit-poor) Fortune, which has more than $40 million in the till. The discussions apparently began when North Star approached Fortune to see if Fortune would be interested in OEMing North Star gear for resale.

SELF PRESERVATION

Although the potential merger between the two companies fell through, that kind of deal could portend the beginning of an important trend in the microcomputer market, as smaller firms band together for self preservation against the onslaught of the multinational megacorporations like IBM and AT&T. North Star would also bring valuable dealer and reseller channels to an eager Fortune.

So far so good. Preliminary congratulations are certainly in order for the new Fortune management team assembled by Jim Campbell, formerly of Xeroid. However, before any of us in the computer trade or business press pronounce Fortune truly resurrected, the firm will have to show a real profit. As we here at UNIX/WORLD and others in the industry have said, we'll give you to the end of calendar '84 to prove you can do it.

On the down side, Fortune's marketing efforts may have been impaired by the recent layoffs. Moreover, the firm seems to have virtually ceased advertising after an aggressive attack on IBM this summer. The campaign featured Charlie Chaplin crossing his fingers as he promised to tie all those stand-alone PCs together.

One is also worried that the time spent negotiating the possible merger with North Star may not be all that well spent. North Star has its own problems, and Fortune executives should not let themselves be distracted from the job that still is half undone in their own backyard.

On the other hand, some of the news this summer indicates a positive direction. Fortune just signed a major agreement with one of its already largest customers, Ford Motor Co. That deal calls for Fortune to deliver 216 XP:30 central processing units to Ford's Body & Chasis Engineering operations by year's end.

So, put bluntly, we wish you luck fellas. We'll be waiting eagerly and hopefully for the third- and fourth-quarter financials, which will tell the tale.

MORE RUMORS ON AT&T AND CONVERGENT

There's been much speculation over exactly what the delayed products Convergent Technologies is building for its OEM AT&T. Here's the latest: Some say Convergent is building an integrated voice and data workstation with a telephone handset based on a machine originally designed and prototyped inside AT&T's Bell Labs.

While we're touching on the subject of AT&T, we understand the firm is also preparing to pick up the Olivetti portable personal computer as well.

SYDIS ENCORE

Sydis, Inc., the integrated voice workstation maker, appears to be on a roll. The firm has reportedly signed a major OEM contract with

CREATE
LASTING IMPRESSIONS
HIEROGLYPH™
UNIVERSAL REPORT PRODUCTION SYSTEM™
INTEGRATED TEXT/DATA/ADVANCED GRAPHICS SOFTWARE
A "report" is information presented in organized form—typically a printed document. If you are a professional whose work involves preparing reports HIEROGLYPH™ is for you. HIEROGLYPH is designed to meet the needs of technical and office report preparation and production where the combination of text, data and advanced graphics are essential elements. Whether you are an engineer, scientist architect, manager of business reports, a writer or graphic artist, HIEROGLYPH integrates all information in your UNIX Universe. HIEROGLYPH combines all the elements necessary to generate the final composition. You have the option of camera-ready copy or multiple copies in Color or Black and White. HIEROGLYPH uses simple, single-stroke commands. Both text and commands can be in English or several foreign languages. The user manual is written for three levels: New, Experienced and Expert. Regardless, you can be doing productive work the first day. HIEROGLYPH software incorporates Text processing, Document Aids, Document Filing, Graphics, Data Handling and Production Tools. Together they comprise the most productive system for creating and producing superior reports—documents that make lasting impressions.
PRESCIENCE
HIEROGLYPH is the product of Prescience, Inc. (pronounced pres-ce-ence) 820 Bay Avenue, Suite 300, Capitola, California 95010 (408) 462-6567.
HIEROGLYPH is referenced in the following publications: AT&T UNIX System V Software Catalog, CONVERGENT TECHNOLOGIES Software Catalog, ENVISION Independent Sofware Supplies Catalog, PYRAMID Prism Program, SUN Microsystems Catalyst Program.
Please circle Ad No. 33 on inquiry card.

the Italtel, the Italian state telephone monopoly. In its first year, the agreement will come near or even exceed the lucrative multimillion pact the firm signed earlier this year with GTE Corp.'s Business Communications Systems operations. The GTE contract calls for the number two telephone company to accept more than $140 million worth of Sydis VoiceStation 1 systems during the first year. But that's not all. Several more lucrative deals with major U.S. OEMs are said to be on the way.

GOULD SOFTWARE?

You've probably been noticing those startling dragon ads from Gould, Inc.,'s Computer Systems Division touting its powerful Unix system-based supermicros, minis, and mainframes. If you thought the company was only in the hardware business, however, you'd be dead wrong.

This fall, the little-known Gould Software Divison, located near Chicago, will bring out a set of Unix system program development productivity tools. These will include a C compiler; a cross development system for C, Pascal and Fortran; and the Unicellerator, which is said to increase a VAX or Gould Concept 32 supermini's performance by up to 65 percent when running 4.2BSD.

It is another product, though, that may prove to be one of the most interesting, particularly to end-users suffering through the lack of applications software for the Apple Macintosh. The firm plans to bring out a 68K/Macintosh Cross Development System that is purported to allow software developers to compile source code on Unix system-based hosts four times as fast as those done on an Apple Lisa-based development system. All those anxious (and probably bored) Mac owners probably just can't wait.

HEARD WANDERING ABOUT THE HALLS

Sage Computer Inc., an M68000/UCSD p-System supermicro vendor, will reportedly join the Unix system bandwagon by the end of this quarter.... National Semiconductor has had to recast the die for its vaunted NS32032 32-bit microprocessor. That's bad news for some Unix system startups, many of whom are using the National chip as the souls of their machines.... DEC-compatible disk drive maker Qualogy of San Jose, Calif., formerly known as Data Systems Design, is reportedly at work on a Unix System V supermicro.... AT&T and IBM plug-compatible mainframe maker Amdahl Corp. may be teaming up in some joint marketing deal.

ON THE RECORD

Zilog's director of marketing, Richard Rubin, when asked at a recent press conference to compare the firm's standing in the supermicro market against the competition: "We subscribe to a number of market research services, all of which believe they have walked down from the mountain with the tablets. Depending on whose numbers you use, we are in the top three, top five, or top seven." □

Gould...Innovation and Quality in UNIX-based Systems

The Firebreathers from Gould blast the competition into oblivion.

With blazing performance.

Great leaps in raw performance are rare in the computer world. Usually, changes occur in increments of half-a-MIP or so. Now real Firebreathers roar into the arena, quadrupling the best the competition has to offer.

With scorching speed.

These creatures don't run a little faster. Running real production code supplied by VAX™ users, our PowerNode™ 9000 benchmarks at 4.5 times faster than the VAX 11/780...at a comparable price! Even our second-in-line PowerNode 6000 runs 1.5 times faster. But with all this power and speed, the PN6000 has a dainty footprint...60% less than the VAX 11/780.

With sizzling simplicity.

Even the hottest firebreather won't fly if it's hard to run. Ours are UNIX*-based, so ease of use and compatibility become part of the power. Plus we offer the "Compatibility Suite", application software packages which are consistent across our entire Gould PowerSeries™ product line...the widest range of UNIX-based systems in the world.

More than a myth.

We know these claims sound outrageous. Make us prove them. Give us your benchmarks and production code. We'll show you numbers that surpass all expectations. Call 305-797-5459 for a test flight on the Firebreathers. Mainframe performance disguised as a supermini. The scorching performance you expect from Gould.

Gould Inc.,
Computer Systems Division
Distributed Systems Operation
6901 West Sunrise Boulevard
Ft. Lauderdale, FL 33313

*UNIX is a trademark of AT&T Bell Laboratories
™PowerNode and PowerSeries are trademarks of Gould Inc.
™VAX is a trademark of Digital Equipment Corp.

GOULD
Electronics

Please circle Ad No. 16 on inquiry card.

mail

This space is yours. Tell us what we're doing right—or wrong. Raise issues and ideas that other readers may respond to. Propose projects. Reject market trends. Take a stand.

ONE FOR OUR READERS

Dear Editor:

At McDonald's Corp., the International Business System runs its application software on several different micro- and mini-computers, thanks to the "portable" Unix operating system. Although all processors use the same type of cartridge tape drive, these tapes are incompatible. This prevents the physical transfer of data between processors of different manufacturers.

There are several advantages to using cartridge tape, including: (1) the drive is normally included in the price of the hardware, since a backup system is required; (2) cartridge tape is less expensive than magnetic tape; (3) cartridge tape is easier to use; (4) cartridge tape is easier to port.

Twenty years ago a similar problem existed with the nine-track magnetic tape. Standardization was achieved by making the tape drive compatible with IBM. Today, in order to make cartridge tape drives compatible between varying hardware, the same need for standardization exists.

What would it take to use cartridge tape as a physical medium to transfer data from one hardware to another hardware? We welcome any comments.

Please direct comments to: Mo Shahmirzadi, McDonald's Corp., One McDonald's Plaza, Oak Brook, IL 60521.

Sincerely,

Martin J. Moderi
Staff Director

A BLURRED IMAGE

Dear Editor:

While we appreciated being mentioned in Vol. 1, No. 3, in both the *On-line* section and the Honeywell article, there were several inaccuracies that require clarification. First, Quadratron is based in Encino not in Torrance. A view from Silicon Valley does blur the image of Southern California, but we appreciate the distinction. Second, we are not a spinoff of Fortune Systems. We had been working on office automation before the creation of Fortune, and while Stefan Zimberoff did leave to write the Fortune word processor in-house and return, it was not as has been described.

Yes, the implication is correct that Quadratron's office automation series, Q-OFFICE, is the product of choice by more Unix hardware manufacturers than for any other application package. It is also time that it is used on more proprietary operating systems and MS-DOS-based systems than Unix.

We appreciate the ability to clarify these issues.

Sincerely,

Karl Klessig
President
Quadratron Systems, Inc.

NCC —APPEARANCES WERE MISLEADING

BY OMRI SERLIN

The general media consensus of the 1984 National Computer Conference (NCC) was that the show was "subdued," meaning no flashy introductions of major new products and relatively tame after-hours bashes. But appearances can be misleading. This NCC was a landmark in several senses.

The two side-by-side demonstrations of *multivendor networking,* using approved IEEE 802 and proposed National Bureau of Standards (NBS) standards within the framework of the ISO Reference Model, were highly significant. Although only a relatively unimpressive file-transfer application was shown, the fact that so many vendors have agreed to invest the not-insignificant effort to make this possible is impressive in itself.

Interestingly, the immediate commercial impact will occur not in the office environment, where local networking was supposed to be heading, but on the factory floor. General Motors tantalized suppliers with promises of future RFQs for something like 150,000 programmable controllers and factory floor computers before the decade is out.

All such RFQs will apparently call for compliance with GM's MAP (Manufacturing Automation Protocols), a planned full implementation of the ISO Reference Model. MAP is using the IEEE 802.2-802.4 token-bus LAN standard at the Layer 1-2 levels and is evidently heading toward adopting the proposed NBS internet and transport protocols for Layers 3 and 4, respectively. At its press conference at NCC, GM left little doubt that the standards train is leaving the station, and those who plan to catch a later one do so at their own, grave risk.

A glimpse into the explosion of activity in *multi-microprocessor* (MMP) systems was afforded NCC goers in the session I had the honor of chairing. The media again missed the significance of this nascent development. As more and more computing power finds its way to the individual user's desk, the role of central multi-user machines is changing, from that of time-shared providers of CPU power to that of custodians of departmental and plant-level databases and hubs of electronic mail systems.

In this type of application, commonly referred to as "on-line transaction processing" or OLTP, the new architectures are much better suited to carry the load. A good many of the new MMP vendors also provide fault tolerance. With businesses becoming more reliant on on-line systems for their daily functioning, fault tolerance will become more important, especially as corporate management begins to realize the extent of escalating downtime costs.

THE BIG SHOWDOWN BEGINS

Finally, this NCC saw the beginning of the vaunted IBM/AT&T confrontation. Somewhat surprisingly, the key battlefield isn't the hardware. (So far AT&T has little to offer that is notable in that area.) The really significant war is being waged around the Unix system. What AT&T is actually trying to do with its horrendously costly "Unix-as-a-standard" campaign is to force IBM to recognize the Unix system as a standard. And there are signs that AT&T may be winning.

NCC keynoter (and IBM President) John Akers' references to the Unix system, in both his NCC press conference and in an earlier interview with *Electronic Business* magazine, suggest that the issue of what to do about the Unix system has reached IBM's highest management levels.

At the same time, various IBM independent business units (IBUs) and departments continue to offer a variety of Unix dialects, capped recently by the VM/IX offering for the IBM 4300 and 308X mainframes. This offering is only a PRPQ (custom job), is bound by a host of limitations, and is underwritten by the relatively unknown Information Programming Services (IPS) operation.

However, when viewed in conjunction with PC/IX (also from IPS), the Series/1 Unix offering from the Telecommunications Carrier Products IBU, and Xenix for the IBM 9000 from IBM Instruments, the shape of a trend may be in the making.

Will "mainstream" IBM ever endorse the Unix system? Given a choice, IBM would clearly rather continue to ignore it, hoping it will fade away as other fads before it have. The trouble is that AT&T's highly visible campaign is threatening to establish a demand for the Unix system at the end-user level.

YOU CAN'T PREDICT THE FUTURE.

BUT YOU CAN BE PREPARED FOR IT.

WITH INFORMIX AND File-it!

THE FIRST DATABASE SOFTWARE FAMILY FOR UNIX AND MS-DOS.

Now OEMs and systems integrators can sleep better at night. Because one company has taken the worry out of buying the right software.

RDS.

The company that produces a family of database software designed to take on the future.

Incompatibility is a thing of the past.

INFORMIX® and File-it!™ are compatible with UNIX,™ MS™-DOS, PC-DOS,™ and PC/IX systems (over 60 micros and minis* at last count).

INFORMIX is a true relational database system designed to take full advantage of the power of UNIX. It includes the most widely used report writer on the market.

Then there's File-it! The first easy-to-use UNIX file manager. Together, they have the flexibility to accommodate novices and experts alike.

INFORMIX and File-it! are fully integrated. Users can upgrade from File-it! to INFORMIX or access data from one program or the other without re-entering data, retraining employees or reprogramming.

Applications can also be moved from MS-DOS to UNIX and vice versa without having to rewrite the application.

Simplify program development.

RDS offers C-ISAM,™ the de facto standard ISAM for UNIX. It's a library of C subroutines with a B^+-Tree based access method that stores, retrieves and modifies data from indexed files. It's embedded in INFORMIX and File-it! Or is available as a standalone product.

Software good enough for AT&T.

AT&T, inventor of UNIX, has co-labeled INFORMIX, File-it! and C-ISAM to run on their full AT&T 3B Computer line (from micros to minis).

Hewlett-Packard, Altos, Zilog, Siemens, Cromemco, Perkin-Elmer, Sydis and General Automation have selected RDS as well.

In fact, INFORMIX has an installed base of over 6,000 copies. And RDS has sold over 35,000 licenses for all their products to date.

But before you make up your mind, check the facts one more time.

There's only one database software family that's UNIX-, PC-DOS-, MS-DOS- and PC/IX-based. It runs on more than 60 systems. And it's ideal for both novice and expert.

Now it doesn't matter where the future's headed. You're already there.

*RDS products are available for the following systems:

Altos 586, 986, 8600, 68000	Hewlett Packard 150, 9000 Series 200, 9000 Series 500
Apollo DN300	IBM PC, PC-AT, PC-XT
AT&T 3B2, 3B5, 3B20, AT&T Personal Computer	Intel System 86/380, 286/310
BBN C machine (all models)	Masscomp NC 500
Bunker Ramo Aladdin 20	Momentum Hawk 32
Charles River Data Systems Universe 68	NCR Tower
Convergent Technologies Miniframe and Megaframe	Onyx C8002, C8002A
Corvus Systems Uniplex	Pacific Micro Systems PM200
Cromemco System 1	Perkin-Elmer 32 Series, 7350
DEC 11/23, 11/34, 11/44, 11/60, 11/70, VAX 11/730, 11/750, 11/780	Pixel 100/AP, 80 Supermicro
Dual Systems System 83	Plexus P/25, P/35, P/40, P/60
Fortune 32:16	Pyramid Technology 90X
Forward Technology 320	Radio Shack Model 16
General Automation Zebra (all models)	SCI Systems IN/ix
	Silicon Graphics IRIS 1400
	Visual Technology 2000
	Wicat Systems
	Zilog System 8000 (all models)

Demos of INFORMIX and File-it! are available. Demonstration software and complete manuals included.

RDS™ RELATIONAL DATABASE SYSTEMS, INC.

See us at COMDEX

2471 East Bayshore Road, Suite 600, Palo Alto, California 94303
(415) 424-1300 TELEX 467687

INFORMIX is a registered trademark of Relational Database Systems, Inc. RDS, File-it! and C-ISAM are trademarks of Relational Database Systems, Inc. UNIX is a trademark of AT&T Bell Laboratories. MS is a trademark of Microsoft and PC-DOS is a trademark of International Business Machines.

Clearly the IBM units mentioned above have already felt this demand and have decided to respond on their own.

Historically, IBM's key asset has been its ability to read the market and to respond to its needs. It has recently become much less partisan and much more pragmatic, offering products not made by IBM (a sacrilege only a few years ago). As long as AT&T's pocket continues to stand the strain of stirring the Unix system caldron, IBM will continue to be under pressure; with PC/IX and VM/IX, Big Blue is testing the waters. As John Akers puts it, "If an operating system is successful in the marketplace, it makes sense for IBM to offer it."

Of course, he was referring to MS-DOS, a product from a small company that IBM didn't need to fear. The Unix situation, however, is complicated by AT&T's enormous size and its recently adopted posture as a direct IBM competitor.

Still, IBM can be relied upon to make a rational business decision to go with the Unix system, in one form or another, when it is convinced that end-users really want it and that offering the Unix system will be more profitable than ignoring it.

SHORT NOTES

Alpha Micro (Irvine, Calif.) will run Unix System V on its M68010-based multi-user supermicros, formally introduced at the NCC. In the past, A-M stressed its proprietary AMOS operating system to its vertical market VARs.

Convergent Technologies (Santa Clara, Calif.) has written off its lap-top Workslate, resulting in an $8.6 million charge against the June quarter, which ended with a whopping $6.7 million loss ($0.27/share) on $85 million in revenues. CT's stock recovered almost $2.50 (to $13) on the news, reflecting relief among investors worried about the impact of continued Workslate efforts.

CompuPro (Hayward, Calif.) introduced at NCC an NS32016-based multi-user supermicro, beginning at $12,495 for 512 Kbytes with a 1-Mbyte "solid state disk" and dual floppies. Initially running FORTH, the machine is eventually expected to run Unix System V.

Cromemco (Mountain View, Calif.) introduced at NCC a 10-MHz M68000-based multi-user supermicro running Unix System V with Berkeley's C-shell, `termcap`, `uucp`, and `vi`. Prices begin at $9,995 for a 512-Kbyte machine with a 50-Mbyte hard disk and a 390-Kbyte floppy.

Microsoft (Bellevue, Wash.) reported revenues of $100 million for the fiscal year ended June 30, 1984. The privately held company did not disclose earnings.

A color version of Microsoft's Windows has been demonstrated at NCC on Tandy's 2000; however, general deliveries of the Windows package are apparently still substantially behind schedule.

Morrow (San Leandro, Calif.) introduced at NCC its 10-MHz M68000-based Tricep supermicro. The machine runs Unix System V but can also accept an Intel 80188 board to run MS-DOS. Prices begin at $9,000 for a 512-Kbyte machine with a 16-Mbyte Winchester. Morrow will also market a portable made by Vadem, a startup founded by ex-Morrow staffers.

Masscomp (Westford, Mass.) has filed with the SEC to go public. The firm, now three years old, supplies engineering workstations and laboratory computers; it hopes to sell two million shares at $11 to $13 each. In the fiscal year ended June 30, 1984, the company lost $1 million ($0.02/share) on revenues of $22 million but earned $345,000 ($0.03/share) on revenues of $8.8 million in the fourth fiscal quarter. Underwriters are Morgan Stanley and Hambrecht & Quist.

NBI (Boulder, Colo.), which previously specialized in word processors, has established a technical products division to market general-purpose supermicros. The division is headed by vice-president and general manager Glenn Edens (ex-GRiD). Its first product, introduced at the NCC, is U! (U Factorial), an 8-MHz 68010-based desktop running a multi-window version of Berkeley 4.2 Unix. With 1 Mbyte, a high-resolution display (1024 x 768), and a 24-Mbyte hard disk, U! retails for $15,495.

Meanwhile, the company announced a further 6-month delay in the introduction of the IWS workstation, originally announced in April 1983. Problems with the multitasking software are apparently the main stumbling block because the "new" U! and the "old" IWS are basically the same product.

SCI Systems (Huntsville, Ala.), which gained wide attention as supplier of the IBM PC main board, officially unveiled at NCC the SCI 1000, an Intel 80186-based, multi-user supermicro. The machine will run a Unix System III port done by

In microcomputers today, UniSoft sets the standard.

AT&T has recently been advertising that their UNIX™ operating system will be the standard OS for microcomputers. That's true. But if you want AT&T's UNIX software on micros today, talk to UniSoft Systems.

UniSoft has been delivering AT&T's UNIX adapted for 68000-based microcomputers for two years. More than 75 different computer systems run the UniSoft software, UniPlus+.™ At each Bell release level, all these systems are object code compatible. This means that applications software developed on any UniPlus+ system will work on any other. This is where software portability pays off.

UniSoft enhances Bell's vanilla UNIX with the best features from the Berkeley BSD research version of the UNIX operating system. IP/TCP networking, record and file locking, and virtual memory from UniSoft turn UNIX into a commercial product. All this added value is still Bell-compatible.

Don't wait six months to get System V running on your hardware. UniSoft's customers can ship it now.

If you're building or selling a 68000-based UNIX system, your operating system should come from UniSoft Systems, the UNIX experts.

UniSoft
SYSTEMS

THE BERKELEY PORT AUTHORITY

739 Allston Way, Berkeley, CA 94710 • (415) 644-1230
TWX II 910 336-2145 • UUCP ucbvax!unisoft!unisoft

*UNIX is a trademark of Bell Laboratories

Please circle Ad No. 53 on inquiry card.

Interactive Systems (which also supplies the PC/IX and VM/IX to IBM). Much of the speculation about an IBM 80186 machine may have been based on the SCI 1000 system. The SCI 1000 retails for $8,995, with 512 Kbytes and a 16-Mbyte Winchester.

AT&T Bell Labs (Summit, N.J.) staffers recently described a set of modifications to the Unix system that allow it to run on the dual-processor 3B20A model. The modifications, consisting mainly of internal kernel semaphores, are general enough so multiple-processor configurations could be accommodated as well. Work is also under way to integrate DMERT, the operating system for the AT&T 3B20D, into the standard Unix kernel.

Figures floating around at NCC suggest that, of the estimated 100,000 Unix system installations worldwide at the end of 1983, 80,000 were Microsoft's Xenix (of which 22,000 were on Tandy 16/16B) and 15,000 on various Altos machines. Interestingly, AT&T is estimating the total 1983 installed base at 70,000. □

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

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

Portability has been an important part of the Unix system's popularity. But what does porting require and why are ports difficult?

WIZARDS' WOES: THE PROBLEMS OF PORTING

BY VANESSA SCHNATMEIER

Portability—the biggest buzzword in the Unix operating system's mystique. How you define portability probably depends on what part of the computer world you hail from. To some sections of the industry, a portable system might be "a system that costs you less to port than it does to write it yourself." To an independent software vendor, code is portable if it takes no more than two to four hours to put up on your system.

However you label it, the idea and practice of portability have been central to the development and sudden surge in popularity of the Unix system. The system's relatively portable nature and its readiness

ILLUSTRATION BY JEF GUNION   PHOTO BY RALPH COOKSEY-TALBOTT

both to be transported from processor to processor and to accept a wide range of C programs have almost spoiled other systems for Unix system programmers, many of whom may get itchy about a port that takes as long as, say, a week. But portability is a reason to work with the Unix system or its look-alikes, not just an end in itself.

Porting the Unix operating system means changing the Unix source code so that it works on another type of processor. (This means more when you remember that the Unix system kernel alone takes up about 15,000 lines of code.)

According to Jeff Schriebman, president of UniSoft Systems—a firm that has done about 80 ports in the last three years—to "port" may take anywhere from two weeks to a year. The amount of work varies directly with the difference between the original system's type of CPU and that of the new system, the memory management unit that works inside or beside the CPU, the number and type of device drivers needed, and the utilities desired.

WHAT PORTING REQUIRES

Porting the Unix kernel usually only requires that 20 to 30 percent of the code be rewritten to suit a new microprocessor. This includes fine-tuning the kernel's operation with the memory management unit and the peripherals. The remaining 70 to 80 percent of the kernel code should remain unchanged, Schriebman says, adding, "The majority of the source code is the same for a VAX as it is for a Bell 3B20 or for a Motorola 68000."

But with a widely different CPU, porting the Unix system to a new microprocessor entails not only remodeling the kernel, but also the C compiler and a host of other tools, such as an assembler, loader, debugger, stand-alone monitor, and stand-alone device drivers for the target system, and programs to download software onto the new machine. The C compiler alone can take from six months to a year to port.

This brings up another kind of portability—the portability of applications from one "flavor" of the Unix system to another. Think about it: The "official" versions of Unix include AT&T's Version 7, System III, System V, System V Release 2, Berkeley 4.1BSD and Berkeley 4.2BSD. Some people even list Microsoft's Xenix with that group.

An incomplete list of Unix look-alikes includes MORE/bsd from Mt. Xinu, Eunice from Wollongong, HP-UX from Hewlett-Packard, and OS/x, a dual port of 4.2BSD and System V from Pyramid Technology. "How much time will it take a software vendor to port its software from one version of the Unix system to another?" the vendors ask with quaking knees. The answer—it depends, but maybe not as long as you thought.

WHAT MAKES A PORT DIFFICULT?

The first problem may come up not with the Unix system, but with the language it was written in, C. "C is a very portable language, but people's C compilers don't conform to any specifications," says Schriebman. There is currently no standard for C, though ANSI's X3J11 C standardization committee is currently expected to produce at least a preliminary standard within a year or two.

Another "difficulty" with C is that it has nothing built-in in terms of I/O, math, or string management —characteristics that are part of other languages. Larry Rosler, supervisor of language systems engineering at AT&T and also a member of the ANSI C standardization committee, says this state of affairs is likely to continue for the foreseeable future.

Rosler says that while the standardization committee's first goal would be, of course, portability, "the number two goal was to preserve something that we all agreed was called 'the spirit of C.' " Rosler added, "The designer of the language, Dennis Ritchie, decided very early on that since C was designed as an implementation language for operating systems, designed to replace assembly language, to impose artificial penalties on it would make it less attractive for that purpose."

Once they have stopped finagling with C itself, those porting the Unix system to another machine run into another set of issues. There is always going to be *some* machine-dependent code in the system. Some code, usually that in the kernel, is of necessity machine dependent, and you'll have to redo it in any case. Utility programs, however, may also hide various forms of machine dependencies, and these must be weeded out before completing the port.

Some examples of specific machine-dependent questions: Byte size has *almost* been standardized at 8 bits per byte. There are a few oddballs, such as Honeywell's and Univac's 9-bit byte and CDC's 10-bit byte, but for the vast majority of hardware, 8 bits is the standard. The standard character set used within Unix implementations, says Rosler, is ASCII—many applications simply don't port onto an EBCDIC implementation.

Sign extension is still a problem. The issue arises when you address a character and treat it as an integer. In the ASCII character set, the 8-bit set really translates to 7 bits when you count the sign bit at the beginning that lets the computer know whether the integer is positive or negative. This won't affect you if you're working only with characters.

In C, however, this explicitly violates the standard, says Rosler, because in C every character must be representable as a small positive number between 0 and 127 decimal.

A STICKY WICKET

It gets sticky when you're dealing with 1-byte-long characters and trying to store integer values in those characters. "That's a problem that really reflects a deficiency in the C language," Rosler says. "There is no way of saying portably in C, 'this 1-byte-length object is really an integer.' "

The industry has standardized on three lengths of integers in C—the long integer (32 bits), the short integer (16 bits), and the plain integer. (If it matters, you pick whichever one is correct; if it doesn't, you let the compiler choose whichever one is most efficient.)

What's not in the language now is a way of saying, "This 8-bit character is an integer and should be treated as having a sign." Rosler says the C standardization committee plans to deal with this by extending the language. Right now, though, any integer of that type will have to sign-extend somehow, via the compiler, whether the hardware supports it or not.

Floating-point arithmetic has no standards yet, neither does the order of evaluating function arguments—that's another price for preserving "the spirit of C," says Rosler. It depends on the order in which the argument stack grows in the particular machine, and to make the compiler writer specify which order is correct "would be a severe penalty on several architectures."

The way characters are addressed in words remains nonstandardized. A 32-bit word in a Motorola 68000 processor may address words beginning with the leftmost byte. On another machine, points out Schriebman, the machine might start with the rightmost byte.

Another glaring area in which many Unix systems lack any firm standard is in the handling of terminals and display screens. The way the Unix system can deal with hardcopy terminals such as printers has been established since the late 1970s—you send the stream of single characters (because the Unix system is oriented not by screen but by character), and there are a limited number of control characters (such as carriage return and bell) for the system to deal with. All very simple. But nowadays that kind of terminal has given way to the cursor-addressable (or "smart") terminal, which may have its own 80186 processor to aid with local editing and data storage. Here portability falls down.

Berkeley's 4.2BSD Unix tackles this weak spot with the `termcap` database, which helps users write terminal-independent software and create their own terminal descriptions to link the local Unix system with their particular brand of terminal. Another package from Berkeley, called `cursive`, also comes close to a standard for dealing with disparate terminals, says Rosler.

Estimates on how long it takes to port a system can vary greatly.

David Mosher, technical manager for experimental computer science at UC Berkeley, describes what `termcap` can do: "When you get to more sophisticated terminals, it becomes an interaction problem between what the processor wants to do and what these things are capable of doing. You end up having to restructure the editor so that you can use these sophisticated commands and then go do the 'dumbified' conversion to deal with dumb terminals. When most of these things were designed, the only things we had were dumb terminals—essentially what we've done is allow every smart terminal to act as much like a dumb terminal as it can."

Terminals, networking, and interprocess communication are all areas the Unix /usr/group is now addressing, and Rosler says something should be coming from the committee within a year or so.

HARDWARE PORTS

In a healthy display of enlightened self-iterest, AT&T is doing its part for hardware porting with a program it instituted earlier this year: a program of so-called generic ports to several major microprocessors. AT&T is working closely with Motorola, Zilog, Intel, and National Semiconductor to get their versions of an overall port to each company's type of processor.

According to Rosler, AT&T will gather up the information it receives from the reports done in these projects and include the resulting source changes into a single source, which will probably be the source for one of the processors AT&T supplies—a portable source, they hope.

"We are on a project of consciously making our source more and more portable as these ports go on, of buying back the changes, and making this so-called generic port," Rosler said. "Ultimately, when an independent software vendor or an OEM wants to buy Unix source from AT&T in order to port it, the source that they will get will be [the generic port], which doesn't deal specifically with any particular processor."

Rosler wouldn't say directly how much work people would have to put in to adapt a "generic port" to their own machines. He said that this hinged on the definition of portability: "The loosest definition of portability is that it's less expensive to use than it would be to write it yourself. Then it becomes a question of degree. You measure the portability by how much it costs you, how much effort it takes to adapt something rather than re-create it."

The C compiler alone can take from six months to a year to port.

As Schriebman would be quick to point out, time estimates can vary wildly on how long it takes to port a system between machines. UniSoft, which over its relatively short life span has honed the art of porting down to a fine science, can do a simple port on easy systems in from two weeks to two months.

One new development in porting may be a new "Unix porting kit" soon to be offered by Palomino Computer Systems. This kit purportedly will allow almost every 68000-based hardware vendor to do its own port in from one to two months! The kit is scheduled for release in the first or second quarter of 1985 and supposedly will be cheaper than UniSoft's UniPlus+.

Schriebman, when asked about the kit, replied that they'd found that, while those experienced with porting can put together a port in two months, when a firm does its own port, even with help from Unix wizards, the port tends to take from nine months to a year. "They'll see," he shrugged.

COMPATIBILITY AMONG UNIX VERSIONS

Despite the furor about portability and the differences between particular flavors of the Unix system, there's less separating the flavors than you might think. (We hesitate to say it's all a matter of taste.)

In fact, both Berkeley and AT&T go out of their way to ensure backward compatibility. Mosher says, "4.2 has a 4.1 compatibility mode—you can transfer into 4.1 and the operating system will allow 4.1 programs to continue to run. And that allows users to advance from 4.1 to 4.2 in their own time scale. Then the system administrator at some point can make a deadline and say everybody has to convert their programs to 4.2 by this time."

