UNIX™ REVIEW

THE PUBLICATION FOR THE UNIX™ COMMUNITY

April 1985 $3.95

RECEIVED
11 JUL 1985
COMPUTER SCIENCE

MODULAR.
INTEGRATED.
NOW.

Handle™ presents a modular software series for office automation.

Available now for several UNIX™ and XENIX™ based systems, the **Handle** family of office automation products may be purchased as individual modules, in combinations, or as a fully integrated system. The **Handle** product series is a powerful set of software tools designed for today's multi-user office environment. **Handle** integrated software can easily share data between modules in the series as well as with outside applications and databases. **Handle's** open architecture is built on a database foundation and is available to outside developers.

Handle Office Automation Software Series product modules include:

- **Handle Writer/Spell™.** Word processing with automatic spelling correction and verification.
- **Handle Calc™.** Virtual spreadsheet with up to 32,000 rows and columns.
- **Handle Graphics™.** Advanced business graphics.
- **Handle List™.** List processing, management, and forms.

AT&T 3B2/300 & 3B5

CONVERGENT TECHNOLOGIES MINIFRAME

DURANGO POPPY II

To be announced?

HANDLE TECHNOLOGIES, INC.

Circle No. 30 on Inquiry Card

850 NORTH LAKE BOULEVARD / PO BOX 1913 / TAHOE CITY, CALIFORNIA 95730 / 916-581-5227

TM— HANDLE, HANDLE WRITER/SPELL, HANDLE CALC, HANDLE GRAPH & HANDLE LIST ARE TRADEMARKS OF HANDLE TECHNOLOGIES, INC. UNIX IS A TRADEMARK OF AT&T BELL LABORATORIES. XENIX IS A TRADEMARK OF MICROSOFT CORPORATION. MINIFRAME IS A TRADEMARK OF CONVERGENT TECHNOLOGIES. POPPY IS A TRADEMARK OF DURANGO.

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

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.

Circle No. 1 on Inquiry Card

UNIX™ REVIEW

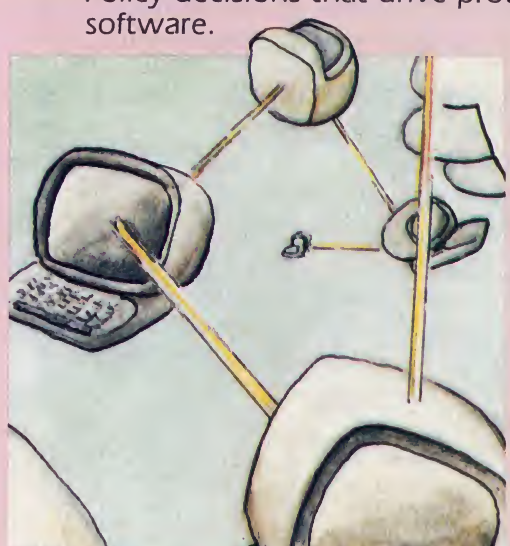
THE PUBLICATION FOR THE UNIX COMMUNITY

Volume 3,
Number 4
April 1985

DEPARTMENTS:

Cover art by Hyon Kim

FEATURES:



THE NETWORKING MAZE

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

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

UNIX REVIEW is sent free of charge to qualified individuals in the U.S. The publisher reserves the right to determine qualification. For those not qualified, there is an annual (12 issues) subscription charge of $28 in the U.S., $35 in Canada and $48 in all other countries (surface mail). Correspondence regarding editorial (press releases, product announcements) and circulation (subscriptions, fulfillment, change of address) should be sent to 500 Howard Street, San Francisco, CA 94105. Telephone 415/397-1881. Correspondence regarding dealer sales should be sent to 901 South 3rd Street, Renton, WA 98055. Telephone 206/271-9605.

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

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

PUBLISHER:
Pamela J. McKee
ASSOCIATE PUBLISHERS:
Ken Roberts, Scott Robin
EDITORIAL DIRECTOR:
Stephen J. Schneiderman
EDITOR:
Mark Compton
ASSOCIATE EDITOR:
Mark Hall
EDITORIAL ADVISOR:
Dr. Stephen R. Bourne, Consulting Software Engineer, Digital Equipment Corporation.
EDITORIAL REVIEW BOARD:
Dr. Greg Chesson, Chief Scientist, Silicon Graphics, Inc.
Larry Crume, Director, AT&T UNIX Systems Far East
Ted Dolotta, Senior Vice President of Technology, Interactive Systems Corporation
Gene Dronek, Director of Software, Aim Technology
Ian Johnstone, Project Manager, Operating Software, Sequent Computer Systems
Bob Marsh, Chairman, Plexus Computers
John Mashey, Manager, Operating Systems, MIPS Computer Systems
Robert Mitze, Department Head, UNIX Computing System Development, AT&T Bell Labs
Deborah Scherrer, Computer Scientist, Mt. Xinu
Jeff Schriebman, President, UniSoft Systems
Rob Warnock, Consultant
Otis Wilson, Manager, Software Sales and Marketing, AT&T Technology Systems
HARDWARE REVIEW BOARD:
Gene Dronek, Director of Software, Aim Technology
Doug Merritt, Technical Staff, International Technical Seminars, Inc.
Richard Morin, Consultant, Canta Forda Computer Laboratory
Mark G. Sobell, Consultant
SOFTWARE REVIEW BOARD:
Ken Arnold, Consultant, UC Berkeley
Jordan Mattson, Programmer, UC Santa Cruz
Dr. Kirk McKusick, Research Computer Scientist, UC Berkeley
Doug Merritt, Technical Staff, International Technical Seminars, Inc.
Mark G. Sobell, Consultant
CONTRIBUTING EDITOR:
Ned Peirce, Systems Analyst, Bell Laboratories
PRODUCTION DIRECTOR:
Nancy Jorgensen
PRODUCTION STAFF:
Cynthia Grant, Tamara V. Heimarck, Florence O'Brien, Barbara Perry, Denise Wertzler
BUSINESS MANAGER:
Ron King
CIRCULATION DIRECTOR:
Wini D. Ragus
CIRCULATION MANAGER:
Jerry M. Okabe
MARKETING MANAGER:
Donald A. Pazour
OFFICE MANAGER:
Tracey J. McKee
TRAFFIC:
James A. O'Brien, Manager
Tom Burrill, Dan McKee, Corey Nelson
ADVERTISING REPRESENTATIVES:
Jules E. Thompson, Inc.
1290 Howard Avenue, Suite 303
Burlingame, CA 94010
Lucille Dennis 415/348-8222
303/595-9299—Colorado
312/726-6047—Illinois
617/720-1888—Massachusetts
713/731-2605—Texas
Jules E. Thompson, Inc.
2560 Via Tejon
Palos Verdes Estates, CA 90274
Ed Winchell 213/378-8361
212/772-0933—New York

VIEWPOINT

Picking through the maze

The cover and lead story of this issue refer to local area networking by way of a labyrinth metaphor. You, the reader, well may wonder why.

Then again, the connection may be all too clear if you've already had occasion to develop local area networks under UNIX. For those who haven't, the lead article by Dr. Greg Chesson and Mark Hall should be quite illuminating.

As the person responsible for much of the software driving AT&T's Datakit, Dr. Chesson has personally tangled with most of the networking dilemmas presented by UNIX. Drawing on that experience, he discusses how—like the labyrinth—UNIX offers a worthy but surmountable challenge to LAN developers. He also charts some possible routes out of the maze.

There *is* hope, after all. Indeed, reassuring signs are all around us. As Dr. Chesson notes in his piece, "It was inevitable that UNIX and local area networking would merge." Fruits of the merger are already visible—as a quick tour around any UNIX trade show will attest. The changes that are necessary for the process to continue seem inevitable—even if they come at the cost of modifications at the kernel level.

One unavoidable question that arises, though, is: why must the process be so arduous? The answer lies in the origin of UNIX. Developed to perform single-processor work at Bell Labs Research, it was never intended for the plethora of applications it now services. The serious commercial role it plays today is testimony to the system's intrinsic value and the considerable serendipity that marked its growth within the Bell System.

For all that, the time for change has come. UNIX is no longer simply a tool for Bell Labs Research. In fact, it might be described as a "market phenomenon". To remain so, it must grow.

So as to explore what's already been attempted in the UNIX LAN realm and to speculate on what may come in the near offing, UNIX REVIEW asked Dr. Chesson and five other experts to discuss the technologies that make UNIX networking possible. Next month, we'll branch off into an exploration of what can be *done* with those technologies by way of distributed resource sharing. In a still later issue, we'll look at long haul networking and the sociological implications it bears.

But I'm getting ahead of myself. In this issue, Dr. Chesson's lead is followed by an account by Bruce Borden detailing trends in network hardware technology. Borden, one of the cofounders of 3Com, brings considerable experience to the piece, having also developed Excelan's IP/TCP implementation.

Steve Holmgren, president of Communication Machinery Corporation, takes up the software side of the story in a later article. His account describes how the art of policy decisions interacts with the science of protocol message exchange.

A totally different tack is adopted by Jordan Mattson in a study of human networks. Mattson contends that properly designed games can fuel healthy social development. If you guessed that he also offers some thoughts on what "proper design" entails, you're right.

Ned Peirce chips in with this month's interview, an illuminating discussion with Sandy Fraser, the director of the Computing Science Research Center at AT&T Bell Labs. As the person responsible for much of the Datakit data transport technology, Fraser has some very definite opinions about network development. Peirce, a systems analyst specializing in UNIX, has some thoughts of his own.

Acknowledgements would not be complete without a tip of the hat to reviewers Richard Morin of Canta Forda Computer Lab, Carl Smith of UniSoft Systems, and Deborah Scherrer of Mt. Xinu.

The sum total, I think, is a painless introduction to UNIX networking concerns. Be assured that the experiences the writings were distilled from were not so easily gained. ■

Mark Compton

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

THREE COMPLEX NEEDS ONE SIMPLE SOLUTION

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

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

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

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

PERFORMANCE

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

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

PC NETWORKING

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

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

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

VERSATILITY

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

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

RELIABILITY

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

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

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

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

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

*OEM Pricing

CYB...Single Source Solution for Systems Integration

UNITE is a trademark of CYB SYSTEMS, Inc. UNIX is a trademark of AT&T Bell Laboratories. Ethernet is a trademark of Xerox Corporation. DEC and VAX are trademarks of Digital Equipment Corporation. IBM is a registered trademark of International Business Machines Corporation. PHILON is a trademark of Philon, Inc. RM/COBOL is a trademark of Ryan McFarland Corporation.

Circle No. 32 on Inquiry Card

THE MONTHLY REPORT

Yet another UNIX success story

by Mark Hall

UNIX, still basking in the afterglow of IBM's announced mainframe support for System V, received the additional blessing of the Pentagon in late February. It is now officially the preferred operating system of the military.

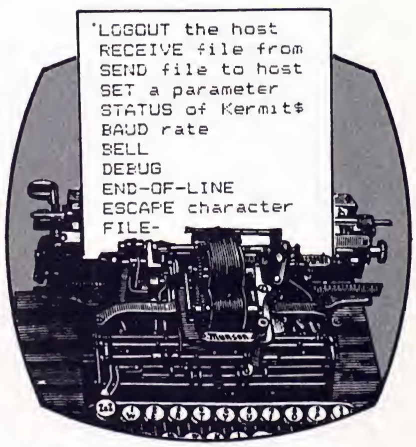
A new ruling, expected to go into effect on January 1, 1986, requires computer manufacturers to provide UNIX as at least an option on all RFPs (requests for proposals) submitted to the Department of Defense. At this writing, it's not known whether the DoD prefers System V or 4.2BSD.

Some observers have speculated that the Pentagon's adamant stance in favor of UNIX may have been part of the reason behind IBM's move to back the operating system.

Meanwhile, the Swedish government has also let it be known that vendors offering to sell it computer systems had best provide UNIX as the operating system—or save themselves the effort.

ALTOS WOOS OEMS WITH 68020 SYSTEM

Phil White, the senior vice president of marketing for Altos Computers of San Jose, CA, has a dream that comes to most marketing executives sooner or later. "In five years," he said, "Altos is going to be a billion dollar company." That's quite an ambitious



statement for a firm that's on track toward $130 million in fiscal 1985.

Part of White's dream hinges on the success of its newest multiuser system, the Altos 3068. Targeted primarily at original equipment manufacturers and operating under UNIX System V with Berkeley enhancements, the 3068 represents "a departure from the traditional direction of our company," White contended.

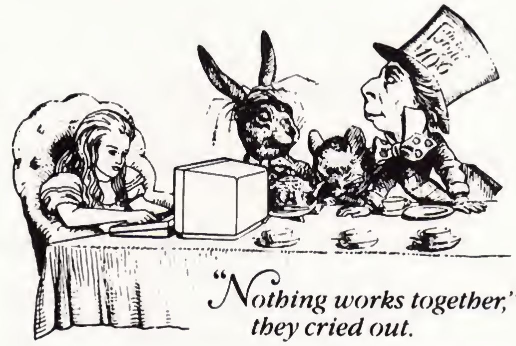
The Altos 586/986 shared-processor, multiuser system, the predecessor to the 3068, uses XENIX and is sold principally through dealers.

What makes the 3068 so appealing to OEMs, White believes, is the system's architecture, which incorporates Motorola's M68020 32-bit microprocessor. "We decided," he said, "that there is a huge opportunity to provide this high-end processor for the OEM market. We think we'll be the first to get the 68020 to market."

Shipments of the new computer are slated to begin in June when Motorola expects to have quantity deliveries of its new chip available for Altos. Sources in the industry say that there are only sample quantities of the high-powered chip available now. Commenting on the microprocessor industry, Jim Farrell, Motorola's manager of technical communication in Austin, TX, observed, "This is a first-come, first-serve business. Altos was among the early customers for the 68020 and they'll get what they need. We're very happy they're among the first."

The M68020 makes the new Altos computer very fast, with a CPU capable of 12.5 MHz in June and upgradable to 16.7 MHz by autumn, when Motorola expects to be ready with a faster version of the chip. The 3068 uses a proprietary 32-megabit internal bus to link users over the eight plug-in boards in the card cage. Users share a disk drive that can store up to 240 MB of unformatted data. The 3068 also includes a 5¼-inch floppy and a 60 MB streaming cassette.

White considers "networking and communications as being key" to the interests of OEMs.



"Nothing works together," they cried out.

"It will now," said Alis.

With Alis,™ everything works together. Text, spreadsheets, drawings, business graphics and database information work together in a single, always editable document.

Alis combines the advantages of integrated PC applications with the information-sharing benefits of communications-based OA systems. It offers the most advanced total office solution ever.

Alis makes it easy for people to work together. It provides integrated electronic mail, calendar and meeting scheduling, and revolutionary Automatic Office Assistants™ to aid management information monitoring and decision making.

Written in "C," Alis is initially available on UNIX* to large OEMs. It's destined to bring sanity to the mad tea party world of office automation.

*UNIX is a trademark of Bell Laboratories. Applix, Alis, and Automatic Office Assistants are trademarks of Applix, Inc.

Alis™

The next-generation office software system from APPLiX

Finally, some answers in Wonderland.

APPLiX, INC., 112 TURNPIKE ROAD, WESTBORO, MASSACHUSETTS 01581 (617) 870-0300

Circle No. 3 on Inquiry Card

With the introduction of the 3068, Altos has also announced some improvements to its local area network, WorkNet. Formerly, WorkNet operated at 800 kilobits per second over twisted pair wire and could be stretched to a distance of 500 feet. The upgrade pumps the data rates up to one megabit per second and extends the distance limitation to 1000 feet. Up to 30 processors can be attached to the new network. Both versions of WorkNet use CSMA/CA (carrier sense multiple access with collision avoidance) as its network access protocol. Peter Kavaler, Altos' manager of networking and communications software, claimed his company has already shipped 600 WorkNets.

Kavaler pointed out that WorkNet allows for "very effective distributed resource sharing" via UNIX. By employing a *super root* function in the directory tree structure, 3068s on a WorkNet can share files by using standard UNIX pathnames to locate files. By inputting the appropriate pathname, the 4.2BSD symbolic links used by Altos let WorkNet cross machine boundries. A "one-time configuration" by the system administrator establishes these connections. "Because so much of UNIX is its file system, you can do much more because the files on multiple processors are linked," Kaveler said.

The 3068 has also made ties to the non-UNIX world with the addition of communication processors. IBM 3270 SDLC and BSC emulation, as well as 3780 and 2780 RJE, is now available. Communications over X.25 networks can be set up through the 3068. The new computer can be configured as a network server for a WorkNet, providing file serving, print serving, and non-UNIX communications capabilities. With the high-speed M68020 microprocessor, White anticipates no network bottlenecks even with a single, centralized server configuration.

Shipments of the new computer are slated to begin in June when Motorola expects to have quantity deliveries of its new chip available for Altos.

One other thing that makes this new system attractive to OEMs is price. The basic system will sell for less than $10,000 in quantitites. The WorkNet costs around $395 per 3068 for the software. Since the system already comes with an RS-422 interface, the network only requires connectors and cabling.

BIG BUCKS FOR RDBS

Britton Lee, Inc., the Los Gatos, CA, provider of relational database software, finished 1984 with growth rates topping 125 percent. Revenues for the year reached $21.6 million with profits of $3.8 million. This compares with revenues in 1983 of $9.5 million and a loss of $1.4 million. Britton Lee claims to have more than 350 of its programs installed worldwide.

A little further north in California, Menlo Park-based Oracle Corp., a privately held firm, is also enjoying a good year with its relational database product. According to spokesperson Judy Diaz, "Right now, we're on target to double our revenues over last year." Last year Oracle took in approximately $13 million. Diaz suggests that relational database software is doing so well because, "IBM has put its blessing on the technology."

From its new headquarters in Alameda, CA, Relational Technology, makers of the Ingres RDBS, reports its best year yet. With a fiscal year closing in June, Peter Tierney, marketing vice president, at Relational, said, "It's no big secret. Generally speaking, the entire industry is doing well. Enough missionary work has been done over the last two years so that relational systems are being recognized for what they can do." Tierney revealed that Relational Technology took in $8.1 million in fiscal 1984. He expects growth curves to reach 250 percent by the end of June.

For its own part, Tierney believes that the growth of Ingres and UNIX have been closely linked. "Since Ingres was developed along with UNIX (4.2BSD) at Berkeley, Ingres grows as the UNIX market grows," he explained. AT&T's endorsement of Ingres hasn't hurt either, he pointed out.

"HEATHKIT" UNIX

Like to get a UNIX system for less than $3000? Check out DMS Design of Cupertino, CA (John Bass, owner/senior engineer, 408/996-0557). The company says it will be shipping a two-user Version 7 UNIX system this month that offers everything but Fortran 77 and floating point. The pricetag will be $2700.

The system comes as an unpackaged computer. The CPU is a M68008 with one megabyte of RAM. It has a 5¼ or 3½-inch floppy disk drive, along with a 30 MB Winchester hard disk and controller. The SCSI standard interface provides for eight devices to link to the system. (The M68008 and the hard disk controller take

INTEGRATED OFFICE SOFTWARE

XED®

Full Screen Interactive Document Processing • Spreadsheet
Calendar Scheduling • Legal & Accounting Applications
Data Base & Forms Management • Telephone Directory
XED PostMaster™ Executive Mail • Typesetting

XED®
INTEGRATED OFFICE SOFTWARE

Box 3938 • Chatsworth, CA 91313 U.S.A. • (818) 884-2000
FAX (818) 884-3870 • TWX 910-494-1716 • Intl. Telex 292 662 XED UR

up one connection each on the SCSI interface.) A four-segment MMU and DMA controller come with the system. A power supply and cables are also delivered.

Later in the year, DMS plans to introduce a M68020 job processor board and a four-plane 16 color RGB graphics board. Both of these are estimated to cost less than $1000 each.

By providing the system in "kit" form—that is, with no enclosure—DMS expects it will save a fortune in packaging and regulatory costs. The product is intended for the serious home user and small OEM.

UNIX TOOLCHEST

Those who browse through the AT&T UNIX Toolchest (described by Mark Sobell in February's *Industry Insider*) may be surprised this month to find source for the much-ballyhooed Korn shell (**ksh**) included in the listing. The new offering, which offers C shell functionality and Bourne shell compatibility, has long been popular within AT&T Bell Labs but rarely has been seen outside.

The Toolchest itself, meanwhile, is going public. In its first few months of life, AT&T's electronic software distribution system was open strictly to "preferred customers".

Among the offerings in the Toolchest are a variety of programs that AT&T views as useful, but not yet appropriate for full development and promotion efforts. Customers are welcomed to purchase *source code*, as is, after locating programs they want via a Browsing System computer containing program descriptions that anyone with a modem can dial up, log on, and scroll through. The number for the system is: 212/522-6900.

The Browsing System list has been bolstered in recent weeks by the addition of the Korn shell, a Lisp interpreter, and nine other tools. Rumor has it that a source license for the shell will sell for $2000 per site.

Browsers who want to buy the Korn shell or any of the other packages can do so by indicating to the Browsing System that they wish to have *purchase authority*. AT&T will subsequently send a licensing agreement in the mail.

Mark Hall is the Associate Editor of UNIX REVIEW. ■

ERROR-FREE
MULTI-USER
EASY TO USE
AVAILABLE
AFFORDABLE
ULTRALINK
UNIX TO VAX/VMS
COMMUNICATIONS
FROM
CREARE

The power of your VAX is a single command away. UltraLink is a cost-effective software solution for error-free transmission of data between UNIX computers and any VAX running VMS. Without any additional hardware, you can transfer files, get print-outs on your high-speed VAX/VMS printer, and run interactive VMS sessions from your local UNIX terminal—all for $1800. Call (603) 643-3800 for more information. Creare Inc., Etna Road, PO Box 71, Hanover, NH 03755.

TRADEMARKS: VAX, VMS: DIGITAL EQUIPMENT CORPORATION; UNIX: AT&T BELL LABORATORIES; ULTRALINK, CREARE: CREARE, INC.

Circle No. 61 on Inquiry Card

FAST FRIENDS

VMS MEETS UNIX

Once not on speaking terms, VMS(tm) and UNIX(tm) are now completely compatible partners, thanks to VAX UNITY(tm) under VMS, our proven bridge between UNIX software and VAX VMS systems.