Rosler says there really isn't much to worry about in porting software from System III to System V or to System V Release 2. "The capabilities of our C compiler changed a little bit to allow names of any length to be significant, but this is something where a properly written program wouldn't have cared. In fact, we went so far in Release 2 as to have a compiler option which said 'truncate all the names the way they were in Release 1.' We kept to our promise of full backward compatibility by doing that, even though we don't think that's the way programs should be written."

Sue Picus, supervisor of software generation systems development at AT&T, says that most of the changes between the AT&T Unix versions are additional capabilities that don't necessarily cause portability problems. Of course, if you use some of the new significant features, you can't go home again. You can't then go back to System III from System V without changing your application code.

Mosher goes even further, saying that the only major divergences between AT&T's System V and Berkeley 4.2BSD are in the areas of signals and I/O controls. "I/O control is one of those things that has a nebulous, endless list of arguments. No one agrees there's a central place to register your local fanfare. . . so that would need some careful mapping, to make sure to map between them to a single I/O mechanism. Our signal mechanism works like hardware interrupts, which people understand."

A couple of firms have taken advantage of the relatively narrow gap between the versions of Unix, such as Pyramid Technologies with its OS/x system, reportedly compatible with both System V and 4.2BSD. This is simple to do, says Mosher, as long as the company carefully tailors and routes around specific system calls to avoid trouble.

WRITING PORTABLE CODE

It is actually possible to write code that can pass muster on most versions of the Unix system on most machines *without* having to spend weeks slaving over a hot kernel. All it takes is some attention from the programmer.

"It's a little bit like folklore in some cases," says Dan Ladermann, vice-president of research and development for The Wollongong Group and a member of the steering committee for the /usr/group standardization committee. He adds, "Some of it is intuitive. You have to ignore your knowledge that you're writing this code for a 68000 or a VAX. People who write knowing that it's going somewhere else don't take shortcuts."

When Unix system people say portability, they may not mean the same thing as an everyday computer industry person. Says Ladermann, who worked for the Defense Department and Ford Aerospace and Communications before joining Wollongong, "A large industry looks at man-years to port a product. Under Unix, a man-week or a man-month is a lot of time."

Ladermann cited the case of a man who recently delivered a paper at a conference on how he had solved a particularly knotty portability problem: "The guy probably spent more time writing his paper than solving the problem."

Sue Picus says that "in the ideal world, what we're shooting for is code that can be ported without change, just recompiling and a new set of header files to hide all the machine-dependent information."

Rosler adds, "Some things simply are not going to fit into that kind of model. If somebody has written an application program on [System V] Unix that makes use of large amounts of memory or relies on a paging system to cover it, and he tries to port that application to an environment or processor that has a very small addressable memory, obviously that is not going to port. . . . I wouldn't call it a violation, but an assumption of capacity that simply isn't there."

Ladermann said that an experienced Unix system user would define as "portable" any program that could be taken off another Unix system-using machine and simply be recompiled in order to make it run.

"Maybe you change the parameter file," he said. A "semi-portable" program shouldn't take more than two to four hours to make it run. By this reasoning, says Ladermann, "unportable" is a program that takes all of several days to convert into a form suitable for the new machine or version of the Unix system. "And this is for a program that took up to six months to write," he adds. Very few application programs, he asserts, would take more than a week to whip into shape.

`lint` AND SCCS

The Unix operating system itself provides two tools for ensuring that a program will be portable—the `lint` and Source Code Control System (SCCS) utilities. SCCS lets you conveniently maintain different versions of a program in historical order; this program is a boon to software developers, who must juggle their products from system to system.

Usually, porting the Unix kernel requires rewriting only 20 to 30 percent of the kernel code.

`lint` examines a program's C source code for errors, inefficiencies, and nonportable structures. (In some other languages, this kind of program is actually built into the compiler.) `lint` won't necessarily prevent programmers from wreaking whatever havoc they choose regarding code portability, but, like the warning on a package of cigarettes, at least it sounds the alarm.

`lint` doesn't only serve as a kind of programmer's spelling and grammar checker, but also as a kind of "writing guide." It can detect if you've used a variable before you gave it a value, or if you've given a variable a name and never used it. These may not necessarily violate portability, but they may violate the intent of the program. Rosler compared `lint` to the Unix Writer's Workbench application that combs through a piece of writing and pinpoints phrases that might be trite or sexist and says: "They're all correct and legitimate, but maybe that's not what you really mean."

The idea of portability is inextricably linked with the ideal of standardization. Portability can only exist if everyone agrees to adhere to a set of basic guidelines (unless everybody is writing code for the same system). According to Alan Mason, a member of the European Unix Users Group at Heriot-Watt University in Edinburgh, Scotland, the goal is not necessarily "to set a standard that will be obeyed by every installation, but one which must be able to be implemented on any system."

The ideal environment, Mason said in the August 1982 issue of *CommUNIXcations,* would be one where users might easily move between different systems without perpetually residing at the low end of the learning curve. Both ANSI and /usr/group are attempting to deal with the issues of standardization and portability.

Mosher lamented the passing of the days that saw simpler relations between Berkeley and Bell Labs. Said Mosher, "At Berkeley, we essentially try to make our interfaces very much like what AT&T is doing, provided that AT&T is willing to show us what they're doing." He added that the UC group had formerly "coordinated very closely" with researchers at Bell Labs, back when the Bell Labs group "was actually producing the Unix distribution."

"We used to do joint research programs with them all the time," Mosher said, "so we knew exactly what was going on, and we would agree on certain things in advance. Unfortunately, with AT&T taking on Unix, they have an entirely different group, and they're not as receptive to this sort of thing as the Bell Labs people were."

Mosher says that at Berkeley "we've tried to change our interface to look as much like System V interfaces as possible." The lack of access to System V code has hindered them, he says, because they've been limited to what code is available in the literature. However, he says a new license agreement has recently been signed, and now that they can see the actual semantics of the code, "there'll be a lot more System V-type compatibility in any future systems that we release." □

A frequent UNIX/WORLD contributor, Vanessa Schnatmeier is a freelance writer who has written for several computer magazines. Her last work for UNIX/WORLD appeared in Vol. 1, No. 4. Her current interests include women's issues and computers.

Gould...Innovation and Quality in UNIX-Based Systems

Now Gould offers the widest range of UNIX-based systems in the world.

Gould's PowerSeries™ computers span the performance range. From professional desk-top computers and high-resolution graphics workstations to shared resource and back-end processors with up to twice the performance of a VAX 11/780.™ And, all with the UNIX* operating system to assure interconnectability and software compatibility.

In addition, there's utility. Gould PowerSeries computers are designed for the way you work. Users have all the advantages of a dedicated system plus the lower-cost-per-user option of sharing resources with Gould's offering of standard networking strategies including Ethernet.™

Computer users also need compatibility. Gould's years of expertise with UNIX software have produced the "Compatibility Suite" of application software packages that is compatible and consistent across the entire PowerSeries product line under our Universal Timesharing Executive, UTX,™ a unique combination of Berkeley 4.2 BSD with selected features from AT&T's System V.

And, for your service and support needs, you can rely on Gould's worldwide customer support network.

Put Gould PowerSeries Computers to work for you now.

Gould Inc., Computer Systems Division
Distributed Systems Operation
6901 West Sunrise Boulevard
Fort Lauderdale, FL 33313
1-800-327-9716

*UNIX is a trademark of AT&T Bell Labs
™PowerSeries and UTX are trademarks of Gould Inc.
™Ethernet is a trademark of Xerox Corp.
™VAX is a trademark of Digital Equipment Corporation

Please circle Ad No. 15 on inquiry card.

GOULD
Electronics

UNIX* + OPERATING SYSTEM

- Advanced Full-Screen Editor
- PC DOS File Transfer
- 8087 Co-Processor Support
- Queuing System
- IBM Support

Now you can have a complete UNIX System III implementation on most models of the IBM PC.

The IBM Personal Computer Interactive Executive† (PC/IX) offers you tools like the C language, programmer's workbench, communication facilities, a text processing system and much more. It's also a multitasking system that offers you the same facilities found on larger UNIX operating systems.

And PC/IX incorporates many significant enhancements. For example, a new full-screen editor helps you program more effectively. It offers such advanced features as windowing, function-key editing, the ability to execute commands from the editor, and automatic file backup. Beginning users of the editor can rely on a variety of "help" screens.

If you've been using PC DOS, you'll be glad to hear that it can co-reside with PC/IX. And that you can transfer files between the two systems.

PC/IX is specifically designed to take full advantage of the architecture of the IBM PC.

It transparently supports the 8087 Math Co-Processor. And PC/IX automates the management of input/output streams through a multipurpose queuing and spooling system.

IBM has other software for PC/IX

IBM's = PC/IX

available, too, including INfort (a FORTRAN compiler with programming tools) and INmail / INnet / FTP (an electronic mail and file transfer facility).†

And because PC/IX is from IBM, you get IBM's quality documentation, as well as IBM service and support.

PC/IX requires 256KB of memory on an IBM PC with fixed disk or an IBM PC XT, PC XT/370 or PC AT.

You'll find PC/IX at your nearest IBM Product Center or your Authorized IBM Personal Computer Dealer. To order, or for more information and the latest list of PC/IX application software, call IBM toll free at 1 800 IBM-2468, and ask for the PC Software Department, Ext. 390. Or call your IBM marketing representative.

IBM

IBM Direct — PC/IX — 11-84
PC Software Dept. 3E/390
1 Culver Road, Dayton, NJ 08810

Please send me more information on PC/IX.

Name____________________

Title____________________

Company____________________

Address____________________

City__________ State______ Zip______

Phone____________________

*UNIX is a trademark of AT&T Bell Laboratories. PC/IX is based on UNIX System III, which is licensed to IBM by AT&T Technologies, Inc.
†Developed for IBM by INTERACTIVE Systems Corp.

AIRBRUSH BY JEF GUNION

MAYBE WE CAN STILL KEEP THE NAME

BY GORDON W. WAIDHOFER AND MICHAEL SWEENEY

Take that favorite version of the Unix system, the one that has all the facilities you like so well, and move it from your M68000 system over to a PDP-11 without any changes. Boot it. If you get a log-in, AT&T would probably like to have a very long and lucrative chat with you. Don't take the first offer.

Now take a handy facility you like, Unify, for instance, or a spreadsheet you know, and move it down that same garden path. If it runs as is, you have portability; if it doesn't run, you don't necessarily have nonportability—you just have a pain in the OS and a problem.

It is that problem that makes portability a matter of degree instead of the "load-it-and-run-it" ideal that lurks in the hearts of marketers everywhere. The fact is that processor architectures differ, sometimes wildly. And even the same processor varies from one integration to another. Because of various implementation differences, an M68000 processor in one box is going to behave differently from the same processor in someone else's box. Even a small variance in that behavior means that a *truly* portable operating system is a pipe dream.

Moreover, the prospect of ever developing "true" portability in any operating system, language, or application software flutters around the dim edges of futility. There are too many "standards" and too many interested parties to permit it. Just in the short history of computing as we know it, there have been word sizes ranging from 4 to 64 bits, including some particularly odd sizes for special applications.

In the last 10 years we have seen micros evolve from 8-bit words to 32 bits with a variety of architectures and memory management schemes. It is probably a safe assumption that we will see a new generation of microcomputer architectures before today's software documentation gets dogeared.

If the only part you can keep is the name, you're talking development, not portability.

DEGREES OF PORTABILITY

We are forced, then, to deal with packages that are partially portable, somewhat portable, and easily portable. Even the Unix system, a purportedly highly portable operating system, calls for an effort on the order of two man-years or more to port it to a new machine. Perhaps an operating system could be developed in which all the machine characteristics could be set as variables, and perhaps it could be made intelligent enough to identify and set the variables by itself. Perhaps that will be feasible when memory comes in gigabyte chips and laser disc storage is old technology.

There are too many 'standards' and too many interested parties to permit true portability.

In practical terms, *portability* means only that a software package is capable of being altered so that it will run on more than one machine or under more than one operating system. Even typical usage of the term carries the tacit assumption of some amount of effort, some degree of change.

How much must be changed is what defines the degree of portability. That is, if it costs half as much to port something as it did to write it originally, it's 50 percent portable. If it costs more, it's not portable at all. Or, for those who object to putting such artistic issues in cold cash terms, if the only part of it you can keep is the name, you're not talking portability—you're talking about development.

Portability can be considered an aspect of software—its "portableness," so to speak. In turn, portability itself has a number of aspects: code, data, and skill, to be specific. Portable code, obviously, is of key importance in any discussion of portability. Without that, the other two are beyond hope since the nature of portability in the last two is somewhat different. Only the cost issue remains the same.

The "portableness" of data resides in the structure or format rather than in the data itself, which makes data portability somewhat paradoxical. Data is probably the most sensitive to varying architectures or operating system processes, but due to its regularity, it is also generally the easiest to port (because it can be done mechanically) and has the longest life span of the three aspects.

However, this doesn't change the fact that making radical changes in computer systems can still be a tremendously expensive and time-consuming proposition for an organization—a proposition to be approached with trepidation and a generous supply of Maalox.

Skill, as an issue of portability, is much less tolerant to change because significantly different architectures or machine characteristics may require a widely variant style of programming. At the extreme, the user may need to learn an entirely new language or a new command vocabulary. It is in this area that the Unix system shines (at least relatively).

If the user can move directly from a micro to a Cray, key in a series of commands, and get the same behavior, there is portable skill. And in this case, the benefits accrue to both the organization and the software engineer since capabilities are expanded and training costs are reduced.

THE PROBLEM'S PITH

And there, in the words *costs are reduced,* lies the crux of the issue. Portability is economically driven. Its purpose is to avoid obsolescence (particularly of data and skill) because with obsolescence comes expense.

On the other side of that same economic coin, the more versions of a package a software house can offer, the broader the market and greater the potential sales. In addition, a given software package or system will have a longer market life span if loyal users can take advantage of new, more powerful hardware without having to buy new, more expensive, and more complex software from some other company.

This point leads to the inescapable conclusion that portability is much more important to developers than to users. But the possibility exists that in striving for portability, developers do themselves and their users a disservice. After all, the purpose of developing software is to provide functionality. Functionality (performance plus effectiveness) is the trade-off for portability.

This relationship can perhaps be illustrated by applying some systems theory principles. Systems theory holds that systems (an organism, for instance) that are highly specialized to function efficiently in a particular environment will function *less* efficiently (if they can survive at all) when the environment changes significantly. On the other hand, systems that are highly adaptable (not as specialized) have a better chance of surviving an environmental shift but don't function at high efficiency in either environment.

Since the effect of porting a software system from one machine to another is to change the environment, we can generally expect that we will have to make the package highly specific to the new environment in order to maintain efficiency. Otherwise, we can expect it to be less efficient—if it can survive the shift to begin with.

A case in point is the widely reported situation with Lotus 1-2-3. When first developed in Pascal, a portable language, the package was painfully, unusably slow. When rewritten in assembly language and made specific from machine to machine, it became a trend-setting, very valuable package. The lesson here is that if portability means significantly degraded performance, then portability itself has no value.

BARE TREES?

Yet, portability may prove to be a dead branch on software's evolutionary tree. If anything, we are further away than ever from standardization of hardware architecture or system software characteristics. Also, a trend developing in hardware manufacturing is to put more and more proprietary microcode instructions in read-only memory (ROM), instructions that carry a great deal of the power of the machine or even the operating system.

The best that other developers can hope for, then, is to emulate the functionality of the hardware/software system. If this trend continues, it may eventually become impossible to effectively port a different operating system or even applications software to a particular machine. Portability would then become a hollow echo of someone's Utopian dream, a "wouldn't-it-be-nice-if . . ." topic for late nights in the computer room.

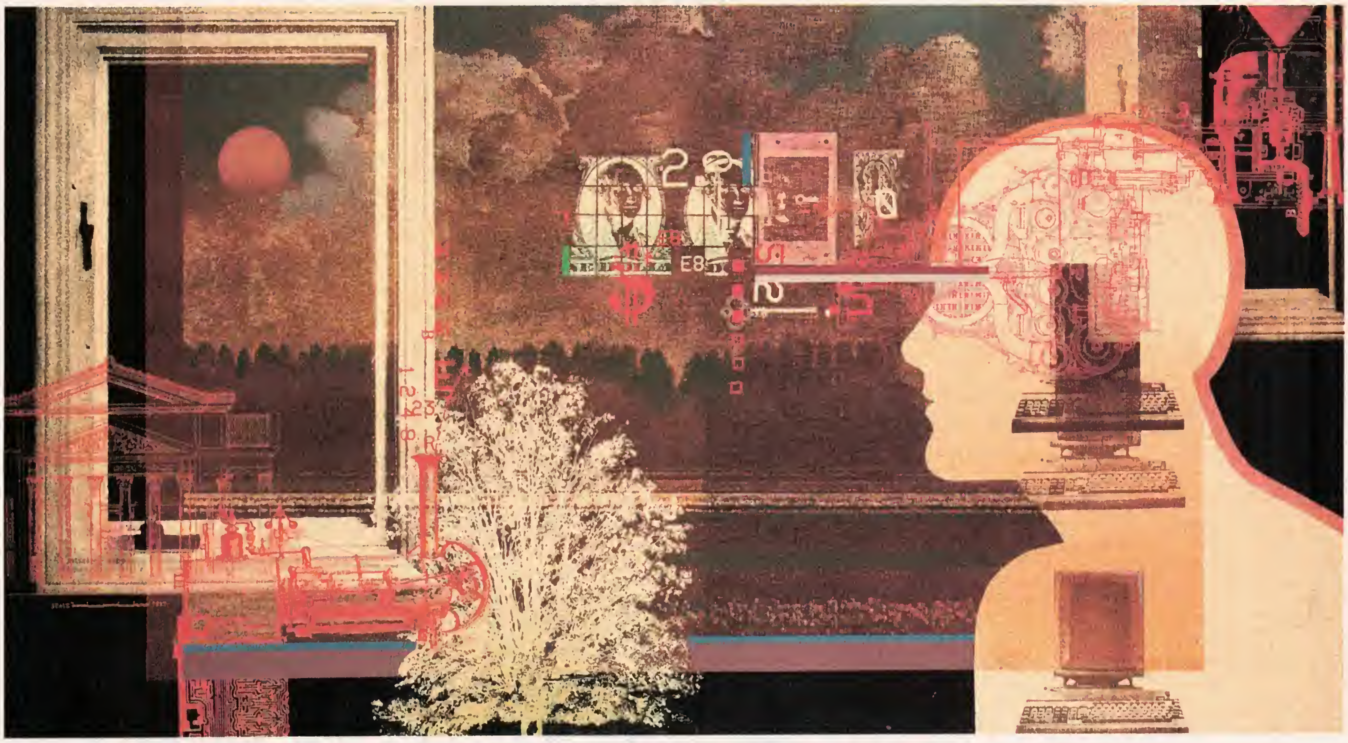
We in the software development community would do well to remember that portability is now selling computers, and there is already an existing demand for the Unix system, no matter whose box it's on. It would not be the first time that a company saw an opportunity to sell hardware to people who wanted a specific kind of software.

As we go about improving the Unix system and making it more efficient for specific machines and applications, we might also remember that we run the risk of improving ourselves out of an emerging standard. It is we who must find the point at which portability does not overshadow functionality as a goal. It is we, the software development community, who must guard against overstepping that point. □

Gordon Waidhofer is the director of software services at Voelker-Lehman Systems in Fremont, Calif. He has been an independent consultant on Unix systems software and previously was a software engineer with The Wollongong Group. Michael Sweeney, a former market communications analyst with Rogers, Kirkman and Associates, is director of creative services at Voelker-Lehman.


SERIX
ANNOUNCING SYSTEM V UNIX™
THE CHARIOT CUP 08-22 Runner's World
...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
Please circle Ad No. 6 on inquiry card.

WHY BUY A DATABASE SYSTEM?



TO BUILD APPLICATIONS!

PROGRESS™ is the high-performance database system and fourth-generation application development language for multi-user microcomputers.

PROGRESS gives you all the flexibility and control of structured programming languages, but uses plain English syntax for simplicity and ease of customization. You and your staff will learn to use PROGRESS quickly and benefit immediately from its state-of-the-art features.

Why build business applications in C, BASIC, or COBOL when PROGRESS will increase your productivity 10 to 40 times? PROGRESS makes this possible by integrating these powerful features:

- English-like application language
- Flexible relational DBMS
- Integrated data dictionary
- Automatic screen and report formatter
- Full-screen syntax-checking editor with on-line help

Once your application is built, PROGRESS offers:

- High performance
- Comprehensive multi-user capabilities
- Automatic database recovery
- Easy modification and customization
- Ad hoc query and report-writing capability
- Portability to a wide range of machines

PROGRESS is available today on major UNIX™-based super-microcomputers.

Power. Performance. Ease of Use . . . Now, that's PROGRESS! You can't afford to build applications without it.

DATA LANGUAGE CORPORATION
5 Andover Road, Billerica, Ma. 01821
617-663-5000

PROGRESS

A NEW APPLICATION ENVIRONMENT

Want to make PROGRESS in a hurry?
Call Area Code 617-663-5000 today.

Or just fill out and return the coupon below.

DATA LANGUAGE CORPORATION
5 Andover Road, Billerica, MA 01821
(617) 663-5000

☐ Yes! Please send me a demonstration disk and documentation for PROGRESS. I enclose $295.

☐ Please send more information about PROGRESS.

☐ I am interested in PROGRESS seminars.

Name__________ Title__________

Company__________

Address__________

City__________ State_____ Zip_____

Telephone (_____) __________

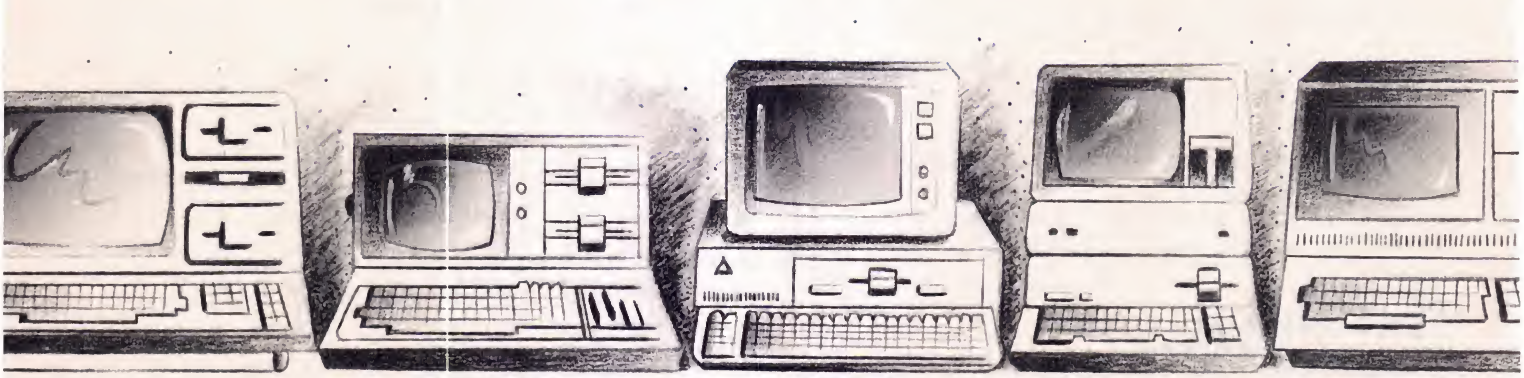
3

Please circle Ad No. 12 on inquiry card.

This software vendor was seeking a refuge from the whims of Big Blue. With the Unix system and C, he found what he was looking for.

A SOFTWARE VENDOR'S SEARCH

BY NEAL NELSON



It all began in Chicago. I had opened a small software house there in January 1973, functioning primarily as a third-party software vendor installing smaller IBM business systems. But it was hopeless. IBM was frequently announcing new families of computers, and each new family was not only more powerful, faster, and cheaper than the previous machines, it was incompatible with them as well.

Big Blue, which was dominating the marketplace, seemed intent on offering an incompatible language/operating system on each new generation of small computer. This meant that small software vendors would be forever doomed to continually rewrite the same basic applications in these new languages for these new machines.

For each machine type, I was spending 14 months training my people, building a library of programs, and converting my business applications. Even with a small staff of four to six programmers, it would cost at least $100,000 in training to bring up a new machine and language combination.

Big Blue seemed intent on offering an incompatible language/operating system on each new generation of small computer.

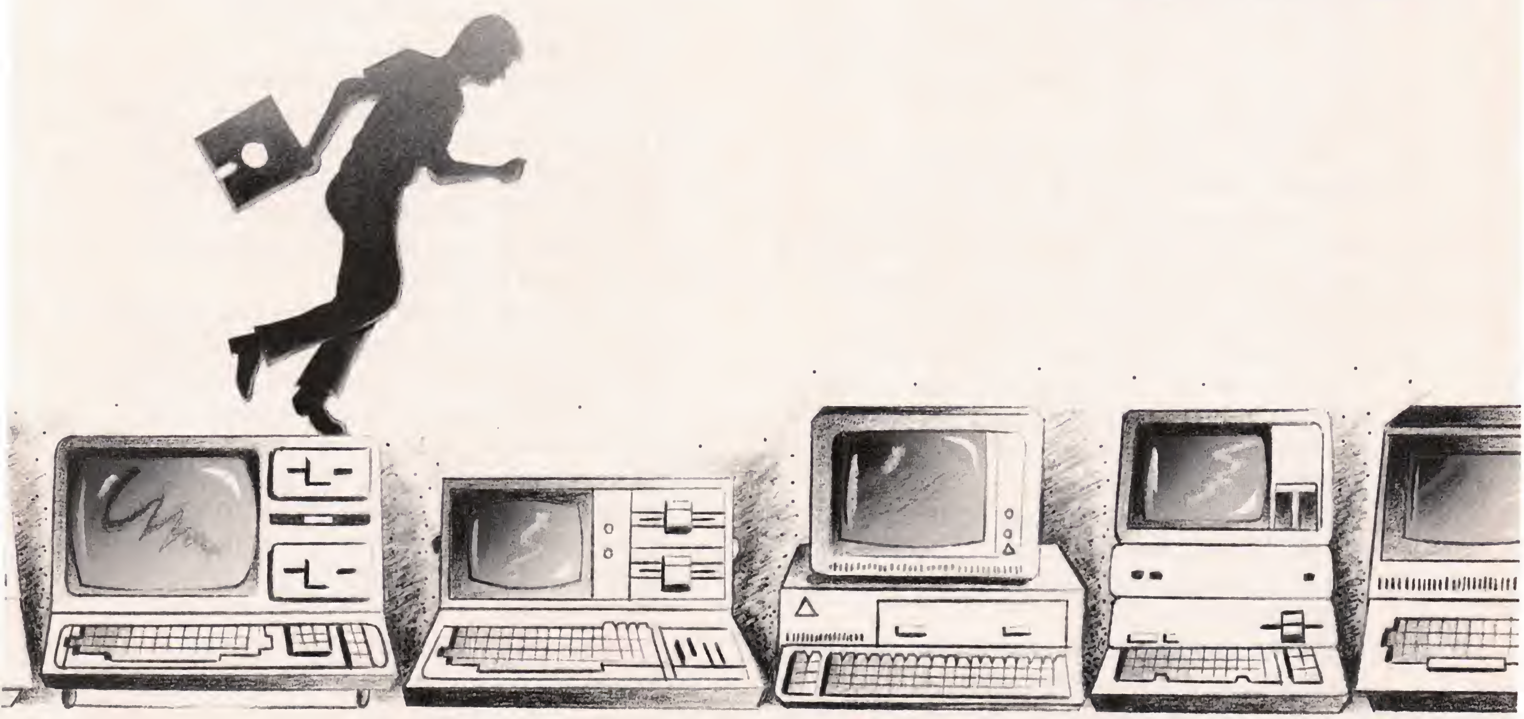
By 1981 I had gone through this development and training cycle a number of times, including five major efforts for various IBM machines (System/3, System/7, System/32, System/34, and Series/1). In spite of the work and expense required, I typically had only 18 to 24 months with a machine before the market moved to something new. You can see how it was extremely difficult for a small software vendor to survive when 14 out of every 18 months were spent developing and testing the same basic applications in different languages under different operating systems.

I decided to make one more attempt to have a decent life in the software market, and if I failed, I resolved to get out. I was even considering getting a normal job—it was that bad.

FIVE CRITERIA

As part of my last-ditch effort, I identified five criteria to help me in my search for the right programming environment. If I could find an environment that met these criteria, I knew I could develop software that would have a useful life longer than 18 months. My five requirements follow:

(1) I needed more than a standard language—I needed a standard



CHOOSE A COMPUTER FOR ITS MIND NOT ITS BODY.

Big. Lots of muscle. Sexy. All ways we describe the computer of our dreams. By those standards, the humble Reduced Instruction Set Computer (RISC) seems like nothing to write home about.

Unless someone in your family designs computers.

Because in high-level language benchmark tests* on the Berkeley campus, RISC outperformed heavyweight superminis and micros by as much as four to one. (Incidentally, most benchmarks were written in C, the same language used to implement UNIX.)

Don't be fooled. What RISC lacks in sex appeal it makes up for in shrewdness.

Instead of using raw muscle to speed up memory references, RISC uses logic to eliminate them. Two hundred fifty-six registers, not memory, hold data for processing.

Parameter passing schemes worthy of a doctoral thesis are replaced by simple hardware. A sliding "register window" easily redefines one procedure's temporary data as another's parameters.

And while hundreds of complex instructions on most computers are never used by high-level languages, every super-optimized RISC instruction is. All 31.

But how would RISC perform when it left college and tackled its first real job?

Just fine, we thought. So when we considered architectures for our high-performance UNIX supermini, the Pyramid 90x, we felt quite comfortable taking a RISC.

We combined its large register stack (528 in our case), register window, and reduced instruction set (about half the number in a typical supermini) with some common computer techniques. Like virtual memory, cacheing and floating point.

The results were uncommon.

Two to four times the performance of popular UNIX hosts on standard benchmarks like Ackermann's Function. In less than half the floor space and at about half the price.

The first commercial implementation of RISC worked as well in the computer room as it did in the classroom.

And proved that you don't have to have brawn to make a good UNIX computer. Just brains.

We'll be glad to send you more information on how RISC works, including a reading list of recent research, if you send your name and address to RISC, Pyramid Technology, 1295 Charleston Road, Mountain View, CA 94043.

Your comments and questions are always welcome. Just call (415) 965-7200.

PYRAMID TECHNOLOGY

*Write us for test details. UNIX is a trademark of AT&T Bell Laboratories. © 1984 Pyramid Technology Corp.

operating system as well. During my eight years in purgatory (or hell), I had written applications in Data General COBOL, IBM Series/1 COBOL, and IBM 370 COBOL. The manufacturer and operating system had made these three versions of COBOL so different that moving from one machine to another was almost as difficult as starting over in a new language.

(2) I wanted a machine environment that was offered by multiple equipment manufacturers. I had experienced situations where a manufacturer had delivery, marketing, or quality problems that prevented it from selling competitive machines in a timely fashion. In these cases, when the manufacturer sneezed, the software house got pneumonia.

(3) I was looking for a multiuser system because I felt strongly that most businesses would require multi-user capability.

(4) I wanted a language/operating system that had some type of credible support from a large organization. I had spent many years with customers who feared that the computer company might go bankrupt, leaving the customer with a very expensive doorstop.

(5) I wanted fast, powerful, competitively priced hardware. I've learned that if the equipment performs well, programmers are more productive and customers are happier.

I looked at all the common language/operating systems (CP/M, Oasis, Pick, etc.) and could not find one that met all my criteria. However, because some of the articles I had read about Pick and Oasis referred to the Unix system, I decided to investigate that as well.

After exploring the combination of the C language and the Unix operating system, I knew I had found it. The Unix system offered these advantages: (1) it was comparatively standard; (2) it was available in 1981 from more than 30 equipment manufacturers (a number that has since grown to over 80); (3)it was a true multi-user operating system; (4) it was developed at Bell Laboratories; and (5) there were a number of inexpensive and powerful Unix system-based computers already on the market.

In June 1981 we began to program and install Unix system computers to perform normal small-business accounting work. During the next 2 1/2 years, we installed 10 Unix system-based computers in a variety of office environments, including manufacturing and service, cities both local and remote, and medium- and small-size firms. We closely monitored the problems, frequently revising and refining the program package to provide more features and to improve performance.

By November 1983 we had proved conclusively that business packages written in C under the Unix system could perform well in a normal commercial environment.

COMPLETING THE MAJOR TASKS

Until this point, we had installed the same computer (Onyx model C8002M) for all of our customers. We realized that all versions of the Unix system were not exactly alike, but we did not want to confront these problems until the major coding, training, and debugging tasks were complete.