UNITY features:

- concurrent and reliable use of both VMS and UNIX
- AT&T's UNIX System V, Release 2, with Berkeley 4.1 commands
- easy installation and maintenance
- an unchanged, DEC-supported environment
- no VMS modifications or special permissions needed; UNITY runs as a VMS user level application
- cannot crash VMS
- complete and correct UNIX functionality
- significant recent performance improvements
- direct access to the ever-growing library of UNIX software
- no retraining needed for UNIX-trained personnel to use VMS

Backed by HCR's high-calibre support, UNITY brings to VAX VMS sites our ultimate product...

PARTNERSHIPS IN UNIXCELLENCE.

Human
Computing
Resources
Corporation

HCR

10 St. Mary St., Toronto, Ontario, Canada M4Y 1P9 Tel. (416) 922-1937 Telex 06-218072

VMS is a trademark of Digital Equipment Corporation. UNIX is a trademark of AT&T Bell Laboratories. UNITY is a trademark of Human Computing Resources Corporation.

Circle No. 4 on Inquiry Card

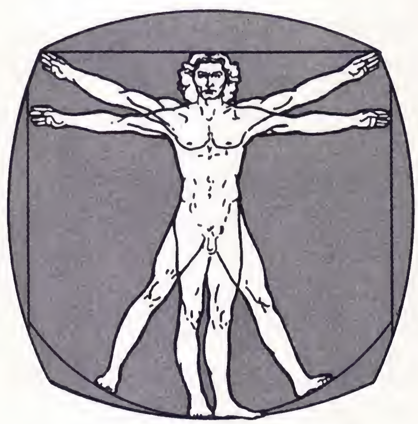
THE HUMAN FACTOR

The human side of networking

by Richard Morin

I have some good news and some bad news for you. The good news is that you're going to be on a network soon—if you're not already. The bad news? If you're not already on a network, you soon will be.

Networks can be very useful. They allow speedy and generally reliable communication. But, like any powerful piece of technology, they affect the way people operate.



IMMEDIACY

In large companies, a great deal of business is transacted by means of interoffice mail. A memorandum is sent, and a response is generated and returned—a process guaranteed to take several days.

Since recipients know this, they can plan accordingly. Even if the request is made in urgent terms, a delay of a few hours or even a day is generally anticipated.

Truly critical requests are made by telephone. If a client calls with a request, it is (usually) handled promptly. The hassle of reaching somebody by telephone, though, keeps the number of calls within reason.

Enter computer networks, offering transmission times measured in seconds and probabilities of reaching intended recipients that are quite high. Networks are a good thing, in general, but they also require new strategies.

How will we know how to prioritize requests? If we take days to handle urgent requests, someone will surely notice. If we treat all requests as urgent, we'll go crazy trying to answer them all immediately.

Different strategies will evolve. People will try to estimate the urgency of a request from the way it is stated. They are also quite likely to prioritize by the identity of senders, and to treat low priority messages much as they treat other junk mail.

JUNK MAIL

The postal system is a good example of an efficient and economical network. With few exceptions, mail is delivered to its destination within a few days. The cost of sending mail is low, usually far less than the cost to generate it.

This low cost encourages the proliferation of junk mail. Special rates make bulk mailings especially attractive. We are thus beset by piles of mail that are neither requested nor desired, and must be sifted through for the few letters that bear reading.

Meanwhile, our own companies produce brochures, catalogs, catchy letters, and other promotional literature that we're sure everybody on *our* mailing lists wants to see.

Now consider the telephone. If your telephone were to ring right now, you would probably put down this magazine to answer it. Most of us allow telephones to interrupt our meals, our rest, and (occasionally) even our intimate moments. We are *trained* to answer telephones, and it takes a great deal of self-control to ignore one.

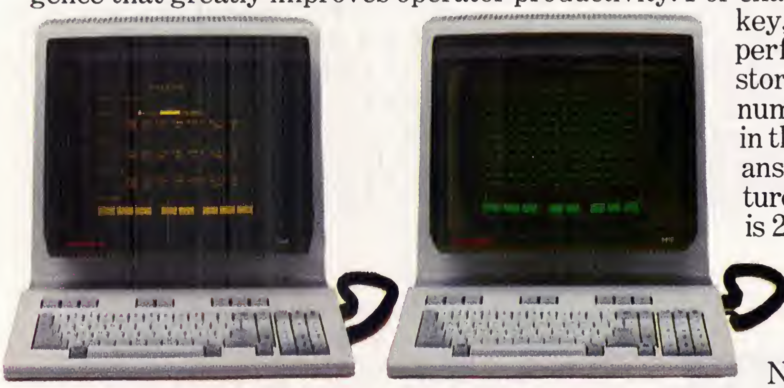
Fortunately, the number of "junk calls" is fairly low—largely because the labor involved is expensive. The perfidious automatic calling machines solve this, at the risk of retaliation from outraged victims. (Another use for auto-dial modems...)

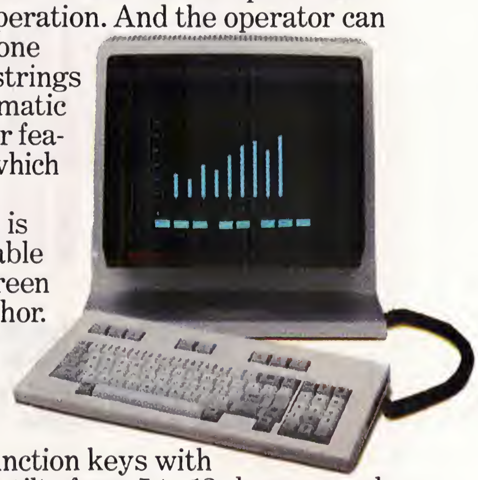
Computer networks offer a whole new avenue for junk mail. They provide immediacy at low cost. How long will it be before

WHAT'S BEHIND OUR CONVERSATIONAL TERMINAL MAKES WHAT'S IN FRONT OF IT MORE PRODUCTIVE.

At Teletype Corporation, we just made another intelligent move. We added an integrated modem to the 5410 conversational terminal—making it an even better value.

In addition to saving desk space and being cost-effective, the modem gives the 5410 "built-in" intelligence that greatly improves operator productivity. For example, all the operator has to do is push a single key, and the modem will dial a host computer and perform the logon operation. And the operator can store up to three phone numbers and logon strings in the modem. Automatic answering is another feature of the modem, which is 212A compatible.





The 5410 is also now available with a white, green or amber Phosphor. No matter which color you choose, your operators will appreciate the crisp, easy-to-read characters. High resolution is maintained even when switching from an 80 to 132 column mode.

Other features that enhance productivity include 8 programmable function keys with matching screen labels; an English option menu; a detachable keyboard that tilts from 5 to 12 degrees and has tactile feedback; standard character sets; and the list goes on and on.

In the interest of operator productivity, write to Teletype Corporation for more information on the 5410 at: 5555 Touhy Ave., Dept. 3223-OO, Skokie, IL 60077. Or call 1 800 323-1229, ext. 115.

TELETYPE: VALUE SETS US APART.

"Teletype" is a registered trademark and service mark of Teletype Corporation.

AT&T
Teletype Corporation

Circle No. 17 on Inquiry Card

junk mail starts showing up on our computer screens?

Probably a while, fortunately. There is a strong social stigma attached to abuse of privileges on the existing networks. Even so, some network members find themselves awash in unsolicited mail.

Worse, an even greater threat looms on the horizon. Commercial networks, where no social constraints exist to civilize mailers, have begun to enter the picture. In time, many of our computers will be linked into both commercial and non-commercial networks. What piles of trash will we then inflict on each other?

Mail screening programs are beginning to appear, but they can cause problems of their own. Poorly designed screening programs can discard useful information or let junk mail through. (Is it true that DARPA is funding development of an Anti-Bombastic Missive (ABM) system?)

> Your connections depend on your connections.

Perhaps the best idea would be an automatic filing and deleting system for junk mail. A set of keywords at the top of the message would allow the system to index the material. Any junk mail sent without keywords would be stored in */dev/null*.

Given this incentive, vendors would quickly learn to use keywords. Recipients would be able to find needed information quickly and cooperating vendors would be delighted to find telephone calls coming from prospects who had actually saved and read the appropriate literature.

FUTURES

Non-commercial and private networks are just beginning to link up. There are gateways connecting Arpanet, Bitnet, CSnet, usenet, uucpnet, and others. Access to these networks tends to be limited: your connections depend on your connections.

In addition, one often has to know routing information. You can't use a gateway if you don't know its name. To send a piece of mail over usenet, you must specify a path between the sending and receiving sites.

This is changing, however. Through the efforts of Mark Horton and others, a domain naming facility is being developed for usenet. Eventually, this will allow addresses to be specified without regard to paths.

Mail will then pass freely among the cooperating networks. Using an address such as *DOMAIN.site!user*, one will be able to send mail to anyone from Alfred Aho to Patrick Winston. (Imagine Dennis Ritchie's screening program.)

Commercial networks have yet to break into this circle, however. My UNIX mail software can't talk to my CompuServe EMAIL account. In fact, as far as I know, no gateways whatsoever currently exist for commercial networks.

Consequently, there are no commercial services available over these networks. There is certainly no facility for value-added services. Try to sell services over a net and you will find that your mail will disappear, or worse.

WHAT MAKES SOFTRAK PROJECT MANAGEMENT SOFTWARE UNIQUE? UNIX.

AND A NO-RISK OFFER.
CALL TODAY. 1-801-531-8550

SofTrak
Application Software for Project Management
1977 West North Temple, P.O. Box 22156 AMF, Salt Lake City, Utah 84122

Circle No. 35 on Inquiry Card

COHERENT™ IS SUPERIOR TO UNIX* AND IT'S AVAILABLE TODAY ON THE IBM PC.

Mark Williams Company hasn't just taken a mini-computer operating system, like UNIX, and ported it to the PC. We wrote COHERENT ourselves. We were able to bring UNIX capability to the PC with the PC in mind, making it the most efficient personal computer work station available at an unbelievable price.

For the first time you get a multi-user, multitasking operating system on your IBM PC. Because COHERENT is UNIX-compatible, UNIX software will run on the PC under COHERENT.

The software system includes a C-compiler and over 100 utilities, all for $500. Similar environments cost thousands more.

COHERENT on the IBM PC requires a hard disk and 256K memory. It's available on the IBM XT, and Tecmar, Davong and Corvus hard disks.

Available now. For additional information, call or write,

Mark Williams Company
1430 West Wrightwood, Chicago, Illinois 60614
312/472-6659

Mark
Williams
Company

COHERENT is a trademark of Mark Williams Company.
*UNIX is as trademark of Bell Laboratories.

Circle No. 6 on Inquiry Card

The non-commercial networks, as mentioned above, have reserved a private circle of Hell for despoilers and commercializers.

The bad news? If you're not already on a network, you soon will be.

On usenet, for instance, one is allowed to post, once, a brief announcement of a new product. Anything more will cause instant howls of outrage.

Usenet mail is supposed to be strictly non-commercial in nature. After all, why should *ucbvax* forward internal memos for Unixystems, Inc? One suspects, however, that a certain amount of commercial traffic slips into the stream.

There is an accepted way to handle commercial mail between UNIX sites. The sites set up a direct link using **uucp**. No other sites are burdened with the mail, and the messages travel very quickly.

This is difficult to maintain, however, if many sites are involved. In addition, broadcast messages are not handled efficiently by direct transmission. The Unixystems main office doesn't want to make a separate call to each of its customers to distribute a bug report.

A MODEST PROPOSAL

What if we instituted a new network (bucknet?), using the existing **uucp** software? Sites could link up to it in the same way that they now link up to usenet. The only difference would be that services would be provided on a commercial basis.

Some firms might wish to enter the mail forwarding business. Usenet members do this for free now, but they resent handling even quasi-commercial mail. A commercial mail forwarder would have no such compunctions. It could even forward to other commercial networks.

Other firms might sell batch processing, typesetting, or even consulting services over the net. Some effort would certainly be involved. We would need to develop software for accounting, mail tracking, and so forth.

The new software could be layered onto the current software, however. This would reduce the effort involved, and simultaneously guarantee compatibility.

Complications would arise, of course. Rules would have to be worked out to control the passing of data between commercial and non-commercial networks. These guidelines could be developed, however.

Some appealing hybrid services might be touchy. What about an indexed network news database? Or a commercially edited version of the news? Would these be unacceptable "commercializations" of usenet?

A commercial version of usenet could provide services that usenet itself cannot provide. It could also provide an economical and legitimate path for some of the commercial and quasi-commercial mail now transmitted surreptitiously on usenet.

Perhaps the most interesting thing about bucknet is that it could develop in the same anarchic way that usenet and uucpnet did. Anyone wishing to start up a bucknet site would simply be able to do so. Pick a service, announce your **uucp** login details, and you'd be in business.

A bit of preliminary coordination and discussion may be in order, however. I'd be happy to field any comments, flames, or expressions of interest. All these will be gratefully accepted. Given sufficient interest, a followup column will appear.

How long will it be before junk mail starts showing up on our computer screens?

IN SUMMARY

We have just begun to tap the combinatoric power of computer networking. The next several years will see the development of global networks with tens or hundreds of thousands of sites. New strategies, traditions, and—yes—laws will develop to handle the problems created.

Long-distance networking was once strictly the province of research scientists and military personnel. The UNIX community and the educational establishment have now begun to join the fray. Other groups will join over time.

See you on the net . . .

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

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

4.2BSD

FROM NOW ON, CONSIDER IT SUPPORTED.

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

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

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

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

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

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

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

"We know UNIX™ Backwards and Forwards"

MT XINU™

UNIX™ SUPPORT FROM BERKELEY

Circle No. 7 on Inquiry Card

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


H. KIM (85)

THE UNIX NETWORKING LABYRINTH

Thoughts on the LAN challenge

by Dr. Greg Chesson and Mark Hall

I exist as I am, that is enough,
If no other in the world be aware, I sit content,
And if each and all be aware, I sit content.

Walt Whitman,
Leaves of Grass

Around 1700, the English governor of a backwater of the British Empire known as "Virginia" planted a large garden to entertain guests at his Williamsburg mansion. Among the well-manicured lawns and flower beds grew an extensive labyrinth of tall hedges. Upon entering the labyrinth, the governor's friends could take a number of different paths and wander for hours. The governor was often amused by the plaintive cries of visitors lost in the labyrinth, begging for a guide to lead them out to afternoon tea. For users and system developers working through the maze of UNIX local area network options, the frustrations those visitors felt must seem familiar.

It was inevitable that UNIX and local area networking would merge. Neither are particularly new and both have large and growing followings. UNIX, as we know, can be traced back to the 1960s. It has since been ported to a multitude of computers. Local area networking is equally mature. Ethernet, for example, was developed at the Xerox Palo Alto Research Center in the early 1970s. Arcnet, Datapoint's LAN, has been in use for close to a decade.

Both system manufacturers and LAN firms are well along in the production of technologies that integrate UNIX and local area networks. Nevertheless, the nature of UNIX complicates the development of these products—a fact not masked even by the availability of numerous proprietary network systems.

> Like the character in Whitman's poem, UNIX exists by itself and is quite content.

THREE WALLS BETWEEN UNIX AND NETWORKING

UNIX aficionados praise the level of performance, adaptability, control, and variation the system offers. The "point of view" UNIX has of itself—that is, the means by which the kernel manipulates its directories, utilities, and files—is essential to these capabilities. To a UNIX system's way of thinking, everything that "exists" does so under its aegis. UNIX is the master of all that it "sees". There is no other system with which to share control. Neither is there any provision within the UNIX kernel for *recognizing* the existence of any other system. Like the character in Whitman's poem, UNIX exists by itself and is quite content.

When UNIX refers to something in its system object base, such as addresses or names, a request is perceived and executed with the assumption that it is a local request. Similarly, all system calls under UNIX are assumed to be local. What's more, a number of operating system commands are ill-defined in the context of a network. For instance, what does it mean to do a network **fork** or **exec**? It's not clear. Dilemmas like these commonly confront network developers.

UNIX was conceived without networking in mind. Its origins as a single-processor operating system preclude a simple transition to multiprocessor environments. As local area networks have flourished, the impetus to integrate separate UNIX-driven computers has increased. To accomplish this, communications specialists must first clamber over the name/address/object space wall that stands between them and a system that recognizes a peer.

The next barrier to getting UNIX on a network is the lack of a suitable IPC (interprocess communication) facility. In the world of UNIX, the *pipe* is the alpha and the omega for IPC. But to service a network, a system must have the ability to address other processors, processes, and resources. In addition, each processor must be able to exchange data with other processors and retain control of information as it moves about the network.

To address the IPC problem, many alterations have been made to the system over the years. The RAND ports, the University of Illinois IPC Mechanism, the Version 7 **mpx** Multiplexor, the Carnegie-Mellon ports, the USG (UNIX Support Group) Messages, the Berkeley 4.2 Select Mechanism, and the Berkeley 4.2 sockets, among many others, attest to the inventiveness of the software mind. They also create a new problem. With so many ways to solve the IPC puzzle, which one is right? No approach has been successful enough to be considered a standard. And each reflects a technique and style of IPC so radically different from the others that it's difficult to imagine a hybrid approach.

Daunting as the IPC challenge is, it is not the only formidable one to confront UNIX network developers. A third wall results from the need network software has for asynchronous I/O to effectively manage more than a single I/O stream. However, UNIX I/O is synchronous, making it difficult—and, in some cases, impossible—to manage multiple I/O streams. This is the reason why most of the IPC mechanisms that have been added to the operating system employ some technique for supplying asynchronous I/O or an equivalent.

All of this is not to say that designing a local area network for UNIX-based processors is impossible. It's been done in the past and there's no reason to think it won't be accomplished in the future. As mentioned above, many developers have adopted their own variations for linking UNIX machines in a network. But therein lies the rub; the variation is too great.

The commonality of UNIX as an operating system can grow fuzzy as discrete communications protocols are introduced. The value an organization gains by standardizing on UNIX can be easily undercut by disparate methods of connecting computers.

The market is replete with possibilities for integrating hardware capable of offloading network protocols.

Part of the problem stems from the long-standing need to provide network services to UNIX users. The need is substantial enough that a wide range of solutions has come into play. The **uucp** utility was, in its fashion, one of the earliest attempts to communicate data on both local and wide area scales. Many UNIX networks operating today use the **uucp** protocol to transmit data to machines within a single room, building, or collection of buildings.

The **uucp** utility offers low-level file transfer capabilities—such as **rmail**—that resemble many of the applications level programs that are available on microcomputer LANs. The **uucp** facility can even execute commands on remote machines.

But **uucp** has its limitations. For one thing, it was designed to operate in a low-speed telecommunications environment, typically at 1200 or 2400 bits per second. Bulk file transfers are not well suited to such a network. Furthermore, the error checking capabilities of **uucp** are not perfect. The file queuing mechanisms of **uucp** represent a significant system load, which is perhaps appropriate for a low-speed network. But on a high-speed LAN, it would seem that better techniques could be found.

STREAMLINING COMMUNICATIONS

One of the most recent changes in UNIX that might guide LAN developers out of the labyrinth to a pleasing afternoon tea is the *stream* I/O mechanism. Streams, developed at AT&T Bell Laboratories by Dennis Ritchie, has the advantage of simplifying kernel I/O. It provides a place for protocols to live under UNIX by offering a procedural interface to processing functions. With streams, device drivers become simpler, I/O overheads grow smaller, and protocol layering can be accomplished with ease. The modularity of streams is especially well-suited to offloading network activities to an outboard processor.

But even this method has its drawbacks. Streams does not provide for asynchronous or concurrent operations. It needs to use the **select** call or its equivalent, requiring twice as many system calls as were previously necessary to get something done. Also, **select** is only available with 4.1/4.2BSD. Another problem with streams is that it does not solve the name/address/object space dilemma that was discussed earlier. It also requires that protocol code be inserted into the kernel, making the kernel more complex than it already is.

There are several ways to reduce these networking software problems. One is to rewrite the kernel. Another, as applied by streams, is to replace structure in the kernel. These approaches don't make the problem go away, however. They merely change the environment where the work must be done to solve the puzzle.

The availability of advanced hardware that can be used to off-load protocol and related processing to intelligent peripherals offers a viable approach to networking in the UNIX realm. The software issues that exist in UNIX at the macrocosm level, however, cannot be purged this way in the intelligent peripheral microcosm. Still, if an intelligent processor implements several layers of network protocols, it simplifies the software adjustments that must be made in the operating system. Established Ethernet manufacturers, such as ACC (Santa Barbara), Bridge Communications (Cupertino), CMC (Santa Barbara), Excelan (San Jose), Interlan (Littleton), Ungermann-Bass (Santa Clara), and 3Com (Mountain View), all offer intelligent peripherals that can perform the role of implementing several layers of network protocols. Indeed, the market is replete with possibilities for integrating hardware that is capable of offloading network protocols. Intelligent controllers do a fine job of handling a single network protocol, but UNIX often needs to communicate using multiple protocols. And granted that intelligent network peripherals have made key contributions to moving software out of the kernel, there still remains plenty of room for adjustments in the three walls: name / address / object space, IPC, and asynchronous communications.

CIRCULAR PATHS

It seems that no matter how you attack the UNIX networking problem, you work your way back to these three barriers. Because of this, when higher level networking functions such as distributed file systems are generally available, a rewrite of the UNIX kernel seems inevitable. But, of course, that won't be the first time the kernel has been modified.

> **Just as the colonial governor of Virginia established and enhanced his labyrinth, programmers have long been carving their initials into the UNIX tree.**

Why rewrite the kernel? Think of the three walls. When a user program on a network requests a file, the operating system has to know whether the file is local or remote. To get UNIX to accept the concept of "remoteness", the kernel has to be modified.

But even after the system can recognize that a file is remote, it also must be able to find a network address for the file. There's nothing in traditional UNIX to provide for that. So, once again, we see that the kernel needs to be changed.

Even this, though, does not complete the process. Once a file is located, communications with the remote file server must be established. This requires adjustments to UNIX IPC and additional concurrency. It's time to crack open the kernel again.

Finally, after all of this has been accomplished, there is still the matter of data security to confront. Like all operating systems, UNIX has its security weaknesses. To address this, the kernel must be changed.

These points have already led to a vigorous evolution of UNIX. Software developers have always wanted to change UNIX, to tweek it, to make it better. Just as the colonial governor of Virginia established and enhanced his labyrinth, programmers have long been carving their initials into the UNIX tree. This evolution has strengthened UNIX, just as the carefully designed shrubbery gave dimension and added challenge to the Williamsburg garden.

Like that garden, UNIX flourishes on change. That's why, despite all the barriers to networking, UNIX will continue to be adapted to LAN needs. The changes are necessary if UNIX is to continue growing.

Dr. Greg Chesson serves on the Editorial Review Board of UNIX REVIEW and is presently Chief Scientist at Silicon Graphics, Inc., in Mountain View, CA. In the distant past, he worked as a drummer in the Woody Herman Band and the C.C.Riders. While a member of the Computer Science Research Department at Bell Laboratories from 1977 to 1982, Dr. Chesson developed the packet driver protocol used by ***uucp*** *and the* ***mpx*** *files in the Seventh Edition of UNIX, as well as the original protocols and software implementations for the Datakit network. At Silicon Graphics, he has implemented XNS and other network protocols and is currently developing new technologies.*

Mark Hall is the Associate Editor of UNIX REVIEW. ■

COSI

T A N G O™

Use Tango to:

- Connect IBM·and compatible PC's running DOS to UNIX systems.
- Offload processing to PC's.
- Control data and applications on remote PC's.
- Distribute processing between UNIX and PC's.

Buy Tango for:

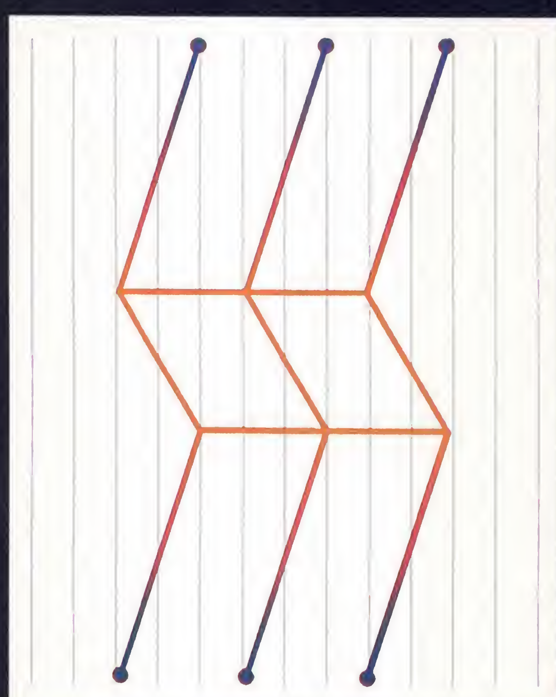
- Execution of DOS programs on the PC under UNIX control.
- Simple elegant file transfer under error correcting protocol.
- DEC, IBM, and Tektronix (graphics) terminal emulation.

Tango utilizes a standard RS-232 serial port on the PC and connects to the UNIX computer via a modem or direct connection.

COSI

313 N. First St.
Ann Arbor, Michigan
48103
(313) 665-8778
Telex: 466568

Tango is a trademark of COSI. UNIX is a trademark of Bell Laboratories.



The PC-to-UNIX™ Connection

Circle No. 8 on Inquiry Card

The Potpourri of Networks

System protocols vary so widely that communications among them have long been difficult—often excruciating. At times, they've actually been impossible. In light of this, the International Standards Organization (ISO) appointed a committee in 1977 charged with establishing an architecture to enable applications operating on different computers to exchange usable information.

The result of ISO's labors is the now-famous seven-layer Open Systems Interconnection (OSI) reference model. By structuring all data communications in accordance with this model, systems and software developers stand a good chance of being compatible in a multi-vendor environment—so long as other systems also comply with the ISO's architecture.

From bottom to top, the seven layers defined by the OSI are:

- Physical
- Data Link
- Network
- Transport
- Session
- Presentation
- Application

The *physical* layer deals with the interface between devices. This is often defined as the *media* level, and includes such familiar connections as twisted pair, coaxial, and fiber optic cabling. It establishes guidelines for signal voltage, bit duration, and the mechanical and electrical characteristics of the interface.

The second level, the *data link* layer provides for error detection. It also enables, disables, and maintains the physical link while data is being transmitted.

Level three, the *network* layer, offloads concerns about the physical connections between systems from the higher layers of the model. A portion of the switching and routing function throughout the network takes place at this level.

At the *transport* layer, level four, the end-to-end transfer of data between processes occurs. Packets are checked at this level to ascertain data integrity.

At the fifth layer, the *session* level, communications between applications are managed. It's here that the patterns for one-way/two-way concurrent or two-way simultaneous communications are set. The session layer is also where a transmission can be re-established should a disconnection take place. Data encryption during transmission also occurs at the session level in many implementations.

The *presentation* level, the sixth layer in the model, handles code conversion for non-compatible communicating devices, such as those using EBCDIC and ASCII.

The top layer, the *applications* level, provides end users with usable programs, like electronic mail or network management functions.

PROTOCOLS BY THE BUSHEL

Since data had been transmitted for a good many years prior to the activities of the ISO in 1977, a number of non-conforming approaches had already gained a foothold. The one that is most widely used and emulated today is Systems Network Architecture (SNA) from IBM. In contrast with the OSI model, SNA architecture consists of five layers: data link control, path control, transmission control, data flow control, and presentation services.

Unlike the approach taken by ISO, which intended for its communications architecture to be hardware and software independent, SNA is very hardware dependent. For example, SNA defines a complex interrelationship between *physical units*, called *PU.types*. A particular processor, say an IBM 4341, would be configured as a PU.T5 and a cluster controller like the IBM 3274 would be configured as a PU.T2. These classifications are critical to establishing communications in the SNA world: physical units that can't be configured as PU.types can't play ball.

Another communications architecture comes from Digital Equipment Corporation. Digital Network Architecture (DNA) is an eight-layer method for transmitting data between DEC or DEC-compatible systems. Beginning at its lowest level, DNA consists of communications facilities, physical links, data links, routing, end communications, session control, network applications, and network management layers.

Xerox Network Systems (XNS) offers a four-layer architecture. Level zero, as Xerox has dubbed the lowest level, depends on transmission protocols, such as Ethernet or X.25 to prepare packets for transmission over a given media. Level one handles the addressing and routing of packets. Level two uses a store-and-forward algorithm to route datagrams across the network. And level three provides control protocols and data structuring to packets on the network.

At Wang Laboratories, yet another proprietary method has been adopted. Under the rubric of WangNet, a three-tier architecture controls the flow of data in a network. The transport layer, the first level, fulfills the role played by the first four layers of the OSI reference model. WangNet's services level handles the activities provided by the OSI's fifth layer, as well as part of the sixth layer. The rest of the OSI's sixth layer responsibilities and all of the seventh layer duties are picked up by WangNet's applications level.

Still other architectures have been adopted by companies such as Burroughs, Control Data Corp., and Data General, to name just a few. But the yardstick for all architectures, regardless of origin or form, is the ISO seven-layer reference model.

GETTING ON THE NETWORK

Just as there are many approaches to network architecture, network access also has its myriad of techniques. One of the most popular access mechanisms is CSMA/CD, Carrier Sense Multiple Access with Collision Detection. Ethernet employs it, as does the IBM Personal Computer local area network. A variation is CSMA/CA, Carrier Sense Multiple Access with Collision Avoidance. Both are arbitration schemes, meaning that devices contend for control of communications channels before broadcasting data. The former method is more sophisticated and offers a higher probability of getting data from point *a* to point *b*. The Institute for Electrical and Electronic Engineers (IEEE) has established the 802.3 committee to review potential CSMA/CD standards.

Another approach that has a lot of promise is the token ring. The promise comes from the fact that it's supported by IBM. The IBM Cabling System, the company's sanctioned local area network uses token ring access techniques, because of the strengths they offer to networks requiring deterministic operations. That is, in contrast to CSMA/CD's "everyone has an equal shot to send data" method, token technology establishes paths for data and a hierarchy of devices that can transmit on the network. The IEEE 802.5 committee is studying possible standards in this area.

The token bus, like token ring mechanisms, employs a deterministic access to the communications media. It differs chiefly in its topology—that is, in the way the media is logically distributed. The token bus approach has been championed by companies like General Motors because of the strengths it brings to factory networks. The IEEE 802.4 group is studying this technology.

MORE TO COME

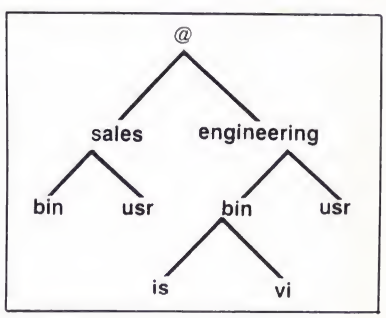
The various communications architectures and techniques mentioned here are far from exhaustive. With advances in microprocessor technology and software development coming at an accelerating rate, it's likely that even more will become available with time.

Mark Hall

ALTOS

WANTED:
UNIX ENGINEERS

Altos is a leading supplier of UNIX*-based microcomputers. We have positions open for Senior UNIX Engineers who can help us develop new 286 and 68020 based UNIX products. Projects include design and coding of virtual memory, dual-processor Kernal design, and porting of Berkeley 4.2 file system enhancements to Xenix and System 5. We also have openings in our communications and applications groups.



The diagram on the left tells you everything you need to know about our state-of-the-art network product: WorkNet. Developed entirely by Altos, WorkNet links multiple UNIX systems in a single distributed file system. With an installed base of over 500 networks, it represents one of the many exciting products you can help us develop at Altos.

A BS or MS in computer science is required, with several years of experience with 3270 BISYNC, 3780 BISYNC, SNA or X.25 protocols. Our applications group is looking for experienced developers of friendly office software. Please send your resume to: ALTOS COMPUTER SYSTEMS, 2641 Orchard Parkway, San Jose, CA 95134. An equal opportunity employer. Principals only, please.

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

Circle No. 28 on Inquiry Card

NOW THAT THE PC FAD IS OVER, IT'S TIME TO GET DOWN TO BUSINESS.

Like hordes of locusts, the PC swept the business community. Corporations bought them like electronic calculators by the thousands to improve the productivity of their executives. Portables were carried home from the office every evening and on trips. Computerization was even affordable to the small business for the first time. Programmers put their unique genius to work to develop some of the best software ever written. Productivity tools like word processing, electronic spread sheets, data base management and accounting was placed into the hands of new computer users. Productivity improved for everyone. From the CEO . . . to his staff . . . to the salesman . . . to his secretary. Forecasts for continued PC growth were nothing but highly optimistic. One at every desk. One in every home. What happened?

> *"Networking won't solve the multiuser problem either economically or functionally."*

Like the first crust of any marketplace it saturated quickly. Those that are the first to buy almost anything new and promising, bought. There are no more computer hackers and hobbyists to sell to. They all have one. Applications for the home that made any sense didn't develop. Corporations found that they needed PCs to "talk" to each other. That solution is distant because networking won't solve the problem either economically or functionally. Most available networking does nothing more than messenger floppies around. The small business found that as soon as its first PC was operational and productive, a second one was needed to satisfy demand usage. The PC, with all its promises, turned out to be a dead end for the business environment. The PC and clones just haven't been the godsend for business that was predicted. Why?

The PC is a personal computer. Just that. Not a business computer. That's because PCs are single user computers with single user software. Good for one person but not good enough for a whole company. Even if the company is two people.

Every computerized business has someone entering information while someone else is looking up information. That's two users. And every business has more than two users who need access to the computer. That's a multiuser computer environment.

> *"The small business needs a second PC as soon as the first one is working."*

It's now hard to justify PCs in a business environment. A multiuser computer capable of supporting up to five users is available for the price of a single IBM PC XT. It has more storage and a business oriented operating system. Supermicros are available that have the power of minicomputers without the accompanying price tag. Ten unconnected PCs, sitting around worth about $50,000, doesn't make sense when for much less you can get a lot more computing power in a supermicro that accommodates 20 or more users. But don't let even that price tag scare you. On a per user basis, multiuser computers cost about $1500 less than a PC. New users can be added for less than $600 with a dumb terminal. And they're upgradable.

> *"A six port multiuser computer is now available for the price of a single IBM PC XT . . . microcomputer systems cost $1500 less per user than multiple PCs."*

Multiuser computers communicate with each other. They share the same data base, software and peripherals. They have sophisticated business features such as record locking, user accounting privilege levels and system security. They are business oriented and priced well within the reach of the first time computer user.

But what about all the PCs already in place? Don't ask the PC manufacturer for a solution. They're concentrating on selling more single user systems. The real solution is to get started with a true multiuser computer in the first place. With multiuser business computers now in the same price range as a PC, it doesn't cost any more to make the first step the right step.

The PC has seeded the next wave. It's here now. Supermicro multiuser computers that can support up to 32 users. If you don't believe it just look at the new product introductions from IBM, DEC and AT&T, let alone the smaller companies like Altos, Plexus and IBC. Big system features for every end user. Software for every conceivable specialized business application. That's not the end of it. New challenges are there for everyone. Opportunities abound. Software companies are already applying their talents to multiuser operating systems, disk conversion and even more powerful and productive software. Companies are shifting their emphasis to provide multiuser system enhancements as they did for the PC. Value added resellers and specialist dealers will give the end user the support that's been terribly lacking from department store retailers. It's a great day for someone who needs a multiuser computer. And everyone does.

> *"Multiuser computers share everything . . . they have business features such as record locking, user accounting, privilege levels and system security."*

Thanks PC! You've whetted the appetite of a large new business environment for computerization. One that is bigger, more demanding, and more sophisticated than we've ever seen before. There's no turning back now. You were a fad, but now it's time to get down to business . . . multiuser business.

Randy L. Rogers
President and CEO
IBC/Integrated Business Computers
Manufacturer of Multiuser Computers
Chatsworth, California.

Circle No. 65 on Inquiry Card

RULES OF THE GAME

Shifting software

by Glenn Groenewold

Local area networks are a logical extension of computer technology. From the standpoint of efficiency, it makes a great deal of sense to link CPUs and users in this fashion. As in other areas of computing, though, we find that our burgeoning technology may have outrun existing concepts of law.

Because of this, a showdown may be looming that could be as explosive and far-reaching in its impact as the ongoing struggle between the owners of copyrighted films and the manufacturers and users of home VCRs. This controversy recently resulted in a Supreme Court decision—the *Betamax* case—that left film copyright owners on the ropes, for the time being at least.

The immediate issue in the *Betamax* litigation was whether VCR owners could legally record copyrighted material off television broadcasts. Underlying the case was a more basic question: to what extent could copyright owners continue to control the use of their property once they licensed it for telecast?

The Court was not persuaded by the owner's argument that the effect of widespread off-the-air copying would be to deprive them of future sales. Instead, it looked at the innocent nature of the activity constituting the alleged "infringement" and concluded that

American video enthusiasts were not nefariously appropriating someone else's property for commercial gain.

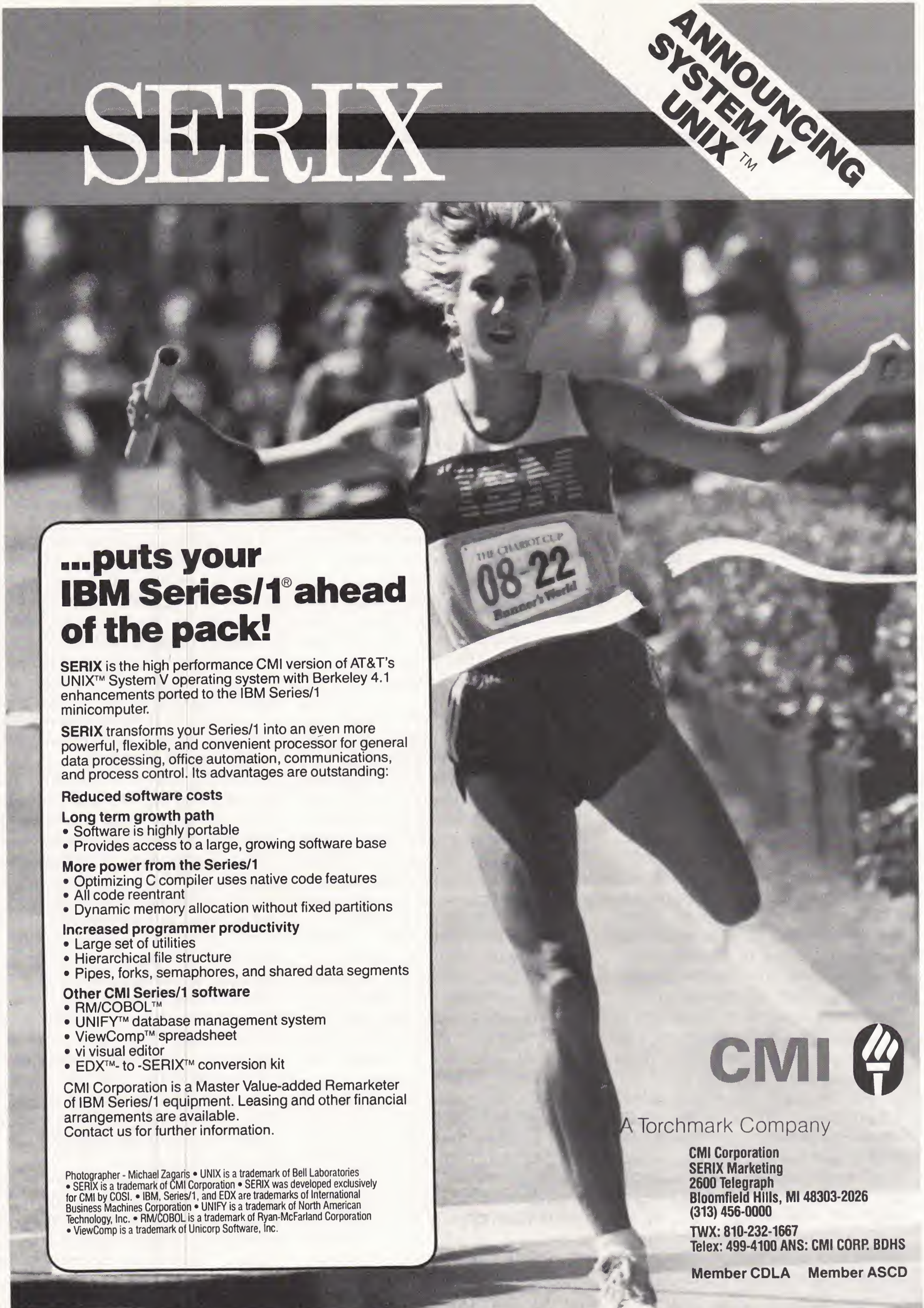
Parallels between the position taken by the Hollywood copyright owners and the manufacturers of computer software seem obvious. Software developers generally insist that the copies of software they provide to customers are merely being *licensed* for use, not sold. Typically such "licenses" purport to restrict the number of CPUs on which the software is to be utilized, the number of users that can access it, or both.

These limitations buck up against the realities of networking technology and the reasons networks came into existence in the first place. For instance, if one can transmit data to a CPU for processing, doesn't it follow that one would wish to initiate the work without physically having to move to the remote unit? Likewise, wouldn't someone attending a meeting on the first floor of company headquarters prefer to avoid traveling to the fifth floor to obtain information if it were possible to move the data between networked computers? Nevertheless, this sort of use of the capabilities of local area networks may violate the restrictions purportedly imposed by software licenses.

Networking technology, of course, lends itself to less innocent forms of licensing violation. If it's possible to utilize software that's licensed to a remote CPU, it's also possible to copy it onto the machine in your work area. As long as it's all in the family, what's the harm?

Plenty, most software manufacturers would say. Given the ease with which much software can be copied, it's unlikely that most people would seriously consider paying for additional copies of a program if they were free to transmit it over a network. This, indeed, would be to the economic detriment of software developers. But the user might well respond, "Why on Earth should *I* have to pay for another copy of something I've already purchased?"

If the software providers' lament sounds reminiscent of the position taken by film producers, it's still not possible at this point



Circle No. 9 on Inquiry Card

to say with certainty which of these views would prevail in a legal showdown. The laws provided by Congress are susceptible to contrary interpretation, and the courts have not yet ruled on these questions with regards to software.

LICENSE OR SALE?

Increasingly, software is distributed in sealed packaging containing a visible advisory to the effect that by opening the package, the user agrees to accept the restrictions set out in the included "agreement". It's ludicrous to pretend that trade secret protection can be preserved through use of these shrink-wrap agreements, and hardly anyone does. This illustrates a major problem with the trade secret concept: adherence to its accountability requirements becomes impossibly cumbersome in a mass-marketing situation.

As a result, software producers generally place their bets on the protection afforded by copyright. But there's a huge pitfall in copyright law that they must avoid if they hope to restrict the use made by a customer of a "licensed" copy of a computer program.

The law says that the owner of a first copy of a copyrighted work can dispose of it. That is, he or she can sell it, lend it to a friend, rent it to someone else, or do almost anything with it except make copies for distribution to others.

To appreciate the rationale for this, we have to remember that copyright law originally was intended to apply chiefly to published books. By providing that the owner of a copy of a book could lend it, give it away, resell it, or the like, Congress merely was making the law conform to what people were actually doing. And since until quite recently, making a facsimile of a book was a rather involved process, the prohibition against copying operated only as a barrier to commercial piracy.

> Congress might be receptive to carving out a special niche in the copyright system for computer software.

If the copyright law hadn't permitted the owner of a copy of a book to deal with it in these ways, public libraries as we know them could not exist. Nor could used bookstores and garage sales operate.

Copyright law no longer is limited to books. Neither is the law's provision giving an owner control over his or her copy of a copyrighted work. This is why the burgeoning video rental industry has been able to come into existence. The proprietors of video emporia, having purchased copies of video cassettes, legally may rent them out as they please.

This is the spectre that haunts the software industry. If a copy of a computer program is to be considered as having been *sold*, purchasers legally can treat it as they would a book. This means they can rent it for use by others, resell it, or even give it away.

Even more horrible from the standpoint of the copyright owner is the prospect that the purchaser of a program might even be able to sell *copies* made from it. This prospect arises because one of the few portions of the copyright law specifically dealing with computers permits the making of backup copies, and *also* permits their lease or sale in the event the right to use the original program is transferred. (The exact meaning of this whole area of copyright law has yet to be defined by the courts.)

NO ANSWERS SO FAR

Unless Congress, through additional legislation, clarifies the legal status of software copies distributed for utilization, we will have to wait for a definitive court decision in order to know the extent to which software producers can control the use of "licensed" software.