Late in 1983 we began porting the program packages to different brands of equipment and to different versions of the Unix system. Our first port from Onyx to Computer Consoles Power 5/20 was a breeze. Both machines ran Unix System III, and we just copied the programs over and compiled them. They ran with virtually no changes.

Our second port—to an Intel computer running the Xenix operating system—was a little more difficult. The Xenix release we were using was closer to Version 7 with some Berkeley enhancements, and we had to make some changes so the programs would compile properly in both environments.

> **I was even considering getting a normal job—it was that bad.**

At this point we started including in some of our library of support modules two or three alternate sections of code that were selected by the "#ifdef" feature of the C compiler. This allows us to have a single library that can accommodate the proper programming technique for each of the various versions of the Unix system. The changes were minimal and were limited to support routines (such as the indexed access method), so we were quite pleased. The basic application package (over 100 programs) required no changes.

However, our joy was somewhat premature because a number of programs provided incorrect results when we conducted a complete system test, even though the programs compiled without errors and the initial programs we tested operated correctly. Detailed investigation revealed that the problem was caused by one major machine-related area of incompatibility —arrangement of string fields in memory.

It turned out that we had inadvertently written our indexed access method with a dependence on the string handling that had been implemented by Zilog on the Z8000 processor used in the Onyx. By coincidence, CCI had used a processor chip having the same type of string processing.

Some processors store character strings as a sequence of bytes in memory, while other processors store character strings as pairs of bytes within sequential words.

Within each word the bytes might be "reversed."

A FLIP-FLOP PROBLEM

I transferred my sample files in binary form from the Onyx to the Intel when I noticed the "flip-flop" problem. I initially did not fully understand the nature of the problem, so I wrote a quick "flop-flip" utility program for the receiving machine.

This very quick and dirty program reversed the bytes in each word of a file. When I ran the utility and ran some samples--BINGO! The text was perfect, and the numbers were all screwed up. At this point I realized that the "flip-flop" applied only to character data, not numeric data.

I began to understand why I had heard people explain that portability can be achieved only when data is moved between machines entirely in text file format. No real problem, I thought. I wrote several sets of utility programs, one set for each data file. One program converted the various fields of the data file into text strings. The sister program would convert the strings back to binary data fields. I could "unload" the files to text files on the source machine, transfer the text files, and "reload" the text into binary files on the destination machine.

I finished my utilities, unloaded the files, transferred the text, reloaded the files, and everything went smoothly. I could read and write records, and all data was displayed correctly. I was home free, almost.

I noticed that files occasionally would not print in the proper sequence. They always printed in the same sequence. It would be correct for a while, jump out of sequence for a while, then jump back in for a few records. All the records were there, but they just didn't seem to be in the proper sequence. Slowly I began to realize that there was another aspect to the flip-flop problem.

My indexed access method used a character string compare to determine greater than/less than when inserting records in an indexed file. This works well if both alpha and numeric data are sequenced in memory the same way (as is the case with the Z8000 chip). However, if the character strings are flip-flopped (as with the 8086), a character compare of a numeric field reverses the bytes, giving incorrect results.

This whole flip-flop experience had so far taken place in one bad weekend. It was 9:00 P.M. Sunday when I realized that I still had not completely solved the problem and that I would have to make substantial changes in the indexed access method to accommodate the way character strings were handled on the 8086. I made the additional changes and completed porting my program library and sample files by 8:00 A.M. Monday.

If this whole experience sounds rather harrowing, realize that I moved an entire application library (A/R, A/P, G/L, Job Costing, and Payroll, consisting of over 120 programs with 12 sample data files) from one manufacturer's product line to a different manufacturer's product line in three days with just over 42 man-hours of effort. By comparison, when we moved the same applications written in RPG II from the IBM System 32 to the IBM System 34, it required approximately 900 man-hours of effort and over 5 months to complete.

PORTING TO PLEXUS

After I finished the Intel port, I set out to port the packages to Plexus. I felt this would proceed smoothly because both Onyx and Plexus advertised Unix System III and, from a review of the manuals, the system calls appeared to be identical.

The port went beautifully. I transferred the source, compiled the programs, and almost everything ran the first time. I did not have to change any of the application programs or support routines. The only "false step" I encountered was that I had specified an incorrect option for the C compiler, and four of the larger programs did not compile. After I read the manual and selected a different compiler flag, these four programs compiled and executed correctly.

After the Plexus port was complete, I decided to port to the Altos 586. I expected this port to go smoothly since Altos was running Xenix with the 8086 processor, and I had completed the port to the Intel machine with Xenix running on the 8086 chip. Indeed the port was quite routine.

By this time I had begun to make quite extensive use of compiler "#if" and "#ifdef" statements. I had established conventions for isolating changes that I made to accommodate the Unix system version I was using (System III, Xenix, etc.).

I would bracket any of these machine- or version-dependent routines within compiler "#if, #endif" control statements. Then, when I moved the programs to the Altos, I "turned on" the sections of code required for the Intel/Xenix combination, and the programs just compiled and ran. When I moved to the Plexus, I set the switches for System III, compiled the programs, and—voilà!—they worked the first time.

My next port was to a DEC PDP--1134 running a pure Version 7 Unix system that had been loaded straight from the Bell Labs distribution tape. I had to make several changes for the Version 7 port to accommodate the "older" formats of some of the system I/O statements, but I realized that the older formats were fully supported by System III. Thus, I could code the routines to Version 7 specifications, and they would function for Version 7, System III, and System V.

My next opportunity to port came with a request to run the programs on a DEC VAX 11/750 running Berkeley 4.1. By this time I was fairly well versed in the Unix system, but I had never before used the much discussed VAX or any Berkeley version of the Unix system.

A MIXED BAG

The port was sort of a mixed bag. The Berkeley release was not all that different from the various other versions of the Unix system I had used, but I had not previously ported to a 32-bit processor with the requirement to align "longs" and "doubles" on 32-bit boundaries. This caused some problems with the layout of fields within my data records.

I also uncovered some places where I had accepted return codes from function calls as "short" rather than "int." On the 16-bit machines, "short" and "int" are the same size, and they would return the same result. On the VAX, however, "long" and "int" are the same size, and occasionally I did not receive return values correctly. I am much more careful to check details like this when I am coding system calls these days. The VAX/4.1 port required about three days and was not all that bad (it was a "learning experience").

After this I ported to a VAX running Berkeley 4.2. I felt that the port would be quick, and it was, except for one problem. All of the versions of the Unix system I had used had handled signal catching the same way. If a signal was caught while a read or write was pending, a return at the end of the signal processing function would cause a "-1" return code for the read or write.

Berkeley 4.2 changed this. I had to use `setjmp longjmp` instructions instead of the technique that had worked on all the other versions and flavors of the Unix system I had tried. The 4.2 port was completed in less than two days.

After completing the 4.2 port, I moved the programs to a Perkin Elmer 3210. I found the Perkin Elmer to be very similar to a VAX running Version 7. When I set my compiler switches for this combination, almost all the programs ran perfectly. The Perkin Elmer is a little more fussy about the 32-bit alignment than is the VAX. I found some data fields that were not properly aligned, even though the VAX accepted them. So far, the Perkin Elmer port has been my all time fastest, taking 26 hours flat.

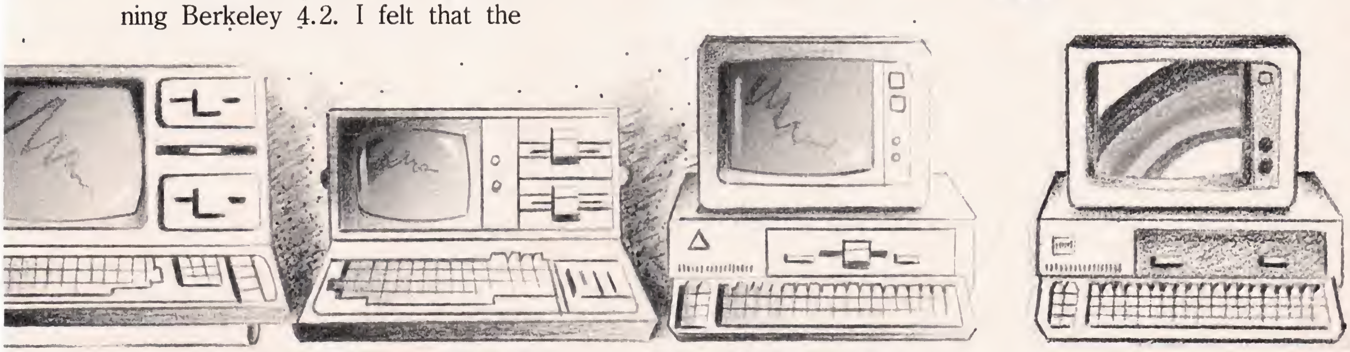
I have completed a port to the AT&T 3B2 300 running Unix System V, and it was extremely routine. I had been told that programs that ran under System III would run under System V, and that is exactly what my experience was. I set my configuration variables for a 32-bit processor and System III Unix, and the programs compiled and ran.

When IBM announced the PC/AT, I smiled all day long because I knew that I could set my configuration table for "Xenix 8086" and that I could move my entire library to this new machine within 48 hours.

It's hard for me to remain cool, logical, or businesslike when I think back to the eight years of my life that I spent writing and rewriting the same application programs in COBOL, RPG, EDL, BASIC, and Assembler for various brands and models of computer equipment and compare that to the 26 hours I spent moving these applications to the Perkin Elmer 3210. The Unix system has given me the freedom to enhance and improve my programs rather than constantly rewriting them in the same form for some "new" language or operating system.

In spite of occasional (and oft discussed) problems and limitations, programs written in C can be very portable under the Unix system. The total environment of productivity and portability offered by the Unix system and C is so much better than anything else I have seen or tried that I cannot imagine ever using anything else.

Neal Nelson graduated from Purdue University in 1970 and started work at the Taylor Forge Division of Gulf & Western. In January 1973 he formed the software house whose story he tells in this article.



One is the latest and greatest, while the other is sophisticated and elegant. But which is better?

4.2BSD VERSUS SYSTEM V

Unix System V is the latest, the greatest, and the best conceivable operating system on the market —capable of doing everything in the universe better, faster, and cleaner. Or so says AT&T.

Unix 4.2BSD is the most sophisticated, flexible, elegant, and downright honest operating system around. That's what you hear from people who use 4.2. Can both be the best? An odd kind of family feud is in progress, with the genealogy of these Unix systems resembling something from the dark side of Greek mythology, replete with incest and Oedipal complexes.

In brief, from the mythical roots of the Unix system in Ken Thompson's search for a better space game, two parallel lines of evolution have developed—one in Bell Labs and the other in the various universities that found in the Unix system (which AT&T distributed, to its credit, in source form for a nominal fee), enough substance

BY KATHLEEN B. WALLACE


NEW & IMPROVED
4.2BSD
System V

TOWER 1632. THE SHAPE'S THE SAME, BUT IT'S GOT AN ENTIRELY NEW DIMENSION.

The most memorable shape in the computer industry just became even more unforgettable.

You might think it would be hard to improve on the computer that took the OEM world by storm only a short year ago, but we've done just that.

First, we added two 46MB, 5¼″ Winchesters to bring total storage capacity to 92MB (unformatted) in a single Tower unit.

Then we added a ¼″ streaming tape for faster, more convenient back-up. The tape's 45MB capacity allows you to dump the entire contents of a Winchester on one tape cartridge.

Of course, this enhanced Tower still offers the UNIX*-based Tower Operating System for maximum software compatibility and flexibility. And unsurpassed interfacing capacity. And the NCR name. Your guarantee of support. Now and in the future.

And it all still comes in that famous box. You know. The one that towers over the minicomputer.

Call us at 1-800-222-1235 to learn more about the Tower's exciting new dimension.

*UNIX is a trademark of Bell Laboratories.

BUILT FOR SYSTEMS BUILDERS.
TOWER 1632.

NCR
OEM Systems Division

NCR Corporation, World Headquarters, OEM Systems Division, Dayton, Ohio 45479. Nationwide (800) 222-1235. In Ohio (513) 445-2380. In Canada (800) 268-3966.

Please circle Ad No. 29 on inquiry card.


NCR
Disk
Communications
Fault
Battery
On
Off

The "Personal" Mini

A 32-bit multi-user virtual memory microcomputer by LMC.

The MegaMicro is a "big" computer in a small box. It allows one or up to 32 users to run big applications programs (ones so big they can't even be compiled by smaller 8- or 16-bit machines) simultaneously. Because the MegaMicro is a multi-user system, it allows easy sharing of data bases and peripherals—obstacles that soon haunt business and scientific users of "personals" who find a need to "network" or to add devices such as laser-printers, multicolor plotters and the like.

LMC's MegaMicro is built around the newest state-of-the-art VLSI logic—the 16000 family developed by National Semiconductor. Each MegaMicro is supplied with UNITY—HCR's full Bell-licensed UNIX operating system—as well as FORTRAN AND C. Also standard are hardware virtual memory and hardware floating point, a half Meg. of RAM and a very fast 33 Meg. Winchester hard disk. The result is a computer with the performance of a large mini, at a "micro" price. For example, the MegaMicro does 161,000 double-precision (64-bit) floating point multiplications per second. All this at a realistic price, and even less with government and quantity discounts. The result is a cost per "work-station" far lower than similarly configured (and less powerful) "personals."

The MegaMicro is powerful, inexpensive and designed around the Multibus (IEE 796) which means it has a completely "open" architecture.

LMC MegaMicros *The Logical Alternative*™

LMC™

The Logical MicroComputer Company
A member of the Marmon Group of companies
4200 West Diversey, Chicago, IL 60639, (312) 282.9667, Telex 887787 LMC

SEE US AT COMDEX/FALL

Please circle Ad No. 24 on inquiry card.

Look closely at System V and Release 2.0 and you'll see the encroachment of Berkeley-style enhancements.

to warrant serious scientific effort.

While many universities have partaken of the fruit from Ma Bell's tree, UC Berkeley, with its BSD (Berkeley Software Distribution) releases of the Unix system as modified by the Computer Science Research Group, has become the most vocal and well-known standard-bearer of the university lineage.

THE ROYAL HOUSES OF EUROPE

The complexities become obvious when we begin to trace the cross-fertilizations between the two lineages. In keeping with the tradition of the royal houses of Europe, the relationships are often obscure, but no less significant. Within Bell Labs, Version 6, which was assembled and distributed in 1975, diverged in three directions:

First, an internal research and development system under the patronage of Bell USG (Unix Support Group);

Second, Version 7 (if you had a PDP-11 and ran the Unix system, you ran this one), released in 1979;

Third, Unix 32V (a mildly virtual version of Version 7 for the VAX, which served as the basis for Berkeley 3.0BSD).

In early 1982, courtesy of the Justice Department and *The Great Divestiture,* AT&T released its first commercial implementation of the Unix system—System III (sounds impressive, doesn't it?). System III featured nicely printed manuals, bug fixes, and enhancements deemed desirable to the commercial user —such as the restricted shell, improved system accounting, error logging, and commands to make system administration more routine.

The introduction of named-pipes, which provides a rudimentary facility for interprocess communication, and the new "inittab," which provides the facility for "changing states" and implies the existence of states other than single and multi-user mode, represent System III's major technical enhancements.

In early 1983 AT&T released System V (System IV, if it ever existed, must have fallen into the proverbial bit-bucket) and followed that with Release 2.0 in early 1984. In its latest incarnation, Release 2.0 continues the AT&T trend of making the Unix system more palatable to the commercial user. Let's face the facts: Until AT&T took hold, the Unix system was an R&D operating system, which, by its very (source and portable) nature, was designed for tinkering.

Yet, as a result of all that tinkering, the Unix system is also a very "workable" operating system: It provides an environment in which anybody, commercial customers included, can get something done. On the other hand, no commercial customer can afford to tolerate downtime or things that bump in the night just because a computer genius had a brainstorm after eating an anchovy pizza.

Enter AT&T to save the day: Release 2.0 has lots of bug fixes and some new utilities, plus slick documentation, "standards" in the form of standard disk and tape names, and a promise from AT&T that future modifications to the Unix kernel will not imperil programs that depend on the current kernel implementation.

A METAMORPHOSIS AT BERKELEY

While AT&T was enduring what it refers to as "the recent events," the Unix system at Berkeley was undergoing a metamorphosis of a different nature. While early Unix had settled on the PDP-11 for a long period of evolution, and though AT&T did produce a "virtual" version of the Unix system for the VAX, AT&T, as of this writing, has never released a full-blown, demand-paged virtual memory version of the Unix system for the VAX.

The Computer Science Research Group (CSRG), funded by the Defense Advanced Research Project Administration (DARPA), brought to that task the energy, spirit, and freethinking that abounds in an academic environment. The result was Berkeley 3.0BSD, which was released in late 1979 and which was followed in 1980 by release 4.0. This release was in turn updated by the release of 4.1 later that same year.

The final release of 4.1BSD represents a truly well-tuned and stable operating system that, in contrast to AT&T's releases, offers the C shell with job control, the `vi` full-screen editor (it is almost incomprehensible that until System V, "standard" Unix subsisted on a line-oriented editor), `termcap`, and `curses`, Pascal, LISP, `sdb`, and approximately 80 new commands.

While AT&T releases for the VAX remained encumbered by the hardware restrictions of the PDP (AT&T releases did not include 1K-blocking until System V), Berkeley brought the Unix system into the technological age of the VAX. The point must be made that Berkeley has been conscientious in its effort to release 4.1 as a mature, solid operating system, while at the same time pushing the edge of scientific investigation.

THE OBJECTIVES

Mike Karels, principal programmer at CSRG, says: "The idea with 4.2BSD was to stand back from the system and see what it really needed....The goal was to provide a new basis—to redefine everything that needed to be redefined, all at once. Then, instead of having a constantly evolving, changing interface, there would be only one upheaval when everyone converted."

The new 4.2 is a completely reworked operating system. CSRG looked at code that had come down virtually unchanged from Version 6 and evaluated that code in light of the needs of a larger, more complex computer with many more users. 4.2 differs quite a lot from the previous Berkeley releases mostly because of the local-area networking enhancements to the kernel and the fast file system, which pushes the disk blocking factor to 4K.

As aptly demonstrated at the USENIX conference in June 1974, CSRG regards 4.2 as being equivalent to 3.0. Just as 3.0 was the first in a series of releases that introduced major new computing concepts that were refined in subsequent releases, 4.2 introduces newer, more radical concepts that will be refined in future releases. CSRG is fully funded and is currently working on kernel refinements and new applications such as windowing and virtual file systems.

Doug Kevorkian, of Bell Technical Labs, says: "Our major objective is to make Unix the software environment of choice across the range from micro to main-frame computers. That philosophy carries with it some very important sub-goals: namely to maintain out customers' investment in existing Unix applications with a commitment to upward compatibility to the extent that it is reasonable . . . also, to maintain an open approach to the architecture and the source code of the Unix system, encouraging it to be ported across to the widest variety of hardware possible."

So now we have 4.2 and System V Release 2.0. While there are a raft of bright, freewheeling freethinkers within AT&T who are doing fine work to advance the art of computer science, the focus of AT&T's interest in the Unix system is on the dollars of Unix.

The slant of AT&T's releases is toward solid, commercially desirable features. AT&T is, after all, in business, and its business is to make money. The people who buy are in business, and their business is to make money. They want to spend their money on a product with which they feel comfortable, and in Unixland that product is Unix from AT&T....or is it?

Look closely at System V and Release 2.0 and you will see the encroachment of Berkeley and Berkeley-style enhancements—the `vi` editor, `pg` (a `more` by any other name is a `more`), a new form of the shell that offers job control, a `curses` package based on terminfo from Purdue University, and shell functions, which are a generalization of "aliases" from the C shell.

> **No commercial customer can tolerate downtime just because a computer genius had a brainstorm after eating an anchovy pizza.**

Similarly, AT&T has responded to market demands for demand-paged virtual memory. Doug Kevorkian of Bell Technical Labs says: "We are very actively moving along the path to a demand-paged implementation in System V to eliminate that shortcoming."

Crying "Uncle!"? No, AT&T is merely demonstrating a sensible responsiveness to market demand for excellent products developed in the research environment. The technical management at Bell expresses a real respect for their colleagues in academia; however, in spite of such advances, AT&T's current offering is deficient in some rather significant areas.

NETWORKING OFFERINGS

Right now, System V's networking offering is minimal. Whereas 4.2 addresses networking in the broadest possible terms by providing support for TCP/IP and other protocols, System V designers within Bell are currently limited to the perceived short-term needs of their customer base.

3Bnet, announced at NCC in July as "Ethernet-compatible," links AT&T PC users who need to share files or exchange data to a central AT&T computer that acts as a file server. The 3B software interface for the PCs is built directly on top of System V and allows the user to switch between terminal emulation on the central Unix machine and the native PC operating system, DOS.

THE ENTITIES BEHIND THE ACRONYMS

AT&T: **An acronym for Bell Labs, Western Electric, AT&T Technologies, and all those other names that once stood for Ma Bell.**

DARPA: **Defense Advanced Research Project Administration; they have the money.**

CSRG: **Computer Science Research Group; they received the money and did the work that produced the Berkeley Software Distributions.**

One or more of the DOS directories can reside physically in the central Unix machine, and a program running in the DOS environment can utilize files resident on the Unix machine. That's great if you're into DOS, but it's a far cry from remotely logging into a networked 4.2-based machine and transferring files at the speed of light.

So what's the bottom line? AT&T has wisely digested the popular "fruits" that have grown in its university garden. But by the unique nature of portable Unix, the source license holder has an unusual choice. The systems-builder who elects to follow AT&T as a Unix system "standard" must trust that AT&T will make wise choices in selecting developments from the university environment, in funding its own internal research endeavors, and in releasing the results of those endeavors to the user.

One must also trust that AT&T will make these choices in consideration of the Unix system in general and not just for the betterment of AT&T's own computer product line. Or the systems-builder can take the maverick approach and use Berkeley distributions as the foundation for a software product line and get the innovations from the (real) source. To be counted among the mavericks are DEC, Sun, and National Semiconductor.

At the heart of this dilemma resides a nasty little conundrum: The Unix system is great not only because it implements elegant concepts, but because it is portable and is available in source. And because it is available in source, it is modifiable— it can become anything you want it to be. But if it becomes "standard," will it still be Unix? □

Kathy Wallace is a Unix Applications Engineer at National Semiconductor Corp., Santa Clara, Calif. After graduating from UC Santa Barbara with a degree in history and anthropology, she taught for 2½ years in an elementary school program for the educationally handicapped. She now lives in a Victorian in Santa Clara.

Intriguing parallels abound between the development of AT&T's Unix standard and IBM's earlier standardization with System/360.

UNIX, C AND APL

BY DAVID SAUNDERS

Lost. That's where IBM and its customers were 20 years ago—lost in a wilderness of incompatible hardware. Because IBM models were built differently and worked differently, converting programs from obsolete machines to new machines was painful. Many customers, who were apprehensive of change, lost faith in the durability of new products and shied away from new machine purchases.

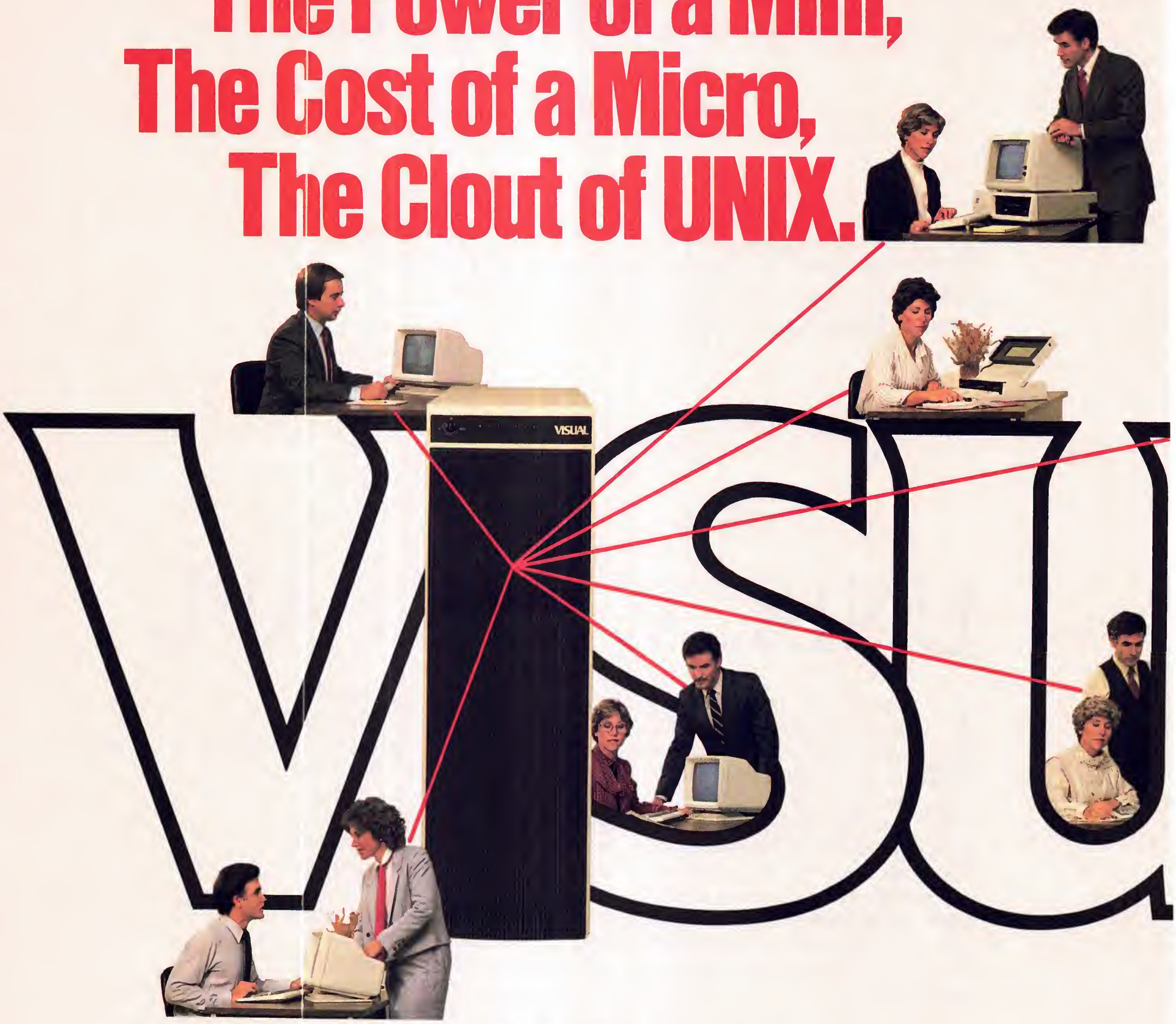
IBM embarked on a serious quest to create a standard mode of industry compatibility: It went to the mountaintop, saw the light, and came down to announce System/360. With System/360, all computers worked the same way from the user's viewpoint. There would be no more conversions, and future upgrades would be effortless. All a user had to do was make that one last conversion to System/360.

The gamble worked. System/360 was, and still is, a huge success. Users responded to the benefits of portability and a growth path. IBM has set standards that persist to this day in models derived from the 360 series. Now, when IBM introduces new hardware technology, it may go faster or cost less, but function is rarely affected. IBM knows better than to ask users to change again.

In solving an in-house compatibility problem, IBM set a precedent that the rest of the computer industry must follow in order to compete. The same need that produced System/360 persists today in the problem of intermachine incompatibility.

The solution clearly must be achieved through software, not hardware. No one can legislate hardware standards for a whole industry as IBM can internally. Seemingly by chance, the challenge of making software universally portable was taken up by the other industry giant —AT&T—but the result is similar. IBM introduced a single instruction set for all its computers—

The Power of a Mini, The Cost of a Micro, The Clout of UNIX.



Introducing The Perfect System Builder

With the VISUAL 2000, we've designed the perfect tool for system builders and integrators. A multi-user computer system that supports more terminals, offers more expandability, and gives you more configuration flexibility than any other system in its price range. Truth is, it outperforms a lot of higher priced systems as well.

What that means for you is an important competitive edge in delivering integrated business solutions. An edge that comes not just from the VISUAL 2000's surprisingly low price. But from its higher technology. Technology that gives you . . .

Power to support more on-line users in demanding applications—up to 16 independent terminals or workstations.

Flexibility to accommodate a remarkable variety of application requirements, thanks to an open architecture, compatibility with mainstream industry standards, and a full spectrum of configuration options.

Expandability that can take you further than any other system in its class—an unrivaled upgrade path to protect your system investment.

Put the Full Power of Intel's 286 in Your Systems

The Intel 286 is today's chip of choice for UNIX-based systems. Only the 286 gives you: on-chip memory management; an instruction set optimized for multi-tasking; pipelined architecture; and an optional 287 numeric coprocessor which can speed up floating point by a factor of 10. What do these features mean to the end user? Faster response time, more users supported, and lower system cost.

A "Mainframe" on a Single Board

The VISUAL 2000's single-board base-level design is the key to higher performance, greater reliability and lower cost. A single high-density board includes the 286 CPU; 512KB-2MB of RAM; controllers for Winchester, floppy and streaming tape; an intelligent communications processor; six RS-232 ports; and a parallel printer port. There's even a real-time clock with battery backup. A few short years ago a system with these features might have been called a mainframe.

From a Multi-User System to a Multi-System Network

With the VISUAL 2000 you never pay for more system than you need, and never have to settle for less. Up to 6 megabytes of RAM. 4 Winchester disk drives. Floppy. Streaming tape backup. And mainframe communications. All in a small stand-up enclosure which looks right at home next to a desk.

Use it as a central processor for inexpensive video terminals, or as a database manager for a cluster of intelligent workstations or personal computers, including VISUAL's own IBM®-compatible COMMUTER™.

And if a fully expanded VISUAL 2000

See us at COMDEX, Booth #1836

AL 2000

isn't enough, you can connect up to 254 systems in an ARCNET® local area network.

In all configurations, the VISUAL 2000 achieves an exceptional ratio of price/performance.

Software: A Complete System Builder's Tool Kit

Ultimately, the strength of a system is the strength of its software. And there too, the VISUAL 2000 really shines. Start with XENIX™, Microsoft's enhanced version of UNIX™, which gives you everything you expect from UNIX *plus* greater speed, reliability and ease of use in business applications. Then add all the tools you need to deliver end-user applications quickly and easily. Languages such as C, SMC BASIC, RM/COBOL™, TOM BASIC™, SOFTBOL™, and MicroFocus Level II COBOL™ to give you instant compatibility with hundreds of proven business applications. Plus sophisticated system foundations such as INFORMIX™ database management and RealWorld™ modular accounting. And professional productivity software such as 20/20™ spreadsheet modelling and XED™ word processing.

The Bottom Line

High performance. Superior flexibility. Extensive software. And low cost . . . VISUAL 2000 systems start at under $10,000, suggested list. No one gives you more in a UNIX-based multi-user system.

And when you consider that VISUAL has long been a leader in video terminals, and can supply a full range of compatible terminals and workstations, we think you'll agree that VISUAL is the system builder's system supplier.

The VISUAL 2000.

It began with a better design. And it's available *now* to help you build better systems. But don't take our word for it.

See for yourself.

Call VISUAL today.

Please circle Ad No. 48 on inquiry card.

VISUAL See for yourself®

Visual Technology Incorporated
540 Main Street, Tewksbury, MA 01876
Telephone (617) 851-5000. Telex 951-539.

REGIONAL OFFICES:

Northwest:	(415) 490-1482
Southwest:	(213) 534-0200
North Central:	(513) 435-7044
South Central:	(214) 255-8535
Northeast:	(201) 528-8633
Southeast:	(301) 924-5330

THE CODATA DIFFERENCE

A Major Improvement in OEM Microcomputers.

With so many microcomputers on the market, and with each manufacturer claiming to have the solutions to everyone's problems, why are more and more OEM users choosing CODATA?

Very simply because the CODATA 3300 works!

That's an important part of what we call THE CODATA DIFFERENCE - a refreshing, new concept in OEM marketing.

More performance...Lower price...Extended warranty.

We've increased the performance of the CODATA 3300 and, at the same time, we've lowered the price. That means our CODATA multi-user, multi-tasking 16-bit computer systems have price/performance ratios nobody else can match. We're so confident in our system's integrity that we've extended the warranty on every CODATA 3300 to 180 days. That's the longest warranty in the business and, let's face it, you've got to be pretty sure of your product to guarantee it for six full months. We're sure.

These are more elements of THE CODATA DIFFERENCE.

An uncomplicated pricing policy.

We have only two price sheets: one for suggested retail, the other for OEM prices. Simple, straightforward, easy-to-understand. Beyond that, there are no quantity commitments, no bill-backs, no hassles.

Full support and technical assistance if needed.