Legislation theoretically would be preferable. That's because there's a problem inherent in relying on judge-created law, since the rules handed down often are strongly influenced by the facts peculiar to the particular lawsuit before the court. If the facts are extremely one-sided, the law may go heavily in that direction without the opposing point of view ever really having a chance.

Still, legislative action may not be forthcoming. As the underlying conflict of interest between software providers and users becomes apparent, Congress—judging by its past reluctance to act as arbitrator in similar quarrels—may be unwilling to intervene. But if a consensus could be reached, Congress might be receptive to carving out a special niche in the copyright system for computer software. It has done this for other industries—an obvious example would be the "jukebox" provisions that were added to copyright law—and it recently has done so for the computer industry by creating a special category for microchips.

However, if we assume that there will be no assistance from Congress, the problems emerging from local area networking technology have the potential for bringing matters to a head.

GUESSING THE COURTS

Predicting what courts will or won't do is at least as hazardous as guessing next weekend's weather. Of course it is fairly safe to prognosticate that Rio won't get a blizzard, and that Nome won't get a week of 110 in the shade. Within such extremes, though, we can have only general expectations.

Contrary to what a number of people think, judges are not fond of creating absurd distinctions. So if a case should arise involving the peddling of software from a rack next to the checkout counter at the supermarket, it doesn't seem likely that a court will take seriously the manufacturer's claim that it was only granting a license, not selling a copy. Clearly, the software industry would not regard these as good facts for a landmark legal decision.

Even with a less extreme factual situation to work with, it's anybody's guess how the courts would handle the licensing restrictions typical in the industry. There's really no precedent, but then there's no precedent for computers, either.

It's possible that the courts, as they often do, may undertake to cut the baby in half. Judges generally are loath to smother emergent technologies by saddling them with oppressive restrictions. And it's difficult to imagine that the courts would not yield to the practical realities of such innovations as networking—at least to some degree.

So the courts conceivably might uphold the licensing concept, but void its more onerous restrictions. This approach will ensure that nobody will be satisfied.

Predicting what courts will or won't do is at least as hazardous as guessing next weekend's weather.

The courts might, for instance, zero in on whether a particular practice can be said to confer a significant commercial advantage on the user (in which case he or she pays up) or instead is merely a matter of office convenience. If a legal test resembling this should be adopted, the courts then might set aside limitations on the number of terminals or users able to access a program, but enforce restrictions that prohibit transfer of software to an additional CPU.

But then, nobody really knows. In the absence of industry agreement, the only thing that can be said with relative certainty is that somewhere down the line, a few lucky lawyers are going to make a big wad of money representing the litigants in that first major lawsuit.

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

LOOKING FOR THE PROMISE OF UNIX™ ?

YOU JUST FOUND IT.

UNIX promises to let all sorts of people do all kinds of things quickly, flexibly, and efficiently. For three years BASIS has been delivering what Unix promises to individuals, small businesses, large corporations, government agencies, universities, and resellers all over the nation. We sell:

- complete Unix computer systems
- carefully selected software
- expert consulting
- a full range of timesharing services

Our reputation for providing outstanding customer support is unparalleled. Call or write for more information.

BASIS
SPECIALISTS IN UNIX COMPUTING

1700 Shattuck Avenue Berkeley, California 94709 415 841 1800

UNIX is a trademark of AT&T Bell Laboratories

See us at UNIX Systems Expo, San Francisco, Booth #210

Circle No. 29 on Inquiry Card

DEVIL'S ADVOCATE

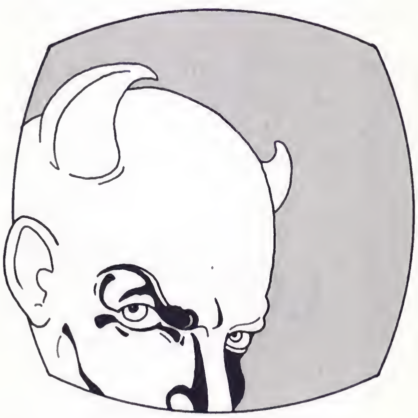
Many happy returns

by Stan Kelly-Bootle

Was it not Dylan S. Eliot, the Taxpayer's Bard, who warned us: "Do not go, April, into that cruel month"? Despite his warnings, millions of wage-earning hedonists, referred to by politicians as "Fellow Americans", rush each year, at the drop of an Easter Bonnet, to complete a self-taxing but fun-filled exercise known as "The Filing of the Return".

What was formerly a painful annual chore imposed by a sadistic IRS has become a welcome opportunity for loyal citizens to hone their data processing skills, reduce the budget deficit, trim the National Debt, and flush out those uncomfortable, guilty lumps from beneath the mattress. Just as the medical profession recently rediscovered the efficacy of the leech, so are psychosocial-econometricians now convinced that the regular bleeding of our surplus dollars is vital to the health of our body fiscal — "*interim sanum in portfolio sano!*" The old alienation between "us" and "them" is thus melting away under the Freedom of Information Network.

We now know precisely what they do with our individual contributions. Your personalized nametag could well appear on a Navy hammer-shaft, in the identifier list of a DoD Ada module or on the hinge of an Air Force toilet seat (unless, of course, you've ticked



the box saying, "I want my contribution to remain anonymous."). The interflow of money and government services holds the nation together as surely as gluon binds the modern atom.

But of more relevance to this column, a major reason for this dramatic change in our attitude to "Filing the Return" is that both Taxer and Taxee are now computer literate. This spread of computer literacy, although gained at the expense of literacy in general, is reflected in the RPG-styled worksheets that have replaced the traditional IRS tax forms. The underlying language, formulated to appeal equally to structured and misguided form-fillers, is a powerful blend of C, BASIC, Algol 68, and pedagese, and is known simply as T.

The documentation for T is a model of sublime clarity. After struggling through UNIX manuals through the rest of the year, it is indeed a pleasure each April to turn to such precise, reader-friendly material as *T by Example* (formerly called *IRS Publication 334 (Rev. Nov. 84): Tax Guide for Small Business*).

T is a strongly-typed language, but it has only three types to offer:

FP$	(floating point dollar)
INT$	(integer dollar)
FSF	(filing status flag)

Some character strings are allowed for **NAME ID** but these are immediately transformed into integer values.

There is also a user-defined, binary-valued **ICON** for "box-checking". Almost any non-frivolous mark can be made or not made as the mood dictates; its significance varies according to the box to be checked. Frivolous marks, said to "spoil the box", incur a fine of $500.

The "check-box" on the new 1040 that has attracted the most attention is:

Do you want an Audit?. . .

YES ☐ NO ☐

Some 90 percent of all taxpayers proudly say, "YES", but unfortunately the IRS is physically unable to meet this demand. I understand the frustration: let's say you have produced a brilliant algorithm in the new, fashionable T language, and you seek the approbation, nay, the admiration of your peers. The months drift by with no call from the auditor/referees. Self-doubt grips you. Did I miss a **goto** in the Bows and Arrows Excise Tax module (Chapter 35, page 149)? Did I *dispose* of, or merely *deplete*, my standing timber between June 22, 1984 and December 22, 1984 (Chapter 22, page 94)? In the event of errors, there's always next year.

T carries some of the spice and mystery of assembler by using line numbers as either variables or pointers to variables, according to the context. For example:

> **14. Add Lines 1 through 12. Enter here and on line 4, page 1.**

Note the natural, conversational form of the **FOR-NEXT ADD** loop and the exciting hint of VM, with its possible page faults. In my own case, for example, I was unable to locate page 1, and entered a halt-state until the Post Office opened the following morning.

The metasyntactical device <**here**>, as in "Enter here", refers to the cell pointed at by the value of the VPC (Virtual Program Counter), namely "14". This has caused some trouble for M68000 implementations of T since it violates the rule that relative addressing modes not be valid, alterable, data-effective destination addresses. It remains to be seen whether Motorola will recall all M68000 chips for a free microcode fix.

Conditional statements in T also have a gentle, informal surface syntax:

> **31. If the total amount of lines 27 and 29 is larger than line 28, enter AMOUNT OWED.**

The M68020 is strongly recommended for dynamically resolving these odd-even address conflicts on the data bus.

T compilers here will identify the *local* variable, **AMOUNT OWED**, with the cell at address 31. The variable **AMOUNT OWED** appears in many procedures where it is patriotically maximized by reiteration. The greatest of all these values is then sent to the check-writing module.

Great care is needed if some variables have been declared as **INT$** (accepting the IRS invitation to conserve memory by rounding to the nearest dollar) while others remain in floating point format. Remember that Schedule A has the instruction:

> **4. Multiply the amount on Form 1040, line 33, by 5% (.05).**

which may require the temporary storage of four decimal places.

An obvious joy behind the new algorithmic approach to taxation is the simple and generous rule governing investment tax credits and expense deductions for using the T software. For T compilers bought before August 31 and residing on discs not exceeding 3¼ inches in diameter (provided that the track density/sector occupancy ratio conforms with the limits set out in Table 89.3 Rev 4), the loading time or 50 msecs (whichever is smaller)—excluding any concurrent processes that relate to personal use—can be added to compilation time for the purpose of estimating your total deductible computer usage. The Bellsoft TaxLog program, in fact, will figure all this out for you automatically.

The success of the new IRS regime proves that if you treat people as honest hackers, their response will amaze you.

The latest Coding Reduction Act calls for a simplified four-line schedule, written in *tiny-t*, to replace all current variations. The new schedule is shown in Figure 1.

Beta-site testing of tiny-t has revealed a tiny bug in the schedule, but the general feeling is that, after all the boring consistency of C and UNIX, programmers will welcome a language with a spark of personality.

Stan Kelly-Bootle has diluted his computer career by writing contemptuous folk songs for Judy Collins ("In My Life," Elektra K42009), The Dubliners and others. He is currently writing, with Bob Fowler, "The 68000 Primer" for the Waite Group, to be published by Howard W. Sams in the Spring of 1985. ■

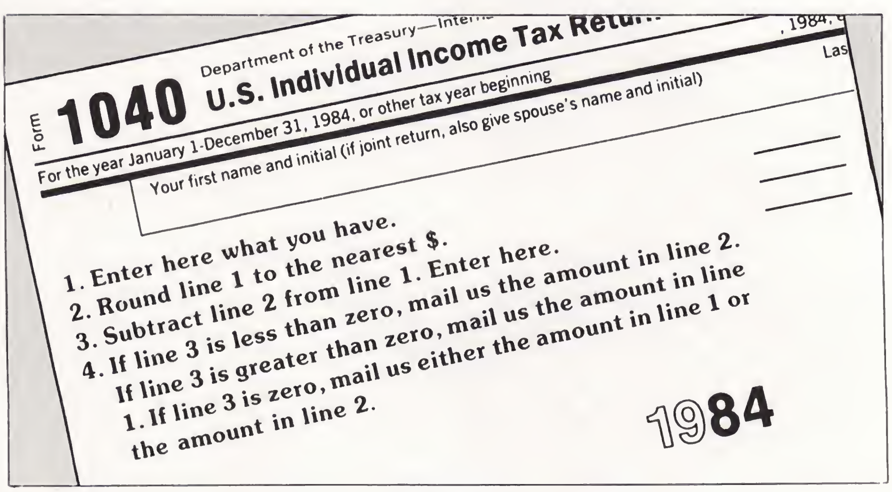


Figure 1 — *Shades of the future. For the IRS, simplicity is bliss.*

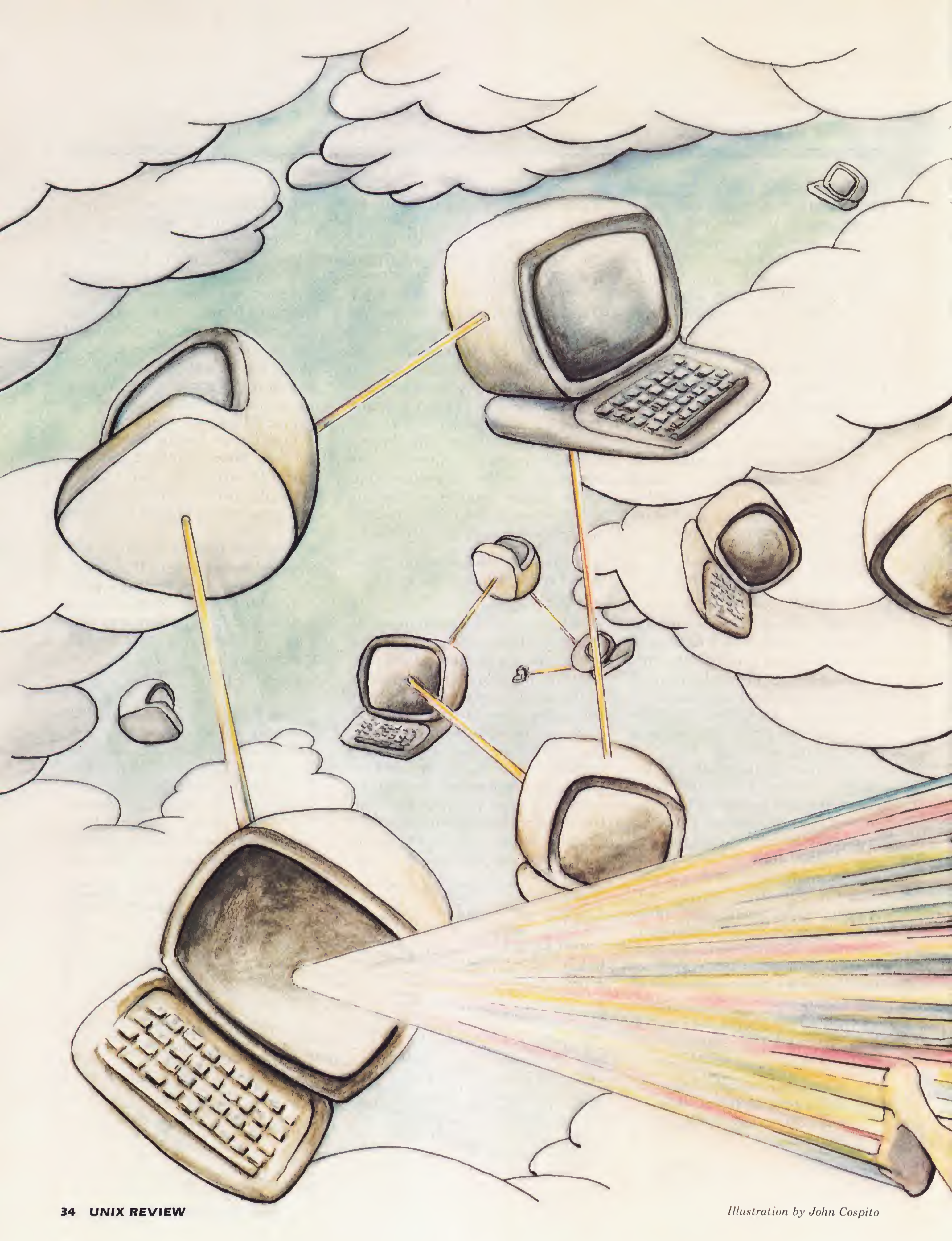


Illustration by John Cospito

THE ART OF PROTOCOL MESSAGE EXCHANGE

Policy decisions that drive protocol software technology

by Steve Holmgren

Each software protocol implementation calls for a unique set of decisions. These decisions are not part of the protocol specification itself, but they have a profound effect on the size and performance of the protocol implementation. Beyond affecting the quality of the resulting system, some also affect system evolution.

The range of implementation-based considerations can be broken down into four major categories: 1) how to deal with protocol layering, 2) how to deal with service interfaces, 3) how the protocol should use the supporting execution environment, and 4) how to implement the protocol itself.

The important thing to understand from the outset is that a protocol reference model is *only* a design tool. In implementing protocols, clean protocol boundaries can offer both advantages and disadvantages. One obvious advantage is that clean boundaries are more amenable to protocol modifications. This may be important since competing protocol architectures imply transitions. The other side of the coin is that, in implementing an entire protocol suite, performance can be enhanced by taking advantage of known interactions. Since the advantage of implementing a heterogeneous protocol suite lies in the assurance that communications will be able to occur across system boundaries, an optimal implementation of a particular protocol family may be more important than providing the "hooks" to facilitate a protocol family change. Moreover, transition from one protocol family to another is not likely to occur in stepwise fashion.

Note also that when making layering decisions, lower layers pose different problems than higher layers. Higher protocol layers can usually be treated simply as applications. Lower transport level protocols, though, hold the key to system design. At the transport level and below, a strong communications base must be formed. Protocol implementations at this level can make or break the system. It's particularly important that these layers be isolated since the underlying communications media and access mechanisms can and will change with technological advances and emerging services.

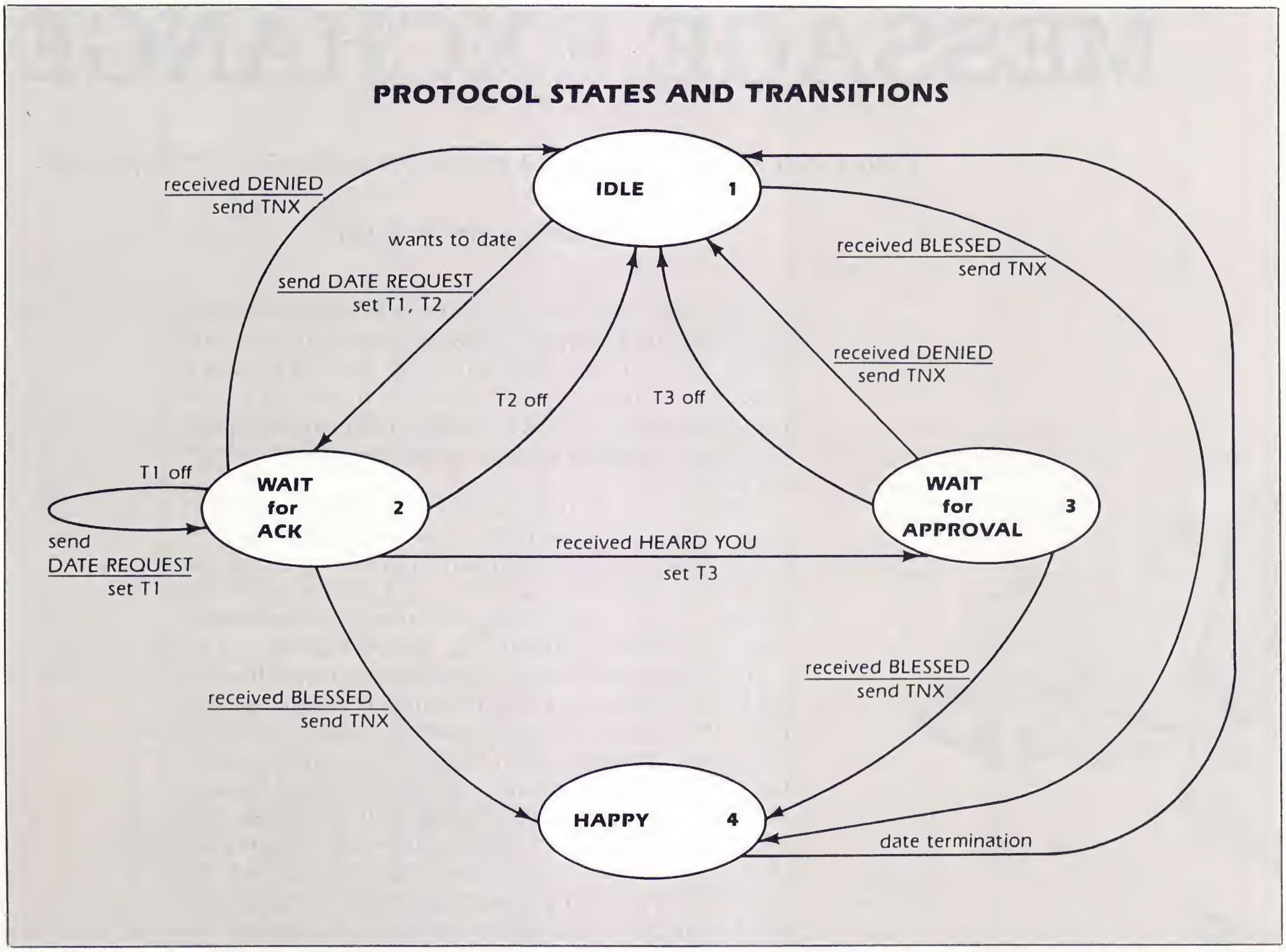


Figure 1 — *The states and events that cause transition within the dating protocol.*

The interfaces a protocol implementation offers to the higher level user and the lower level network are equally important. These interfaces play a critical role in determining how useful an implementation will be and how well its data will perform. Because of this, a concentrated interface design is a requirement of a good implementation. The variety of interfacing strategies, though, make for a complex topic that is somewhat beyond the scope of this article.

Instead, the emphasis here will be on the core protocol implementation issues that effect the execution environment and the protocol mechanisms themselves. This discussion is framed in the context of a simple dating protocol example (as excerpted from a paper presented by Danny Cohen at the Fourth Berkeley Conference on Distributed Data Management and Computer Networks, San Francisco, August 28-30, 1979).

In the hurlyburly of Silicon Valley, dating arrangements are often neglected until the last minute. Thus, to simplify the search for partners and avoid the traffic snarls associated with hopping between local social establishments, a dating center was recently established. Its operation is a fairly straightforward one: when individual X is interested in a date, a request is sent to the center for a date with individual Y. The date is usually blessed unless it is found to be in conflict with center policy. Typical policy variants include a mismatch in preferences for operating systems or differing views on programming languages called by a single letter. Assuming, though, that the parties are found to be compatible (that is, if their most serious disagreement proves to be over the hidden meanings of the phrase, "a user friendly command handler"), a meeting time is assigned and letters are sent to X and Y notifying them of the upcoming event.

PROTOCOL DATA UNITS

The dating protocol uses the following packet types and exchanges:

```
x:  <date request>, x, y, xid           ->  c
c:  <heard you>, xid                    ->  x
c:  <blessed>, x, y, time, xid, cid     ->  x
c:  <blessed>, x, y, time, xid, cid     ->  y
x:  <thanks>, cid                       ->  c
y:  <thanks>, cid                       ->  c
```

In the case that a policy variant is detected, a denial message is sent to X:

```
c:  <denied>, xid, cid                   ->  x
x:  <thanks>, cid                        ->  c
```

The different states and events that cause transition to other states are shown in Figure 1.