Our customers select the CODATA 3300 because it fulfills their needs, because it does what we say it will do, and because we have taken great pains to assure that it is functionally simple and capable of performing its assigned tasks with no problems. But, if technical support is required it's only as far away as your phone and available at a moments notice via our TOLL FREE number during normal business hours in every U.S. time zone.

A multi-user, 33-megabyte system for only $9,600.

The CODATA 3300 is a powerful 16-bit, 68000-based, MULTIBUS system that can effectively accommodate up to ten users. It's a complete UNIX system that runs full ANSI standard FORTRAN-77, RM/COBOL, BASIC+, SMC BASIC, APL, and PASCAL. The 3300 provides up to 33 megabytes of unformatted on-line storage via an integrated, high speed Winchester drive, and a removable, quad-density 5-1/4" floppy disc system. The 3300 features 320 K bytes of parity protected RAM.

An expanded CODATA 3300, with 84 megabytes of disc storage is priced at $13,500. The 3300 is also available as a 12-megabyte system for as low as $7,800.

LET US SHOW YOU THE CODATA DIFFERENCE.

Write or call us for more information.

CONTEL CODATA

285 North Wolfe Road, Sunnyvale, CA 94086
1-800-521-6543. In California: 1-800-221-2265

Please circle Ad No. 8 on inquiry card.

KEYBOARD AND MONITOR OPTIONAL

System/360 BAL; AT&T has effectively introduced a single new machine code for all computers—C.

By creating a single user interface uniting many different machines, AT&T's Unix operating system, itself written in C, takes unification one step further than IBM did. As with System/360, the attraction to users is the same —portability. Because Unix with C provides this key feature, it seems likely to become a universal standard. Having the colossal might of AT&T behind it makes this an almost inevitable result. The Unix system has become a bandwagon that none can ignore.

THE IMPACT OF THE UNIX STANDARD

The Unix system completes an evolutionary cycle for operating systems and systems programming languages. It lifts the entire industry to a single universal standard. Because this standard-setting happened before, at IBM, we can draw some parallels.

Unix plug-compatible hardware: The IBM 360 standard spawned a whole industry of hardware vendors building IBM plug-compatible hardware. The Unix system creates the same opportunity. With one virtual machine code—C—and Unix system calls, all computers become Unix system machines. Computers are close to becoming interchangeable, so users can mix and match with hardware—and new machines can quickly capture Unix market share.

Because the Unix system is an uncomplicated, open standard, it is easy to become a Unix system vendor. Users will benefit as new technology spirals and vendors compete to deliver faster machines at lower costs. Peripherals from plug-compatible manufacturers (PCMs) will come with software that uses Unix as a software bus, and users will be able to install them with little effort.

Unix absorbs other operating systems: The earliest IBM 360 models allowed programs from older IBM computers to run unchanged via special emulation modes. Similarly, the Unix system will be adapted to allow other operating systems to run under it as a conversion aid. This will ease migration to the Unix system just as IBM's emulation modes facilitated upgrading to System/360.

Microsoft's MS-DOS and Digital Research's CP/M are both evolving toward the Unix system. Unix system developers will offer Unix shells that support MS-DOS and CP/M programs to allow users to migrate to new Unix system machines. If vendors don't help their users migrate to the Unix system, others will do it for them.

C becomes a system programming standard: C will be the first (and perhaps only) language implemented for any new processor, so that it can quickly grab its share of the Unix pie. Software vendors will implement their wares in C in order to be on new machines fastest and to have the widest possible market. C may be less efficient than assembler, but portability is more important.

C effectively makes assemblers obsolete, and new machines may not have assemblers at all—they will just come with C. For example, Pyramid Technology's Unix supermini uses a RISC (Reduced Instruction Set Computer) architecture microcoded to optimize the machine for C.

HIGH-LEVEL LANGUAGES

Assemblers, the lowest level programming languages, have instructions that correspond one-to-one with specific machine codes. High-level languages (HLLs) like BASIC, COBOL, FORTRAN, Pascal, and PL/1 are problem oriented rather than machine oriented. Instructions in these languages often translate into many machine instructions.

C is better designed for applications programming than most HLLs. This qualifies it as a HLL. But C's real job and true strength is in systems programming on the standardized portable Unix machine. C is a one-to-one assembler for the universal Unix "computer" more than it is a HLL.

In some ways C is the lowest-level HLL possible—it has to be for it to be efficient as a systems programming tool. But what would be the highest-level language possible? What would be its purpose? And how would it relate to C?

It is difficult to do fine thinking about an application while writing in assembly language because the mind has to span the full range from the subtlety of the problem to the grossness of the hardware. Very high-level languages (VHLLs) remove the computer as a distraction and approach a one-for-one correspondence with human thinking. APL, Prolog, and Smalltalk are the languages most often cited as approaching this ideal.

The evolution of software is analogous to progress in the semiconductor industry, which started with the transistor, a discrete component. Integrated circuits and large-scale integration improved on discrete components, but VLSI takes integration to the logical conclusion, building a complete end-user function in a single package. Similarly, assemblers are the first step away from this detailed level. VHLLs complete this evolution, giving the user maximum function.

VHLLs employ the highest level of abstraction possible—furthest from machine activities. It is possible to think in the notation of VHLLs, as mathematicians have for centuries thought and communicated in mathematical notation. In fact, most

VHLLs trace their roots to the tradition of pure and applied mathematics, as do our everyday arithmetic and logic.

AS SIMPLE AS POSSIBLE

Once a VHLL is implemented, computers can execute instructions in the VHLL. This brings computers to the point of executing the expression of thought, the notation in which we think. With this, the effort required to communicate with machines is minimized, making man-machine communication as simple and productive as possible. As Einstein said, "Everything should be as simple as possible, but not simpler."

C is ideal for implementing VHLLs because it gives them portability. C is the first stable layer in the evolution of programming languages, uniting diverse hardware with a common operating system, Unix. This unification acts as a foundation to bridge the gap between the human mind and VHLLs. Most computer languages are computer-oriented; VHLLs are oriented to the human intellect.

Much current research is directed toward computers that will understand "natural" languages and human speech. It may be too soon and rather anthropomorphic to expect computers to duplicate the whole range of human behavior. Early attempts at manned flight tried to mimic birds, but today's airplanes do the trick without flapping their wings.

True natural language capabilities for computers require more levels of programming and hardware evolution. It will first be necessary to stabilize this level of efficiently communicating with humans through VHLLs and enhancing productivity by amplifying the human intellect. In this, VHLLs are as important a precursor as standardizing with C.

It may be that, as in the case of C for HLLs, one VHLL will emerge as a clear leader. As with C, it doesn't have to be perfect, but it must be adequate for the job and have sufficient backing. APL is an excellent example for the following reasons:

(1) APL is IBM's VHLL. It shows how the computer giant has developed productivity following standard-

APL'S FUNNY SYMBOLS

People either love APL, or they hate it. "APL is a mistake," says Edsger J. Dijkstra, father of structured programming, grudgingly admitting that the mistake is "carried through to perfection." What disturbs people about APL is that it is so different from other programming languages. This is because APL was not designed as a computer language in the first place. APL was originally known as "Iverson Notation," when Kenneth Iverson developed it at Harvard as an effective notation for describing algorithms in applied mathematics.

APL had a healthy gestation away from the constraints of real computers and was only implemented on a computer when Iverson moved to IBM. The name "A Programming Language" was perhaps coined to emphasize its relevance to IBM. But APL was sufficiently ahead of its time that it was rarely actively promoted —indeed, it had to fight off attempts at suppression in favor of PL/1.

An early critic asked: "Why do you insist on using a notation which is a nightmare for typist and compositor, and impossible to implement with punching and printing equipment currently available?" This need for special hardware is unique among programming languages. It hinders experimentation with APL and conversion to it, and has been a real obstacle to its growth. But typesetting, printing, and punching were all once struggling new technologies themselves.

Some say APL is unreadable, that it is a write-only language. This is unfair; while other computer languages are superficially readable, they are no more comprehensible to nonprogrammers and are harder to learn. APL users quickly get used to APL's terse form, simple rules, and the graphic symbols that concisely represent the concepts of computing. One screen of APL can contain programs that would take many pages in other languages.

Maybe APL's very power has temporarily worked against it. Only recently have personal computers (and their APL implementations) become available that are powerful enough to execute APL. New bit-mapped displays and printers now handle APL characters more readily, removing the limits to APL's growth. Emerging trends in computer graphics make APL's use of graphic symbols seem almost prophetic.

In other areas, too, APL may be about to come into its own. There are many countries where students have to learn English before they can learn programming languages based on English. APL's international mathematical symbols bypass this, giving it a further advantage wherever English is not the mother tongue and building strong followings in Europe and Japan.

It is a tribute to APL's power and to its users' dogged devotion that APL has grown despite its seeming initial handicaps. APL's unusual origins have helped preserve a distinctive character that makes it very different from other languages. Being different is no guarantee that APL is better, but it's a good start. APL's funny symbols may soon prove to be its success symbols.

ization with System/360.

(2) APL has a strong commercial following. One in three mainframe IBM sites uses APL, according to data compiled by Guide, an IBM user group.

(3) Most APL authors are not professional programmers, but instead are end-users. This fact shows that computer expertise is not necessary to use a VHLL.

(4) APL's unique characteristics highlight the differences between HLLs and VHLLs.

(5) Advanced APLs written in C for the Unix system are now becoming available.

HOW APL WORKS

Try adding two numbers in Roman numerals, say CVIII and MCMLXXXIV. Tricky, isn't it? Now try multiplying them! In Arabic numerals, adding 1984 and 108 is easy, and multiplying them is at least feasible. This shows how the right notation amplifies the power of the mind. In the present information revolution, proper thought tools are as important as machine tools were to the industrial revolution. Using APL, one simply types 1984 + 108, as on a calculator. Multiplying is just as easy.

APL extends the range and power of mental tools by simplifying and extending math notation even further. Most other computer languages channel our thinking to fit the computer. APL expands the range of human thought by clearly identifying mathematical concepts and assigning symbols to them so they can be recognized and manipulated. Then APL brings computers up to that level.

As a computer language, APL is disarmingly simple—one types expressions in APL notation just like using a calculator. Beginners can make use of APL within minutes and become proficient in days. Extensions to the calculator concept are added in a consistent way and can be learned as needed.

APL has a repertoire of around 50 math functions, many of which are already familiar from school. Each function has a symbol and a name describing the fundamental operation it performs on data—mathematical, logical, relational, or structural. Familiar and useful combinations are built up, like molecules in chemistry or idioms in language, to process APL data as required.

APL expressions can be entered and executed immediately in its calculator mode, or scripts of expressions can be entered and stored for later execution in APL's equivalent of programming. As with Unix shell programming, a simple syntax minimizes interpreting overhead compared with the amount of work that is done.

APL data (called arrays) and programs (functions) live in file folders called *workspaces*. Each workspace contains all the APL functions and variables relevant to a particular application. Workspaces are kept in libraries, like file folders are kept in file cabinets. Workspaces can be loaded—placed on a desktop—modified, experimented with, used, and replaced in libraries in their new form. Program debugging in this simple interactive environment is quick and easy.

Unix shell programs process whole files of data, filtering them in the process, and piping results to the next program. In a similar way, APL takes whole arrays of data, either in pairs (for example when adding two numbers or matrices), or one array at a time (as when reshaping a matrix), passing the result on to the next function. Like shell programs, APL programs rarely need to loop.

THE MOST USEFUL FUNCTIONS

Many of APL's most useful functions work at this structural level of data, making the creation and transformation of data structures as easy as operations on their contents. Other languages tend to work on a single number—or datum—at a time. APL works in the plural—on data—allowing users to think holistically and freeing them from concern with computer-oriented details. Storage allocation, data representation, and so on are all taken care of by the APL interpreter system.

APL unifies the writing, testing, and running of programs under a simple built-in command system that is part of the language. This minimizes unnecessary interaction with the host computer system. APL includes many powerful features for interactively developing, debugging, and running applications. APL's command language has always used the simple file-cabinet and desktop metaphor that's now becoming popular on other systems.

APL revolutionizes math in the same way the introduction of Arabic

WHY USE APL?

APL's funny symbols often discourage people from further investigation, but inquisitive individuals quickly find their reward. APL's strengths can be summarized as follows:

(1) APL is easy to learn—anyone can begin to use it immediately and can learn more as needed.

(2) APL is visually precise—it represents the basics of data processing pictorially.

(3) APL is internationally standardized—APL software is fully portable.

(4) APL is concise—it condenses full pages of conventional programs into a few lines.

(5) APL is productive—projects that would take weeks can be completed in mere days.

notation did. It has important consequences for math education, too, uniting it with computer science. Further exploration of APL is richly rewarded. As Iverson says in his book *A Programming Language,* "The descriptive and analytic power of an adequate programming language repays the effort required for its mastery."

The idea of mastery of a language is fascinating. The highest level of accomplishment in most computer languages is appropriately called *hacker.* APL's features delight beginners, and they interest and inspire users at all levels. APL, building on the tradition of mathematics, is one of those rare creations with the intelligent design and creative potential to continually challenge the user and to offer the potential of genuine mastery.

Various enhancements to APL have been discussed in the APL community during the last few years, the objective being to further simplify the language and extend its capabilities. These discussions have resulted in the release of second-generation implementations of APL that enhance its power and the scope of its potential application.

The seal of approval for this new generation of APLs came in the form of IBM's recent announcement of APL2, a program product to supersede its current mainframe product called VS APL. APL2 conforms to the proposed ANSI/ISO standard for APL, but its advanced features go well beyond the minimum requirements of the standard, creating a new standard for second-generation APL.

APL2's most significant feature is the extension of APL's array model of data. Formerly, APL arrays were either all numbers or all characters. APL2 arrays can contain mixtures of both so that a single APL matrix can, for example, directly and powerfully represent a spreadsheet or a simple database containing different types of data.

More significantly, the elements of arrays can themselves contain arrays. This brings to APL a recursive data structure that supersets Lisp data structures. A single APL variable can contain any combination of lists and tables, both numeric and character, nested to any level. APL functions work consistently with these structures, and new APL operators are included to manipulate them.

This comprehensive scheme of extensions to the APL language makes APL2 even more suitable for expert systems and artificial intelligence applications as well as for complex databases. Users claim that APL2 gives a whole new level of power and capability over existing APLs, like the increase in power they gained when first moving to APL. It seems likely that, in time, APL2 will supplant APL.

MORE INFORMATION ON UNIX APL

Dyalog APL currently runs on Cadmus, DEC VAX, Fortune, Gould, HP 9000, NCR Tower, Perkin-Elmer, and Zilog computers. Ports are planned to Amdahl, AT&T 3B series, and others. Dyalog APL is available from Notation, 10 Jackson St., Los Gatos, CA 95030, 408/354-3274, and from MIPS Software Development, 31555 West 14 Mile Road, Suite 104, Farmington Hills, MI 48018, 313/855-3552.

APL*PLUS/UNX currently runs on AT&T 3B series, Cadmus, Fortune, HP 9000, Spectrix, and Wicat machines. Ports are planned to DEC VAX, IBM PC/AT, IBM 9002, and a number of others. APL*PLUS/UNX is available from Notation (see above), and from STSC, Inc., 2115 East Jefferson Street, Rockville, MD 20852, 301/984-5123.

TWO ADVANCED APL'S FOR UNIX

The good news for Unix system users is that APL and even APL2 are not restricted to IBM mainframes. APL has been available on some Unix systems for a while. Recently, two vendors announced second-generation versions of APL for the Unix system, comparable with IBM's APL2. Both are compatible with the proposed ANSI/ISO APL standard and are implemented in C.

Dyalog APL, from Dyadic Systems in England, is being distributed in North America by Notation Corp. Dyalog APL features a full interface to the Unix system. Shell commands can be executed directly from APL and the result captured as an APL matrix for further processing. Communication with other processes and files is simple, and subroutines written in C can be called directly from Dyalog APL. It includes an editor that works like the Unix system's `vi`, allowing full screen editing of APL functions.

The APL*PLUS/UNX system comes from STSC, Inc., a leading APL time-sharing and software vendor. APL*PLUS/UNX is modeled on STSC's advanced mainframe APL*PLUS version of APL. Like Dyalog APL, it features a simple and powerful interface with the Unix shell and a full-screen front-end based on the Berkeley `termcap` library.

Because both APLs have advanced language features, Unix system users will reap all the benefits of IBM's most successful interactive language on the processor of their choice. This is a fascinating merger of IBM's best VHLL technology and AT&T's software standard.

IBM "bet the company" on System/360, convincingly demonstrating the value of standardization. AT&T has taken this demonstration a stage further to unite the whole computer industry under the Unix system umbrella, providing a firm foundation for future progress in which VHLLs will play a key role. APL is an excellent VHLL, already popular with IBM customers and recently greatly enhanced. It fulfills the requirement for a VHLL to bridge the gap from universal portability (granted by the Unix system and C) to universal ease of use.

APL and Unix, both written in C, are equally readily ported to new machines, and APL is a fine environment for testing and using routines written in C. Unix system users gain the advantage of APL, a powerful and productive interactive VHLL. APL users get the advantage of all the features of a Unix system (including C and shell programs) available in a familiar and productive form.

The combination of Unix, C, and APL makes a uniquely powerful and synergistic unity. The overlap between Unix and APL use is currently limited because APL is largely an IBM phenomenon and Unix is everywhere else. But it is in this growing overlap that the synergy between IBM's and AT&T's software technologies is greatest. □

References

Iverson, K.E., and Falkoff, A.D. (1981): *A Source Book in APL.* Palo Alto, Calif., APL Press. Includes Iverson's Turing Award paper, "Notation as a Tool of Thought."

Gilman, Leonard, and Rose, Allen J. (1984): *APL, An Interactive Approach.* The most popular text on APL, over 350,000 copies sold.

Martin, James (1982): *Application Development Without Programmers.* Prentice Hall, Englewood Cliffs, N.J. Introduces APL to managers as a productivity tool.

David Saunders discovered APL after spending seven years in management science and data processing in England. He is president of Notation Corp. in Los Gatos, Calif.

The AT&T 3B2/300

BY BRUCE MACKINLAY

This month our crazed reviewer gets his hands on the new AT&T 3B2/300 and takes it apart to tell you about this new supermicro.

The sleeping giant has stirred. Yes, AT&T has come out with a very strong machine, but all is not copacetic. Although the hardware, especially the memory, is very interesting, the lack of a floating-point co-processor makes the machine unsuitable for the lab.

The machine I reviewed was a 3B2/200 with 1-Mbyte internal random-access memory (RAM) and 32 Mbytes of Winchester hard disk. My thanks go to the UX-Software people, of Toronto, and Don Feith, of Feith Systems. They supplied the 3B2 when AT&T could not find one. I am glad to do a gloves-off, hands-on review of the new 3B2. Feith Systems included a 4-channel serial card so I could test it.

The total system, including all Unix system software (and the C compiler) costs $16,000. With this machine, you can run six users and two parallel printers—a lot of bang for your buck. Next month I will review the UX-BASIC. (What? BASIC on Unix. Is this heresy?)

For a little machine, the 3B2 is fairly fast, in many ways outperforming a VAX 730 running Ultrix. It was about two times faster than an M68000 running System III (the Scorpion from Honeywell). However, I rather doubt the widespread belief that the machine can support 18 simultaneous users. It can support 14 terminals; however, if 14 people get on those 14 terminals at the same time, they had best be patient.

Multi-user editing (that is, running many editing processes using ed) showed quite a bit of a slowing down when I simulated eight simultaneous users. Figure 1 shows the spread between the one user editing one task and eight users editing eight tasks. Graphed is a ratio showing the stretching out caused by multiprocessing. The graph shows that the system does not have any sign of thrashing (signaled by a sudden decrease in response time), but that the system has hit its limit at about seven users. The high points of the benchmarks were the system calls, forks, and function calls per second. The WE32000 processor was designed with Unix and C in mind, and this shows in these areas.

FIGURE 1: 3B2/300 SIMULATION OF MULTI-USER CAPABILITY

MULTI-USER EDITING STRETCH

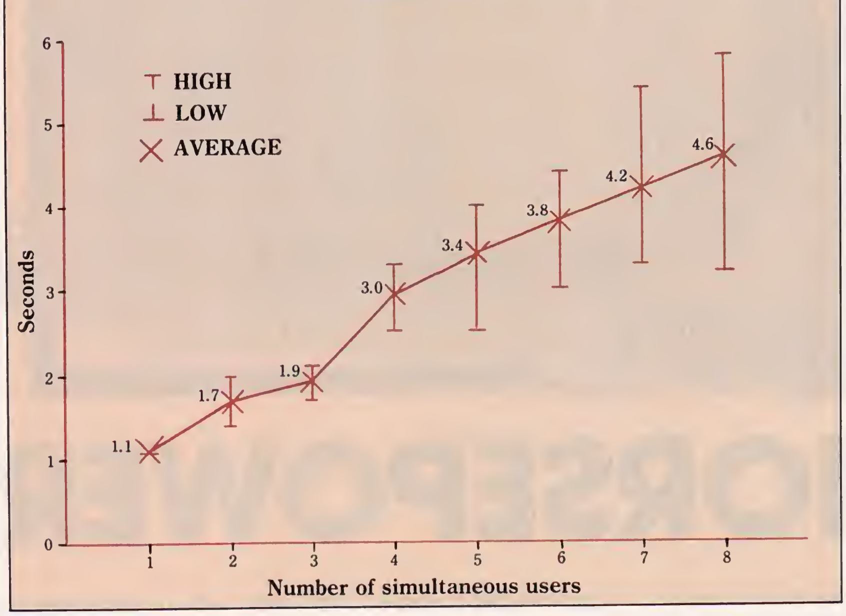
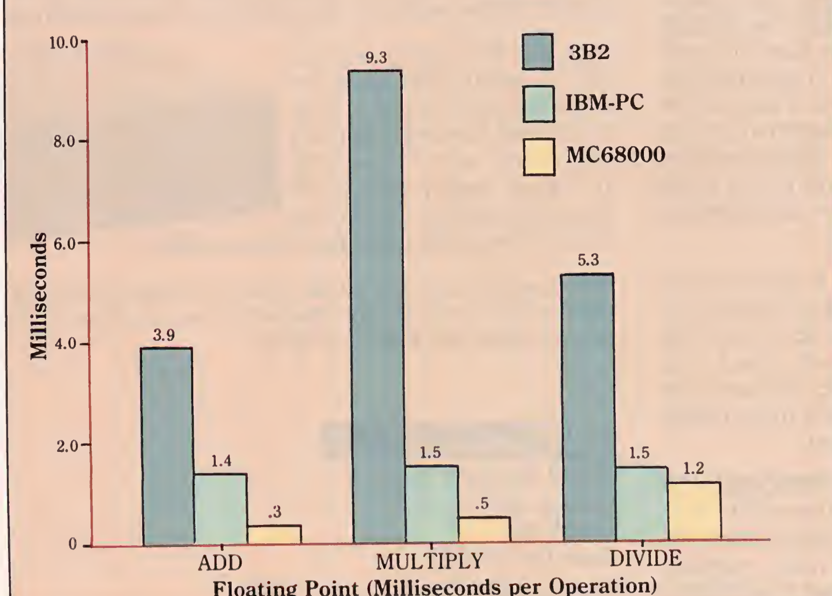


FIGURE 2: AT&T 3B2/300 FLOATING POINT RESULTS
COMPARISON OF 3B2, IBM-PC AND MC68000 (8MHz)



FLOATING POINT SPEED

One fairly surprising benchmark result was the floating-point speed. The 3B2 performs a floating-point multiply six times slower than an IBM PC/XT running Unix (see Figure 2). Both machines perform the multiply in software. The IBM PC uses 16-bit registers and a single bit shifter, while the 3B2 uses 32-bit registers and has a 32-by-32-bit barrel shifter.

After running the benchmarks, I just did not believe the numbers. First, the machine was much too slow; second, the single precision was slower than double precision. I was sure that there must be a bug somewhere or that I was doing something wrong. I called Ram Chelluri (a benchmark expert) at AT&T Technology and confirmed the numbers. The only thing he could say in defense was that the 3B2 conforms to the IEEE floating-point standard.

I went through my notes of past benchmarks to find the numbers for the Scorpion (reviewed in Vol 1., No. 3). The Scorpion performs full IEEE floating-point operations 19 times faster than the 3B2. Granted, the Scorpion is an 8 MHz M68000, while the 3B2 is a 7.5 MHz WE32000, but that does not account for the tremendous difference. Both use 32-bit registers and should have very similar performance.

If you were hoping to use the 3B2 in your lab, you had better get used to waiting. It took almost a full

UNIX™ HORSEPOWER!

There are a lot of UNIX based systems on the market today claiming to be "SUPERMICROS". But do they really have what it takes to run multi-user UNIX well? The IBC ENSIGN™ does and here's why:

FAST MEMORY: No computer running at **any** clock speed can run faster than it's overall memory design. The ENSIGN has up to 8MB of 120nsec memory with dual bit error correction. With IBC's proprietary memory management, **all** of this memory runs with no wait states as fast as the 68000 CPU will go. Compare this to other systems running only small cache memories at full speed. Other multiple user systems cannot load all their programs into a small cache memory. Their systems slow down considerably under a heavy multi-user load.

INTELLIGENT SERIAL I/O CONTROLLER: Even the fastest CPU will slow down when it's trying to handle interruptions from multiple on-line users. The ENSIGN provides slave serial I/O CPU's **and** FIFO buffering for both input and output. The result is the ENSIGN's ability to support up to 32 users, with heavy serial I/O demand, while leaving the main 68000 CPU free to run with little serial I/O overhead.

INTELLIGENT DISK CONTROLLER AND HIGH PERFORMANCE DISK DRIVES: The ENSIGN has a slave CPU to handle all disk operations, **plus** 16K of disk buffering. IBC's proprietary disk DMA allows high speed data transfer to main memory without slowing down the main CPU. Further, the ENSIGN supports SMD type 8" hard disks with much faster seek times and transfer rates than 5¼" hard disks usually found in personal desk top computers.

THE RESULTS: The IBC ENSIGN runs multi-user UNIX at performance levels not attainable by other supermicros.

Call IBC and get a copy of IBC's multi-user benchmarks—benchmarks that test 8 users running large CPU programs, with heavy disk I/O **and** heavy serial I/O simultaneously. You'll find that nothing can compare to the ENSIGN.

If you want to run multi-user UNIX on a high performance system with up to 32 users, 8MB memory, and over 1,000MB disk storage, see the IBC ENSIGN.

IBC/Integrated Business Computers
21621 Nordhoff Street
Chatsworth, CA 91311
(818) 882-9007
Telex No. 215349

UNIX is a trademark of Bell Laboratories

Please circle Ad No. 4 on inquiry card.
Please circle Ad No. 21 on inquiry card.

minute to invert a 10x10 matrix and over an hour to invert a 50x50 matrix (see Figure 3). AT&T plans to add a floating-point co-processor and to do the operations in hardware in the near future, but they would not give me an approximate date.

HARDWARE OVERVIEW

The hardware consists of one small unit, somewhat smaller then an IBM PC, yet it contains a 5¼-inch quad-density floppy disk, a 32-Mbyte Winchester hard disk, up to 2 Mbyte of memory, room for 18 serial and 4 parallel ports, and a full 32-bit virtual memory and direct-memory access on all devices. It might be small in physical size, but it is large in capability. The main system printed-circuit card contains the floppy-disk and hard-disk controllers. In this way, the standard peripherals do not take up expensive extension bus space.

The hard disk comes in two flavors, 10 Mbyte and 32 Mbyte (formatted). 10 Mytes is really too small for most multi-user applications; in fact, 32 Mbytes is a mite too small as well. AT&T will soon announce an expansion box that will add up to two more 70-Mbyte disks and a streamer tape drive for backup. The tape drive is a welcome sight because, even with the large 720-Kbyte quad-density floppy disk, it would take 40 disks to back up the 32-Mbyte hard disk.

After uncrating the machine, I immediately opened the unit to get a look at the machine's guts. AT&T's expertise in chip manufacturing is clearly demonstrated. Looking inside, your eyes are drawn to an unusual card carrying six chips, including the processor, memory management unit, and bus interfacing chips (see Photo 1).

The second thing you notice is the 1-Mbyte memory card, which is an unbelievable 8-by-1.75-inch rectangle. This memory card costs only $2,200 (quantity one). Above the main circuit board are the I/O expansion cards. The system I reviewed contained one serial expansion card. This allowed me to measure the throughput of the serial ports on the expansion bus.

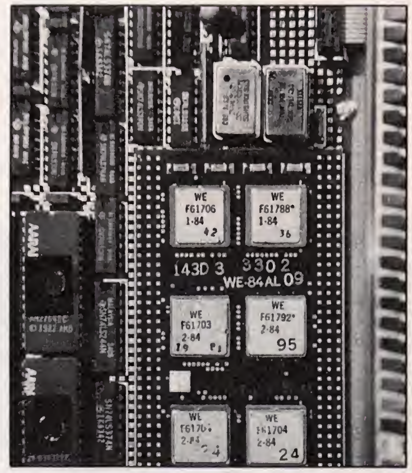
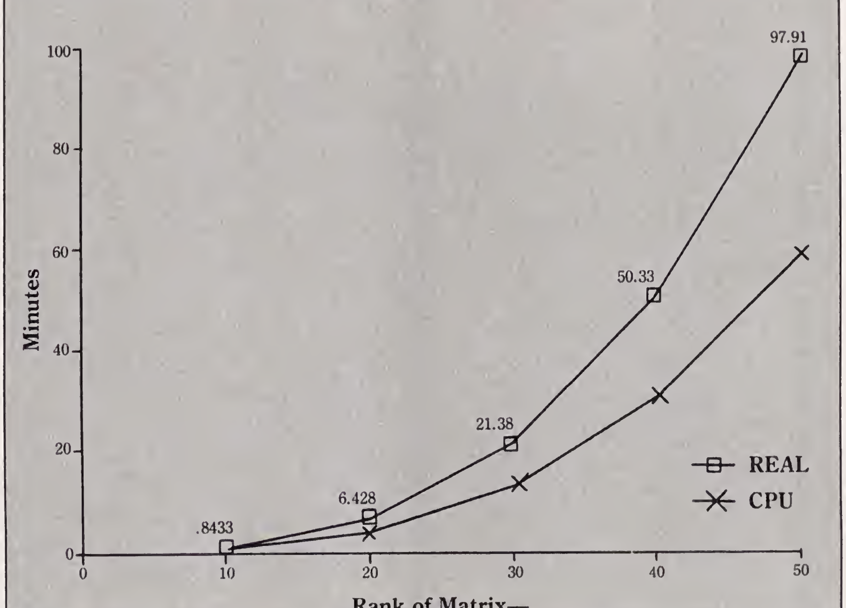


PHOTO 1: VIEW OF THE INSIDE OF 3B2/300

FIGURE 3: TIME TO INVERT VARIOUS-SIZED MATRICES ON 3B2 FLOATING POINT INVERSION TIMES

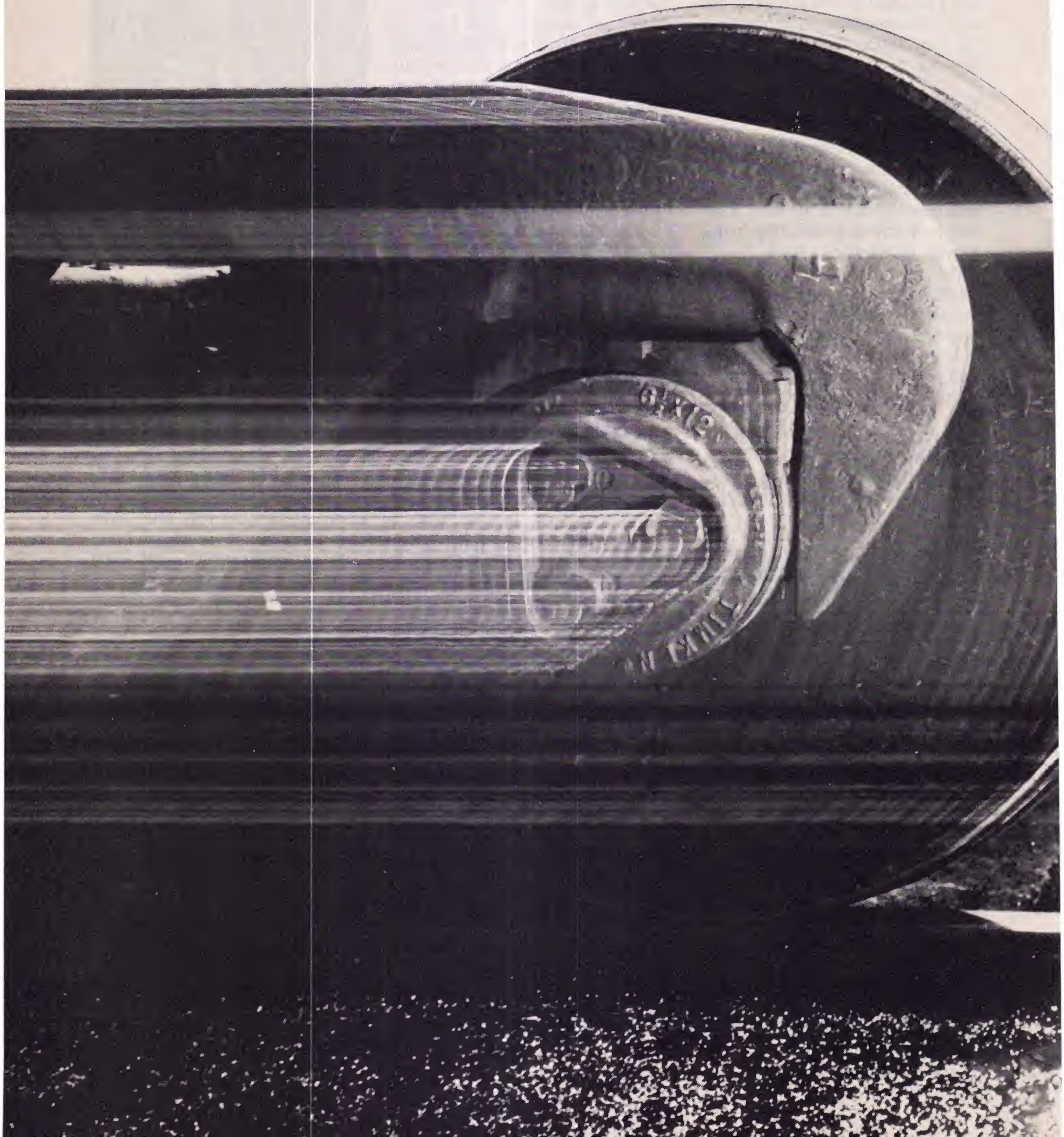


The system provides two buses, a 32-bit and a 16-bit bus. The 32-bit bus is for memory, while the 16-bit bus is for I/O expansion. The 16-bit I/O bus has only four slots. Currently there are only two types of I/O expansion cards, but AT&T plans to release the specifications for its expansion bus, allowing third-party hardware vendors to produce their own cards.

The most common expansion card is a serial card adding four serial and one parallel ports. The parallel port is only for attaching a parallel printer (another reason not to get it for your lab). This card is interesting because it contains its own CPU, using an 80186 16-bit processor, 32-Kbyte local memory, and 16-Kbyte read-only memory.

Each serial line is rated to handle up to 19.2K baud with a maximum throughput of 3840 characters per second for a single expansion card. The card uses DMA to greatly reduce the overhead that a normal serial line places on most systems. The benchmarks show that the system could print 866 characters per

The Association of
didn't spin its wheels with



American Railroads the wrong microsystem.

We're Dual Systems.

And when the railroad people needed a super microsystem to run real-time testing of train wheel axle assembly units with measurements taken at the rate of 500 times per second, they turned to us.

After all, our super microsystems have enjoyed a rather enviable track record with Boeing and Sony and Ford and United Press International, just to name a few.

And for three years we've been delivering 68000-based systems, full UNIX* Version 7 with Berkeley enhancements, and IEEE-696/S-100 bus conformance. That's longer than anyone else.

Which brings us to our newest high-performance micro, the 83/80. Featuring the full UNIX capabilities of Version 7 or System III, the 83/80 can handle up to twelve users and provides a standard 512 KB of dynamic RAM with parity that's expandable to 3.25 MB.

And we've backed it with a full one-year warranty.

Our 83/80 incorporates a high-throughput SMD controller and an 80 megabyte Winchester disk drive with 20-25 milliseconds average seek time. And our backup memory is well worth remembering — it consists of an 8" floppy disk with 1 MB of storage.

In addition, you'll find our 83/80 delivers increased performance through its Dual ported full-track disk buffer and proprietary controller circuiting. More users can access with better response time.

It's also very well-educated. Our 83/80 can read or write up to an entire track of data in a single disk rotation, regardless of where the disk-head settles on a given track.

That's smart.

The C programming language comes standard with UNIX, of course. Other optional languages include FORTRAN-77, PASCAL, RM/COBOL®, LISP and BASIC. And that's just for starters. Optional software includes data base and administrative packages like INGRES and UNIFY.

And the sticker price? Quantity 10 at $14,693.

As you can see, we had our own wheels turning when we designed the 83/80.

For further information, please write or telephone our Marketing Department at 415/549-3854.

We'll be glad to give you more than just a whistle-stop tour of one powerful super microsystem.

DUAL

Dual Systems Corporation
2530 San Pablo Avenue
Berkeley, CA 94702

*UNIX is a trademark of Bell Laboratories.
® RM/COBOL is a registered trademark of the Ryan McFarland Corporation.

Please circle Ad No. 13 on inquiry card.

second through a single line (when the baud rate was set to 9600 baud); this is 90 percent of the rated speed. Printing to two lines on the same card produces 85 percent of the rated throughput.

NETWORK INTERFACE CARD

The other AT&T expansion card is the network interface card that connects the 3B2 to Ethernet. AT&T is keeping its networking plans under wraps and would not say more than that it plans to announce a full X.25 implementation for the 3B2 in either the first quarter or second quarter of 1985.

They would not even say if this network would be a high-speed local-area network (LAN), or if the network would use the network interface card. Internally they have been using a version of X.25 for quite some time now, but I got the impression that the internal implementation of X.25 (called BX.25) was not robust enough for external release.

There are two slots on the 32-bit memory bus for expansion. Each

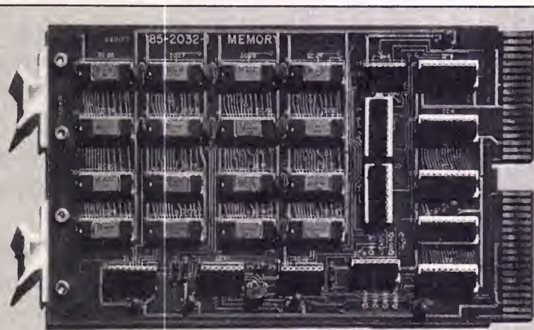


PHOTO 2: A 4-KBYTE MEMORY CARD FROM AN OLD DEC LSI-11/23 MINICOMPUTER

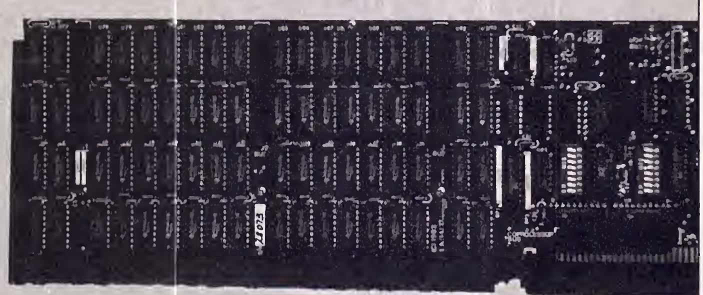


PHOTO 3: A 512-KBYTE MEMORY CARD FROM A RECENT IBM PC

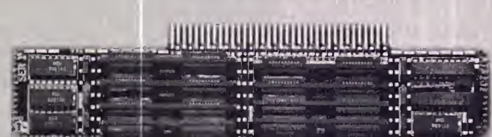


PHOTO 4: THE 1-MBYTE MEMORY CARD FROM A 3B2/300

slot can contain a 1-Mbyte dynamic RAM card, for a maximum of 2 Mbyte of internal memory. If for some reason you want less memory, there is a 514-Kbyte memory card. But 1 Mbyte seems to be a good minimum amount for a multi-user system. Many other "32-bit" processors are limited to 16 bits on the memory bus. Having the full 32 bits on the memory bus should improve performance.

The compactness of the 1-Mbyte memory card is incredible. For comparison, I found a 4K memory card from a very old DEC LSI-11 machine (my first 16-bit computer) and a 512-Kbyte memory card from an IBM PC. You can see the shrinking memory sizes from Photos 2, 3, and 4. Gone are the days when 56 Kbytes of main memory were the most I could put in my machine; now I feel limited by the 640 Kbyte of memory.

The secret of AT&T's super-high-density memory card is the use of 256K-bit chips, packing four chips to a single in-line package. The second trick was to place the package vertically instead of horizontally. Each package contains 1 million bits of storage (with eight packages you get a million bytes of storage).

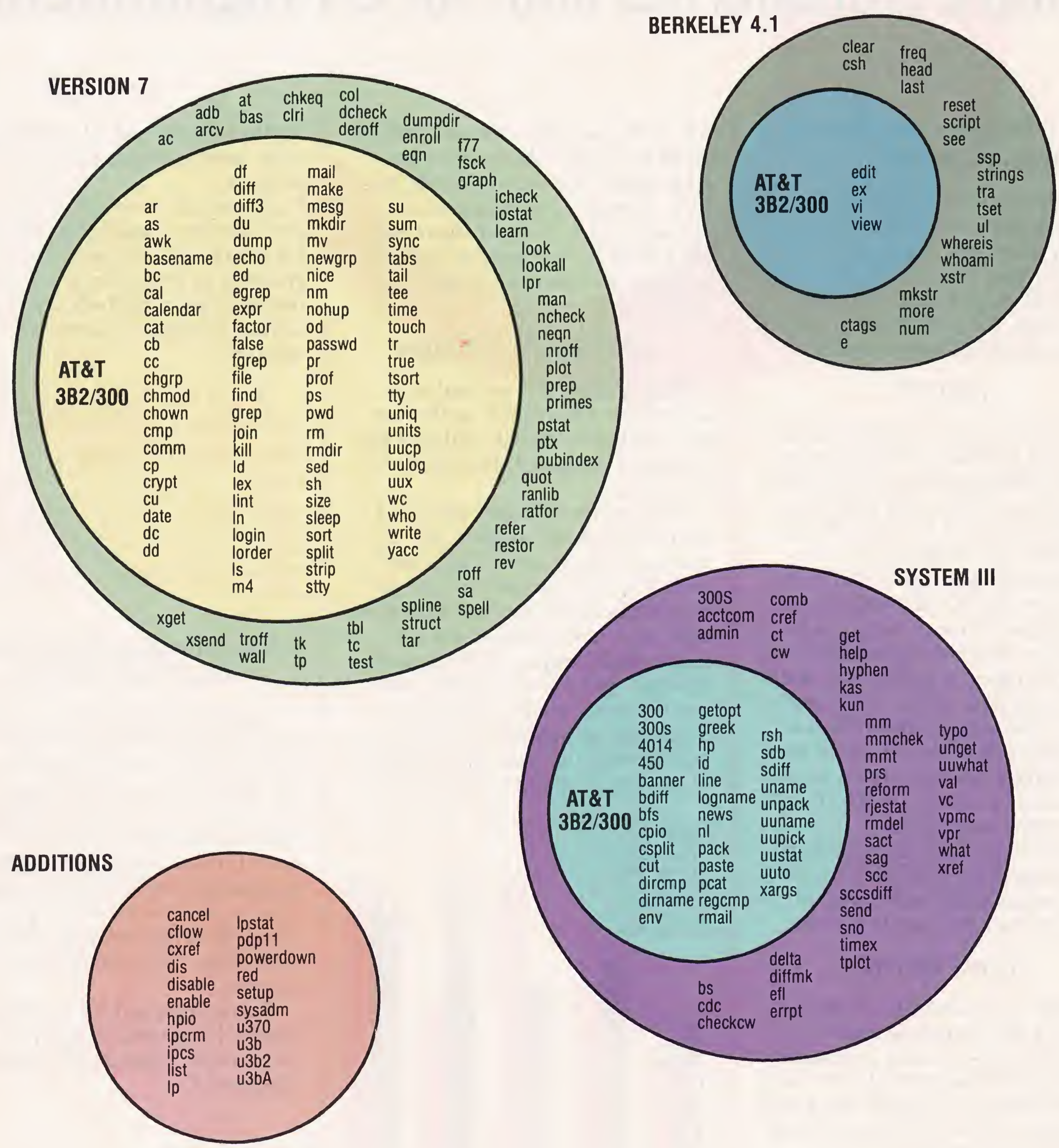
Along with the shrinking size, there has been increasing speed. My LSI-11/23 memory card can access a byte in 340 nanoseconds (ns), according to DEC-reported specifications. The chips on the IBM PC memory card are rated at 200 ns, and the AT&T 256K-bit memory chips are rated at 105 ns.

Another important aspect of the 3B2 is that it contains a FULL virtual-memory management hardware. Unix System V does not use this hardware to full advantage (yet), but in theory this allows a single program to use the full 4-billion bytes of addressable storage. In the past, this feature has only been available on mainframes or superminis.

THE CPU ARCHITECTURE

The heart of the 3B2/300 is the WE32000 processor chip, a true 32-bit microprocessor. The processor clock runs at 7.5 MHz, which is amazing because the processor is implemented using advanced complementary metal-oxide semiconductor (CMOS) technology. AT&T has invented (and patented) a new form of CMOS called "Twin-Tube" CMOS.

UTILITY COMMAND COMPLETENESS



Office Automation Software. . .

Selected for Distribution by AT&T.

HANDLE is the number one office automation or "integrated" software system available for the UNIX™ operating system. Selected by **AT&T** Information Services for distribution on the 3B2/300™ and 3B5™ UNIX systems, **HANDLE** is the most powerful full-featured office automation system in the market today.

Some of **HANDLE**'s unique features include:

- Dynamically integrated word processing, graphics, list management, virtual spreadsheet, and spelling correction with a powerful "document metaphor" that provides a common method for creating, printing, merging, archiving off-line, filing under multiple topics and retrieving documents.
- Full multi-user capabilities with extensive security — documents, or portions of docuents, can be marked "author only", "read only" or "general access". **HANDLE** provides its own record locking.
- User preferences — **HANDLE** conforms to each individual's preferences for such things as preferred printers and individual document catalogs. User restrictions can control such things as the ability to delete documents, access archive catalogs, etc.

The AT&T 3B2/300 computer with HANDLE on screen and integrated graphics printouts.

The first Office Automation System for UNIX.

HANDLE CORP, 140 MACKINAW ROAD, PO BOX 7018, TAHOE CITY, CA 95730 • 916-583-7283 **Please circle Ad No. 18 on inquiry card.**

This allows AT&T to make chips with the advantages of CMOS (coolness, low power, quiet) without being very slow.

The CPU contains 180,000 transistors, not a record, but still a lot, especially for CMOS. As a comparison, the Intel 80186 contains 55,000 transistors and the 80286 130,000 transistors. It is not the number of transistors that counts but the overall design, or architecture, that matters. By definition, a bad design produces poor performace.

In July 1981, W.A. Wulf published a seminal paper that presented the essential principles for designing a CPU for programming languages and modern operating systems. Most computers previously had been designed for machine or assembly language programming. The WE32000's design, however, was based upon Wulf's principles.

The WE32000 was designed to run C code efficiently. For example, the string operations and string format of C are directly supported in machine code, and there are simple instructions to implement the C functions `strlen` and `strcpy`. Furthermore the bit field types are directly supported in hardware and machine code. The WE32000 has a full 32-by-32-bit barrel shifter. And finally, the C function and subprocedure linkage is directly supported via a set of calling and returning primitives.

One of the most important areas described in Wulf's paper is that of arithmetic and logical instructions. On older machines, it was hard to write compilers, and producing fast, compact code was almost impossible. On the WE32000, compilers should produce very fast and efficient code. Even though the designers worked with C in mind, this does not mean that Pascal or other programming languages will produce slow or inefficient code. The designers try to speed up all languages, but C got special attention.

PRODUCT OVERVIEW

Hardware	
Model:	AT&T 3B2/300
Price:	$16,000
Configuration:	1-MB RAM, terminal, 720KB floppy, 32MB disk
Processor	
CPU:	WE32000
Cycle Time:	7.5MHz
Bus:	32-bit and 16-bit proprietary
Main Memory:	514KB to 2MB 32-bit wide RAM
Floppy:	1 720KB 5¼ inch (formatted)
Winchester:	10-33MB 5¼ inch (formatted)
Software	
Version:	System V
Shell:	Bourne only
Languages:	C, f77, and third party (BASIC)
Third Party:	many packages

Older CPUs are very slow when switching from one process to another. This overhead can decrease the throughput of time-sharing systems by up to 20 percent (or even more on some older operating systems). This is important wasted time. The WE32000's designers have tried to streamline and speed the process switching. You can see the effects of this careful design in the benchmark results. The operating system calls and forks per second are very fast.

If you want more information about the design of this interesting processor, I suggest that you read the paper printed in the "Proceedings of the Symposium on Architectural Support for Programming Languages and Operating Systems," Vol. 10, March 1-3, 1982, pp. 30-38, Association for Computing Machinery.

UNIX ENHANCEMENTS AND PROBLEMS

The 3B2/300, of course, runs Unix System V, but there are some surprises in this implementation. First of all, Unix comes unbundled. This means that if you are used to having all of the Unix system's tools and programs on your machine, you are going to have to purchase a number of software packages.

The system comes with a Unix Core that contains the essential utilities (and not much more); you can add to the system with 18 additional software packages. For example, `lpr`, `dump`, `nice`, and many others are not considered essential. Some unbundling makes sense, but AT&T has taken the idea to too great an extreme. It makes sense to separate the development tools from the

BENCHMARK MEASUREMENTS

Arithmetic Instruction Times (microseconds per op)

	short	*long*	*float*	*double*
+add	2	1	3922	2128
*multiply	7	4	9346	7463
/ divide	16	13	5319	3663

Memory Loop Access Times (microseconds per byte)

	read	*write*	*copy*
Char type	944 ns	5	6
Short type	871 ns	3	6
Long type	448 ns	653 ns	2

Input/Output Rates (bytes/sec)

	read	*write*	*copy*
Disk	47K	29K	17K
Pipe			154K
TTY 1		866	
TTY 1+2		1631	
RAM 1-byte			154K
RAM 4-byte			517K

Array Subscript References (microseconds)

short[]	long[]
7	5

Function References (microseconds/ref)

0-parameters	1-parameter	2-parameters
funct()	funct(i)	funct(i,i)
19	24	29

Process Forks (540 Kbytes)

23 per second

System Kernel Calls (calls per second and microseconds per call)

getpid() calls:	3 kcalls/sec or	345ms/call
sbrk(0) calls:	2 kcalls/sec or	513ms/call
create/close calls:	159 pairs/sec or	6289ms/pair
umask(0) calls:	2 kcalls/sec or	421ms/call

C Compiler Time Estimates:
Compiler can handle 1801 symbols
CPU: 13 sec + 13 lines/sec
REAL: 1.3 x CPU

production tools, but to take the development tools and break them into four separate packages makes me wonder.

AT&T even went as far as separating the system administration tools from the essential tools. I can't imagine anyone using the 3B2 without the system administration tools. In defense, I must say that AT&T has priced these additional software packages very well. The line printer spooling package, for example, costs only $50.

AT&T has added some interesting capabilities to the 3B2. For example, a software package allows you to use the Teletype 5620 bit-mapped terminal as a windowed graphic system. This allows you to open multiple windows, each one running a different program. However, I really can't say much about how this works because the machine I reviewed did not include the terminal.

In the manuals I received, there were pages discussing the 3B network capabilities. As I understand it, this is the internal implementation of the BX.25 networking. Although I could not test the networking, it sounded very good. If the final released software is better than this, then the people who sell Berkeley 4.2 Unix had better look out. For those of you who are addicted to `uucp` and `usenet`, there is a basic network package containing all the traditional `uucp` tools.

The lineprinter spooling package contains a large number of commands, but I can't make a judgment because the software and the manual were not included. `vi` is available, but `csh` is not. SCCS is not available for the 3B2.

One improvement over the typical Unix system environment is in the area of system administration. There is a new `sysadm` program that allows you to do most system administration via a menu. With this program, an intelligent novice (most secretaries) could perform all the functions of the traditional system administrator/operator. What is lacking for this intelligent novice is instruction of when and why to use `sysadm`.

RECONFIGURATION UTILITIES

Another improvement that deserves passing comment is the reconfiguration utilities. This will allow a third-party hardware vendor to sell you extension hardware with the necessary software, and you, the owner of the 3B2, could install the hardware directly into Unix. In essence, the reconfiguration utilities will allow the owner to re-link the Unix kernel to take advantage of changing hardware. Without this capability, it has been hard or impossible for owners to customize their Unix machines.

There are two interesting additions. In both cases, I did not have the software and could not test them, but they sounded good on paper. The first is an interprocess communication package that allows separate processes to communicate via a queue of messages. Pipes are the traditional Unix system interprocess communication method, but for some applications a pipe just will not do.

Pipes are one-directional, byte-oriented channels between two closely related processes. Message queues allow two (or more) distantly related processes to communicate. With message queues you can establish a single service, like a database server. When any process wants to use this service, it can simply send a message to the server and wait for a reply.

The second interesting package consists of interprocess semaphores. These are very low-level routines that allow two parallel processes to coordinate or synchronize their tasks. One of the biggest problems with older versions of the Unix system is the lack of any synchronization and mutual exclusion primitives.

`lock` files are used in older versions of the lineprinter spooler and networking (`uucp`) programs. These `lock` files were used to make sure that only printer or sender is running at any instant, but `lock` files can, and often do, fail. This locking is a form of mutual exclusion or synchronization.

Another area where the Unix system lacks mutual exclusion is in file- or record-locking. Without true mutual exclusion, two processes can update (write on) the same file at the same moment, causing a collision and corruption of the file. It could be possible to implement full file- and record-locking using semaphores. In November AT&T plans to release a new version of System V that should contain the /usr/group's version of file-locking.

The addition of these two very powerful tools makes the Unix system much more robust. Again, I did not actually receive the software packages and could not test the software. And I am concerned about the amount of system overhead that these tools will add. Generally when you add new tools and system calls, you tend to make the rest of the system more complex and therefore slower.

DOCUMENTATION

AT&T has completely rewritten the manuals for System V. Gone are the large fat binders with the traditional Unix system manual pages and reprints of internal and academic reports. Now, sections for each software package present the software for the beginner. In the back of each manual is an appendix containing the traditional manual pages from the original Unix system manual. The binders are smaller, about the size of the IBM PC manuals.

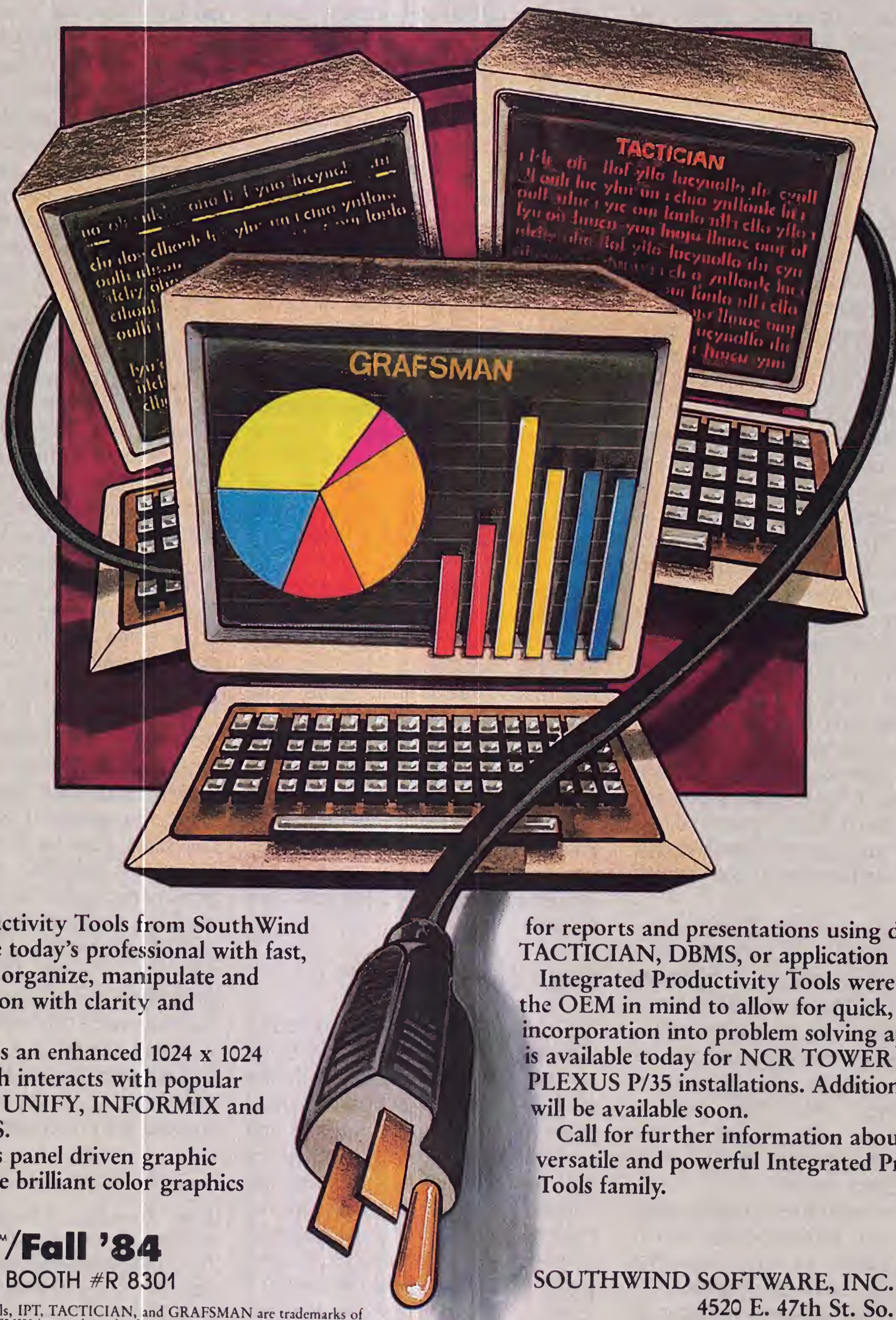
Overall, the manuals are much easier to read, but it has become harder to use the manuals as reference manuals. When I was running the benchmarks, I found it difficult to know where to look for specific information. This problem could be solved very easily by expanding and improving the index. In fact, the current index was very helpful. If the people at AT&T want to improve an already excellent set of manuals, then effort spent on the index would go a long way.

One of the areas where AT&T has made a good beginning is the owner/operator manual. This manual is still a little thin, but it is a godsend when compared to the void that existed in the past. What is lacking in this manual is the realization that the intelligent novice will be reading the manual. These novices need very clear instructions for the care and feeding of a 3B2 and the Unix system. The novice might not know when or why to back up the system regularly.

This machine is a good one, but I expected more. If IBM or Digital Equipment had produced it, then I probably would have given it a rave review. But after years of reading about the advanced research in the Bell Labs, I felt let down. When you judge the 3B2/300 purely on its price/performance, it is an excellent machine. Consider that it is just the beginning for AT&T; it is the giant's first sleepy steps. When you consider plans for enhancements to the 3B2/300 and AT&T's support, it is an excellent choice for a small business or a department of a larger corporation. In fact, I want one, and I will suggest it to my clients. □

Bruce Mackinlay, a frequent contributor, is a senior partner in NOVATECH Systems, Inc., a software development and consulting firm. He is currently working with UNIX/WORLD Magazine in developing our computer facilities.

Fast...Versatile ...Powerful Tools



Integrated Productivity Tools from SouthWind Software provide today's professional with fast, versatile tools to organize, manipulate and display information with clarity and effectiveness.

TACTICIAN is an enhanced 1024 x 1024 spreadsheet which interacts with popular UNIX DBMS ie. UNIFY, INFORMIX and MICROINGRES.

GRAFSMAN's panel driven graphic editor helps create brilliant color graphics for reports and presentations using data from TACTICIAN, DBMS, or application files.

Integrated Productivity Tools were designed with the OEM in mind to allow for quick, easy incorporation into problem solving applications. IPT is available today for NCR TOWER 1632 and PLEXUS P/35 installations. Additional IPT ports will be available soon.

Call for further information about the fast, versatile and powerful Integrated Productivity Tools family.

See us at
COMDEX™/Fall '84
Riviera Hotel BOOTH #R 8301

SOUTHWIND SOFTWARE, INC.
4520 E. 47th St. So.
Wichita, Kansas 67210
316-788-5537

1-800-346-3025 EXT 234

SOUTHWIND SOFTWARE

Integrated Productivity Tools, IPT, TACTICIAN, and GRAFSMAN are trademarks of SouthWind Software, Inc. UNIX is a trademark of AT&T Bell Laboratories. UNIFY is a trademark of the Unify Corporation. INFORMIX is a trademark of Relational Database Systems, Inc. MICROINGRES is a trademark of Relational Technologies, Inc. NCR TOWER 1632 is a trademark of the NCR Corporation. PLEXUS P/35 is a trademark of Plexus Computers, Inc.

Please circle Ad No. 38 on inquiry card.

Locus

A TRANSPARENTLY CLEAR SOLUTION

With its distributed version of the Unix system, Locus says it has the remedy for networking ailments.

BY JEFFREY COPELAND, VANESSA SCHNATMEIER, AND ALAN WINSTON

While some companies map out the perfect paths to connect micros to mainframes, larger firms struggle with the question of how also to link mainframes and minis, or even mainframes and mainframes. Locus Computing Corp. thinks it has arrived at the solution for companies that utilize the Unix system in all levels of computing—a distributed version of the Unix system called, not too surprisingly, *Locus*.

Locus' goal is to present all users in a local-area network with a "single-system image." Locus wasn't designed to be just another version of the Unix system, but a network transparent down to its internal kernel data structures, files, blocks, and i-nodes. Through Ethernet, the system shares not only files, but tasks as well. A device available at one site can be available at any other.

The Locus system automatically replicates stored data on a file-by-file basis. Users can set a "dial" to indicate the number of copies they wish to make and at which sites. As a result, all data is much more available and reliable because a crash at one machine only means that the network retrieves a properly updated copy of the then-open files from somebody else's disk.

Heterogeneous CPUs (M68000s, VAXs, and, we suspect, 3B series machines from AT&T) can also be linked in a way that keeps the environment exact, though there are some limitations. As far as heterogeneous flavors of the Unix system go, the version we saw was a superset of Berkeley 4.1. By the time you read this, however, Locus should be fully compatible with System V and downwardly compatible with 4.1, although Locus Vice-President Charles Kline admitted that there were still some details to be ironed out. In short, Locus bills itself as almost a panacea for the woes of networking—and it largely fulfills its claims.

TRANSPARENCY

Locus operates through what company President Gerald Popek refers to as "transparency," making a collection of machines look like a single machine with a single root file system. This extends to programs too—any program can be run on any machine in the system. We saw the way the system operates at the company's offices in Santa Monica, Calif., where Bruce Walker, senior systems architect, put Locus through its paces.

The first aspect of functional transparency is data transparency, which is achieved through name transparency and uniform access to data regardless of location. Having only one root file system for multiple machines allows a unique name for each file anywhere in the system. Like any other Unix system, the file can be found by a unique pathname. An obvious way to do this would be to put all files on the machine huey on a file system named `/huey`, the files on dewey under a file system named `/dewey`, and so on.

However, this scheme would prevent data from being replicated across multiple machines—one of Locus' primary features. Instead, the file system is structured so each machine has a copy of the root and each can convert a pathname into a pair of file system and i-node numbers from which the data can be found anywhere on the multi-machine system. This means that the file accessed from huey may actually be located on both dewey and louie.

However, data transparency extends further on Locus. The single file system gives not only name transparency, but location transparency as well. For example, on DECnet, moving a file changes its name from `HUEY::DBA1:[SYSEXE]FOO.EXE` to `DEWEY::DBA1:[SYSEXE]FOO.EXE`. A file's physical location on Locus can change from huey to dewey, and its name will still be in `/bin/foo`.

A general result of this is complete device transparency. It is possible to access louie from my terminal attached to dewey, or louie can access the line printer attached to huey. Similarly, the disk attached to quasi can be read from modo.

This is all possible because Locus is constructed so that remote operations are efficient. The networking is firmly embedded in the operating system kernel, and, as a result, all messages are from operating system to operating system. All network transactions are unseen by the user program. The data transfer protocol is specifically tailored for fast operation over Ethernet.

RELIABILITY BY SYNCHRONIZATION

Since files can be in multiple locations, the reliability of the system is aided, but the system uses extra care to make sure that all files agree. At the heart of this is atomic file update, which ensures that, if a failure occurs while an update is in progress, the file is left in either the old state or the new state, not something in between. Once the copy in use has been updated, the copies elsewhere in the system are updated. In the case of a crash, information in the i-node allows the system to select the latest version of the file.

A benefit of atomic updating is that it removes the need for things such as editor checkpoint files. If the file on disk is updated immediately when the write is performed [Locus adds a `commit()` system call to force a file's changes to be committed to disk], then saving the keystrokes from an editor session isn't necessary because the changes to the file are already recorded on disk.