The first goal in any protocol implementation is to correctly follow the prescribed course message exchange. Typically, this is where naive protocol implementors bathe themselves in surface level simplicity and underestimate the job before them. Symptomatic of this is not only the tendency to overlook the importance of correctly following a set of message exchanges, but also to shrug off the accepted fashion of using the message exchange process.

In the same sense that it is possible to use the dating protocol correctly and yet be an abysmal social failure, it is possible to correctly implement a protocol specification and yet fail to provide the necessary execution environment for the protocol application.

Half the battle in implementing a protocol lies not in writing the implementation itself, but in mating it to the protocol processing environment. This is achieved through the art of protocol implementation. The success or failure of the policy choices made in this process will determine the success of the implementation as a whole. This article describes a few common strategies for protocol implementation, and then relates the art of policy decisions to the science of protocol message exchange.

ALTERNATE IMPLEMENTATION APPROACHES

Before embarking on a lofty discussion of the interface between art and science, let's first examine the scientific component of protocol implementation.

Even though there are any number of ways to implement a particular protocol, it's surprising how often one of two approaches is selected. The first, which is usually chosen by people doing their first implementations, is to use a set of IF-THEN-ELSE conditions following a protocol state diagram. The second alternative is a finite state machine approach. Both approaches can effectively carry out the task of supporting protocol control and data exchanges, but—as you may have guessed—both also have their drawbacks.

THE IF-THEN APPROACH

The IF-THEN approach offers a natural way to implement protocol software. Under it, each state and event possibility is exhaustively tested as an explicit IF-THEN-ELSE sequence. A partial implementation of the dating protocol with the IF-THEN approach is represented in Figure 2.

Popular variants of this approach have large IF statements containing a particular state that precede a series of IF-THEN-ELSE statements that in turn decode a particular event. A more progressive variant uses a "switch" statement to decode the state before using "sub-switch" statements within a case to decode the event.

The IF-THEN approach has much that makes it appealing. It's very easy to generate and understand, and it leaves no mystery about what is supposed to occur once the circumstances of an event are known. The concern here, however, is not to make life easy for programmers, but rather to create

```
If( state == IDLE && event == WANTSDATE )
{
        send date request
        state = WAITFORACK;
}
else
if( state == WAITFORACK && event == RECVHEARDYOU )
{
        set timer t3
        state = WAITFORAPPROVAL;
}
else
if( state == WAITFORACK && event == TIMER1 )
        retransmit date request
else
if( state == WAITFORACK && event == DENIED )
{
        send TNX
        state = IDLE
}
else
if( state == WAITFORACK && event == BLESSED )
{
        send TNX
        state = HAPPY;
}
```

Figure 2 — *A partial implementation of the dating protocol using the IF-THEN approach.*

```
Dating Machine

dodate()
{
        while(1)
                (*actions[ getstate() ][ getevent() ])();
}

int (*actions[][])() =
{
/* IDLE */      &p1, &p2, &nullact, &nullact, &nullact,
/* WACK */      &p3, &p4, &p5, &p6, &p7,
/* WAPP */      &p8, &p8, &p10, &nullact, &nullact,
/* HAPP */      &p11, &nullact, &nullact, &nullact, &nullact
};

p1()
{
        send DATE Request
        set T1, T2
        state = WAITFORACK;
}

p2()
{
        send TNX
        state = HAPPY;
}

p3()
{
        send TNX
        state = IDLE;
}

p4()
{
        send DATE Request







        set t1
}

p5()
{
        send TNX
        state = HAPPY;
}
```

Figure 3 — *A partial implementation of the dating protocol using the finite state approach.*

an environment in which the program can run and where access to the resources required to execute that program can be facilitated.

The principal difficulty with the IF-THEN approach is that most of the machine language generated for such an implementation simply evaluates and branches the IF expressions. Relatively speaking, very few program statements are actually spent on the execution of protocol action.

A less obvious difficulty with such implementations lies in the symptomatic probability that little attention has been given to the operational policy choices programmers must make when boundary conditions rudely greet them in the execution environment. The longterm effect of this is that such software has a way of maturing badly as programmers out in the field hack quick fixes to it. It would be far better if boundary conditions received more attention on the development bench, where time and programmer ego are available for structural improvements as well as bug fixes.

THE FINITE STATE APPROACH

The finite state approach has a distinct data orientation, focusing on events, states, and actions. In this, it differs from the programming language orientation of the IF-THEN approach. The finite state approach takes a current state and event, and produces an action and a new state. The implied software architecture with such a model boils down to an array of subroutines indexed by current states and event types. The resulting implementation is a series of small action subroutines and a larger matrix of subroutine addresses.

A partial implementation of a finite state is shown in Figure 3. The natural benefit of this approach, of course, is that little time needs to be spent in analyzing the current state or what to do with it. The current state and event are evaluated a single time and the proper action is immediately dispatched. When compared with the machine code that must be generated for the IF-THEN approach and the number of IF-THEN expressions that might need to be evaluated simply to determine the current state and event match, the benefits of the finite state approach become relatively clear.

The chief drawback of the finite state approach is that within a very short time, it becomes almost totally unfathomable—even to the original programmer. The improvement won by doing a straight index to obtain the required action has the cost of losing the context in which each of the smaller action routines is executed. While this approach is sophisticated in its use of environment, it suffers over the long haul since very few of those who inherit

Circle No. 62 on Inquiry Card ►

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

GOULD
Electronics

the original programmer's code will take the time to understand and maintain it, opting instead to replace it with an "invented here" version.

The finite state approach is also plagued by the nagging concern that the original programmer really *did* maximize the efficiency of the original implementation and either got very "tricky" in a coding style or simply lost it altogether and failed to implement required protocol actions. This fear arises from a perception that the loss of context not only impedes the understanding of others but hampers the original programmer's efforts to correctly code and debug a product.

IMPLEMENTATION BALANCE

The best state of affairs seems to be a compromise between the IF-ELSE and finite state approaches. A hybrid that does as effective a job as the finite state approach in dispatching event processing while maintaining context within source, as with IF-ELSE, would seem to be the ideal solution. The difficulty with the hybrid approach, though, is that again it lures the programmer away from hard, fixed responses to problems with policy choices that allow artistic impulses to creep into the development process.

The typical split between the IF-THEN approach and the finite state approach is to use the IF-THEN concept for processing different event types, and employ the finite state approach for handling the current state processing within each event type.

The following shows a partial implementation of the dating protocol with the hybrid approach:

```
if( event == HEARDYOU )
{
        state = WAITFORAPPROVAL;
        set t3
}
else
if( event == BLESSED )
{
        send TNX
        state = HAPPY;
}
else
if( event == DENIED )
{
        send TNX
        state = IDLE;
}
```

In the dating example, explicit event processing was included within the IF-THEN. More complicated protocols would use "switch" or finite state arrays of subroutine addresses indexed by the current state.

The importance of the hybrid approach is that context is maintained by using the IF-THEN event processing while program code efficiency is generally preserved by the use of finite state subprocessing.

OPERATIONAL CONSIDERATIONS

Given the hybrid approach, it is possible to construct decent software capable of implementing the correct exchange of protocol messages. Difficulties arise, though, when it comes to integrating the completed protocol processor into its operating environment.

Principal among these integration concerns is the availability and management of data buffering space in the execution environment. While this concern varies from system to system and little can be said of general applicability, particular care should be taken to handle no-buffer situations. Surprisingly, one of the largest systems level problems with protocol implementations results from data flow locking up or buffer space running out. In protocols that have packet sequencing, buffer starvation can be gracefully handled by artificially treating a packet as "out of sequence" when a buffer is unavailable.

> A protocol reference model is *only* a design tool.

INTELLIGENT PACKET ALTERNATIVE

Most protocols are designed around an active node/passive packet model. Figure 4 offers a graphic representation of the model. Under this design, each node has processing capabilities to account for all the protocol states and all the possible event types it might receive within a state. This leads first to large implementations that, when compounded by programmer judgments etched into the interface between protocol implementation and execution environment, can cause significant difficulties in implementing the protocol. This, in turn, can lead to difficulties in getting the protocol to interoperate with implementations built by other individuals.

An alternative protocol model can be constructed to take advantage of the understanding the originator of a protocol message has of the protocol mechanisms to be applied to each packet. This model revolves around the idea that separate protocol message encoding can be included with each packet to request the activation of distinct protocol mechanisms. For example, the encoded message, "Send a message acknowledgement; sequencing is unimportant," might be included. This sort of encoding is the signature of the passive node/smart packet model. Figure 5 provides a graphic representation of this model.

The advantages of this approach to protocol design are many-fold. We don't have the space to explore all of them here, however. Without a complete discussion of the pros and cons, suffice it to say the intelligent packet model simplifies protocol implementation judgments to the level of judgments typically required in implementing a programming language.

PERFORMANCE

The metrics used to evaluate protocol implementations take into account a number of levels of performance. At minimum, one should be able to establish that an implementation correctly installs specified message exchanges. Deeper evaluation considers data transfer performance and implementation size. Experience in tracking performance has shown that protocol software rarely behaves internally as designed. There is almost always some surprise about the number of times a subroutine must be executed or the amount of time an area of code takes to complete. Thus, it is strongly recommended that a second pass be made over a "correct" implementation in order to discover areas of concentrated execution. These then must be improved.

The adage that 10 percent of the software is executed 90 percent of the time well applies to protocol implementations. Finding the 10 percent and improving it can typically lead to a doubling of the data transfer performance.

Protocol implementation at first appears to be very straightforward. Indeed, it is almost seductive. What is usually perceived on first glance is simply the need to exchange messages with another implementation in a very structured fashion. What is usually missed is the need to integrate a correct implementation into an execution environment and to handle the boundary conditions that exist within that environment. This is especially true in the area of data buffer memory allocation and management. As a result, sophisticated systems level judgment is called for in the integration process.

Performance, of course, is the real measure of an implementation. For this purpose, "performance" is defined not only in terms of correct message exchange but, more importantly, in terms of the impact the implementation has on its execution environment and its data transfer performance. Significant performance improvements can be found in as little as 10 percent of the software. Take the time to find and improve that code.

Finally, pay attention to program structure. Efforts spent in this area during development will be conducive to the longterm life of an implementation. Generally speaking, by providing a traditional IF-THEN framework for a finite state-based sub-structure, you can go a long ways toward improving a program structure's life expectancy.

Steve Holmgren is President of Communication Machinery Corporation of Santa Barbara, CA, a producer of high performance communication software and hardware. Prior to coming to CMC, Mr. Holmgren served as President of QMI, where he developed a single-chip TCP implementation. He holds a Bachelor's degree in Mathematics and Computer Science from the University of Illinois in Champaign-Urbana, where he went on to interface UNIX to the Arpanet at the Center for Advanced Computation. Mr. Holmgren has also done advanced technical assessments and prototyping for military procurements through the Mitre Corporation. ■

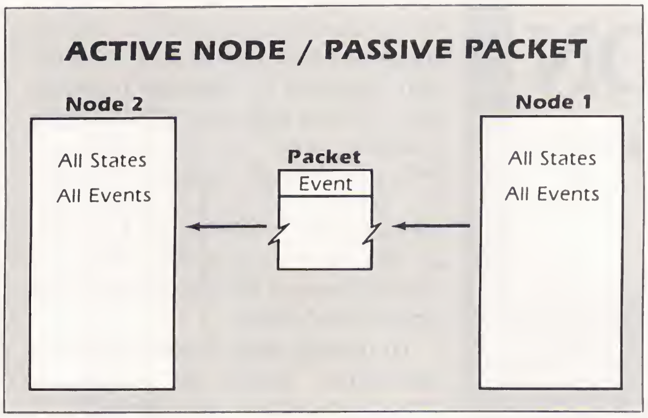


Figure 4 — *The active node/passive packet model.*

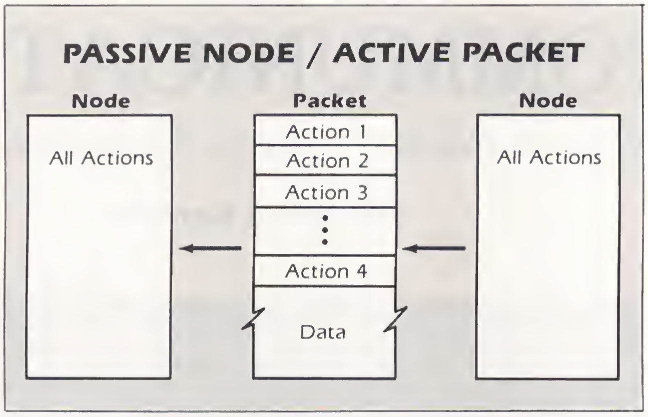


Figure 5 — *The passive node/smart packet model.*

LINES OF COMMUNICATION

A look at trends in network hardware technology

by Bruce Borden

Illustration by Victor von Beck

Over the past 15 years, network hardware technology has kept pace with the flurry of changes in digital and computer technology. Today, transmission rates of up to 50 million bits per second are common, with error rates below 1 in 10^{10} bits.

As transmission rates have continued to increase, other layers of network implementation have become bottlenecks. Complex protocols and poor system architectures often waste bandwidth, thus undercutting any gains realized by using costly high speed media.

To see why careful architectural design and improved protocol performance is essential if higher bandwidth media is to be of any use, let's take a closer look at the changing technology of network hardware.

PERFORMANCE

Fifteen years ago, a continuous megabit per second (mbps) feed to a peripheral device was considered a fast transfer rate. Today, an 8 mbps transfer rate between computers over a local area network is generally assumed to be the "state of the art".

The HYPERchannel, though, has been transferring 50 mbps between computers for years. Why shouldn't that be considered the "state of the art"?

Clearly, a definition is in order. What exactly is a "transfer rate"? The Ethernet has the physical capacity to transfer data at 10 mbps, but the actual end-to-end transfer rate is typically much less than 1 mbps. This should serve as a warning that the media transfer rate is *not* the end-to-end, memory-to-memory transfer rate. This is not to say that media transfer rates are unimportant, because they *do* indicate the upper limit on the end-to-end transfer rate. But it also should be noted that media transfer rates are usually *much* higher than actual end-to-end rates.

In talking about networks, it is generally useful to think of "transfer rate" as referring to end-to-end, host memory-to-host memory throughput. For example, which has the highest transfer rate—a 10 mbps proteon ring, a 10 mbps Ethernet, or a 3 mbps Ethernet? Running IP/TCP protocols on a VAX/780 under 4.2BSD, all three networks actually perform equally, at about .5 to .75 mbps.

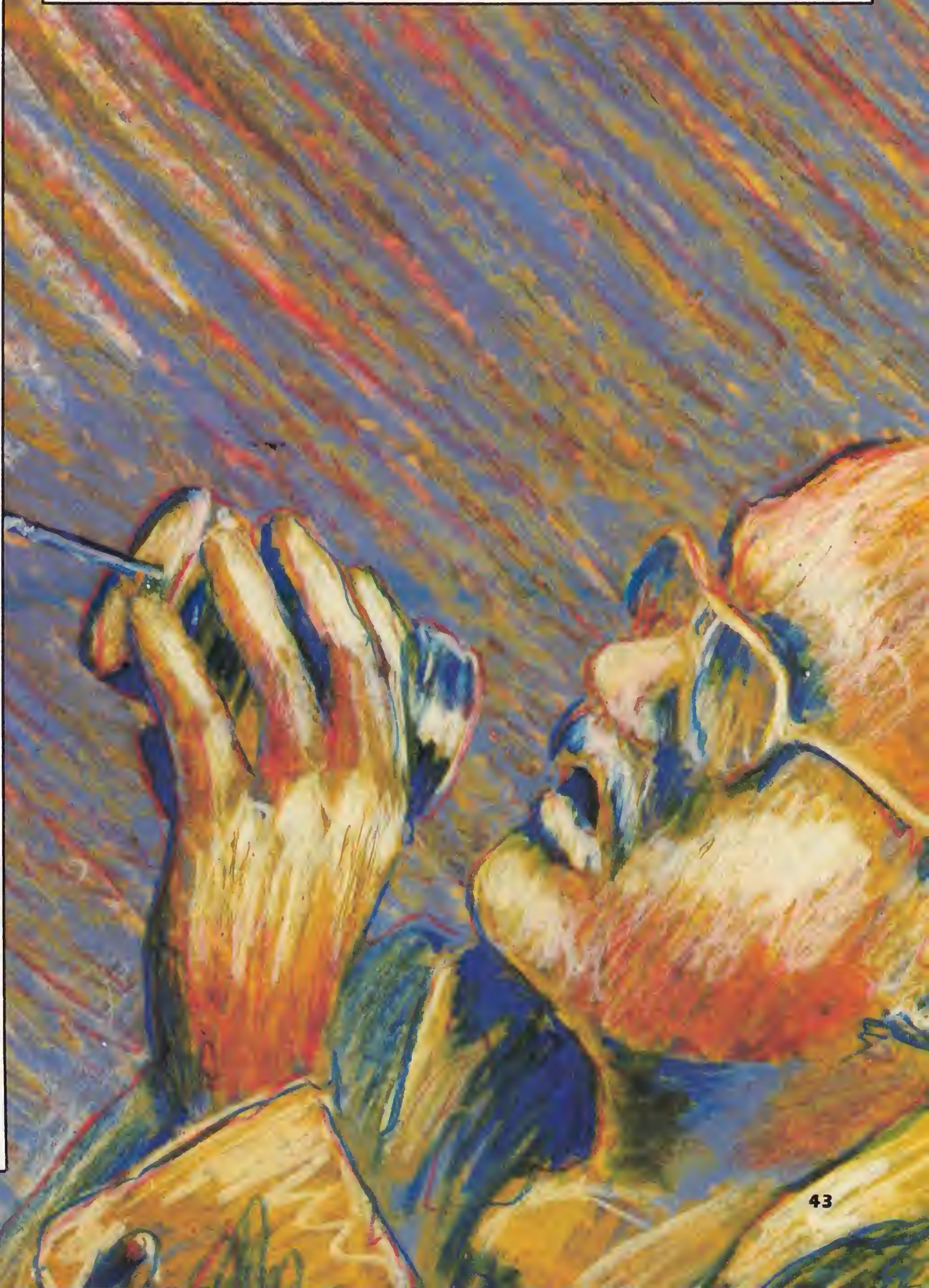
Why so slow? The end-to-end time to move data from one computer's memory through some protocol code, a network controller, a network, another network controller, and some more protocol code before finally placing it into another computer's memory is much greater than the transmission media time.

Even if you have a local, high speed disk drive that can transfer data at 15 mbps (such as a Fujitsu Eagle), you can be assured that your file system is doing a superb job if it can use 4 mbps (1/4) to 6 mbps (1/3) of the available bandwidth. Yet, disk interfaces are far less complex and require less overhead than the typical communications protocol.

One need only look at the seven layers of the International Standards Organization's Open Systems Interconnection model to gain an appreciation for this complexity. [See the article entitled "The Potpourri of Networks" elsewhere in this issue for a more detailed look at the model.] Most communications protocols follow at least a few of the seven layers.

For example, consider a simple network application such as file transfer, using the IP/TCP protocols over an Ethernet. Once a connection has been established between the file transfer program on one end and the file transfer server on the other, the file transfer application program (layer 7) calls a network write routine (layer 6), which passes the data to TCP (layer 4), which adds a header to the data and passes it to IP (layer 3), which adds its own header and passes it to a device driver, which hands it to the Ethernet hardware (layers 2 & 1) for transmission. At the other end, the whole process

There are several ways that hardware and systems architecture can improve end-to-end throughput—*and* reduce CPU overhead at the same time.



is reversed. (Note, standard IP/TCP implementations do not make use of an identifiable session layer.)

Is it any wonder that time is lost in the transition?

At the interface between the presentation and transport layer, the data is usually copied (and a copy or pointer to the data is typically squirreled away for retransmission later in case of lost or damaged data). The data may be copied yet another time at the network layer—and there's still more to come. Most Ethernet hardware copies it a third time from host memory to on-board controller buffer memory. This last copy may be via DMA or a host processor copy loop.

Once the packet is in the controller, it tends to go out on the Ethernet almost immediately. On the other end, this whole process is repeated in reverse order.

A standard IP/TCP packet on the Ethernet will be about 1024 bytes long. At 10 mbps, it takes only a single millisecond to transmit each packet. If the protocol processing and data copying operations (including the device DMA) take 4 milliseconds on each end, then the fastest that data can be moved from one computer to another is about one packet every 10 milliseconds, or about one hundred 1K packets per second. If the protocol implementation is well designed, most of the CPU time on the sender and receiver can be overlapped, thus improving throughput by a factor of two.

Just think about it. Let's say you have a pair of VAX-11/780s running 4.2BSD—a fair amount of computing power, right? But to copy a file from one to the other using IP/TCP over an Ethernet, .12 mbps may be the best you can achieve using *all of the CPU cycles on both systems*! What a waste of CPU! What a terrible state of affairs for the system developer to tackle!

AVAILABLE RELIEF

There are several ways that hardware and systems architecture can improve end-to-end throughput—*and* reduce CPU overhead at the same time. One proven strategy is to include more hardware support—in the form of "co-processor" or "front-end" protocol implementations—for what is normally considered software protocol processing.

For an example of hardware support, look at the Ethernet. Most Ethernet controller "hardware" implements the link layer protocol. This reduces the CPU time required to use the Ethernet. There are also several new controller boards for the Ethernet that contain microprocessors and on-board implementations of IP/TCP protocols, meaning they can take care of functions up through layer four of the OSI model.

As transmission rates have continued to increase, other layers of network implementation have become bottlenecks.

With more of the protocol processing being pushed down into silicon (new custom chips), and the rest being done by high-performance microcomputers on front-end processors, where are networks headed?

It is critical to network performance that front-end processors have protocol compute performance equal to or greater than the processor(s) they are serving. For example, even a well-implemented front-end controller based on the Intel 80186 or the Motorola 68000 will probably be unable to achieve anything greater than 1 mbps IP/TCP throughput *unless* special hardware aids are also implemented. These might include scatter/gather DMA support (both to the host and the network) and hardware checksum support. Once the Motorola 68020 is available for service as a front-end processor, it should offer performance well-matched to the Ethernet and current protocols.

Before custom chips became available for Ethernet, though, these new front-end processors were not possible. Discrete Ethernet implementations took up too much board space and were quite expensive—even without their own processors. Now that the chips are available, though, they or custom ALUs will perform more and more future protocol processing.

What else might we expect to see in the networks of the future? A look at the media chart shown in Figure 1 should offer some clues.