On a standard Unix system, this isn't possible because the actual write-to-disk can take place at any time after the `write()` system call has been executed. A `vi` that did a `commit()` every 20 keystrokes would be worth having since a system crash couldn't destroy more than a few minutes' work.

Because each site has a copy of the root, each site has a complete file system, and sites can be dropped from the network for repair or replacement without damaging the system as a whole. And the system contains a substantial amount of software to reconfigure itself as nodes are removed and replaced. Popek explains, "Whenever you take the machine off-line for testing, you just push the reset button."

Locus also provides tasking transparency. Popek says: "You can `fork`, `exec`, `migrate` tasks in the network, and, most importantly, the environment is identical so that when the program moves, the correct operation continues. If it had open files, and it was sharing the open files with other programs, all that is preserved. Not only are the kernel calls transparent, but [so are] the internal kernel data structures, file block, i-node, all of that." Locus does this by adding a few system calls to the normal complement.

`fork` and `exec` work as on a standard Unix system and have identical calling sequences. `fork` has been modified to allow the child to execute at a foreign site of the same machine type. The only change to `exec` has been to allow executing of the load image at a different site, which doesn't necessarily need to be the same type of machine.

The first of the new calls is `setopt`, which provides a list of preferred sites for the other calls. Using `setopt`, it's possible to say, "The next time there's an `exec`, make sure that the task goes to `dewey`." The `run` system call combines the functions of `fork` and `exec`. The most important of the new calls is `migrate`, which takes the currently running process and moves it to a new site on the network.

A program can be migrated from one machine to another as long as they are the same type of machine. However, an executable program for one type of machine can reside on a different type of machine, and it can be run by transmitting the program across the net and remotely executing it, or by executing the task remotely.

One of the simpler new application programs that Locus provides is *on*, which simply does a `migrate` and runs the named program with the given arguments on the specified machine; for example, `on huey cc /usr/bob/mech.c` would compile the named program on huey.

PERFORMANCE

Comparing a full-scale Locus system to an ordinary Unix system implementation is like comparing a bunch of grapes to an orange, and comparing it to other networking environments is like comparing those grapes to a bowl of mixed fruit.

To take the problem one grape at a time, we can compare a single Locus system to a comparable Unix system. According to tests Locus has run, Locus and the Unix system run locally at very much the same speed. Locus sometimes wins, but not by much. The Unix system also sometimes wins, but again not by much. Elapsed time is very much the same, although Locus spends more time in the kernel in I/O-intensive operations.

The comparison to standard network environments is much harder to do. A certain amount of cunning can speed up any benchmark run anywhere, and one could probably find a configuration that, with the appropriate tweaking, could beat Locus on each—though not every—kind of test.

Locus runs efficiently without much manipulation because it uses a low-overhead strategy for network I/O. Instead of a complex handshaking protocol with requests for acknowledgement, Locus sends out its I/O request directly. The originating CPU is responsible for retrying the operation if the request isn't fulfilled in a reasonable time.

The network hardware over which the Locus system runs (it supports several types as of this writing) can make a big difference to performance. It turned out at one point that block data transfers ran at about half the speed using 3Com rather than Interlan Ethernet boards because the Interlan interface board did DMA transfer, while the Ethernet required a character-by-character copy into the CPU. The critical factor became the speed of the interrupt loop to load the board's buffer, rather than the protocol.

A reasonable comparison—one grape at a time—is that of Locus running locally to Locus running remotely. The results Locus provided show that remote Locus is usually slightly slower than Locus running locally and that it has noticeably more kernel overhead. The percentage increase in elapsed time is extremely small, seldom exceeding 3 percent.

The remote processing overhead can pay for itself in load leveling. A command called `fast` will run your program on the CPU with the lightest load, so elapsed time can be minimized at the expense of machine cycles in transferring the program there.

RELIABILITY AND FLEXIBILITY

Locus' support of atomic transactions at the kernel level is an enhancement we'd like to see as part of standard Unix. Using this facility, you know your file will be left completely updated or not updated at all in the event of a system crash. This makes recovery considerably easier.

Reliability is also provided through file replication. Since the CPU owning the only copy of a necessary file might be unavailable, multiple copies can be made in a transparent manner. They are all referred to by the single file name, and a write to one is a write to all. A read from one of them is a read to the closest version. A lot of care has been taken to keep these copies in synchronization and to get them back automatically into sync if they get out of it.

In addition, Locus keeps running without modification even if one node of the network goes down. The other machines don't need to be rebooted, and they don't come down—even if processes using resources on that remote machine have to be aborted. When a CPU rejoins the network, copies of files that live there are automatically brought back into synchronization.

Locus is designed to run a network of general-purpose machines. The installation is not required to dedicate machines as file servers. However, if there's some benefit to be gained by doing so, Locus supports that very readily. You simply ask for a file that's on the machine being used as a server.

If one machine in the network has an array processor attached, one need only request that a number-crunching program be run there. This kind of optimization can readily be done ad hoc, without the massive software changes that might be required by another scheme.

Because resources have unique names, programs that use them can run on any machine in the net. If a tape drive is taken off a particular machine and put on a different one—during maintenance, for example—programs run unchanged, which they could not do if the device were identified by its location.

In addition to the benefits derived from the ease of system use and the ability to get some value out of a network without rewriting all one's existing applications—benefits which would be difficult to put a dollar value to—there are very tangible advantages in reducing the number of peripherals required, at least those peripherals that aren't in constant use.

Block data transfers are sufficiently cheap that the beta test site at UCLA shares one tape drive among 17 VAXs and uses it to take dumps from all the machines.

At Locus' offices, their five-CPU net shares a single laser printer. These peripherals look local to processes that use them, and no special coding has to be done in the application to make the peripheral-sharing work. This could make it worthwhile to buy an expensive device that can't be justified by the need of a single CPU, but which can be shared among many. □

Jeffrey Copeland is a member of the technical staff at Interactive Systems Corp., Santa Monica, Calif. Vanessa Schnatmeier is a frequent contributor to UNIX/WORLD who also authored this month's cover story on portability. And co-author Alan Winston, a freelance writer and full-time programmer/analyst, has written several articles for UNIX/WORLD.

LOCUS GENESIS

The idea of a transparent network is something that Locus' president, Gerald Popek, had been kicking around since the late 1970s. Popek first noticed the difficulties inherent in working with less-than-ideal networks during his stint at Xerox's Palo Alto Research Center (PARC), "back in the 3-megabit Ethernet days."

He remembers being shown the hardware at PARC and talking with his colleagues there: "I sat down at a bit-mapped display, the what-you-see-is-what-you-get editor, and it was just delightful. Then I said, 'Gee, how do I say *who*?' and they said, 'Well, you don't.'

"Okay, my local disk is filling up, how do I get more space? 'Well, we'll go over and we can talk to so-and-so, and we'll arrange an account on the file server off the other machine. And we'll make you a file directory and you `ftp` your stuff over, and you `ftp` it back. If you've already got a copy over here and you `ftp` the older version back and smash the most recent version, it's your problem if there's not enough space....'

"That was the genesis of this work. It occurred to me that it should be no more difficult to write software to access a remote resource," Popek said.

Guided by his own experiences with single Unix system machines that broke down and prevented him from submitting a $600,000 proposal, and with the six months and 3,000 lines of code it took to write a printer spooler for a one-megabit network, Popek believes that "people should not be thinking about machine boundaries, they should not be thinking about replication. Computers ought to do that for people. You clearly want to simplify people's view of the world."

Popek, a specialist in data security, brought computer science graduate students with him from UCLA to found Locus Computing. The company's products center on the idea of network transparency. They quickly learned there is transparency and *transparency*; that is, some parts of achieving this Holy Grail of transparency were easy, while some were not.

Other groups have attempted to achieve this ideal, with varying amounts of success. One major military project, the National Software Works, aimed to provide transparency over the ARPAnet, but that project was terminated. "It wasn't even *declared* a success," Popek said.

Locus has decided not to market its version of Unix singlehandedly but instead has approached original equipment manufacturers (OEMs) with the lure of a non-exclusive marketing agreement. This method, Popek said, gets Locus out of two sticky ends of the software business—and allows the firm to concentrate on being a software house.

The company took a similar stand with its other major product, the PC-Bridge, another package centered on transparency. The version of the PC-Bridge that AT&T is marketing permits IBM-PC users to access Unix files on the AT&T 3B computers as though the files were another PC-DOS disk drive. Locus is reputedly working on other versions of the PC-Bridge for other vendors, though no names are currently forthcoming.

As for the Locus product, no vendor names had been announced for it at this writing. AT&T, the company's only announced vendor to date, would seem an obvious choice. All that the firm would say is that news on a Locus vendor ought to be available by October 1984.

Popek is sure that his company's product fulfills a clear and barely explored niche in the market, a niche that must be exploited if distributed computing is to succeed. He asked rhetorically: "Where are those distributed applications going to come from? Are we going to build them anew? Are these the IBM 360 days all over again? Or are we going to recycle existing software?

"It's clear, either you're going to run existing single-machine programs in a distributed environment, or distributed computing is not going to happen very rapidly."

THE LOCUS PC BRIDGE

Locus' interface bridges the gap between PCs and mainframes, and between MS-DOS and Unix. This LAN punches through the data and price barriers.

BY ALAN WINSTON

Want to run a spreadsheet, report generator, or color graphics package with data from your Unix file system? Want your personal computers to share data without swapping floppies back and forth? Want to run twice as many `vi` tasks without bogging down your Unix system? Locus Computing Corp.'s PC Bridge may be for you.

This software package allows MS-DOS personal computers to share data directly with a mainframe host or hosts across a local-area network (LAN). The product's unique aspect is that the personal computer—and the system utilities and applications programs that run on it—treat the Unix system host as one big disk drive, even though the connection to it is through an Ethernet, Omninet, or RS-232 port. Off-the-shelf PC applications—like Lotus 1-2-3, dBASE II, and WordStar—run handily, loading from the Unix system or from a floppy, taking data across the LAN, and using the MS-DOS system calls to create, destroy, or update files on the host.

The PC Bridge is divided into four main parts. First, there is a program the PC user must run to make a connection to a host system. This program, called Bridge, asks for user ID and password, gives a menu of host systems available, and logs-in to the host system selected. The session will then have whatever restrictions the user would have if he logged-in directly—this isn't a back door by which PC users become superusers. Bridge is network dependent, and a new version must be written for each configuration.

Bridge establishes a connection to the Bridge Server running on the host. The Bridge Server is about 30K of reentrant code that translates and performs I/O requests from the PC. It does the Unix system calls to implement I/O and performs other functions required by the interface. For example, the Bridge Server hashes Unix file names—even those that won't fit into the more rigid MS-DOS format—into unique MS-DOS names. The Bridge Server can also run a filter to transparently convert Unix text lines (ending with linefeed) into MS-DOS text lines (ending with carriage return and linefeed).

The I/O requests from the PC come from the Bridge Handler. This program takes up about 17K and runs essentially between the application or utility and the operating system on the PC. The Bridge Handler examines each OS call from the application, acting on those that concern the host and passing the rest on to MS-DOS. In this way, the application program can, for example, copy a file from the host to a floppy, or vice versa, without having to know that it isn't a completely local operation.

A TERMINAL EMULATOR

The fourth part is a terminal emulator, which runs through the LAN port. As a result, some screen I/O can be blocked, speeding up activities like scrolling in an editor, graphics screen refresh, or anything else that requires redrawing a large portion of the entire screen. Because the data is sent blocked, only one interrupt per screenful is required, and load on the host is substantially reduced compared to running the same tasks on conventional terminals.

The PC Bridge is only available for non-exclusive OEM licensing, so pricing is not yet known. At this writing, the first licensee has not yet announced their end-user price—but when AT&T does announce it, you'll probably hear about it.

The potential audience for this package seems large and varied. In the commercial environment, the businessman who wants to run a spreadsheet or report generator on corporate financial data can read the system files. He can also develop an interpreted BASIC program to do calculations or graphing since any language that the PC can compile or interpret—including LISP, PILOT, LOGO, C, PROLOG, Ada, and many others—can process data from the Unix host.

It could also be a good environment for MS-DOS software development, given cross-assembler support on the host. Multiple programmers, talking to the host through the terminal emulator, could work on various modules of a major program product and the version control problems that would arise if all were on separate PCs would be handled automatically. It would still be easy to test the programs: get out of the terminal emulator and download the latest version from the host system.

The transparency of the system and the unrestricted access to the files one is authorized to use yield several benefits. PC users don't have to learn anything about Unix to use the Unix file system, so they can concentrate on the idiosyncrasies of the MS-DOS and whatever application program they choose. The PC program gets the latest information from the mainframe without requiring any programmer intervention to produce floppies with the necessary data and without the time lag such a process would demand. PCs can share data

The bridge makes the Unix file system transparent so users only have to deal with the idiosyncrasies of MS-DOS.

without moving floppy disks around; the file to be shared is placed on the host, and each PC may have access.

MINIMIZING BACKUP PROBLEMS

Backup problems—the bane of PC users—may be minimized if all data is placed on the host system. The system manager could back up everybody's files simply by backing up one disk pack. Security problems could be addressed by centralizing the PC data in the computer room, where access can be restricted.

Using the terminal emulator, in some modes, results in less load on the host—enough to effectively double the maximum number of `vi` users. This doubling has the advantage of making the most of the hardware already in place, although it naturally won't be any help if the host is CPU bound rather than I/O bound. PCs on the system can be minimally configured with a single floppy drive rather than the hard disk of the XT. This cuts the price of a PC almost in half.

Locus officials demonstrated the product for me at their offices in Santa Monica, Calif. It certainly appears to do what it was designed to do. I saw it working with an off-the-shelf version of Lotus 1-2-3. That program has a data management menu designed to insulate users from the vagaries of MS-DOS, allowing them to delete or create files with a few keystrokes. With "drive D" (the Unix host—an AT&T processor) as the attentive drive, 1-2-3 performed the same function in the same way—with Unix files! Users never need to know what their current directory is; their only concern is with their current drive.

I also saw WordStar running, saving text files to the Unix system. This suggests that a user could prepare a document on his home PC, bring it in to work, and store it on the host, where it would be available to all. dBASE II also runs well, certainly quicker than on a floppy-based system. These three very popular PC programs convinced me that the interface really is transparent.

Performance is quite good. PC programs downloaded from the host load noticeably quicker than from a local floppy, and somewhat quicker than from a local hard disk. Programs that use a lot of disk I/O run better through the interface than to a local disk because of the greater speed of the big disk drives on the host. There's no particular difference in running a computation-intensive PC task since that's all local processing.

The performance impact on the host is no more than having that many users logged in, and usually less, since the typical PC user—using the standard interface and not the terminal emulator—won't be compiling or kicking off CPU-intensive programs. The terminal emulator, as explained above, can actually lighten the load on the host processor, at least in comparison to dumb-terminal sessions.

I have only two complaints about this product. First, even though all the MS-DOS commands and programs may be stored on the host, and none of it has to be brought in locally, you still can't run an entirely disk-less PC workstation. A floppy disk is still required to run Bridge because you can't download anything until after Bridge has been run. Second, the product is so far only available through AT&T, and only for their hardware as host. Other OEMs ought to check this package out. □

LOCUS PC INTERFACE SPECIFICATIONS

	HOST	PC
Memory	30K (re-entrant code)	17K on each PC
Operating System	Unix or Locus (distributed Unix)	MS-DOS or PC-DOS, Version 2 or later
Hardware	To buy from AT&T, a 3B series processor. For other OEMs, anything that runs the Unix system and has an Omninet or Ethernet connector or an RS-232 communications processor.	IBM or IBM-compatible 8086-based personal computer with Omninet, Ethernet, or RS-232 port. AT'T's Olivetti PC will support this product. One floppy drive is required.
Price	From AT&T: to be determined. From Locus: negotiable	

Alan Winston is a freelance writer and full-time programmer/analyst who has been interested in the Unix system for several years.

All About Device Drivers

BY RIK FARROW

What's missing from the Unix kernel when you add a new piece of hardware? The device driver—the part of the kernel that knows about your hardware.

The owner of a Unix micro has discovered a marvelous hard-disk system that includes its own tape-streaming backup. The customer representative explains that the desired hard-disk system is compatible with the present system; however, the Unix operating system will need to be changed before the hard disk and backup system can be integrated. The customer rep explains that this will cost about $5,000 and that the end-user cannot make the changes himself.

This scenario is not unique to Unix systems and is familiar to almost everyone who wants to add anything more complicated than a serial printer to their computer system. Most operating systems can be easily changed to add additional hardware, and the Unix system is no exception. But the Unix kernel itself must be modified. This means that the owner must have access to the necessary Unix source code, and access to the required code depends on the type of license the owner has.

No matter what version of the Unix operating system you use, there are three ways you can be licensed: *binary, configuration,* and *source.* The most common of these is the binary, or object, license. Binary licensees get all the programs and tools that are included with their version of the Unix system. What the licensee does not get is the source and linkable object files that are necessary for changing the Unix kernel. Binary licensees are out of luck if they want to add some new hardware, unless a kernel that has been changed to accept the hardware is available.

Systems that come with configuration licenses include a directory containing the source and object files necessary for changing your system to accept your new hardware. Typically, this directory is called /usr/src/sys. You know that you have the configuration license if you have this directory.

Source licensees are few and far between because source licenses are both expensive and restricted. They cost about $50,000, compared to around $1,000 for a binary license. Source licensees have the source (the uncompiled programs) for not only the kernel, but for *every* program and tool included with the Unix system. You must sign nondisclosure agreements before you can even look at any source programs. Usually, only software houses and universities (who get special price treatment from AT&T) can afford the source licenses.

Some Unix distributors have eliminated the limited binary licenses and sell only configuration licenses. For example, the Unix system for the IBM PC (PC/IX by Interactive Systems) and UniSoft's System V Unix include the files necessary for adding new hardware. If this is actually the beginning of a trend, then most future Unix systems will be reconfigurable.

DEVICE DRIVERS

What's missing from the Unix kernel when you add a new piece of hardware is the device driver, which is the part of the kernel that knows about your hardware. This hardware, such as your hard disk or modem, looks like several memory (or I/O) addresses to the kernel. Device drivers control your devices (hard disks, modems, printers, etc.) by reading from and writing to these memory addresses.

The device driver controls the device by writing to memory addresses. After a command has been written to memory, it is converted into electrical signals that are sent to the device. The device may respond by returning other signals, which are converted into information stored in memory. The device driver needs to know how to interpret this information (and what commands to send) to operate the device properly.

You may be wondering if you've ever seen a device driver. Probably not, but you have certainly experienced the effects of them. You've all worked with the Unix system, entering your lines of commands to the shell. A device driver is activated every time you type a character.

Suppose you've just typed the letter *c*. The byte representing *c* travels through wires into your computer. The letter itself is stored in a memory address, another memory address is changed to show that a character has been received, and a signal is sent to the processor. The device driver is activated by this signal, and it collects the character from the memory address and passes it to the kernel.

The kernel stores the character in a list of input characters, adds it to another list for characters to be echoed back to you, and resumes working on whatever program was interrupted by your keypress. Another part of the device driver is activated every time the system is ready to send you the character, until finally (several milliseconds later) you see the character you typed.

Something very important has happened in our example. Please notice that it's the action of typing a key or the terminal being ready that activates the device driver. The device driver doesn't wait for a character to be typed; rather, it is actuated by typing a letter. I'll talk more about his later.

DEVICE TYPES IN UNIX

The Unix system treats hardware as two totally different types: block and character. Character devices receive and transmit single characters, which, for example, may be part of a burst of characters refreshing your display while using an editor. But the important thing is that the unit of information is a single character. Terminals, printers, and modems are all character devices.

The kernel can't protect you from changes that you introduce to it.

Block devices deal exclusively in block-sized chunks of characters. The block size corresponds to the size of sectors used in disks, such as 512 or 1024 bytes (and occasionally larger). Block devices don't transfer single characters but rather transfer information a block at a time. Thus, to change a single character in a file, the sector containing the character is transferred into memory, the character is changed, and the sector is written back to the disk.

The distinction between the two device classes is simple: Character devices deal with one character at a time, while block devices transfer block-sized groups of characters.

Both block and character device drivers need memory for temporary storage of data. Memory used for data waiting to be used is called a *buffer*. The Unix kernel program provides special functions that manage the buffering of data.

In the above example, I mentioned that the received characters are passed to the kernel by a device driver. The kernel program maintains lists of characters that have been received until they are called for by a user program. The kernel also has lists of characters to be output that have been written by user programs. These character lists are called *clists*.

A different buffering technique is used for block devices. The Unix kernel maintains a collection of block-sized memory areas (known as *block buffers*), which are used by the device drivers during block reading and writing. Thus, several read/write requests from the same disk block may be performed without accessing the disk more than once since the copy in memory can be used. This buffering technique is known as *caching*.

The kernel program's buffering mechanisms save device drivers from having to handle their own buffer management. At first glance these mechanisms may seem complicated, but, since they do much of the work involved in device handling, they make writing device drivers easier. They also provide the unified approach to buffering that is characteristic of the Unix system.

There is a third class of device called a *raw* device, which is actually a special kind of block device. Raw device drivers transfer blocks directly between the disk and a user program, bypassing the kernel's block-buffers. Raw devices are used to speed up disk-intensive system operations like `fsck`, the file system check program.

THE HOOKS INTO DEVICE DRIVERS

In the Unix kernel are two tables known as *switch tables* that list functions available in the various device drivers. One table lists the device driver functions for character devices, while the other table lists the functions for block devices. There is a line (actually a C structure) in these tables for each device driver. Figure 1 simulates such a table for character devices.

The kernel accesses a device driver by referring to the desired function in the line entry for the device. Each line in the table is identified to the kernel by the *major device number*. For example, major device number 2 refers to the line for the modem device driver in Figure 1. Major device numbers start at 0 for both the character and the block device switch tables.

You can see examples of major device numbers by making a long listing of the device directory, `/dev` (see Figure 2). The first letter in the long listing shows whether the device is block (b) or character (c). The major and minor device numbers are listed before the date. In this figure the major device numbers are zero (0). Note that multiple entries in the device directory with the same major device number and type (block or character) use the same device driver.

Immediately following the major device number comes the *minor device number*, which is used by the device driver only. Most often, the minor device number selects one of several similar devices. In Figure 2, for example, the minor device numbers show different terminal ports, `ttya`, `ttyb`, and `ttyc`.

Another reason for the minor device number is to communicate information to the device driver. One bit of a minor device number might be used to signal to the device driver whether a floppy disk is expected to be single- or double-sided, while the other bits might specify which drive to actuate.

FIGURE 1: CHARACTER DEVICE TABLE

ttyopen	ttyclose	ttyread	ttywrite	ttymode	--	0, terminals
clkopen	clkclose	clkread	clkwrite	clkmode	--	1, clock
mdmopen	mdmclose	mdmread	mdmwrite	mdmmode	--	2, modem

function names
major device number
device name

FIGURE 2: LONG DIRECTORY LISTING FOR `/dev/tty` (TERMINAL) DEVICES

crw------- /dev/ttya	2 root	0,	0 Jun	14	1984
crw-rw-rw- /dev/ttyb	1 rik	0,	1 Jun	14	1984
crw-rw-rw- /dev/ttyc	1 becca	0,	2 Jun	12	1984

THE MOST DIFFICULT ASPECT

Now we have mentioned most aspects of device drivers in the Unix system. Perhaps, writing a device driver doesn't look that difficult. But we have yet to mention the most difficult aspect of device drivers, which we'll soon get to.

There are three main reasons why writing device drivers is difficult. First, you are adding untested software to the kernel. The kernel normally protects you from making mistakes that can bring the system crashing down. User programs can't make changes to the kernel that will cause problems.

But the kernel can't protect you from changes that you introduce to it. A serious error in writing the driver or editing the device switch table can keep the kernel from being loaded successfully. This means that you can't even run your modified kernel to see what's wrong.

Second, hardware devices themselves are tricky to handle. The device driver must send the correct commands in the proper sequence at the proper time to the device, or nothing will work. The device driver code must also be able to handle any error condition that might occur while interacting with the device. If either the commands or error conditions are handled incorrectly, you may have a device driver that works part of the time and that might even crash the system.

INTERRUPTS

Third, and most difficult, are interrupts. Recall our discussion of the character device driver example for a terminal. Two parts of the device driver get activated each time you type a key at your terminal: a part that handles character reception and a part that sends (echoes) the character back to the screen.

While the CPU follows program instructions in a step-by-step way, characters are typed asynchronously with a randomness associated with living organisms. Other delays, for example, between the initiation of a disk operation and its completion, or between printing one character and the next, are not coordinated with the orderly operation of the CPU.

The hardware signal that notifies the CPU of real-world events, like key pressing, is called an *interrupt.* As you can see from our example, an interrupt may occur at any time. It is aptly named because the interrupt forces the currently executing program to be interrupted.

I'll try to make things clearer with a simple example. Suppose you are using an editor on your Unix system. Every time the editor program needs a response from you, it requests that the kernel give it the next character that you press. Unless you have already responded by typing ahead, the kernel notes your request and starts some other user's program.

The other user's program is interrupted when a key is pressed. The interrupt signal activates the interrupt handler, which collects the character. The user's program continues until the kernel restarts your editor program and gives it the requested character. In this example, when you pressed a key, your editor program wasn't even running; some other program was.

Interrupt handlers are difficult to write because the program serviced by the interrupt may not even be running at the time the interrupt occurs. The programmer must always keep in mind that, when the handler is called via the interrupt, nothing is known about the program currently running. The interrupt handler must act like a stalking Indian, performing its work unseen and leaving no traces behind.

They must also be swift. Interrupt handlers, particularly terminal device handlers, make up some of the most often used parts of the kernel code. Poorly written interrupt handlers seriously impair system performance. The difference between a good or bad implementation of the Unix operating system is often related to the quality of the device drivers and their interrupt handlers.

As if this wasn't bad enough, interrupt-driven software is more difficult to debug. The random nature of interrupts makes them hard to capture using ordinary debugging techniques. Sometimes, for example, an interrupt may work fine because printing a debugging string will add enough delay to make the interrupt work properly. Of course, as soon as the debugging statement (or debugging trap) is removed, the interrupt handler fails again.

WRITING THE DEVICE DRIVER

Most device drivers are written by companies that transport (or port) the Unix system or that perhaps write their own variation of the Unix system. Porting the Unix system to a new machine requires compilation of the source code for the Unix system into the machine's native (assembly) language and writing any necessary device drivers. The term *porting* itself refers to downloading (serial port to port copying) of the translated Unix system, complete with device drivers, to the target machine.

Once the C compiler has been written for a particular processor, the main task left for the porting company is to write the memory management code and the device drivers. This is why porting companies soon become experts at writing device drivers.

Device drivers are written by system programmers with C language and Unix system experience. In some ways, it's easier to add device drivers to the Unix system than to some other operating systems. The interface (the device switch tables) between the device driver and the rest of the kernel is clearly defined, and the kernel provides functions that do much of the work.

Some changes to a device driver may be made by someone, such as a system administrator, who doesn't know how to write a device driver. For example, Unix systems that have only one hard disk divide the disk into at least two parts: a root and a swap device. This division, called a partition, is defined by the software of device driver, not in disk hardware.

The system administrator can change the partitions of a hard disk by editing a configuration file that is part of the device driver. Then a

Continued on page 117

SOFTWARE TOOLS USERS GROUP

BY DEBORAH K. SCHERRER

In 1976, when the Unix philosophy had yet to reach beyond a dozen or so universities, Brian Kerninghan and P. J. Plauger, both with Bell Labs backgrounds, produced their now famous little book *Software Tools*. The book's original purpose was to teach good programming practices through the examination of carefully written codes that were clean, efficient, easy to read, and easy to modify. As examples, the authors chose pared-down versions of the Unix utilities they had come to rely on, rewritten in a system-independent manner. The book, along with a magnetic tape containing the programs, became an immediate best seller.

As users began to experiment with and enhance the programs, they realized that the tools offered more than simply a useful set of utility programs. Researchers, primarily at Lawrence Berkeley Laboratory, expanded the original package to include a powerful subroutine library, a Unix-like shell, and many more of the Unix utilities.

By providing all three levels of a user interface—command interpreter, utilities, and programming library—the tools could offer the functionality of the Unix system in conjunction with virtually any operating system. The enhanced package was widely distributed and implemented on a diverse assortment of machines, ranging from Crays to micros.

The result has been the availability of Unix system capabilities on a large number of non-Unix system machines and, perhaps more importantly, a consistent user interface across many different systems. This consistent interface is the key not only to being able to move programs from system to system without change, but also to move users between systems without costly retraining. This emphasis on making people portable, as well as their code, differentiates the Software Tools movement from other vendor or operating system-dependent projects.

Aside from its consistency with the Unix interface, the tools package became widely popular for two major reasons. First, portability—the package could be implemented on virtually any operating system. This portability was achieved by using a programming language available on nearly all machines and by isolating system independencies into "primitive" functions calls that must be implemented separately for each different system.

Second, since the source for most of the tools originated from government-funded laboratories or universities, it was public domain, and users did not have to pay the high price typically charged for products built upon many, many years of effort.

The need for co-ordination and co-operation among implementors and users of the tools led to the formation of the Software Tools User Group (STUG), originating at Lawrence Berkeley Laboratory and initially funded by the Department of Energy.

THE GROUP'S FUNCTIONS

Since its inception in 1978, the group has become an international body performing the following functions: (1) establishing and publishing standards for the primitives and utilities; (2) collecting and evaluating new utilities, enhancements, and variants; (3) holding semi-annual meetings in conjunction with the USENIX Unix users group; (4) publishing a newsletter and software catalogue; (5) collecting and distributing information on current developments to avoid duplication of effort; (6) distributing tapes containing collections of utilities from different sides, both those standardized upon and those available for experimentation and evaluation.

The Software Tools package represents a unique compromise between the desire for user control and the need for vendor support. The Software Tools Group collects and distributes new utilities and enhancements early in their development phase. This allows users to experiment with new ideas, incorporating valuable ones into their own projects and rejecting those that prove unportable or undesirable.

Utilities and enhancements that receive popular support are eventually incorporated into the STUG standards. Thus, the standards are always based on ideas tested and proven by the user community, rather than on newly designed products or untested concepts. And the users generally have access to the (high-level) source code, allowing for tailoring to individual needs.

On the other hand, though, much of the tools source is public domain, the primitives (that is, machine-dependent portions of a library, which form the backbone of the system interface) are often vendor developed, licensed, and supported. To handle issues of portability and implementability of different systems, STUG supports an implementors group consisting of representatives from industry and institutions with an interest in the package's future. This group carefully evaluates changes and extensions before vendors incorporate them into the supported primitive sets.

STUG conferences are the primary forum for exchanging new ideas and co-ordinating development of new projects. Because the meetings are held in conjunction with the USENIX conferences, advances in the Unix community can quickly be examined and evaluated for possible inclusion in the tools community. The Unix system community, in turn, benefits from the discussions on portability and user interface methodology addressed by tools researchers.

As with the Unix system , the original utilities were primarily oriented toward program development. However, the tools approach is applicable to many software projects. Among those of current interest to STUG researchers are networking, electronic mail systems between heterogeneous systems, experimental shells and high-level user interfaces, data management, word processing, and data analysis packages.

The widespread availability of the Software Tools package has given many users their first introduction to the Unix system environment. The compatibility of the package with the local operating system encourages experimentation and learning of the new "world view" without the traditional difficulties in introducing and moving to a completely new system.

Many early tool users became enamored of the Unix environment and have subsequently moved to it. For those whose needs require that they remain with non-Unix systems, the tools can still provide the elegance and functionality of the Unix environment while also providing a consistent interface across systems from micros to mainframes. □

While a research computer scientist at Lawrence Berkeley Laboratory, Deborah K. Scherrer was one of the original researchers to experiment with the software tools concept. She founded the Software Tools User Group, serving as its coordinator for four years, and is a contributing editor to UNIX/WORLD Magazine. Ms. Scherrer is currently a computer scientist for Mt. Xinu and is president of Carousel MicroTools, Inc.

For Software Tools Users Group membership and service information, write:
Software Tools Users Group
1259 El Camino Real
P.O. Box 242
Menlo Park, CA 94025

MARKETPLACE
CAREER OPPORTUNITIES

Readers who have computer merchandise or services to sell, buy, or trade and who want to be included as a classified advertiser in UNIX/WORLD Magazine should write to: Tech Valley Publishing, 444 Castro St., Suite 1220, Mountain View, CA 94041, or call 415/964-0900. The rate for the classified section is $12 per line. The deadline for ads is the first of the month prior to publication. Please include complete name and address in every ad.

Cucumber Books—UNIX and C Language books. Whether you need one of fifty, we stock all titles. Cucumber Bookshop is located at 5611 Kraft Drive, Rockville, MD 20852, (301) 881-2722. Mail and phone orders.