Looking at these media comparisons, it's obvious that we are far from reaching the transmission speed limits of available media. Satellites, microwaves, and fiber-optics all promise bandwidths much greater than those realized by current network implementations. Imagine a fiber-optics network that can operate at 3000 mbps! By way of comparison, consider that the VME bus (a common small computer backplane bus) can only move data at 320 mbps. How could such a bus ever make use of a 3000 mbps network? It can't, of course; but on a network with hundreds of nodes, it might reasonably expect to move 80 to 160 mbps of data from place to place. With transfer rates like these, truly distributed computing would become available.

The challenge for networking hardware will be to produce very high speed custom circuits to move data *reliably* from node-to-node at hundreds of mbps. This will be done by integrating as much high-level protocol design as possible into silicon, by utilizing wide data paths to the host (32+ bits), and by totally offloading the host CPU of all protocol-related processing.

The host's network model needs to look like any high speed peripheral interface. No programmer would think of writing a 200 inch-per-second 6250 GCR tape drive in 512 byte output requests, yet most current protocol implementations *limit* block size to 512 or 1024 bytes per record. You can look for this block size to increase at least 100 times, both at the front-end interface level and on the physical media itself.

FUTURE TOPOLOGY

The choice of whether to use a bus or a ring is made less by rational analysis than by emotional preference. It's very much like the choice between Coke and Pepsi, or **vi** and **emacs**. Although data may be available to help with these decisions, most choices will still be based on personal preferences. Since buses and rings both work, system architects can choose between them based on the hardware realities of the network media and implementation costs.

Looking at the media comparison chart, note that all of the highest speed media lend themselves to point-to-point and ring topologies. For this reason, it's likely that rings will come to dominate local networks over the next five years.

For a number of applications, though, expense is more important than the actual data transfer rate. To support, say, a collection of data query and control terminals on a manufacturing plant floor, an inexpensive low speed network that offers high reliability may suffice. For such an application, an overhead infrared link may be the best technology. It isn't fast, but terminals can be moved around freely without the bother of re-working network connections or overcoming routing problems. Using infrared, it is only necessary that each active terminal have a line-of-sight link to a central overhead master transmitter/receiver. Infrared is not susceptible to electromagnetic interference, and thus is a good choice for some environments. On the other hand, a paint plant with lots of particulate matter in the air would probably not lend itself to this type of network solution.

Another area for system architects to consider is the contention method. One of the two major options is CSMA/CD, or carrier sense multiple access with collision detection. Best known as the mechanism used by the Ethernet to arbitrate multiple user (host) access to the network media, CSMA/CD performs very well until it gets close to the saturation level of the media bandwidth, but it cannot guarantee any one user uniform access to the network. This means that although the network utilization may account for 90 percent of the media transmission rate, one host may be getting 10 percent while another receives only 5 percent.

Token passing, on the other hand, implements much "fairer" media access, providing a round-robin arbitration between all hosts wishing access to the network. The drawback it suffers from is that it makes error recov-

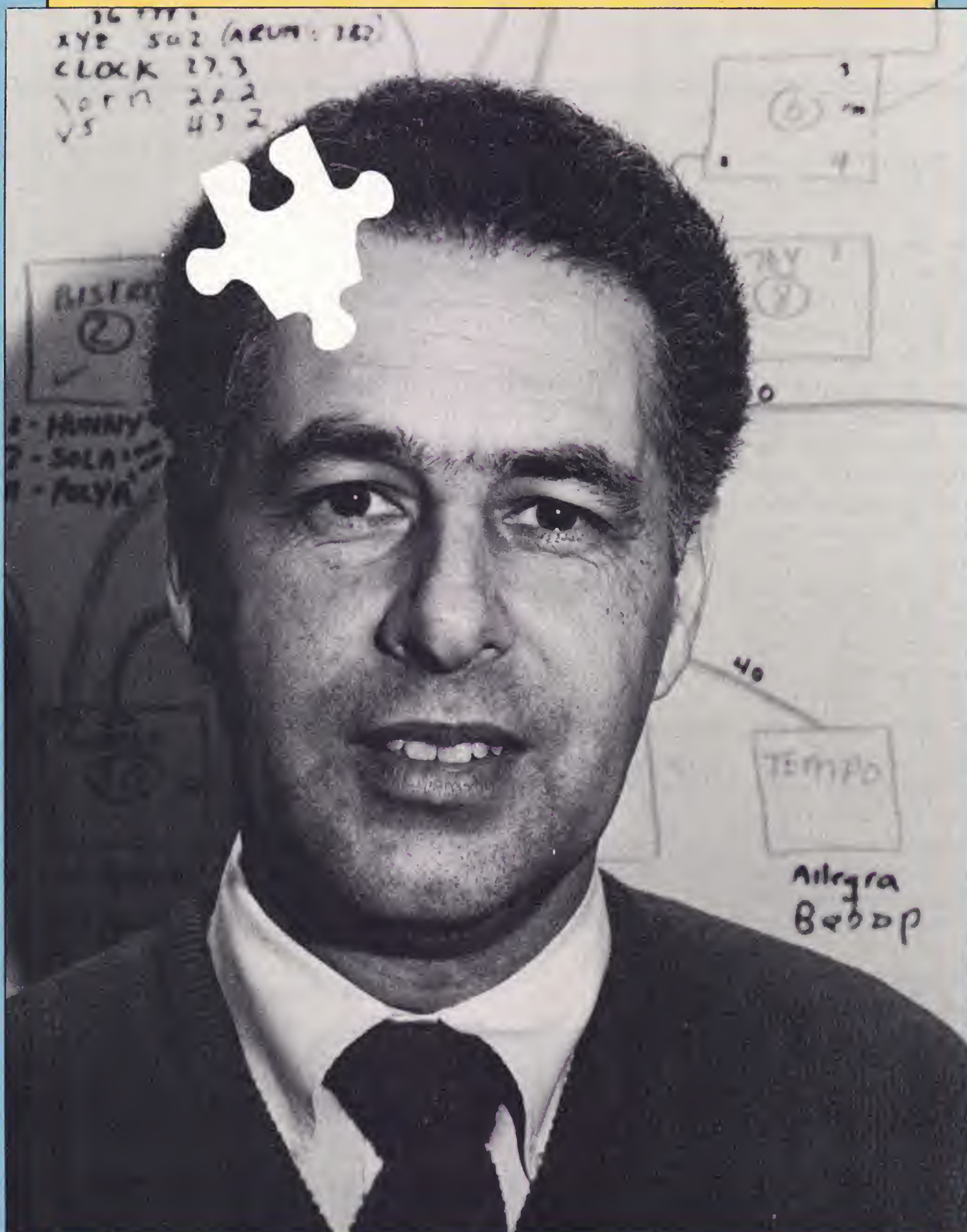
Continued to Page 102

MEDIA COMPARISON

Media	Capacity*	Topologies
FM Radio	.020	Point-to-Point, Star
Satellite	.056 274.000	Point-to-Point (T4)
Packet Radio	.100	Point-to-Point, Star
Infrared	.250	Point-to-Point, Star
Laser	1.500	Point-to-Point
Microwave	1.544 274.000	Point-to-Point (T4)
Twisted Pair	10.000	Ring, (Bus), Point-to-Point
CATV Cable (Broadband)	20.000	Ring, Bus, Point-to-Point
Coax Cable (Baseband)	50.000	Ring, Bus, Point-to-Point
Fiber Optic	3000.000	Ring, (Bus), Point-to-Point

*in mbps

Figure 1 — *A comparison of available media.*



PIECING THE PUZZLE TOGETHER

An interview with Sandy Fraser

Photos by George Peirce/Studio 84

As director of the Computing Science Research Center at AT&T Bell Labs, Sandy Fraser isn't responsible for all of the computing science research done at Bell Labs, but he does handle a big chunk of it. Most of his own research has focused on data communications and network development, particularly as it relates to common carrier networks. The development of Datakit VCS and the Information Systems Network, AT&T's trademarked data network products, is largely the result of work Fraser has done to develop data transport technology.

To solicit his thoughts on UNIX networking, UNIX REVIEW asked Ned Peirce, a systems analyst specializing in UNIX, to interview Fraser in his Murray Hills office. Although reluctant to speak about the commercial aspects of networking and computing, Fraser spoke freely about his last decade in communications research.

REVIEW: *Let's lead off with a little history. Can you describe Datakit and its evolution?*

FRASER: I joined Bell Labs in 1969 with a belief in the opportunity provided by the emergence of computing and data communications, and the convergence of the two. At the time, very little research was devoted to packet switching and I had an interest in distributed operating systems. So to make that happen, I had to assemble a communications network.

I picked up on some work being done by Wayne Farmer and Ed Newhall, two members of the technical staff at Bell Labs. A year or so earlier they had designed what I think was the first asynchronous ring ever made. It was what is now called a *token ring*, but that piece of terminology didn't come until later. I made some changes to the design of the token ring, mostly with the intent of understanding how to integrate it with a wider area network. It seemed clear that we needed to understand how to make larger networks—certainly ones that were larger than the one they had started with.

It turned out that Bell Labs had a fairly substantial number of minicomputers at the time. Here in the research area, we had enough machines that I was able to talk about a dozen people into collaborating with me to the extent of putting their machines onto this ring. That was the beginnings of an experimental network we called *Spider*.

Spider was a heterogeneous network that included PDP-11s, Honeywell 516s, and a PDP-8, as well as a considerable mix of operating systems. At the time, I thought that the inclusion of all these different operating systems would aid insight. But, in retrospect, I can see that I introduced myself to far too many problems all at once.

Spider ring actually provided a sort of service to the research area. I guess it became operational in a service-like way around 1972 and it lasted until we killed it in 1978. But, by 1978, it had sort of withered. About the time that it became robust enough that people could get some sort of service out of it, I had learned most of what I was going to learn from it.

That's when I decided that I had pretty much done the wrong thing in building it. I was simply on the wrong technical tack.

REVIEW: *What were the problems?*

FRASER: Well, there was the generic problem that seems to afflict any local area network designed only for that purpose. I don't think computing can be regarded as strictly a local phenomenon. We at Bell Labs have installations at several places in New Jersey, Pennsylvania, and Illinois. No sooner had I gotten the ring up here than people were asking me to make extensions to computers in Holmdel and Reading, PA. Well, the ring was a 1.5 megabit ring. Its architecture assumed certain

> I was trying to do the impossible in attempting to design a switching machine that was good for everyone and yet was economical.

properties of the environment which were true inside the building but not true for a wide area network. So I simply wasn't able to make a practical extension either into Pennsylvania or to South Jersey.

REVIEW: *Did you see the ring architecture as the problem?*

FRASER: Network design is a multi-disciplinary issue; there are a lot of factors that you have to play off against one another. In starting with a rather narrow framework, like joining minicomputers within Murray Hill together, one ignores a whole class of factors important for wide area networking. You come up with a solution that works very well in that particular environment but doesn't work well in the larger scheme of things.

For example, the protocols designed for the ring were only suitable for a high speed network. But at the time, no one around here was prepared to pay for a high speed line to Holmdel. A 9.6 Kbaud line was probably about the fastest link we had. The protocol simply did not make much sense at those speeds.

REVIEW: *Too much overhead?*

FRASER: Well, the delay would have been too long. We used the network in an extremely interactive way. The protocol just was not appropriate for most transmission lines.

REVIEW: *It involved more than file transfer? Were you also running processes remotely?*

FRASER: Well, there were a number of things that we tried to do. It mostly involved file sharing, remote file access, and remote job submittal—that sort of thing.

One of our problems was that I simply had not thought about maintenance. If a network is contained entirely within an area

It seemed clear that we needed to understand how to make larger networks.

that a person easily could walk around in with an oscilloscope, maintenance is a very different business than it is when things are spread out over a long distance. We simply hadn't thought about that thoroughly enough when we set the thing up, so we failed to put in enough monitoring and maintenance to manage fault isolation and detection reasonably.

Another mistake that dawned on me about that time was made apparent by work done by some folks I was sharing a lab with at the time. They were building an early experimental time division multiplexed switch (TDM). You may not know what that is, but it's a machine that's now widely used for telephone switching. A very small unit does the real switching function and a large number of circuit cards interface to the communication lines that radiate out from the switch.

The switching unit itself is typically very small and yet it offers a very large bandwidth in comparison with packet switches. The contrast with the packet switch I had was striking. The packet switch attached to my ring had only about a megabit of throughput but it occupied a 19-inch rack that stood six feet high. The performance-to-weight ratio difference struck me as an indication that we were doing the wrong thing.

I thought, suppose we were to build a network for the United States. It is fairly clear that you would need high capacity switches to do the long haul switching. You would also need to have a much more efficient technology for switching than we were using at the time.

For those reasons, and a few more, I pretty much abandoned research on the Spider network, though it continued to provide service. I then started to work on a new network architecture in an attempt to solve the problems I'd found.

REVIEW: *Did you envision a single architecture for both local and long haul communications?*

FRASER: Well, the word "architecture" is actually ambiguous in that context. We can talk about that a little later.

I thought it would be possible to design a switching machine that would provide effective service for a wide range of computers and terminals. I tried several designs, including one for a piece of machinery that provided for packet switching entirely in hardware. But, after a couple of years, I came

Only Sperry can make the following four statements.

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

Our 4 new microcomputers run the UNIX system.

Our new minicomputer runs the UNIX system.

Our Series 1100 mainframes run the UNIX system.

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

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

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

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

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

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

©Sperry Corporation 1985.

SPERRY

Introducing an idea that makes obsolescence obsolete.

The UNIX* operating system from PC to mainframe.

Circle No. 21 on Inquiry Card

In the distant future, data and voice may well merge, but for now I don't see any particular virtue in it.

to the conclusion that that, too, was the wrong idea.

There is just too much variety in the computer business. There is a tremendous range of size and performance. We have a Cray downstairs and an AT&T PC here in the office. We have terminals costing a few hundred dollars and others costing several thousand. It seemed impractical to me to engineer efficiently for a wide range of requirements.

If you build a machine that is cost effective for switching traffic from Model 33 Teletypes—which is what we had then—then it is not going to have the performance you need for interfacing with a Cray. If you build something with the performance needed for a Cray, you'll find that you can't afford to put Model 33 Teletypes on it. I finally convinced myself that I was banging my head against a brick wall. I was trying to do the impossible in attempting to design a switching machine that was good for everyone and yet was economical.

Out of that frustration, Datakit was born. In fact, the idea came to me on an aeroplane coming back from UCLA. Somewhat earlier, I had been watching my kids playing with a Leggo set—you know, those little colored plastic blocks you put together—and it suddenly occurred to me that to get the functionality, performance, and cost effectiveness I wanted, I needed to think of the network switching machine in terms of modules like the ones in the Leggo set.

The question became: "What are the modules? What are the piece parts that you glue together?" The idea was to have some high performance piece parts that were very expensive and some real cheap, low performance piece parts. Then, if you had 25 CRTs to join together, you could use a set of low cost, low performance piece parts. If you had a couple of Crays to join, you could use more elaborate modules.

REVIEW: *Would the terminals then be able to talk to the Crays?*

FRASER: Yes, at that point I had come to believe in the idea of Universal Service in the sense that we have for telephony. In fact, the real problem with the Spider network was that we had no terminals on it. The terminals were on the computers, but not on the network. I had come to the conclusion that what society was going to need was a single network that would allow any computer and any terminal to talk to one another, irrespective of location. That was my objective, but in order to make it economically viable, I needed to come up with a modular approach so that we could configure particular pieces of the network to meet particular performance requirements. That is how the *kit* part of Datakit came about.

REVIEW: *Still data versus voice?*

FRASER: As a matter of fact, the potential for integrating voice and data in one network has been obvious to just about everybody. I think we came up with a design that could handle voice very well. However, we continued to emphasize data. The reason was simply that so much is known about telephony—how to switch it, how to manage it, and so on. So little is known about data, even today, that unless we focus on solving the data problem, we can easily get submerged in the much more thoroughly understood telephony problem.

REVIEW: *Even though telephony voice is largely data now?*

FRASER: Oh yes. You will find that among data networks, Datakit probably handles telephony better than most. It was actually designed to carry synchronous traffic of that sort. But, at the present time, there are very economic solutions to carrying voice. There are many known techniques for handling voice. It doesn't seem to make much sense to make a big deal about putting voice on the data network at this time. If you are engineering to fit an environment and a large fraction of the environment is all of one type, you can optimize for that traffic. That is what we are trying to do. I think for the short term, separate voice and data networks make much more sense. They may share transmission lines perhaps, but have separate switching systems. In the distant future, data and voice may well merge, but for now I don't see any particular virtue in it.

REVIEW: *Even though in the "office" they are trying to integrate the two?*

FRASER: Well, I am actually talking about switching machines. It is not too difficult for the transmission lines to merge the two sets of traffic.

REVIEW: *Even though you end up with two different sets of modulators and demodulators?*

FRASER: There are a variety of ways to do it. You can make multiplexors that merge the voice and data together. The switching machines and the interface units are the most complicated and least understood parts of the puzzle now.

That is where our focus is at the moment. Datakit was invented in 1975 and we have been working on and off on it ever since. In recent years, it has become a product or rather, the basis of two AT&T products [Datakit VCS and the Information Systems Network].

REVIEW: *When was it first made a product?*

FRASER: It was announced December 8, 1983.

REVIEW: *That must have given you some pleasure.*

FRASER: Yes, but it had been such a long struggle, it was sort of an anticlimax.

There is a long complicated story about that eight-year period, but we are not likely to have time to cover all of that.

Let me instead go back to your architecture issue. I am still convinced that we, as a society, need a plan that will get us to Universal Service for data at least and then eventually integrate voice as well. That plan has got to include some fairly fundamental conceptual agreements.

You understand what a telephone network is though you may not understand or care what "4 KHz" is. You know that the network has a certain capacity. You know that it doesn't carry parcels or music. Most people know what it is capable of, but they wouldn't be able to offer a very scientific description of its capability. For people who build telephone equipment, there is a very precise description of what a telephone network is capable of that involves the definition of *bandwidth* and a reference to the frequencies used and so on. We need to come to a similar definition for data. We don't have one today. We have a variety of definitions, but no single one we can agree on. We can't get Universal Service without a single definition, the pieces have to be able to talk with one another.

> There is just too much variety in the computer business. There is a tremendous range of size and performance.

REVIEW: *In point of fact, there are many definitions for the telephone network as well, depending on what a subscriber requires. For example, you can buy a conditioned high speed line or settle for a standard line.*

FRASER: You can get different qualities but the basic notion of the information being carried is now pretty much a standard worldwide. It is a 4 KHz circuit, plus or minus a little, wherever you go in the world. Because of that, we've been able to build reasonable interfaces between the various national networks. It is true that different telephones provide different facilities: some have hold buttons, some don't, and so on. But, those are supervisory functions on top of the fundamental transport capability. There is pretty much a worldwide agreement on what that capability is.

The telephone network is over 100 years old and yet it is still a flourishing business—unlike the railroads or the canals. I believe that accomplishment has a lot to do with the fact that the business is defined in a way that is independent of the technology that implements it. The telephone business would not be competitive today if we had to do it with the technology that Alexander Graham Bell came up with. Somebody would have found a way of using electronics to do something to supplant the telephone business. The fact of the matter is that the telephone business has embraced new technology. We've put satellites, fibers, and all sorts of other things into the telephone system without changing it as far as the customer is concerned. That has allowed AT&T to remain competitive by taking advantage of new technologies and economies of scale, without losing its customer base.

Imagine, for example, if in order to install a radio channel from New York to Washington, we had to get an agreement that on Friday night everyone who made

©1984 AT&T Technologies, Inc.

UNIX™ SYSTEM V. FROM AT&T. THE

There's only one UNIX System V. The one created and supported by AT&T. The one fast becoming a universal standard.

It's another reason why good business decisions are based on UNIX System V.

In more ways than one, it pays to make certain that the products you offer are based on UNIX SystemV. It is the stable foundation on which all of our future enhancements will be developed. It's here to stay because we pledge that our future software enhancements will be compatible with it.

Configure your systems now with products based on UNIX System V and you'll be at the forefront of its establishment as a worldwide standard. AT&T has expanded its role in markets in Europe, Japan, and the Far East to create opportunities for machine-independent, portable solutions to operating system needs. Around the world you can profit from our initiatives when you invest in UNIX SystemV solutions.

On the inside track

If you develop products based on UNIX System V, you'll be in on the latest advances from the custodians of the UNIX System standard. A growing number of microprocessor ports to choose from. Expanded networking capabilities. A growing number of applications packages from software vendors. And, the power of the system that spans generations of machines as well as models and sizes—from micros to mainframes.

Service from the source

Along with its various technical and commercial advantages, UNIX System V offers something the other systems can't match—

STANDARD BEARER.

the full backing of AT&T.

That includes worldwide support and training programs. Our commitment to work with computer manufacturers and resellers as well as user groups to develop standard products for the future. And for software vendors writing applications programs, a complete service and support program from AT&T.

The one and only from AT&T is UNIX System V—backed by AT&T service and support—designed to make UNIX System V the one for one and all.

For more information on UNIX System V from AT&T, send in the coupon.

UNIX System V. From AT&T.
From now on, consider it standard.

AT&T

Please send me "UNIX Software."
Mail to: AT&T, Box 967, Madison Square Station
New York, New York 10159

Name________
Title________
Department________
Company________
Address________
City________ State________ Zip________
Phone________
UNIX™ System Licensee ☐ Yes ☐ No
☐ Don't Know

Circle No. 11 on Inquiry Card

phone calls from New York to Washington would have to change the way they made them. Or what if they had to change their handsets? Given those circumstances, we would never have been able to put radio channels in the telephone network. The reason we were able to do it was simply because we understood what telephony was in a very generic way. So we designed radios to carry 4 KHz channels. As long as we did that, we could put them in the system without telling anyone, which is exactly what we did. In my belief, that understanding of telephony is the fundamental reason this company has done so well. The definition of the service has transcended the technology.

I don't know if you call that "architecture" or what, but, in my estimation, that is a key component of a successful data network.

REVIEW: *Do you have any expectations of an agreement on standards?*

FRASER: Having gotten this far, I don't think it is going to be achieved through international standards organizations. I am hopeful that will be achieved, but I think that they are going to be achieved in much the same way that UNIX has become a widely accepted standard. Standards organizations did not cause UNIX to become a standard. UNIX achieved that status because there was a reason—a need—for its existence. It technically met that need.