Dyalog APL—Today's most productive application development tool for commercial and educational users. Implemented in C, it gives you the power of UNIX directly from within APL itself. It is a new generation APL plus UNIX, NC. Available for VAX, HP-9000, Gould, Zilog, Fortune, Xenix, and Unisoft. Call or write to: Notation, 10 Jackson St., Los Gatos, CA 95030. (408) 354-3274.

WMZ Consulting Group—System Consultants for total implementation of information systems. WMZ provides design, implementation, training, and ongoing support. We have experts in UNIX on staff and we are not committed to any software or hardware. We sell solutions not iron. Call (415) 945-2070, or write WMZ Consulting Group, 1483 Ashwood Drive, Martinez, CA 94553.

SENIOR SYSTEMS PROGRAMMER —DEPT. OF PSYCHOLOGY AT NYU. Position involves software maintenance and development on several PDP-11 and VAX systems. Desired skills include thorough knowledge of UNIX, operating systems internal, realtime laboratory programming and training users. Work involves installing, maintaining and developing scientific systems software including realtime control of experiments. Dr. Michael Tandy, 6 Washington Pl., Rm. 961, New York, NY 10003.

THE LATEST FROM THE COMPUTER INDUSTRY

DEC'S VENIX FOR PROFESSIONAL SERIES

Digital Equipment Corp. (DEC) has announced a new Unix system-based operating system for its Professional Series of personal PDP-11 computers. Called Pro/Venix, it was specifically developed for Digital Equipment by VenturCom, Inc., of Cambridge, Mass., and will be marketed, distributed, and supported by Digital through its Classified Software Program.

Priced at $800, Pro/Venix is the newest addition to Digital's Unix system offerings. Digital now markets the Unix system on all of its processors, from VAX to the Professional 350. Pro/Venix, like Digital's Ultrix offering, is a multitasking and multi-user operating system based on Version 7 AT&T Unix and includes the more popular UC Berkeley Unix system enhancements. Pro/Venix is distributed with an AT&T Technologies System V license to accommodate upgrades to newer Unix system releases from AT&T.

Digital is targeting Pro/Venix, in combination with its Professional 350, as a Unix workstation for schools, laboratories, government agencies, and technical and commercial original equipment manufacturers (OEMs). Pro/Venix allows the user to run both a foreground and multiple background tasks simultaneously. Any I/O or process such as a program compilation or disk backup can be run in the background while the user performs other tasks in the foreground.

Pro/Venix takes advantage of the Professional's PDP-11 processor and memory management hardware in a multi-user environment. It also supports both the communications and printer RS232 ports, to which any serial device, such as a modem, printer, or terminal, can be attached. Standard licensing allows for two users; special licensing arrangements are required for attaching more users.

A complete software development environment is included with Pro/Venix, and standard languages include C, FORTRAN-77, Pascal (ISO standard), BASIC, and Assembler. The standard Unix system utilities, libraries, and system calls are included. Some extensions to Version 7 Unix include shared data segments, semaphores, and code mapping.

Pro/Venix requires that the Professional 350, which has 512 Kbytes of memory and floating point adapter standard, be configured with a 10-Mbyte Winchester hard disk. It also supports the extended bit-map option, color and monochrome monitors, VT100 and VT52 terminals, LA50, LA100, and LQP02 printers, and the DF02 and DF03 modems.

For more information, contact Digital Equipment Corp., Maynard, MA 01754, 617/480-4821.

Please circle Reader Service Number 100.

NCR ANNOUNCES NEW TOWER SOFTWARE

NCR Corp. has announced WordMarc, a multi-user word-processing system for the NCR Tower 1632. Developed by Marc Software International, of Palo Alto, Calif., WordMarc is an office-grade word-processing system developed for the Unix operating system. Terminal and printer independent, it features full-screen display, a spelling checker, interactive spelling correction, and the ability to horizontally scroll to 163 characters.

The system also has 13 levels of scientific notation and mathematical character and Greek character support for scientific applications. In addition, WordMarc offers the following: the ability to interface with up to eight languages at one time; single-key functions from terminals; no document size limit; changeable menus and message formats; search and replace; dictionary and glossary creation; and automatic letter and address merge.

For more information, contact NCR Corp., Dayton, OH 45479, 212/599-6966.

Please circle Reader Service Number 101.

MORROW'S 'TRICEP' SUPERMICRO

Morrow, Inc., a maker of small business systems, has introduced a multi-user supermicro that for the first time brings the cost per user in the Unix system environment under $2,500. Morrow's Tricep supports four to eight users running the Unix System V operating system on the 16/32-bit M68000 microprocessor. Prices start at under $9,000 retail, and under $5,500 in OEM quantities. In addition, optional slave processor boards based on the 80188 CPU and running the MS/DOS operating system were to be added in September, broadening Tricep's application base.

Tricep hardware includes the following: an M68000 CPU running at up to 10MHz, with an on-board M68451 memory-management unit; 512 Kbytes of main memory (expandable to 2 Mbytes); an I/O controller with 4 RS232 serial ports (expandable to 8 ports); a Centronics-compatible parallel printer port; hard and floppy direct-memory access controllers; and 80188-based slave processors with 128 or 512 Kbytes of on-board dual-port RAM. All boards plug into the 14-slot IEEE 696 (S-100) bus backplane.

Mass storage includes one to four 16- or 32-Mbyte 5 1/4-inch Winchester disk drives (for maximum 128-Mbyte storage); an optional 5-Mbyte removable hard-disk drive; and one to four optional 1.3-Mbyte 8-inch floppies.

Company officials said Tricep's Unix System V port, sub-licensed from Unisoft Systems of Berkeley, is fully compatible with Bell Labs' System V and that Unisoft, which has done more than 30 System V ports optimized for the 68000 chip, has added such enhancements as record-locking and IEEE floating-point capability. An optimizing C compiler comes with the system; other available languages include BASIC, COBOL, FORTRAN-77, Ada, and Pascal.

For more information, contact Morrow, Inc., 600 McCormick St., San Leandro, CA 94577, 415/430-1970.

Please circle Reader Service Number 102.

GOULD'S 32-BIT VIRTUAL MEMORY PROCESSOR

Gould, Inc., has announced the PowerNode 9000, reputedly the industry's fastest 32-bit virtual-memory processor. Topping off Gould's line of compatible Unix system-based distributed systems, the PN9000 completes the Power-Series family strategy that was started last year at the National Computer Conference. The PN9000 computers offer up to 10 times the performance of a DEC VAX 11/780, Gould said.

The unit derives its processing power from four architectural features: emitter coupled logic (ECL) 10K chip technology, large cache memory, four-state instruction pipeline, and a large virtual memory.

The 10K in the PN9000 operates on a 75-nanosecond cycle. Combined with a four-stage instruction pipeline, this allows it to complete an instruction every 75 nanoseconds after the pipeline has been primed. While one instruction is being fetched, a second is being de-coded, a third executed, and the results of a fourth instruction are stored away.

The cache in the basic PN9000 processor includes 32 Kbytes of high-speed memory, the largest cache in any superminicomputer on the market today. Gould also offers an additional 32-Kbyte cache memory module as an option, for a total of 64 Kbytes. Because of this large cache memory and a very efficient cache controller, the PN9000 has demonstrated cache hit rates in the high 90 percent range.

The PN9000 is available in three basic models: the PN9005, PN9050, and PN9080. The PN9005 is an entry-level system and includes the CPU, 32 Kbytes of cache memory, and 4 Mbytes of main memory.

The PN9080, the most powerful of the three, includes a second processor called an internal processing unit (IPU). Operating in parallel with the CPU, the IPU provides a performance boost of approximately 80 percent without a proportional increase in price.

Since the CPU and IPU have their own cache memories, a fully configured PN9080 can have a total of 128 Kbytes of cache memory. This is evenly distributed between the CPU and IPU.

The PN9050 provides the same performance as the PN9005 except that it includes an extra chassis to accommodate an optional field-installable IPU.

Optional multiply accelerators, which can be attached to the CPU and IPU, provide for fast execution of single- and double-precision floating-point multiply operations.

The PN9000 was designed to operate in either a stand-alone mode or in a distributed local-area network. In a stand-alone mode, the PN9000 supports multiprogramming and multiple users through local or remote terminals. In a distributed system, the PN9000 acts as a processing resource that can be shared among multiple users accessing it through smaller PowerSeries processors.

UTX/32, the PN9000's operating system, is based on the Berkeley 4.2 version of the Unix operating system, with selected Bell System V features. It supports virtual memory management and tightly coupled dual processors (that is, CPU/IPU combination). UTX/32, in conjunction with the networking software and database manager, provides all of the PN9000 resources to the other network workstations.

Transmission Control Protocol/Internet Protocol (TCP/IP), a Defense Department standard, can be used by the PN9000 to provide interaction within a Unix system-based network. Multiple interactive links between various types of Unix system-based computers can use a variety of physical communications links including Ethernet, RS232, and high-speed parallel ports.

The PN9000 processors are packaged in cabinets 55 inches high, 68.5 inches wide, and 32.25 inches deep. The basic systems include the CPU, space for optional I/O controllers, and up to 16 Mbytes of main memory.

Prices for the PowerNode 9000 systems start at $245,000 for the PN9005. The PN9050 is priced at $285,000, and the PN9080 at $385,000.

For more information, contact Gould, Inc., Computer Systems Division, 6901 W. Sunrise Blvd., P.O. Box 9148, Ft. Lauderdale, FL 33810-9148, 305/587-2900.

Please circle Reader Service Number 103.

ZILOG'S SYSTEM 8000 APPLICATION PACKAGES

Zilog, Inc., is now offering seven new application software tools that allow users to choose from four major application groups—word processing, database management, spreadsheet, and graphics—to run on its multi-user Unix-based System 8000 supermicro computer. All were previously available only from separate sources.

Zilog is offering two spreadsheet packages: Supercomp-Twenty, developed by Access Technology, Inc., of South Natick, Mass.; and Multiplan, created by Microsoft Corp., of Bellevue, Wash. They are priced at $1,100 and $750, respectively.

The two word-processing programs now available directly from Zilog are Q-One from Quadratron Systems, Inc., headquartered in Encino, Calif.; and XED, developed by Computer Concepts Ltd., Chatsworth, Calif. They are priced at $695 and $1,050, respectively.

Unify, from Unify Corp., Sacramento, Calif., and Informix database management system (DBMS), developed by Relational Database Systems, Inc., Palo Alto, Calif., provide a full range of DBMS capabilities to System 8000 users. They are priced at $1,995 and $1,695, respectively.

Now running on the System 8000 is DI-3000 from Precision Visuals, Inc., Boulder, Colo., one of the most popular graphics packages available on larger minicomputer and mainframe systems. It consists of a set of sub-routine libraries that are called from either FORTRAN or C programs to create graphics on a variety of display terminals. It is priced at $2,500.

For more information, contact Zilog, Inc., 1315 Dell Ave., Campbell, CA 95008, 408/370-8000.

Please circle Reader Service Number 104.

CINCOM'S FAMILY OF UNIX-BASED SOFTWARE

The Ventures Division of Cincom Systems, Inc., has released a new family of software systems known as CS/XTEND, designed specifically for Unix system-based operating environments. The fully integrated product family includes both application development aids and end-user productivity tools—CS/DBX, a multi-user, shared database executive; CS/TMX, a terminal management executive; CS/RMX, an end-user information retrieval management executive; CS/XPRESS, an interactive application builder for end-users; and CS/XPORT, a system for managing databases that are distributed across multiple computers.

The CS/DBX database system is fully compatible with Cincom's Total database management system. Mainframe or minicomputer applications that have been designed for use with Total can be easily ported to Unix systems with CS/DBX. Similarly, applications designed for use with Hewlett-Packard's Image DBMS can be easily adapted for use with CS/DBX.

CS/TMX offers facilities for generating, coding, and maintaining CRT screen displays independently of user-written programs. A run-time monitor handles all of a program's terminal input and output activities. CS/TMX dramatically reduces the time application programmers spend writing and maintaining interactive applications using `termcap` or other terminal handling facilities.

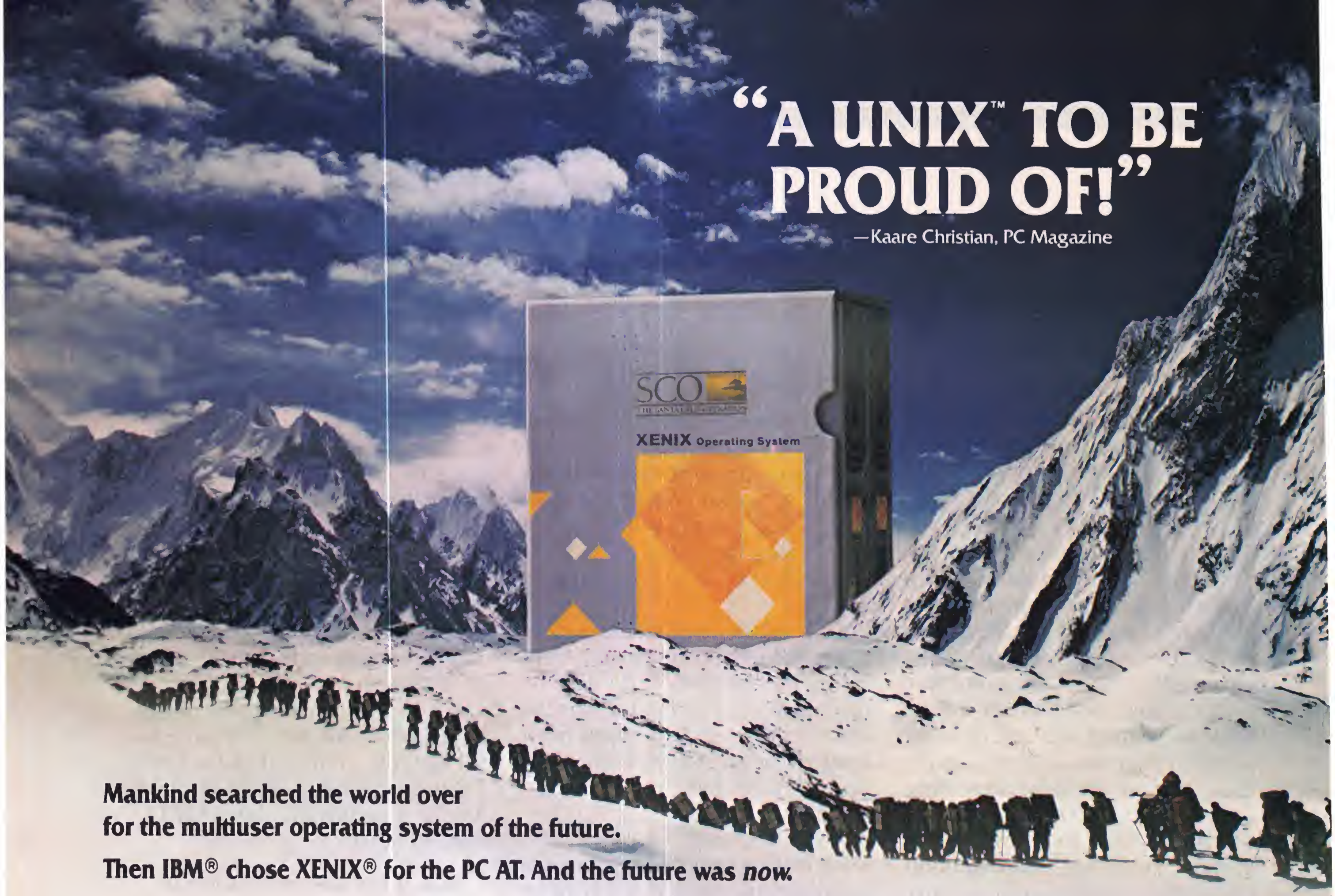
CS/RMX provides relational access to the database for inquiries and report writing. Also implemented on mainframes and minicomputers, the system allows end-users to retrieve information quickly and easily by using English-like commands. Data retrieved from the database can be displayed on a terminal, directed to a printed report, or both.

CS/XPRESS is an interactive, interpretive system designed to make it easy for end-users to create complete application systems. Using this system, a user can interactively design a complete screen format and describe each field's data edit characteristics.

CS/XPORT provides control software to pass information from one computer to another. A program running on one computer (which could be a PC, Unix-based system, or another mini or mainframe) can directly access data maintained in a CS/DBX or Total database on another computer. Users can also programmatically pass data from machine to machine, a message at a time.

Depending on the configuration of CS/XTEND and the number of users supported, the price ranges from $675 to $66,150. For more information, contact Cincom's Ventures Division, 2300 Montana Ave., Cincinnati, OH 45211, 800/543-3010.

Please circle Reader Service Number 105.



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
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!

NOW SHOWING AT SCO QUALIFIED DEALERS!

SCO
THE SANTA CRUZ OPERATION

RELEASED FOR IBM PC XT, AT&T PC, OTHER COMPATIBLES AND APPLE® LISA® EXCLUSIVELY THROUGH THE SANTA CRUZ OPERATION, 500 CHESTNUT STREET, P.O. BOX 1900, SANTA CRUZ, CA 95061. IMMEDIATE SEATING FOR OEMs.

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

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

Please circle Ad No. 37 on inquiry card.

IBM'S NEW PERSONAL COMPUTER

IBM has introduced a new top-of-the-line personal computer with significant performance improvements and an optional Xenix operating system from Microsoft Corp.

The IBM PC/AT uses advanced technology, including the high-speed Intel 80286 microprocessor, a high-capacity diskette drive, and expanded fixed-disk drive options.

The computer delivers almost five times the user memory and more than twice the information storage capacity previously available on IBM PCs. And, in most cases, system performance is two to three times faster.

The system is compatible with most existing IBM PC hardware and software. With prices starting at $3,995, the PC/AT is available immediately through IBM's Product Centers and branch sales offices nationwide.

The PC/AT can be used as a powerful single-user system or shared by up to three users. Because of the unit's performance and storage capabilities, IBM said the new system is ideal for use with the new IBM PC Network, which enables customers to link up to 72 IBM PCs to share programs, information, and peripheral devices.

The basic configuration is the $3,995 model, which includes 256 Kbytes of user memory and a new 1.2-Mbyte high-capacity diskette drive. The second model, priced at $5,795, includes 512 Kbytes of user memory, the 1.2-Mbyte character diskette drive, and a new 20-Mbyte fixed-disk drive.

Both models have an 84-key keyboard and eight expansion slots for additional features, devices, and memory. Both can also be expanded with options to more than 3 Mbytes of user memory and up to 41.2 Mbytes of disk storage. A 12-month warranty is standard.

IBM also announced version 3.0 of its widely used Personal Computer Disk Operating System (PC/DOS). This version has added functions to support the new Personal Computer AT hardware and is compatible with all IBM PCs.

The IBM PC Xenix operating system, also announced for the PC/AT, enables two additional terminals to share the computer's processing power. In either a multi- or single-user environment, IBM PC Xenix allows more than one task to be handled at a time. For example, a program can be compiled while another is being edited. The Xenix operating system is priced at $395 and will be available in the first quarter of next year.

Please circle Reader Service Number 106.

IBM'S PC LAN

In addition, IBM has introduced low-cost products that customers can easily tailor and install themselves to link up to 72 IBM PCs for sharing information, programs, messages, printers, and storage devices.

The IBM PC Network connects IBM Personal Computers, PC/XTs, Portable Personal Computers, and PC/ATs through standard cable television coaxial cable and connection hardware. Specifications released by IBM make it possible for up to 1,000 PCs to be linked in a professionally designed and installed broadband network.

The IBM PC Network includes the IBM PC Network Adapter, an advanced-design feature card that fits in an expansion slot of an IBM Personal Computer system unit and that comes with a 9-foot cable; the IBM PC Network Translator Unit, which provides the required broadband frequency translation for an IBM PC Network and directly connects up to eight PCs; and IBM PC Network Cabling Components, including a base expander and short-, medium-, and long-distance cabling kits for connecting up to 72 PCs within a 1,000-foot radius of the network translator unit.

IBM said in a statement of direction that it will connect the IBM PC Network and a planned industrial network to its future token ring local-area network.

The IBM PC Network is scheduled to be available in October from selected authorized IBM PC Dealers and through IBM's product centers and branch sales offices.

Information is transferred within the network at 2 million bits —about 250,000 characters—per second.

The IBM PC Network Technical Reference Manual, now available, includes the technical data for designing a custom network of up to 1,000 PCs within a three-mile radius of a translator unit.

In connection with the IBM PC Network, the company announced the IBM PC Network Program for sharing data, messages, printers, and other devices on the network. Publications provided with the program contain information needed to develop application programs for the network.

IBM also announced the IBM PC Network SNA 3270 Emulation Program, which can be installed on PCs in the network to give them communications access to a large IBM host computer system via the company's Systems Network Architecture communications protocol.

An enhanced version of the IBM PC Disk Operating System—DOS 3.1 —also was announced to support the IBM PC Network.

The IBM PC Network Program and DOS 3.1 are scheduled to be available in the first quarter of 1985. The IBM PC Network SNA 3270 Emulation Program is scheduled to be available in the first half of 1985.

IBM Product Center prices are $695 for the IBM PC Network Adapter and $595 for the IBM PC Network Translator Unit. Each network requires a translator unit, and each PC in the network needs a network adapter. Prices for cabling components vary, depending on length and function.

Please circle Reader Service Number 107.

VISUAL TECHNOLOGY'S 286-BASED MULTI-USER SYSTEM

Visual Technology, Inc., has unveiled a low-cost multi-user system based on the Intel 286 microprocessor. Housed in a slim floor-standing enclosure, the new Visual 2000 supports as many as 16 terminals under the Xenix operating system, Microsoft Corp.'s popular enhanced version of the Unix system.

In a base-level configuration, the Visual 2000 includes 512 Kbytes of RAM, a 19-Mbyte Winchester disk, a 5 1/4-inch floppy, six RS232 terminal ports, and a parallel printer port, for an end-user list price of $8,990. With a 40-Mbyte Winchester, the system is priced at $10,990. Substantial discounts are available to OEMs, system houses, and distributors in quantity purchases.

Options available for the system include a 1/4-inch streaming tape cartridge drive for disk backup, as well as expansion capabilities that enable the system to grow to support up to 16 terminals with up to 4 Mbytes of RAM and as many as four 5 1/4-inch Winchesters.

Company officials said the system is intended for a variety of applications, ranging from turnkey systems in small businesses to department-level computing in Fortune 1,000 organizations to software development and scientific computing. Customer shipments were slated to begin this fall.

For more information, contact Visual Technology, 540 Main St., Tewksbury, MA 01876, 617/851-5000.

Please circle Reader Service Number 108.

NCR INTRODUCES TOWER-BASED OFFICE APPLICATION

NCR has introduced a software package for the office that has extensive "windowing" capabilities and that can handle up to 14 workstations with quick processing and response time. NCR Officeware, as the package is called, is built around NCR's popular Unix system-based Tower 1632 small-business computer and utilizes industry-standard PCs to increase processing capacity, reduce response time, and provide personal computing applications.

NCR Officeware includes seven integrated modules designed to ease user transitions from one document or application to another. These modules include: Script, a commercial-quality word-processing system; Plan, a financial planning tool; Forms, a data management facility that makes entry and retrieval of data possible through user-defined forms; Graph, which converts spreadsheet information into bar, line, and scatter graphs; Terminal, which allows workstations to operate as host application devices; Desk, an assortment of applications for the individual user; and Network, a set of applications providing the user with interoffice information resources.

Multiple documents, graphs, and spreadsheets can be shown on the screens as windows that can be moved, changed, or filed, or they can be easily incorporated into the word-processing module. Multiple documents can also be worked on at the same time.

For more information, contact NCR Corp., Dayton, OH 45479, 513/445-2075.

Please circle Reader Service Number 110.

CROMEMCO'S UNIX-BASED SUPERMICRO SYSTEMS

Cromemco, Inc., of Mountain View, Calif., has introduced two new series of Unix system-based supermicrocomputers that include the ability to expand memory up to 16 Mbytes of error-correcting RAM. The systems, the System 100 and System 300 series, use the Unix System V operating system with Berkeley enhancements.

Among the extended capabilities and available options are an SMD disk controller that allows the addition of up to two 600-megabyte SMD drives, a nine-track cartridge or reel-to-reel tape backup, a floating-point processor, a high-resolution graphics interface, and a video digitizer.

The operating system, included in the systems at shipment, is the industry-standard Unix System V, with Berkeley enhancements, including the programs `cshell`, `termcap`, `uucp`, and `vi`. Cromemco has also included its proprietary Unix-like operating system, Cromix, on the system.

Cromemco is supporting the new systems with a variety of other software products, including a C compiler and B-NET networking software for remote file and electronic mail transfers.

List prices range from $9,995 to $19,995. For more information, contact Cromemco, Inc., 280 Bernardo Ave., P.O. Box 7400, Mountain View, CA 94039, 415/964-7400.

Please circle Reader Service Number 112.

NEWS FROM AT&T

JOB CUTS, VARS, AND THE OLYMPICS

AT&T, which has reported lower than expected earnings, said that by early next year it will eliminate 11,000 management and blue-collar jobs as the latest in a series of cost-cutting moves. Since the divestiture earlier this year of the local operating companies, the telecommunications giant has been struggling to adapt to the new competitive telecommunications environment and to control its high costs.

AT&T has already frozen the wages of 114,000 management employees, announced plans to close four plants, and cut 5,000 workers from its payroll through a variety of programs. These work force reduction plans include attrition, resignations, and special early retirement programs.

AT&T said the cutbacks would be centered on the 53,000-member equipment maintenance and service operations at its Chicago-based AT&T Technologies. This is the unregulated subsidiary that is serving as the focal point of AT&T's efforts to penetrate the highly competitive computer market. Further staff reductions may be announced by the end of the year.

AT&T VISITS THE OLYMPICS

AT&T's Electronic Messaging System (EMS) transmitted more than 1½ million messages as well as nearly a million additional requests for event results and information during the 1984 Summer Olympic games. The system was part of a telecommunications and information management network that AT&T designed and installed for the games.

AT&T's Olympic Network, comprised of computers, terminals, printers, telephones, and several hundred miles of fiber optic cable, served as two systems in one: EMS and an advanced voice communications system.

Operating with 14 AT&T 3B20S computers, 1,700 Teletype 5410 interactive terminals, and 300 high-speed printers, EMS provided daily Olympic event results, comprehensive progress reports, profiles of athlete performance prior to and during the games, electronic mail, bulletin boards, forms for reserving hotel rooms and ordering meals, and other similar capabilities.

According to an AT&T spokesman, the system exceeded projections. Although officials had expected a maximum of 150,000 transactions a day, the system actually handled up to 196,000 transactions a day. Also, nearly twice the anticipated 50,000 people used it, the company said.

AT&T's voice system handled nearly four million phone calls via 7,000 special phones and more than 20 Dimension private branch exchange PBX switching systems.

A custom-designed fiber optics digital lightwave system, manufactured by AT&T for Pacific Bell, linked all elements of AT&T's Olympic Network, including radio paging and international Telex services.

AT&T SIGNS THREE VARS

Three new value-added resellers (VAR's) including a 160-store computer retail chain, have been signed up for the 3B line of computers. The first two are Systemhouse Ltd. and Computer Integration Associates, Inc. Both have agreed to pick up AT&T's entire line of 3B supermicros and superminis, including the 3B2, 3B5, and 3B20, as well as the firm's 3BNet local-area network, the PC Interface, and the AT&T Personal Computer.

Systemhouse Chairman Rod Bryden said the AT&T 3B line of computers would be used for the firm's Unix-based office automation system. "We are also in the midst of two multimillion-dollar projects employing Unix system development," he added.

Systemhouse Ltd. is an information systems consulting and software development organization with corporate offices in Arlington, Va., and Ottawa, Canada. The company, with 750 employees in 15 offices across the U.S. and Canada, has revenues of approximately $50 million.

Computer Integration Associates, based in Lakewood, N.J., is a supplier of turnkey computer systems throughout the U.S. The 3B products will be used primarily in business accounting systems, medical office management systems, manufacturing, and other business applications. The firm's president, Joshua Beren, said the company has also agreed to adapt certain commercially available software to operate on the AT&T computers.

The third VAR is MicroAge Computer Stores, Inc., based in Tempe, Ariz. A company spokesperson said it has signed a letter of intent with AT&T Information Systems whereby franchises approved by AT&T will be authorized and trained to sell and service AT&T's 3B2 line of business computer systems.

MicroAge said the 3B2 will be available in its stores in the fourth quarter of this year and that marketing support systems are presently being finalized. This agreement is the third between the two companies. MicroAge already markets AT&T's Merlin Communications System and the AT&T Model 6300 Personal Computer. □

PERFORMANCE VERSUS FUNCTIONALITY

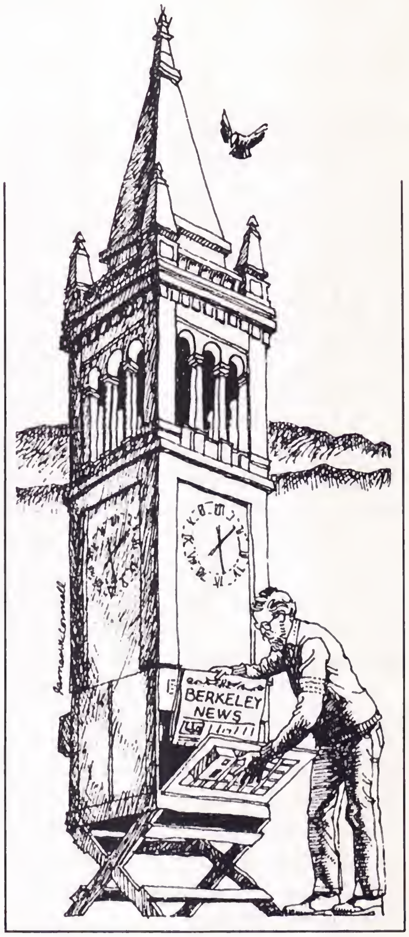
BY ALAN TOBEY

In the normal life of software development, performance is usually the descendant of functionality and not its sibling. That is, adding new functions to existing software often comes initially at the cost of speed and general system performance, and only after the additions are fully integrated and understood can performance issues be addressed. The 4.1 release of "Berkeley Unix" software, for example, was mostly a performance tuneup of 4.0, which was the original virtual-memory VAX Unix release.

4.2BSD, as released by UC Berkeley, represented a largely successful effort to extend the frontiers of the Unix system into networking and other new areas. As could be expected, these extensions had their price in terms of initial system performance.

Users have been fortunate that the 4.2 release included a faster file system that speeded many system functions. The overall result has been that 4.2BSD in general has been performing about as well for most applications as did 4.1, the earlier release. Some relatively CPU-intensive functions run more slowly, while other more disk-intensive functions are significantly faster.

Now that 4.2BSD has been initially absorbed by its user community, the cycle of work is swinging toward improving performance. Efforts are underway at UC Berkeley and at a number of academic and commercial sites to make the system as efficient as possible.



CSRG IS ON THE CASE

The Computer Systems Research Group (CSRG) at UC Berkeley, through a group headed by Mike Karels and Kirk McKusick, has now identified several areas where performance enhancements should prove beneficial. In general, the group's performance tracing reveals that 4.2 spends 10 to 20 percent more time in the Unix kernel than does 4.1, significantly increasing the overhead that causes a general time-sharing system to slow down. This means that relatively less user time and relatively more system time is being utilized in the new release.

The CSRG has also identified a number of performance problems associated with specific user programs:

(1) The mail system under 4.2 seems to be a major culprit. On multi-user machines, *sendmail* can be the heaviest CPU user on the system. Its interaction with the *syslog* logging process also generates increased load.

(2) The network servers that 4.2 added to the system are expensive. Since the system uses one daemon process for each network service request, the spawning of network processes on a multi-user machine adds significantly to the load.

(3) A significant recent finding is that the time required for the use of pipes is longer under 4.2 since 4.2 implements the pipe facility on top of the interprocess communication facilities that require more complex data structures, such as sockets.

(4) Path name translations have become more expensive in 4.2BSD because of the addition of symbolic links between files and a more complex directory format.

(5) Terminal multiplexor handling routines in 4.2BSD, although they included some improvements, were not optimized for the actual use levels they normally experience and thus contributed to increased overhead.

(6) Process table management —largely a kernel scheduling operation—uses linear search techniques that the CSRGhas concluded are no longer sufficient under 4.2.

(7) In a typical UC environment, processing network protocol and input functions seem to add 3 to 5 percent to the cost of overall system functions.

(8) The larger memory require-

Continued on page 113

Experience The Worlds of UNIX* *at*

UniForum™

The International Conference of UNIX Users

January 21-25, 1985 • INFOMART • Dallas, Texas

UniForum '85 is your passport to the fascinating "Worlds of UNIX." You'll examine the growing impact of UNIX in Office Systems, Personal Computers, Technical (Engineering/Programming), and Market Trends at UniForum '85, the largest UNIX event ever held.

More than 200 major vendors, in 850 booths, will display and demonstrate all that's new in UNIX products and applications.

An extensive conference and tutorial program will expand your UNIX database. This program, organized by /usr/group, will include 15 all-day tutorials on user-specific aspects of UNIX...48 in-depth and informative conference sessions... four nationally-known plenary speakers.

In addition, a number of introductory courses in UNIX will be presented throughout the event.

UniForum '85 will be your total UNIX experience. Whether you're just getting started...or are a seasoned UNIX veteran ...UniForum '85 is *the* UNIX event of the year.

For Complete Information, Call: 1-800-323-5155
(In Illinois, 1-312-299-3131).
Or Mail This Coupon Today!

YES! I need to know more about UniForum '85. Please send complete details.

Name

Title

Company

Address

City

State

Zip

Telephone (Area Code)

Electronic Mail Address (one only):

Mail To: UniForum
Suite 205, 2400 E. Devon Ave.
Des Plaines, IL 60018 USA

Sponsored by

/usr/group
The International Network of UNIX Users

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

Please circle Ad No. 41 on inquiry card.

ADVERTISERS INDEX

If you are interested in products or services advertised in UNIX/WORLD, our advertisers are pleased to send you additional information.

Simply fill out the reader service cards and circle the numbers of advertisers you are interested in. Then mail the card and expect prompt, pertinent information.

Reader Service No.	Advertiser	Page No.
1	**Accucom:** /USR/TROFF—Lets your PC set 15½ feet of typesetting per minute.	41
3	**Aim Technology:** Evaluating Unix/Xenix systems? Contact Aim Technology for a free benchmark manual.	98
	AT&T Technologies: Unix System V training—comprehensive, hands-on courses from the creators of the Unix operating system.	25
5	**Cincom Systems:** CS/EXTEND—a new fully integrated family of development and decision support components—makes it easy to create and implement even the most complex applications.	6
6	**CMI Corporation:** SERIX—The high-performance CMI 42 version of AT&T's Unix System V operating system with Berkeley 4.1 enhancements ported to the IBM Series/1 minicomputer.	42
7	**Computer Technology Group/Telemedia Inc.:** Unix and C Training—the training solution that fits the complexities of your Unix-based system.	12
8	**Contel Codata Systems:** The Codata 3300—a powerful 16-bit, 68000-based multibus system that can accommodate up to 10 users.	62
9	**Convergent Technologies Inc.:** The MiniFrame- superminicomputer power and Unix versatility at PC prices. Also part of the Convergent family is the MegaFrame, a revolutionary superminicomputer system.	10,11
10	**Creare:** UltraLink—Strong enough for eight users and crystal clear.	15
11	**Dama Telecommunications Corporation:** DamaNet—Now a national office-to-office digital network. Fully integrated high quality voice and wideband data, on Demand Assigned Multiple Access (DAMA) system.	57
12	**Data Language Corporation:** PROGRESS—The high-performance database system and fourth-generation applications development language for multiuser microcomputers.	43
13	**Dual Systems Corporation:** The 83/80—A new high-performance micro with full Unix system capabilities of Version 7 or System III that can handle up to 12 users.	72,73
14	**Elxsi:** Performance and expansion potential: The 64-bit multiprocessor system 6400.	30
34	**Excelan:** Change a 12-month design cycle to 12 days with EXOS, the technology that provides complete LAN capability.	1
15	**Gould Inc.:** Manufacturers of the Power-Series 20 Computers—Unix-based systems that span the performance range.	35
16	**Gould Inc.:** The PowerNode 9000 and 6000—firebreathers that are scorching old performance standards.	16
17	**Gould Software Division:** New Unix-based software tools that bring projects in on time, within budget.	Back Cover
18	**Handle Corporation:** The first office automation system for Unix—selected for distribution.	77
19	**Heurikon Corporation:** OEMs will like the one-size Minibox, a Unix system multiuser, single-board workstation.	67
	IBM-ISG/DP Professional Software: IBM's PC/IX = Unix + advanced full-screen editor, + PC-DOS file transfer, + 8087 coprocessor support, + queuing system, + IBM support.	36,37
4,21	**Integrated Business Computers:** Ensign—The Unix-based system with up to 32-user capacity, up to 8 Mb memory, and over 1,000 Mb disk storage.	70
22	**ITOM International Co.:** The Supermicro Newsletter—written by the folks who invented the term.	96
23	**Lachman Associates:** The software systems consultant firm that tailors to your business needs.	78
24	**Logical MicroComputer Company:** The MegaMicro—a 32-bit multiuser, virtual-memory microcomputer. Ideal for the OEM.	54
25	**Logical Software:** Softshell—A convenient interface for Unix users at all levels of expertise.	106
26	**Mark Williams:** Use less memory space and run your C programs faster with the MWC86 Compiler.	76
27	**MCBA:** Shrink-to-fit software. Integrated manufacturing, distribution, and accounting packages.	58
28	**Mt. Xinu:** 4.2BSD—Better than just "standard," now fully supported.	26
29	**NCR Corporation:** For OEMs, the Tower 1632. Built for system builders.	52,53
30	**Norm DeNardi Enterprises:** The California Computer Show—Palo Alto Hyatt Hotel, Palo Alto, Calif., December 6, 1984.	90
	Pathway Design: UniPATH—SNA and BISYNC gateways for your Unix-based multi-user system.	4
	Plexus Computers Inc.: Beat our benchmark and receive a Heuer chronograph watch valued at $485.	Inside Back Cover
33	**Prescience:** HIEROGLYPH—The universal report production for the Unix System: Integrated text data and graphics software.	14
	Pyramid Technology: Pyramid Technology 90x—The supermini born to run Unix. No reader service number; please contact by mail or phone.	46
35	**Rapitech Systems:** FORTRIX-C. The program that converts FORTRAN code to C code at approximately 600 lines per minute. Please fill out the coupon and mail for faster response.	87
36	**Relational Database Systems Inc.:** Informix and File-it—the database software family for Unix and MS-DOS.	19,20, 21
37	**SCO, The Santa Cruz Operation:** Presenting XENIX Now! The multiuser, multitasking LPC blockbuster, and introducing C-MERGE as the MS-DOS development environment.	104
54	**Scientific Placement:** National registry of candidates and jobs in the Unix system field.	99
38	**Southwind:** Tactician and Grafsman—fast, versatile, and powerful tools.	82
39	**Syntactics:** CrystalWriter—Complete word processing for the Unix System.	27
40	**Tech Valley Publishing:** Unix shell and vi posters. Please clip and send the provided coupon.	119
41	**UniForum/usr/group:** Presenting the second annual UniForum Conference and Trade Show, Dallas, Texas, January 21–25, 1985.	111
42	**Unify Corporation:** UNIFY—The fast, powerful and flexible relational database management system.	Inside Front Cover
53	**UniSoft Systems:** UniPlus+—Unix adapted for the M68000-based microcomputer.	24
43	**Unisource Software Corporation:** Venix/86 for the IBM PC/XT—offering a 30-day free trial.	8
45	**UNIX/WORLD Magazine:** UNIX/WORLD—Your complete guide to the Unix frontier. Subscribe now and receive a 25 percent discount off the newsstand price.	113
46	**User Training Corporation:** Great learning exhilarates and saves you money—the performance courseware for Unix and C.	108
47	**UX Software:** UX-Basic—Provides application, portability and productivity.	22
48	**Visual Technology Inc.:** VISUAL 2000. The power of a mini, the cost of a micro, and the clout of Unix.	60,61
49	**VLS Inc.:** Cdb C language source-level debugger gets all the bugs before they eat all your productivity.	101
50	**VLS Inc.:** VAX-based systems, custom configuration, sales integration, installation, maintenance.	103
51	**VLS Inc.:** 4.1- and 4.2-based systems—sales enhancements, custom configuration, and support.	105
52	**Yates Ventures:** The Yates Perspective—a monthly newsletter on the software market. Subscribe now and receive a reduced rate.	2

ments of 4.2 generate more work for the paging (virtual memory management) system.

CHANGES

In response to these and other problems, changes have already been made to the research 4.2 systems at Berkeley and at some commercial sites. Both kernel work and C library improvements are being worked on and tested, and notable improvements are already being gained in overall system performance and in clearing the most notorious roadblocks.

The obvious question is when a suitable set of improvements will be available to the 4.2BSD community at large. While CSRG is making no promises—especially about timing—it now seems likely that a "speedup kit" will be made available sometime next year. Rather than waiting longer for a subsequent formal BSD release (none has ever been promised), CSRG will make performance-only software available through a mechanism not yet determined. Also, some of the additional performance work done at other locations should make its way into the world via Usenix contributed-software releases and technical papers at conferences.

The signs, then, are that 4.2BSD is continuing on a normal course of software development—function first, then performance. If, as at least one guru has estimated, the "obvious" performance enhancements can add at least 25 percent in speed and horsepower, 4.2 will be even better positioned for its role as the carrier of technically oriented Unix in the commercial world.

Author's note: I strongly urge that readers and 4.2 users *not contact* the Computer Systems Research Group to ask them about specific improvements or about when the "speedup kit" might be available. They are a *research* group, not a software-support hotline. The results of their work will be released generally when ready; talking with individual user sites is something they simply don't have the time to do. □

Alan Tobey is marketing director of Mt. Zinu, a Berkeley-based company specializing in Unix systems software.

COMMAND ALIASES WITH THE BOURNE SHELL

If you're using the Bourne shell supplied with Bell System V Release 2, you can define *command aliases* analogous to the built-in `alias` feature of the Berkeley C shell. You would use the new Bourne shell *function definition* feature to achieve this capability.

For instance, Figure 1A shows the definition for an "alias" named `dir()` that uses the `-C` option of `ls` to display the directory listing in multiple columns with entries sorted down the columns. The catenated `F` option causes `ls` to supply a slash (/) after each directory name and to append an asterisk to the ordinary filename if it's executable. When you actually use the `dir()` function, simply enter `dir` followed optionally by any number of arguments. Figure 1B shows the equivalent C shell `alias` definition for `dir`.

More than one command can be defined as a single function. Let's say you wish to define a new function, `chdir()`, which changes directories and then displays the pathname of the new current directory. The approach to use is shown in Figure 2A. The individual commands within the braces for the function definition are separated from one another using NEWLINEs instead of semicolons. The ">" is the secondary prompt (value of the PS2 variable). This prompt indicates that the shell expects further input. For comparison, the equivalent `alias` definition for the C shell is shown in Figure 2B.

Contributed by Jon Lundell
President of Opus Systems
Los Altos, Calif.

DEFINING THE `dir()` COMMAND ALIAS:

FIGURE 1A: USING THE SYSTEM 5.2 BOURNE SHELL:

```
$ dir()   ls -CF $@;
$ []
```

FIGURE 1B: USING THE BERKELEY C SHELL:

```
% alias   dir  ls -CF
% []
```

DEFINING THE `chdir()` COMMAND ALIAS:

FIGURE 2A: USING THE SYSTEM 5.2 BOURNE SHELL:

```
$ chdir()   {
> cd $@
> pwd
> }
$ []
```

FIGURE 2B: USING THE BERKELEY C SHELL:

```
% alias chdir 'cd \!*; pwd'
% []
```

DEAR SYSTEM DOCTOR

Question:

How does the shell run a command in the background?

Answer:

Before answering this question, let's first look at how the shell executes a command in the foreground.

An executing program is called a *process,* so one can refer to an executing shell as a *shell process* and to the executing command program as a *command process.* To appreciate the mechanism for executing a command you need to be aware of several simple operations used to manage processes. These operations are implemented by Unix system calls.

A process can duplicate itself. The resulting two processes are indistinguishable, except the original process is identified as the *parent process* and the copy as the *child process.* The duplication is achieved by a `fork` system call.

A process can replace itself. A process reads a program from disk into memory and overwrites itself by using an `exec` system call.

A process can put itself to sleep. For our purposes the parent process can suspend its own execution and wait until the child process has completed execution. This suspension is made possible by the `wait` system call.

A process can terminate itself. When a process is ready to stop, it invokes an `exit` system call.

Now you can understand how the shell executes a program in the foreground. The shell process `forks`, creating an identical copy (a procedure sometimes called "spawning a child"). The resulting child shell executes an `exec` system call to read the program that was specified on the command line into memory and start it executing. This new command process overwrites or replaces the child shell process. Then the command process is the child of the parent shell.

Meanwhile, the original parent shell has suspended its execution by executing a `wait`. The "sleeping" parent waits patiently for the child process it spawned earlier to finish. When the child command process calls `exit`, the parent shell is signaled that its child has "died." The parent shell process "wakes up" and resumes execution. The parent shell shows that it is ready to accept another request by typing a prompt character. These steps are diagrammed in Figure 3A.

A command is run in background simply by *not* having the parent shell wait for the child shell process to exit; that is, the parent shell continues execution after forking off the child shell process. The parent shell will display an identifying number for the child process it spawned and then issue another prompt to show that it is ready to attend to your next request. We illustrate these steps in Figure 3B.

HOW THE SHELL RUNS A COMMAND PROGRAM

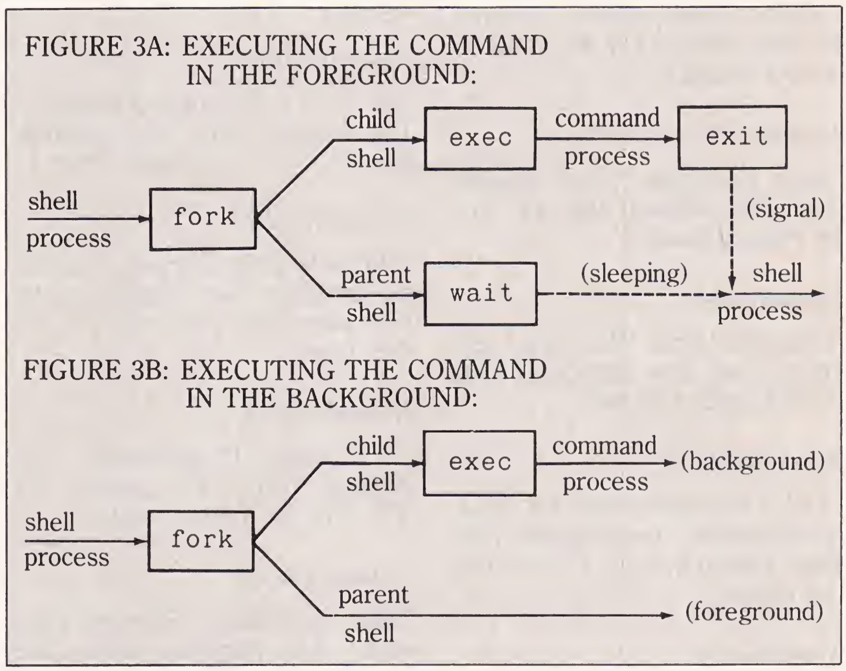


FIGURE 3A: EXECUTING THE COMMAND IN THE FOREGROUND:

FIGURE 3B: EXECUTING THE COMMAND IN THE BACKGROUND:

As an aside, a common misconception is that there are three layers of Unix system software—the kernel program, the shell program, and the command program. Actually there are only two layers—the kernel and either a shell or a command program. The command process replaces the shell that spawned it so the command operates at the "same level" as the shell.

Wizard's Grabbag is a regular feature of UNIX/WORLD. Submit your questions, answers, shell scripts, C programs, or tips and techniques that ease the programmer's burden to the *Wizard's Grabbag*, UNIX/WORLD, 444 Castro St. Suite 1220, Mountain View, CA 94041. Authors of published entries receive $25 for questions, $50 for shell scripts, `awk` scripts, `sed` scripts, `lex`, `yacc`, and C programs, or tips.

Guidelines for reader contributions: Write your shell scripts, C programs, and other code so it is portable across different versions of the Unix system. If possible, it should run without change on Bell Version 7, Systems III and V, and Berkeley 4.x. Thus, you should use "universal" Unix utilities, such as `who am i` (all systems) in lieu of `whoami` (Berkeley only), and the Bourne shell, if possible, when coding shell scripts.

However, C shell scripts are also welcome because most of our readers now have access to this popular command interpreter. Use the standard I/O library when writing C code. Also use the `lint` syntax checker to eliminate nonportable constructions and compile the code with a portable C compiler such as `pcc` to help ensure portability. Hardware dependencies, such as terminal control sequences, should be eliminated or at least minimized and isolated to one code region or to a separate module. Keep your example as short as possible, say under 100 lines of code. □

NOVEMBER

November 5-9

"System Administration," Batavia, Ill. (See calendar key #2, UNIQ, for contact details.)

November 6

"Unix Overview," Los Angeles, Calif. (See calendar key #4, CTG, for contact details.)

November 6-9

"Hands-On Unix Workshop," San Diego, Calif. (See calendar key #5, ICS, for contact details.)

November 7-9

"Unix Fundamentals for Non-programmers," Los Angeles, Calif. (See calendar key #4, CTG, for contact details.)

November 7-9

"Unix," San Diego, Calif. (See calendar key #7, CAPE, for contact details.)

November 12-14

"Unix Fundamentals for Programmers," Los Angeles, Calif. (See calendar key #4, CTG, for contact details.)

November 12-16

"C Language," Batavia, Ill. (See calendar key #2, UNIQ, for contact details.)

November 13-16

"Computer Graphics," Los Angeles, Calif. (See calendar key #5, ICS, for contact details.)

November 13-16

"Programming in C: A Hands-On Workshop," San Diego, Calif. (See calendar key #5, ICS, for contact details.)

November 14-16

"Unix," Detroit, Mich. (See calendar key #7, CAPE, for contact details.)

November 15-16

"Shell as a Command Language," Los Angeles, Calif. (See calendar key #4, CTG, for contact details.)

November 26-27

"Micro/Personal Computer Operating Systems," Los Angeles, Calif. (See calendar key #3, SIA, for contact details.)

November 26-30

"C Language Programming," Los Angeles, Calif. (See calendar key #4, CTG, for contact details.)

November 27-30

"How to Manage Software Projects," Los Angeles, Calif. (See calendar key #5, ICS, for contact details.)

November 27 -30

"Hands-On Unix Workshop," Washington, D.C. (See calendar key #5, ICS, for contact details.)

November 28-30

"Unix," Scottsdale, Ariz. (See calendar key #7, CAPE, for contact details.)

DECEMBER

December 3-6

"Data Communications Concepts," Campbell, Calif. (See calendar key #6, Zilog, for contact details.)

December 3-4

"Shell Programming," Los Angeles, Calif. (See calendar key #4, CTG, for contact details.)

December 4-7

"Hands-On Unix Workshop," Palo Alto, Calif. (See calendar key #5, ICS, for contact details.)

December 4-7

"Programming in C: A Hands-On Workshop," Washington, D.C. (See calendar key #5, ICS, for contact details.)

December 5-7

"Using Advanced Unix Commands," Los Angeles, Calif. (See calendar key #4, CTG, for contact details.)

December 10-11

"Micro/Personal Computer Operating Systems," Boston, Mass. (See calendar key #3, SIA, for contact details.)

December 10-12

"Unix," Somerset, N.J. (See calendar key #7, CAPE, for contact details.)

December 10-14

"Unix Internals," Los Angeles, Calif. (See calendar key #4, CTG, for contact details.)

December 11-14

"Programming in C: A Hands-On Workshop," Palo Alto, Calif. (See calendar key #5, ICS, for contact details.)

December 11-14

"How to Manage Software Projects," Washington, D.C. (See calendar key #5, ICS, for contact details.)

December 11-14

DEXPO West 84, the 6th National DEC-Compatible Industry Exposition, Anaheim, Calif. (See calendar key #1, EI, for contact details.)

December 17-19

"Unix for Non-Technical Users," Campbell, Calif. (See calendar key #6, Zilog, for contact details.)

December 18-20

"Unix Administration," Los Angeles, Calif. (See calendar key #4, CTG, for contact details.)

December 19-21

"Unix," Tampa, Fla. (See calendar key #7, CAPE, for contact details.)

TRAINING CALENDAR KEY

(1) Expoconsul International (EI)
Contact: Steve Barth
55 Princeton-Hightstown Rd.
Princeton Junction, NJ 08550
609/799-1661

(2) UNIQ Digital Technologies
28 S. Water St.
Batavia, IL 60510
312/879-1566

(3) Software Institute of America (SIA)
8 Windsor St.
Andover, MA 01810
617/470-3880

(4) Computer Technology Group (CTG)
Telemedia
Contact: Sandra Smith
310 S. Michigan Ave.
Chicago, IL 60604
312/987-4082

(5) Integrated Computer Systems (ICS)
Contact: Ruth Dordick
P.O. Box 45405
Los Angeles, CA 90045
800/421-8166 or 213/417-8888

(6) Zilog
Contact: Kay Ferrell
1315 Dell Ave.
Campbell, CA 95008
408/370-8000

(7) Center for Advanced Professional Education (CAPE)
Contact: Herb Stern
1820 E. Garry St., Suite 110
Santa Ana, CA 92705
714/261-0140

Continued from page 95

new kernel is generated that includes the modified driver. Changes such as this also require making new file systems on the affected hard disk and rebuilding previously existing file systems from backups.

Other things a system administrator can do to change the Unix kernel don't have anything to do with device drivers. For example, the maximum number of processes can be changed to reflect the addition of more memory to the system. And the number of buffer blocks in the cache can be increased, improving system performance. This type of change to the Unix system is known as *system tuning.*

WILL THINGS GET EASIER?

Adding new hardware to a Unix system is not a trivial task because there currently just isn't an easy way to write and link in the device driver. Device drivers are easier to add to some other operating systems, such as PC-DOS.

In PC-DOS a user adds a device driver by executing the `device` command. Then, during system boot, DOS loads the file containing the device driver into the appropriate part of memory as an extension of itself. The reason this works out so nicely—PC-DOS was written to accommodate changes of this sort and the architecture of Intel's 8088.

Adding device drivers to the Unix system is difficult for two reasons. The device table interface (and the system calls) are currently the way the Unix standard is defined. Changing the device table interface to make adding device drivers easier means changing the Unix standard. This isn't to say that it isn't possible, but rather that a unified approach is called for.

The second reason is that PC-DOS is tied to one processor, the Intel 8088. The Unix system avoids hardware dependence by not taking advantage of the special functions provided by a specific processor. By choosing a set of processor functions that are generally available, Unix systems are not caught in a market controlled by a single chip manufacturer. This lack of dependency on a single vendor has been a cornerstone in the growth of the Unix system.

Will Unix systems become easier to add device drivers to? This question can only be answered by AT&T and the standards committees. If they come to a consensus about what is Unix, perhaps they can also provide a simple standard way of adding device drivers. The viability of the Unix system over the next 10 years may well depend on their solution to this problem. □

A San Francisco resident, Rik Farrow is a freelance technical writer who has written installation and maintenance guides for micro-based Unix systems as well as first drafts of several chapters that have appeared in books on the Unix system.

GLOSSARY OF TERMS

Block device: The device type where the smallest unit of data transfer is a block-size (512 or some multiple thereof).

Buffer: An area in memory used for temporary storage of data.

Cache: A collection of buffers used to increase system performance.

Character device: The device type where the unit of information is the individual character.

Device(s): Hardware, in particular, peripheral devices such as hard disks, modems, terminals, and printers.

Device driver: The part of the kernel program that controls and communicates with a particular device.

Continued from page 120

acerbated by the expectations of co-workers, who *know* that you have the technical ability to respond quickly and who may demand that you do so, even when your responses will necessarily be incompletely developed.

This sort of problem can become very serious when your messages are being delivered to large groups of people. In such situations, massive arguments (known colloquially as *flaming*) can result. Within minutes of receiving these controversial messages, many people zap off responses to them before giving any real thought to a fully reasoned reply.

Considerable time, resources, and money (not to mention human energy) can be wasted in such situations, which are highly vulnerable to high-speed positive feedback growth. The end result of such flaming sessions is usually a great deal of wasted disk space, and not one hell of a lot else.

These situations can also become more complex through the actions of the very filtering effect I mentioned above. Inflections and much emotion are easily lost in most e-mail, and it is only with experience that the e-mail user can be expected to learn the writing "tricks" that can help avoid misunderstandings.

ARGUMENTS AND MISUNDERSTANDINGS

Massive network e-mail arguments have been based solely on a misinterpreted satiric statement. In a commercial environment, with both careers and large sums of money potentially on the line, such misunderstandings could be devastating. It is crucial that e-mail users be aware of the potential for "lost" meanings and misinterpretations that are somewhat inherent in the medium.

E-mail users must also remember that any time someone will be reading your messages, he or she may be judging your professional abilities and integrity based on those messages. In other words, it is very important to remember that *there are real people out there.*

Sitting alone before a terminal, you may not easily visualize the possibly many people who will be reading your words within hours, or even minutes, of your sending them. Always keep in mind that, for all practical purposes, you are getting up on a stage every time you send e-mail messages. Just because you can't hear or see the audience doesn't mean they aren't there!

It's also worthwhile to remember that, in many cases, everything you say in e-mail messages can be permanently archived. Whether online or buried on backup tapes, your words may be preserved for a very long time indeed, and they can be restored to visibility long after you have forgotten the message in question. In some cases, such archivals can be highly useful; in others, they might prove embarrassing if a message were sent hastily.

Two more points to consider, and two that will become increasingly important as e-mail is used for more commercial applications, are the *security* and *authenticity* of messages. Most e-mail sent these days is not encrypted and could be read by numerous individuals other than the actual addressees. Commercial e-mail carriers make little mention of preventing snooping by other subscribers in the event of system security problems (including poor system design).

Perhaps more important in the long run is the issue of snooping by the carriers themselves. I'm not suggesting that such activities are now going on, but it seems pretty clear that any time a third party is moving non-encrypted messages through its computers, the possibility of unauthorized access to your messages is very real.

Most carriers will presumably be making backup tapes for system integrity purposes, so your messages might be preserved by the carrier itself. Obviously, an unscrupulous carrier (or even a single unscrupulous employee of the carrier) could gain access to large amounts of potentially valuable information by scanning subscriber messages for keywords of interest.

THE DANGER OF FORGERIES

E-mail users should also question whether messages *are actually* from their purported sender. If forged messages are not detected, they could be far more serious than spied-upon messages. The relative silence from the commercial e-mail services regarding the security of messages from both external and *internal* snooping or forgery is somewhat ominous.

In fact, the commercial e-mail services seem to make a fairly major effort to distance themselves from any responsibility for anything that might happen to your messages while in transit. This can't be viewed as anything but a discouraging sign.

To the extent that potential e-mail users view these security issues as real, they have three primary choices. First of all, they can avoid the commercial services entirely or use them only for less important messages. Second, they could make full use of the services but simply encrypt their messages first. (Message encryption is a complex subject that I intend to discuss in a future column, but a number of fairly simple, yet reasonable, techniques are available.)

Third, users might choose to design their own private e-mail networks, thus avoiding the problems with third-party carriers entirely. This last alternative might be very attractive to larger organizations, although it could present some logistical, though by no means insurmountable, problems to smaller firms.

E-mail contains the promise of major positive attributes as well as the risk of some potentially serious pitfalls—aspects of both "pleasure" and "pain." Some of the more troublesome elements of e-mail (such as the "flaming" effect) may be mastered simply through good sense. Other aspects (such as security) may require a combination of both technical and perhaps legislative remedies.

Without a doubt, though, e-mail's positive elements—those that guarantee its continued rapid growth—cannot be ignored. With reasonable effort we can get the most out of e-mail without subjecting ourselves to its more negative aspects any more than necessary.

As with any new product, the old adage "Let the buyer beware" is applicable. We must avoid being complacent about the more controversial aspects of e-mail, but neither should we become paranoid with concern. Because so many factors can be both positive and negative in this area, it behooves us to do our best to steer toward the positive to the greatest extent possible. □

Lauren
UUCP:
{decvax, ihnp4, seismo, clyde, allegra}!vortex!lauren

Lauren Weinstein is a computer/telecommunications consultant based in Los Angeles. He has been involved in a wide array of projects ranging from the mundane to the bizarre. He has particular expertise in the fields of computer networking, the Unix system, microcomputer technology, and telecommunications systems.

PLEASURE, PAIN, AND ELECTRONIC MAIL

BY LAUREN WEINSTEIN

Without a doubt, one of the most exciting areas in computing technology is electronic mail. While the monetary volume of such services in the commercial marketplace is still comparatively modest, it appears that commercial electronic mail will represent a billion-dollar (or perhaps multibillion-dollar) aspect of our industry by the decade's end.

Electronic mail (which I'll refer to simply as "e-mail") is fascinating from a purely technical standpoint, as are comparisons of existing systems and discussions of possible future e-mail developments. However, these aren't the areas I'd like to discuss in this month's column.

Instead, let's take a look at some of the less technical aspects of e-mail, in particular at some of the more psychological and sociological factors relating to e-mail and its use. These aspects, which frequently aren't mentioned in the advertisements for these systems, can provide many of the true benefits of e-mail. They can also lead to potential problems, misunderstandings, and other negative results as well, particularly to the newer users of e-mail services.

My particular views of e-mail have been formed over the last decade, during which time much of my own work has been performed via local- or network-based e-mail facilities of one sort or another. Much of this work has related to the development and implementation of e-mail systems themselves. Thus, e-mail has been a significant factor in my life for quite a long period, and I've seen both the benefits and pitfalls of such systems.

Some of the benefits are immediately obvious. Under normal circumstances, e-mail is *fast,* at least in comparison with conventional paper mail. Communication with coworkers or associates on the other side of the planet can progress with much the same speed as communication with individuals closer to home.

TELEPHONE TAG

The old problem of "telephone tag" (the endless returning of phone calls when you never seem able to reach the other person when he or she is available) can also be largely obviated by e-mail. Your correspondents are free to read your messages at their convenience, and they can easily file them for future reference.

Communications by e-mail also have the fascinating property of being highly filtered in terms of extraneous information content. That is, you are dealing almost totally with your correspondents' intellects and not with such trivialities as their speech characteristics, age, skin color, or even physical handicaps.

Writing ability is the primary skill that is showcased in e-mail, and even poor spellers can use various software aids to help with problems in this area. Is the person from whom you've received an e-mail message a corporate official or a high-school student? With e-mail, both may start off pretty much equal in terms of their visible "appearance" on these systems. In a very real way, e-mail has the potential of allowing large-scale, prejudice-free communications of a sort never imagined in the past.

Related to the concept of large-scale communications is the potential ability to reach very large numbers of people at (conceivably) a relatively low cost per person. On various research-and-development-oriented networks, it is already possible to send a single message that tens of thousands of people will see within a day or so of its being sent, if not sooner.

The creation of distributed, special-interest mailing lists on these nets has fostered the development of the best ways to send and process messages to large numbers of people. While we can assume that the cost of such communication will be higher in the commercial environment, the positive possibilities of such facilities are still exciting to contemplate.

HALLMARKS

It's pretty safe to say that the positive factors discussed above are the sorts of things you'd expect commercial e-mail suppliers to proudly promote as the hallmarks of their systems. Oddly enough, however, these same factors can also be *disadvantages*—sometimes disastrously serious ones—particularly to the inexperienced e-mail user.

For example, the speed of communication by e-mail can cause problems as well as provide benefits. Sometimes, having the *ability* to send messages quickly results in people using the ability unwisely, sometimes sending responses to queries, asking questions, arguing, or offering opinions without really taking the time to consider carefully what is being said.

This problem can be ex-

Continued on page 118

Gould . . . Innovation and Quality in Software

CELEBRATE!

New UNIX™-based software tools bring projects in on time, within budget.

Get your hands on our UNIX tool kit. Take out the Gould Software Division 4.2 UNIX with System V enhancements. You'll know it. It has a fast file system, automated system configuration, interactive debugging, and a Source Code Control System that compresses files for more efficient disk storage.

Then grab The Newcastle Connection, an ingenious tool that lets you create a distributed UNIX file system in a LAN by uniting heterogeneous UNIX versions and hardware environments.

Next, try the Unicellerator™, a high efficiency, user transparent tool that dramatically increases productivity under UNIX by offloading CPU intensive jobs from the host. The Unicellerator is available to *all* users of a system simultaneously.

The kit also contains a Cross Development System for developing and interactively debugging C language programs under UNIX that will run on Motorola® and Intel® microprocessors. The heart of the system is a C language compiler that generates compact, efficient code that reduces code space by 20–40% and increases performance by an equal amount. For programmers who want to develop Macintosh™ applications in C under UNIX, our Macintosh Cross Development System will do it four times faster than Apple®'s Lisa™-based environment.

Our UNIX tools are available individually or as a complete, well-integrated set. The sooner you get your hands on them, the faster you'll finish your development projects. And that's worth a toast any time. Contact Gould Software Division, 1101 East University Avenue, Urbana, Illinois, 61801. (800) 952-8888 or (217) 384-8500.

Please circle Ad No. 17 on inquiry card.

GOULD
Electronics