REVIEW: *But, it grew up in a university environment where people were able to assess what their needs were.*

FRASER: Each of these things is going to have to grow up in a different way. All I am really trying to say is that there are many ways in which people come to an agreement about standards. Agreement does not always come by negotiating standards ahead of time.

As a researcher, what I am trying to do is create an option for the world.

REVIEW: *Some people are saying that some foreign countries may force agreement by limiting access to their markets to adherents only.*

FRASER: I really don't want to go into this too far, but I think that a standard is going to result from a mix of different forces, including those things going on in standards organizations. It is absolutely essential that we evolve this concept of what data communications is in a technology-independent way—that is, in a way that can be implemented efficiently today and efficiently tomorrow. That has been a major goal of mine for the last several years in my work with Datakit.

REVIEW: *You don't feel it will be possible to legislate the standards?*

FRASER: You're probably able to understand the subject as well as I do. I don't know what is going to cause the world at large to take the proper path. My impression is that there is also a growing understanding of the need for integrating wide area and local area networks. There is certainly a growing understanding of the value to society of Universal Service. There is a growing frustration that we don't seem to be getting there very fast. As a researcher, what I am trying to do is create an option for the world. I am not very competent to say how that option will get used—or abused.

In that sense, I've got a notion of architecture. But there is a much more concrete notion of architecture that has to do with the sort of switching machine you're using. In that respect, I don't think there's going to be a single standard architecture. There are going to be many switching machines, and each technology generation will be replaced by more appropriate technology as it becomes available.

REVIEW: *How important do you think it is for one network to talk to another? This is largely a rhetorical question, but consider that not all methods for transporting data will be equally compatible with other networks.*

FRASER: Networks that talk to each other are inevitable. The reason for the modularity of Datakit is that we accepted the fact that just about every computer has different software supporting different protocols, and that different terminals support a wide variety of different protocols and physical interfaces. Somewhere you have to build into this network the matching that goes with joining all of these different gadgets in a common communicating environment. In Datakit, what we did was push that problem as far away from the center of the network as possible. It is real tough to do matching efficiently in the center of the network. With today's Datakit, the line card that terminates the wire from the terminal or the host or whatever, contains the translation needed for that host. That was the idea behind the modularity of Datakit. We have, or have under development, line cards for X.25, for bisync, for asynchronous terminals, and so on.

REVIEW: *How would you hook up to an Ethernet network?*

FRASER: You would hook up a line card that talks to an Ethernet. At the moment, one significant mismatch is between Ethernet and the networks that are being used by the common carriers, including Datakit. Ethernet is a datagram network, whereas most common carriers have come to the conclusion that virtual circuit networks are by far the most practical networks to manage. I certainly came to that conclusion quite a number of years ago.

I notice that most of the common carrier networks today are virtual circuit networks. In fact, even in environments where Ethernets are used, a large fraction of the traffic is being carried in a manner equivalent to a virtual circuit. Two machines agree to talk, they talk for a while, and then they quit. TCP is a virtual circuit protocol, IP is a datagram protocol that lies underneath. We have a Datakit network here and we also have an Ethernet. We have the two communicating with one another through a gateway, but it's the least comfortable of all of the gateways we have.

REVIEW: *By "gateway", are you talking about something larger than an interface card?*

FRASER: We're talking about a computer.

REVIEW: *Is that what's required?*

FRASER: No, but that's what we have today. There was some work done some while ago on a more compact dedicated Datakit/Ethernet interface but it was abandoned for unrelated reasons.

REVIEW: *Is the difficulty in Datakit/Ethernet communications largely an addressing problem?*

FRASER: Yes, in the datagram world, each packet is routed independently, carrying its own routing information. In the virtual circuit world, some initial overhead is spent setting up a virtual circuit so that information packets can later be sent over essentially without addressing information.

> Networks that talk to each other are inevitable.

So when a datagram arrives at a virtual circuit network, you first have to take the addressing information off. Potentially, you have to set up a virtual circuit for each datagram. That is not a very efficient way to do things, so you have to have some sort of caching scheme.

REVIEW: *You mean you let the packets collect and then set up a virtual circuit?*

FRASER: Or you can just set up a circuit when you see a new address. But, you don't take down the circuit when that datagram is gone. You can leave it around for a while in the hope of seeing another datagram for the same address.

REVIEW: *These datagrams do not indicate "last packet"?*

FRASER: No, they don't. You have to have a "time out" or some other mechanism for getting rid of the virtual circuit. So, as you can see, it's a bit clumsy.

REVIEW: *It sounds like you end up with an accumulation of unused virtual circuits.*

FRASER: Yes, there are things you can begin to do but you can already see that the gateway is a messy, inefficient sort of gadget.

REVIEW: *In terms of security, what are the relative advantages of datagram and virtual circuit networks?*

FRASER: Virtual circuit networks are much easier to manage, much easier to supervise, much less vulnerable. Security depends on the distribution plan as well. But I think it's much easier to administer a privacy arrangement within a virtual circuit network.

REVIEW: *You are not giving things away with each packet?*

FRASER: If there are a dozen of you all tapping into a single coaxial cable, then you have a perfect right, logically, to listen to each packet, whether it belongs to you or not. If it is a star-connected network out of a switch, then the first thing you have to do is go find somebody else's cable to tap into before you can even *start* listening to what they are saying. That is a property of of the distribution plan—not of virtual circuits per se.

The star distribution plan fits more naturally into the virtual circuit model than it does into the datagram model. It's true, though, that you can make both virtual or datagram work with either distri-

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

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

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

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

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

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

eForum is a communications breakthrough product.

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

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

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

eForum

The Electronic Meeting Manager

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

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

eForum has the flexibility to fit your communications needs.

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

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

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

NETI

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

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

Circle No. 13 on Inquiry Card

bution plan. The other thing is that since star networks have a natural place for supervision in the network, you can begin to provide services in the network that may help security from the point of view of the user—if the user cares to trust the network.

For example, in dialing a call through the telephone network, most people would assume that the number they call is the number they reach. Furthermore, they would probably assume that once the call is set up, somebody else won't take it over.

The telephone system could also provide information about what the calling number was and most people would be willing to believe that that was really the number of the person who was calling. We do these sorts of things in virtual circuit data networks, too. This certainly means that the computer that is called has a reasonably good idea of where the thing is that has called it. That sort of knowledge is less easily achieved in a network with distributed control.

REVIEW: *So in a star network with a central manager, the problem of getting a process on one system to identify itself over a network to a process on a remote machine is easier to tackle?*

I needed to think of the network switching machine in terms of modules like the ones in a Leggo set.

FRASER: It is easier, but it's not trivial. What we've done is make one part of the problem easier, but it's a multifaceted problem. I don't think anyone really knows how to handle it.

REVIEW: *What if you want to add a host to a network that has a central processor? Is that easier than hooking onto a connectionless network?*

FRASER: It has to do with the design of the distribution plan. If you strip the cabinet covers off a Datakit virtual circuit switch, you will find a whole series of line cards. Each card provides a tap onto a bus that runs along the backplane. That tap is very much like a tap onto an Ethernet in the sense that it is passive. You can insert and remove those cards even while the network is running. So we can install a new host without disturbing anyone.

REVIEW: *Much like you would add a telephone?*

FRASER: Yes, that's right. But, it does involve walking over to the cabinet and setting a card into it and then plugging in a cable that runs over to the computer.

REVIEW: *That certainly sounds reasonable.*

FRASER: But not as reasonable as you would like. What you would like is to have some way of prewiring the building so that you could have the same flexibility you have with power outlets. You can buy a toaster from Sears and and plug it right in. You don't have to go down to the basement to put another circuit breaker in the breaker box. We are moving towards that sort of computer environment now. I think that in years to come, we will have the sort of networking flexibility that we have today with the power distribution system—at least from a technical standpoint. I don't believe the special approaches of virtual circuit and datagram networks will have anything to do with this sort of availability.

REVIEW: *I am beginning to understand that one of your main themes is that there are lots of different issues and that some are "locked" and some are not. The argument over connectionless versus connected protocols may be irrelevant to whether or not you can prewire a building or, in fact, provide "Universal Service".*

FRASER: Yes, it is quite certain. We have an arrangement involving wall sockets in the lab that implements a virtual circuit network. It has all the administrative characteristics of an Ethernet from the point of view of making a connection or breaking a connection. It just doesn't happen to work that way.

REVIEW: *So what you do is initially provide enough circuit packs for an entire building and then turn them on as needed, perhaps through software?*

FRASER: Actually, you don't want to invest in circuit packs for unused lines. You don't want to make it expensive. There are actually ways of doing it that are extremely cheap.

REVIEW: *I guess we will have to wait on that one.*

FRASER: Yes, that's for the future.

REVIEW: *Do you have any comments on the ISO model for networks?*

FRASER: Somebody obviously primed you. I expected to need hardware packet switching in order to build very wide-band, low cost switching machines. To accomplish that, I needed to rearrange the lower two layers of the ISO protocol model a little bit. The reason is simple: a modern switching machine is also a multiplexor. Let's say you have a transmission line with traffic on it belonging to 16 people that has to go 16 different places. A switching machine has to be able to demultiplex that traffic. In order to do that efficiently, the switching machine and the multiplexor have to be designed as one.

Within the framework that I've been working in, switching is the most primitive activity in the network. It does not get involved with protocols in any way. That's how we get a high performance switching machine. Unfortunately, the ISO model of computer communications placed multiplexing above the link level protocol. That meant you had to undo all of the link level protocol just to find out who a packet belonged to. That in turn meant that the switching and link level protocols were somehow intertwined.

The only change I made to the protocol hierarchy was to push multiplexing down in the hierarchy and make it a primitive property of the network. I put it right down in the lowest level and pushed error control up the hierarchy to a higher level than switching and multiplexing. In our hierarchy, in order to avoid confusion, I've been calling it A, B, and C instead of 1, 2, 3 or whatever. We have A at the lowest level and C at the highest.

So it looks like:

C - Network level; end-to-end communication
B - Transport level; (switching, multiplexing) link by link
A - Physical interface

In our world, the error recovery/retransmission is all done in level C end-to-end communication. Another reason for doing it that way has to do with integrating voice into the network. Voice requires different treatment as far as error control is concerned. So it fits much more naturally into this scheme. Voice can use levels A and B of the hierarchy the same as data and then make use of a different level C because you don't want to retransmit voice. By making multiplexing and switching functions of the lower levels of the model, we are able to carry both voice and data without getting into the complexities that people encounter if they put voice over a more conventional packet switch.

REVIEW: *Correction is done at the top level?*

FRASER: We do correction end-to-end essentially, we could do it link-by-link but the hierarchy doesn't require it. It means that the user can exercise discretion, and that different applications can be handled differently.

REVIEW: *Do you think your choice resulted from your having a different goal than the international organizations that shaped the ISO model?*

FRASER: I don't know what they were trying to do. I was trying to solve one problem, namely Universal Service.

REVIEW: *And their model didn't fit?*

FRASER: Their model didn't fit and it caused me a lot of trouble. So, I decided I had to change it.

REVIEW: *Has anyone else adopted your approach?*

FRASER: Well, there are a growing number of people who are adopting it in one way or another—but not as a standard. For example, if you look at the X.25 networks, they all use internal protocols that differ from their external images.

In order to get high capacity switching, you need to do the sort of thing that I am doing. Whether you declare it as an external public interface is another matter. There is no reason anyone has to. We have an X.25 interface on Datakit. It's not necessary to tell the world that Datakit has a somewhat reorganized protocol hierarchy.

REVIEW: *Is it irrelevant?*

FRASER: We translate on the line card.

REVIEW: *Which is not a major problem, from what I understand. X.25 works fairly well with Datakit, doesn't it?*

FRASER: There is the potential there to provide a very efficient virtual circuit network supporting X.25. Datakit is an efficient virtual circuit network and X.25 is an interface protocol for virtual circuit networks. The two fit well together.

REVIEW: *X.25 is a rather old protocol. Are there newer protocols supplanting it that may not work as well with Datakit?*

FRASER: The things that don't work as well are datagram networks. IP is a good example. There just isn't a very good fit. You have the problem I mentioned earlier and there are others like it. The real intent was to make our network as transparent to protocols


HEWLETT
PACKARD
Integral Personal Computer
Stop

Almost priceless.

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

Introducing the Integral Personal Computer.

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

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

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

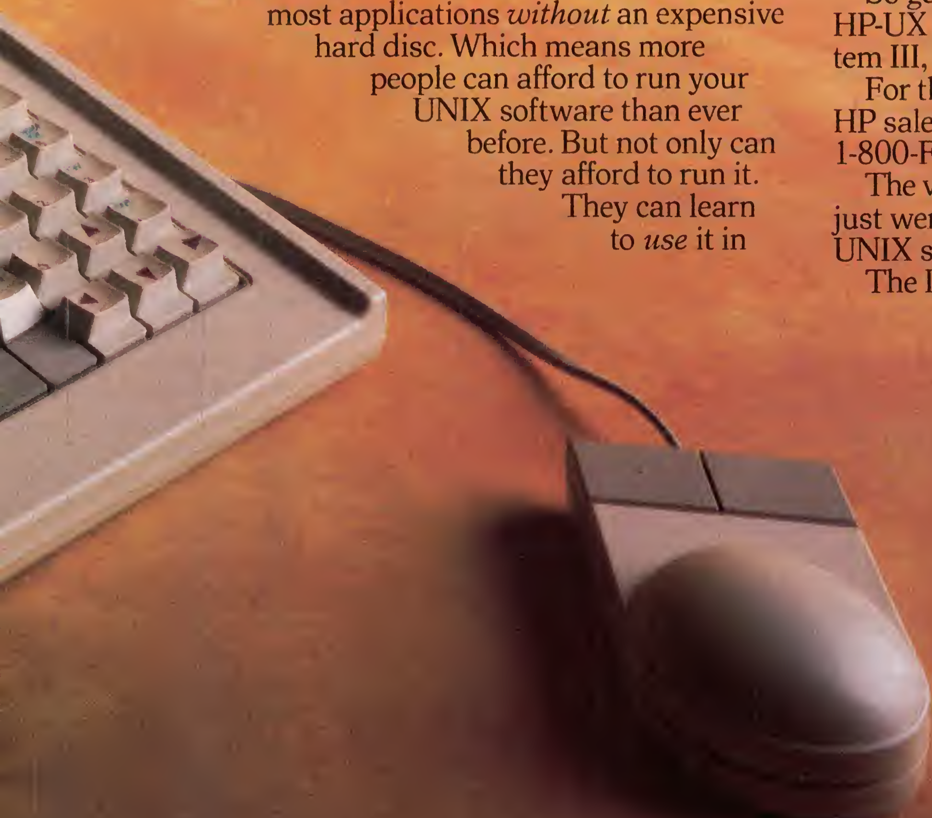
What's more, the Integral PC goes where you need it. The complete system—including our revolutionary ThinkJet printer—packs into a single 25-pound package that takes up less than a cubic foot.

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

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

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

The Integral Personal Computer. Just $4,995*



HEWLETT PACKARD

Circle No. 15 on Inquiry Card

UNIX is a trademark of AT&T Bell Laboratories.

Mouse is optional. *$4.995 is suggested U.S. list price; prices elsewhere may vary

PG02504

as we could. Amongst data networks, Datakit is by far the most protocol-transparent. That is to say, we have a single network that has asynchronous terminals and bisync running over the same trunk, sharing with one another without interfering with one another. Both are carried transparently in the sense that the network doesn't know or care what the protocol is. We can echo characters from an asynchronous terminal or from the host. The transport system is really quite transparent to the end protocol.

We have our own internal protocol that I think has a lot going for it. We have designed it specifically to work with Datakit because it was necessary to come up with a new protocol in order to fully exploit the network. In doing that, we were also mindful of the fact that most networks lose a good fraction of their performance in the host interface. You can buy a 50 megabit coax but get only 1 megabit of throughput. The problem is a mismatch between the design of the network and the design of the host software and interface. We have really worked on that problem. We have our own protocol and a piece of hardware that runs it. The indications are that we are in a very strong position to provide extremely high performance host interfaces that do protocol handling as well as talk to the network.

REVIEW: *Suppose you had a videotex host and you were talking to homeowners. You generally would want to multiplex that sort of traffic.*

FRASER: Yes. We have been using an internal protocol for transport purposes that is designed so that many of the common protocols like HDLC [High-Level Data Link Control, a bit-oriented rather than byte-oriented control] can be mapped into it reasonably efficiently—quite efficiently, in fact. We have just demonstrated experimentally that the internal protocol can be terminated at high speeds in hardware. We haven't made any custom LSI circuits yet, but our experiments give a good indication of the power of that protocol. What we have to do is get some experience mapping it into different end user protocols. We need more of this type of experience, but it is looking quite good at the moment.

REVIEW: *One of the concerns I would have about your position is that if you had to pick a network that is growing in popularity for a broad range of machines, it would be something like Ethernet. It's been hooked up to everything from Crays to PCs.*

FRASER: Yes, and if it does a good job at that, we'll have to worry about it. The question is: is it doing a good job at that?

REVIEW: *Its throughput and bandwidth are comparable to Datakit's, aren't they?*

FRASER: At the moment they are about equal, but not for an equal amount of complexity.

REVIEW: *Datakit is more expensive, isn't it?*

FRASER: The bus interface in Datakit is simpler but today it makes less use of LSI. One of the real differences between Ethernet and Datakit in their current states of development is that Intel has put a lot of money into LSI development for Ethernet. Datakit was designed for LSI but the product that you see today doesn't have any custom LSI in it.

REVIEW: *If we can digress for a minute. Do you think that a network product produced by one, albeit large, company, can compete with other networks being developed by a number of competing companies? The LSI refinements you spoke of in Ethernet, after all, did not come solely from Intel. Without competition, it may appear that there is no great future for Datakit.*

FRASER: It is quite conceivable that through our research we will come up with a better keyboard than the standard QWERTY keyboard. There may well *be* better keyboards than the QWERTY keyboard but it's also perfectly clear that people are going to continue using keyboards they're already familiar with. I don't know if Datakit is in that position or not. It's is too early to tell.

REVIEW: *Is the bandwidth of Datakit necessarily limited to channels of 6–7 megabits per second with a throughput of 1 megabit—or something on that order of magnitude? Is there any bandwidth dependency in the technology?*

FRASER: Be very careful about numbers. The throughput for file transfer between one computer and another is limited to about 1 megabit by the protocol handling software of our host computers. The throughput of a single switch is limited to the bandwidth of the backplane bus which for the prototype is about 7 megabits per second. In Ethernet, you have a piece of wire that potentially has a 10 megabit bandwidth. But you have overhead on top of that, so you never get 10 megabits. The Datakit Information Systems Network product has a backplane with approximately 8 megabits of capacity that you can think of as roughly equivalent to an 8 megabit Ethernet. That number is a random one in the sense that somebody picked it. The technology doesn't force it. In contrast to Ethernet, where it would be hard to go much faster, it is quite clear that it would be possible to go at least an order of magnitude faster with Datakit.

REVIEW: *Would such a network be compatible with 'existing' Datakits?*

FRASER: Yes. In the sense that we were talking about it earlier. There is this concept of Universal Service that we have really been trying to maintain. Datakits can talk to one another over trunks. It is designed so that you can switch directly over trunks. You can have different speed switches on the ends of a single trunk. It's entirely possible to have low speed Datakits, medium speed Datakits, and high speed Datakits.

You can even have things that conform to the service but are not necessarily implemented in the same way that our prototype Datakit is implemented. For example, one of the things that we are interested in doing now is making a network that is suitable for joining our Crays with our other supercomputers. There is no reason at all why that network can't be made consistent with our existing network.

REVIEW: *You will use a different speed backplane?*

FRASER: Yes, an order of magnitude of difference.

REVIEW: *Can that be done now?*

FRASER: Oh, yes. It doesn't even require new technology.

REVIEW: *It will not require special LSI?*

FRASER: No. We don't have the distance problem that Ethernet has. You can increase the speed of Ethernet only if you increase the packet size or reduce the length of the bus. In Datakit, the bus is very short anyway.

REVIEW: *Is that because it's a star network and you're not trying to pack everything on one wire?*

FRASER: That's right. You can make an arbitrarily large network by interconnecting switching machines with transmission lines that go extremely fast. The technology exists to carry data over long distances at gigabit-per-second rates.

You can build high speed interfaces and build a high speed bus. The switch in Datakit is mostly idle. In order to believe that I was

EASIER THAN 1-2-3...

BUT DESIGNED FOR LARGER SYSTEMS

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

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

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

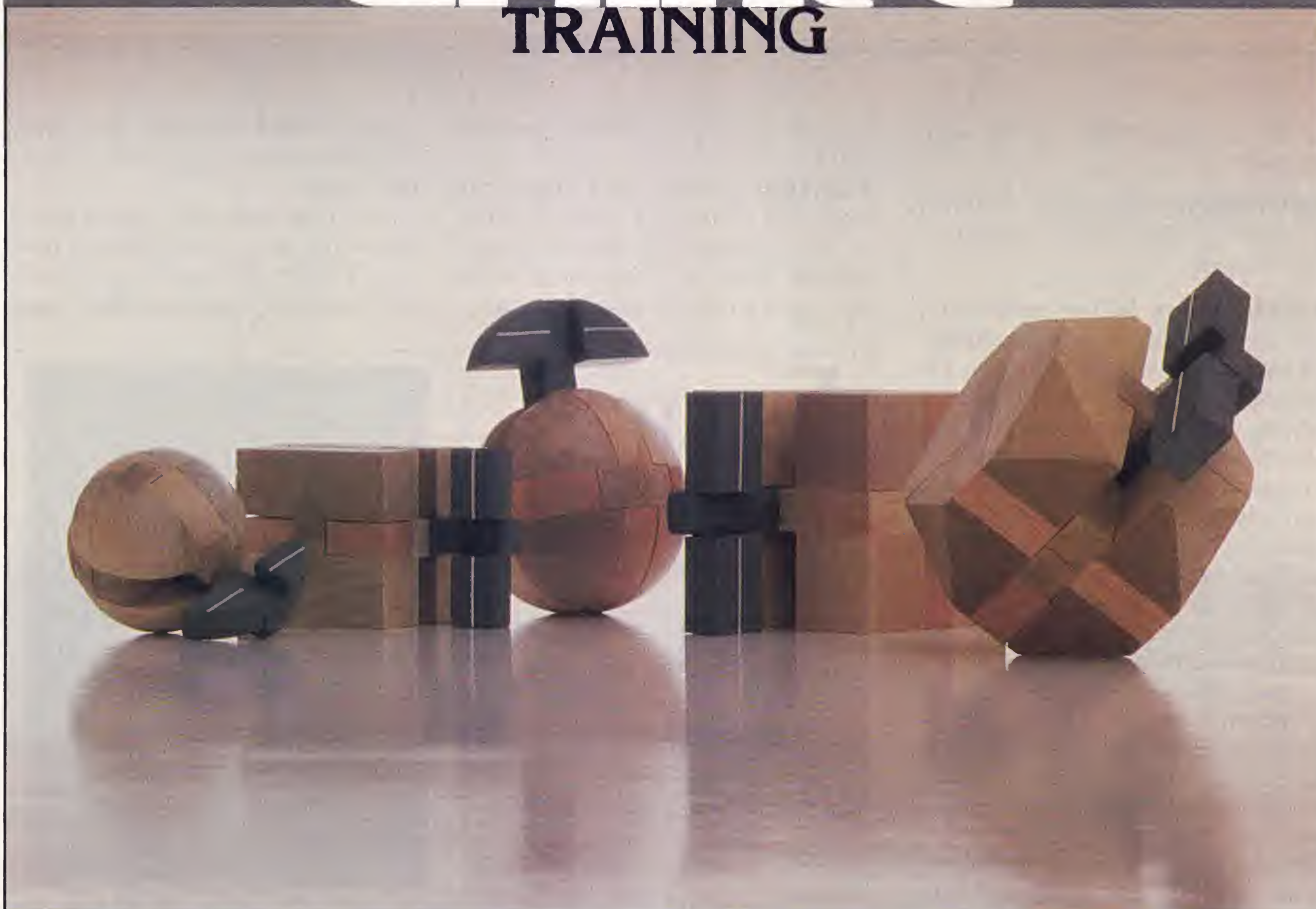
DSD CORPORATION
P.O. BOX 2669
KIRKLAND, WA 98033-0712

EFFECTIVE SOFTWARE FOR BUSINESS

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

Circle No. 37 on Inquiry Card

UNIX™ & 'C' TRAINING



MANY UNIX-BASED SYSTEMS
ONE UNIX TRAINING COMPANY

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

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

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

ONE UNIX TRAINING COMPANY
MULTIPLE DELIVERY SYSTEMS

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

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

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

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

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

™ UNIX is a trademark of Bell Laboratories.

COMPUTER TECHNOLOGY GROUP

Telemedia, Inc.

310 S. Michigan Ave., Chicago, IL 60604

Circle No. 14 on Inquiry Card

on a reasonable track towards understanding how to provide Universal Service, I had to have paper designs for switching machines in all the classes for which the Bell System has switching machines. The paper designs have not all been developed into physical switches because nobody needs them today.

REVIEW: *With the addition of LSI, and the correct scale of manufacturing, do you think that Datakit price/performance will improve?*

FRASER: Yes, very definitely. However, as I understand it, the price/performance is not bad already. It depends on what you want the package for.

REVIEW: *Would you describe Datakit as new technology?*

FRASER: No. Datakit was designed in 1975. It became a product in 1983. Many of the chips that it uses were available in 1975. There have been many changes—and changes still continue—but we've held onto the initial emphasis of making a development quality product using techniques that had already been tried extensively in-house.

REVIEW: *Would you describe Datakit as unexploited technology?*

FRASER: That's another matter. It is certainly not untried. As you may know, Bell Labs is in the process of installing a corporate network.

REVIEW: *Will this replace the Bell Labs Network?*

FRASER: Much bigger than that. It will include about 15,000 terminals and hundreds of computers. It will all use Datakit technology.

In the laboratory, we have come close to demonstrating technology for wiring up offices in a way that is also practical as a method for wiring up your home. It would allow virtual circuit connections that run across the country. We haven't built all the pieces, though. That's another matter. We should build more of the pieces. I mentioned the wall socket, that's clearly key to the future. It will make a big difference in the office, but work in that area is still in the exploratory stage.

I also mentioned the business of providing an interface for supercomputers and that, too, is in an exploratory phase. I mentioned the very high capacity network switches. Where to start? What products is the market ready for? When to launch? And so forth. This is not really a subject I understand but it is a subject that is key to the question you ask. It has to do with whether users' needs match our perception of them. Most users who buy Datakit networks at the moment buy them because they have lots of terminals that they want to talk to host computers. They buy them as an alternative to buying data PBXs for the simple reasons that Datakit also allows wide-area networking and allows machines to do very effective host-to-host communication at high speeds.

REVIEW: *And save on interfacing costs to boot?*

FRASER: Yes. This is something that has already proved itself but it has a lot more potential than has been realized to date. ■

GOLD-UNIX™

PROGRAMMERS - UNIX UTILITY TOOL KIT.

Utilities help to make the programmers task easier, quicker and **cheaper**

Functions Include:
(essential 'C' source provided)

- SINGLE USER, MULTI-KEY ISAM
- HELP SCREEN MANAGER
- I/O CONTROL FACILITY
- INDEX FILE VIEWING FACILITY
- Etc. $299 U.S.

Coming Soon
MULTI-USER, MULTI-KEY ISAM
ORDER NOW
UNIX is a trademark of Bell Laboratories

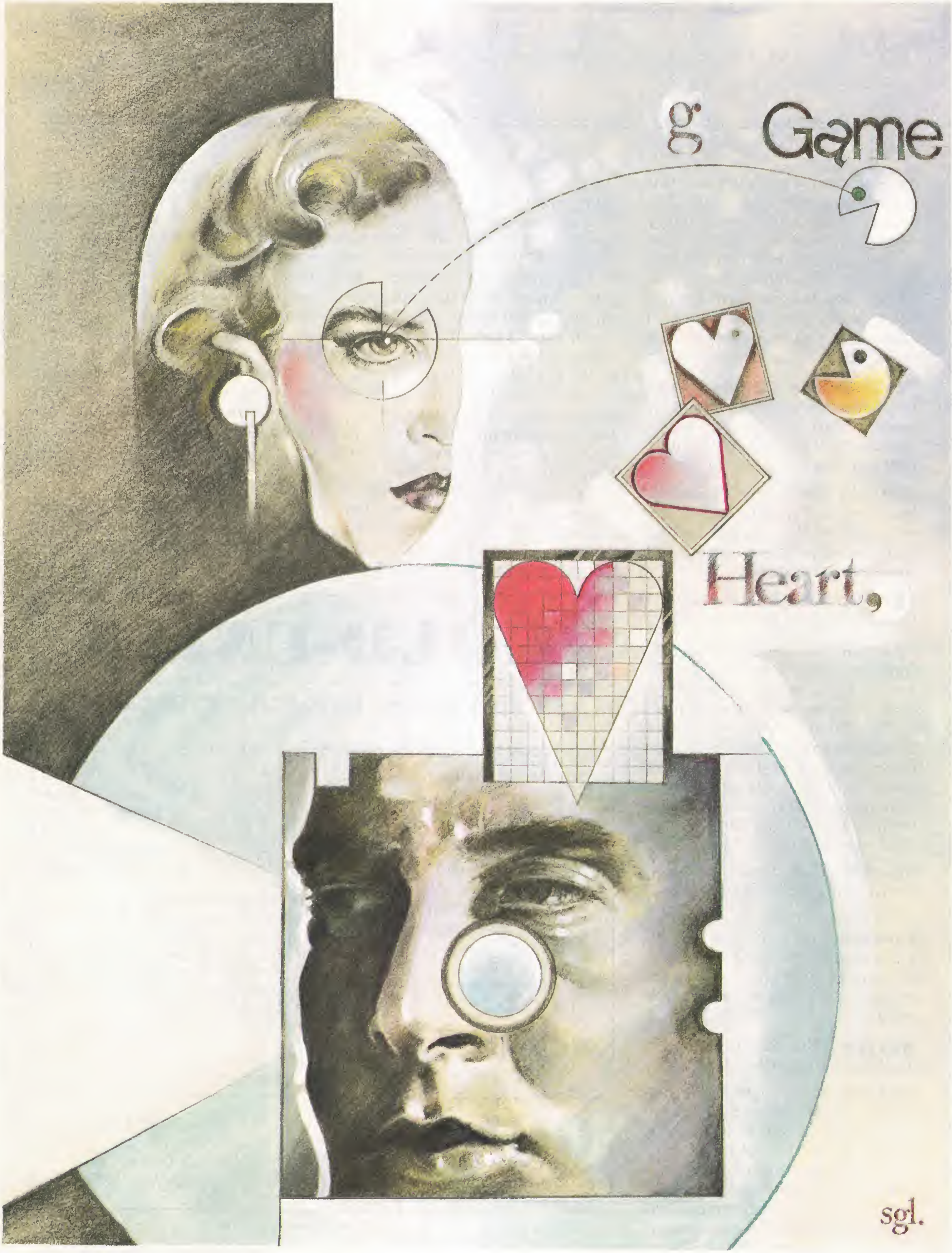
FOR THE PROFESSIONAL OFFICE PROFESSIONAL SOFTWARE

A System to Provide:
- Client Management
- Trust Accounting
- Time Management
- Accounts Receivable
- Disbursement Control
- Billing Management
- Diary (Tickler) System
- Full Help-Screen Facilities

Write Now For More Information

GOLD SOFTWARE CORPORATION
(604) 278-4653
202A 7080 River Road
Richmond
British Columbia
Canada V6X 1X5

Circle No. 36 on Inquiry Card



g Game
Heart,
sgl.

GAMES PEOPLE PLAY

A look at human networks

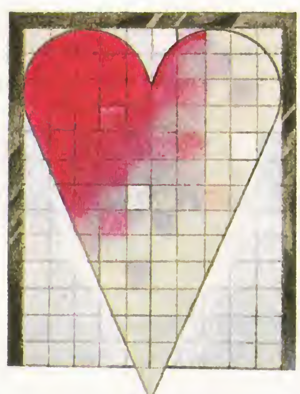
by Jordan J. Mattson

Not all networks are high speed links between computers. In each of our lives, there also exists a web of human relationships that, for all the obvious distinctions with computers, can be much affected by computing. Indeed, in the UNIX community, one of the largest human networks to be found focuses on computer games.

Players and developers alike are linked by their fondness for games ranging from serious simulations such as Ken Thompson's Space Flight Simulator (the program that led to the development of UNIX) to less serious recreations such as Ken Arnold and Michael Toy's *Rogue*. Whatever their flavor, though, the games of Unixland have a significant impact on the human networks touched by UNIX. This simple fact raises fairly complex design issues that responsible games developers do well to bear in mind.

ACADEMIC ROOTS

The first contact most UNIX users have with games comes during their first months of college. It's no coincidence that it is also within the academic setting that games have had their greatest impact on human networks. This owes in part to the general malleability of student networks, but it also serves as testimony to the pervasiveness of computing in the university environment.



Because the academic setting has traditionally been where the "first strike" of UNIX games has been delivered, the clearest examples of the impact of games can be collected there.

Within the academic microcosm, one detrimental effect that can quickly be uncovered is the great isolation and loss of focus many students suffer as a result of game addiction. Over the years, a malaise has recurred at the University of California at Santa Cruz that has come to be known as "freshman game syndrome". Victims become hopelessly hooked on a game (*Rogue* for the most part) and spend many of their waking hours playing it. The results of this addiction are easy to recognize: interest in classes fall, often disappearing altogether; friendships are abandoned as more and more time is spent at a terminal; and contact with other people diminishes.

Once people become addicted to games, their lives usually take one of three paths. The first path leads to deeper addiction, which in turn leads to greater isolation and alienation. This is typically a sure route to probation and, ultimately, to dismissal. Upon dismissal, victims can either reassess their lives or retreat still further into games, denying the reality of what has happened to them.

A second possible path is a backlash to the first. This is the path walked by converts who renounce games as an inherent "evil". For these games teetotalers, there are no shades of gray—only black and white.

A third group of students take another path where shades of gray *are* in evidence. These students perceive the potential for alienation but simultaneously see the potential benefits games can offer, and work to integrate them into their lives.

Illustration by Stephen G. Luker

Among these benefits are the ability to create better environments for work and the opportunity to create healthy, new human networks. Games can promote what Ken Arnold, one of the creators of *Rogue*, calls "user-homeyness". Arnold compares this "user-homey" feeling to one that can come from letting a work space or room "get just messy enough to feel like it's yours." In such an environment, users feel that the system is a friendly place where they can put their feet up on the furniture and thus be happier and more productive. Of course, there are those who feel that games can make systems altogether *too* friendly, wasting precious time and computing resources in the process.

The academic setting has traditionally been where the "first strike" of UNIX games has been delivered.

Arnold takes strong issue with this argument because "games have replaced the water cooler as the place to take a break" in many programming environments. After a grueling bout with an intricate piece of networking code, what better relief than a half-hour spent killing monsters or building an interstellar empire? In this way, UNIX games can pull people away from the head games of the water cooler into something that is more fun, probably more healthy, and most assuredly more relaxing.

HUMAN NETWORK TOPOLOGIES

Games in the UNIX community have also spawned two new forms of human networks. These are characterized by cooperative competition and cooperative creation.

In a cooperative competition network, people are drawn together because of their individual struggles to beat a game and develop better playing strategies. Such network communication can involve the exchange of "vital" information and clues, hints about how best to play a game, and critiques of members' play.

The exchange of "vital" information and clues occurs most in adventure games. A hint can update someone on how you are doing in your game and in turn involve you in theirs. This sort of exchange over a period of weeks or months unsurprisingly has the effect of building a common bond between people. When one person finally does complete—or win—the game, all the people who have participated can celebrate the victory. The ties that grow in this way often carry over to the world outside of games and computing.

The very nature of computer games encourages just this sort of cooperation. Because the computer can't hear you discuss how to beat it and does not mind if you take a half-hour between moves, kibitzing among competitors is commonplace. This exchange of hints and strategy exposes people to new ways of thinking and involves them both as teachers and students.

DUAL

The staff at Dual Systems has compiled a Filesystem Reference Manual for the UNIX™ operating system, both Version 7 and System V. The manual contains three listings of the files distributed on computers using the UNIX operating system. The first is a complete alphabetical listing of all the files, providing pathname, description, origin, and file group. The second listing divides all the files into thirteen application groups such as mail, program development and text processing. The third listing divides the files into six groups based on the originator (e.g., Bell Laboratories, Berkeley enhancements, etc.).

Every user of the UNIX operating system would profit by having one of these books. The drawing showing the hierarchical interrelation of the various files is a great aid in finding one's way around the UNIX operating system. Price is $35.00 in single quantity. Please contact our sales department to order.

*UNIX is a trademark of AT&T Bell Laboratories

DUAL

DUAL SYSTEMS CORPORATION

2530 San Pablo Ave.

Berkeley, CA 94702 (415) 549-3854

Circle No. 33 on Inquiry Card

Another sort of cooperation engendered by games occurs when people work together to develop a new game or improve an existing one. As a creator produces the design of a game, people in his or her cooperative creation network can critique it and suggest improvements. This allows the creator to think through the design, world view, and human interface of the product—important questions if the game is actually to be played by others (and who builds a game except to have it played?). The effort itself also creates a bond between the creator and the people in the cooperative creation network. These bonds can extend beyond the creation of a game to become the basis for lasting friendships and further creative endeavors.

Cooperative competition and cooperative creation human networks sometimes can meld. One example of this would be the human network that created *Rogo-O-Matic*, the expert system that plays *Rogue* and wins. At the center of this human network are Michael Mauldin, Andrew Appel, Guy Jacobson, and Leonard Hamey of Carnegie-Mellon University, but the network extends to any UNIX site where *Rogo-O-Matic* is being used and comments or bug reports are being generated. As in all human networks, some members are more intensely involved than others, but all work as part of a network of people who would probably not be in contact if it were not for the existence of *Rogue* and *Rogo-O-Matic*.

CONCLUSIONS

Since games can have a profound effect—whether positive or negative—on those who play them, developers shoulder a responsibility to design with care and intelligence. The temptation for players to "go it alone" must be discouraged. Incentives to cooperate, on the other hand, should be consciously woven into every game.

> "Games have replaced the water cooler as the place to take a break."

One possible way to achieve this might be to design more games that involve multiple players who must work together to achieve a common goal. Games of this type could cause the flourishing of more cooperative competition and cooperative creation networks. It may also engender the development of such networks beyond the walls of the computer lab.

Jordan J. Mattson is President of System Software Design Group Ltd., a UNIX design and consulting group based in Santa Cruz, CA. He also works as a programmer specializing in office automation systems at the University of California at Santa Cruz. Early in his academic career, Jordan himself suffered briefly from an acute attack of "freshman game syndrome". Mail can be sent to Jordan via **UUCP** *at ucbvax!ucscc!-ucsce:jais, or via surface mail at Crown College-UCSC, Santa Cruz, CA 95064.* ■

XENIX Communications Available NOW!

Put your computers on speaking terms.

$295.00

Introducing TERM. Communications Software

Everyone from the beginning computer user to the expert finds TERM easy to learn and powerful to use Just plug it in and go! In a few keystrokes you can access a remote database or send a group of files to another system

TERM allows your computer to perform efficient, error-free exchange of binary or text files, over phone lines or hard-wired circuits at speeds of up to 9600 baud Available options allow you to include or exclude a group of files for transfer in a single command

TERM's "data capture" feature allows saving transcripts of sessions with remote mainframe and minicomputers to disk for later editing or printout if desired

- Pre-installed and ready to run
- Automatic error checking and re-transmission
- Wildcard (*.*) file send/receive capability
- Xon/Xoff, Etx/Ack Ascii protocols for communications with non-TERM systems
- Full/half duplex emulation mode for remote systems
- Modem7 protocol for remote bulletin boards
- Auto-dial/Answer and Hangup supported on Hayes Smartmodem 300/1200 and compatibles
- Programmable batch file capability
- Unattended file transfer/auto logon
- Translation tables for input and output

TERM is available now on the Altos 586 and Tandy Model 16 along with more than 35 CP/M—80 MSDOS and CP/M—86 systems. IBM PC*/XT, Kaypro, Osborne, Televideo, Victor, Apple* II—CP/M Heath Vector Sanyo, Eagle, Molecular, Altos, and many others

CALL OR WRITE FOR FREE PRODUCT CATALOG

CENTURY SOFTWARE
We make it easy for you.

9558 South Pinedale Circle
Sandy, Utah 84092
(801) 943-8386

CP/M is a registered TM of Digital Research

Circle No. 34 on Inquiry Card

INDUSTRY INSIDER

The diverse network market

by Mark G. Sobell

Manufacturers who can satisfy the growing needs for system interconnection have the world on a string. But before their promise is realized, these companies must first cross some hurdles. One particularly important challenge is the development of standards allowing different systems to be connected. User education is another important issue since sorting out the advantages and drawbacks of the many different types of Local Area Networks (LANs) is already a daunting task—and promises to become more difficult yet. Without education, LAN users will be unable to choose effectively.

The choices range from inexpensive, RS-232-based, low-bandwidth (slow data transfer) systems to costly, coaxial-based, high-bandwidth systems that measure data transfer rates in millions of bits per second.

RS-232 LAN

Touchstone Software Corporation offers one low-cost, low-bandwidth network that can connect a UNIX system to a flotilla of PCs and Macintoshes. The UNIX system functions as a *command server*, responding to requests from the other machines on the network. All that's necessary to set up this network are RS-232 cables and a software package to run on each of the machines in

the network. The software for the UNIX machine costs just under $300 while the packages for the PCs and Macintoshes cost $200 each.

The cost may be right, but the Touchstone system is limited. The speed of data transfer is limited to the speed your terminal normally runs at (usually 9600 bits per second with a direct connection)—meaning that transferring a 200K byte file takes almost three and a half minutes. Touchstone clients are also limited by the number of ports their UNIX systems offer since each member of the network takes up a port that might otherwise be used to service a UNIX user.

BROADBAND LAN

Sytek, which came into the limelight recently when it was selected by IBM to manufacture a LAN for its PCs, has actually been in the LAN business for six years. Among its products is a *broadband* LAN that converts digital signals to radio frequency (RF) signals and transmits them over coaxial cable. As with Touchstone, Sytek's LAN is based on a simple concept: plug every component of the network into a coaxial network cable and give each component an address. Of course, plugging a terminal or computer directly into a network cable simply is not possible. So Sytek manufactures *Packet Communication Units* (PCUs) that connect components to the network cable and perform a myriad of functions, including compensating for different data transfer rates by buffering input and output. They also allow components that use different transmission protocols to work with each other.

Sytek, like Touchstone, uses a system's RS-232 ports as the basis for its network. But, unlike Touchstone, Sytek's product allows you to add components to the network without using up additional login ports. And, because there is no need for special software to run on your computer, you can include *any* type of computer, terminal, or other peripheral device on the network as long as it uses an RS-232 interface. One Sytek installation already connects four separate sites with a total of

EMERALD ONE™

SOFTWARE WITH COURAGE, BRAINS AND HEART

WHAT IS EMERALD ONE?

The most complete integrated office system available today, EMERALD ONE combines the most essential office tasks through six fully compatible and seamless sets of tools. EMERALD ONE runs on a broad range of mainframe, mini, super-micro and personal computers which use the UNIX™ operating system–the emerging standard for the office.

THE TOOLS OF EMERALD ONE

EMERALD ONE integrates your office tasks through:

1. *COMMUNICATIONS,* including Telephone Messaging and Electronic Mail systems,
2. *INFORMATION HANDLING* with EMERALD ONE's powerful Relational Database system,
3. *DECISION SUPPORT* features such as Business Graphics and the Electronic Spreadsheet,
4. *DOCUMENT PREPARATION* with Word Processing and a Cabinet, Document and File Folder system,
5. *TIME MANAGEMENT* tools such as the Personal Diary system and Meeting Scheduler and
6. *SYSTEM ADMINISTRATION* functions that allow a non-technical user to customize EMERALD ONE for the individual, work group and organization with ease.

Circle No. 25 on Inquiry Card

SOFTWARE FOR THE WORK GROUP

EMERALD ONE goes far beyond stand-alone personal computer software by linking individuals and their work groups. With EMERALD ONE, users work as a communicating group, not as isolated individuals. Whether it be a document, spreadsheet or personal diary entry, everything created with EMERALD ONE can be exchanged easily between individuals, work groups and beyond.

EMERALD CITY, THE PEOPLE BEHIND EMERALD ONE

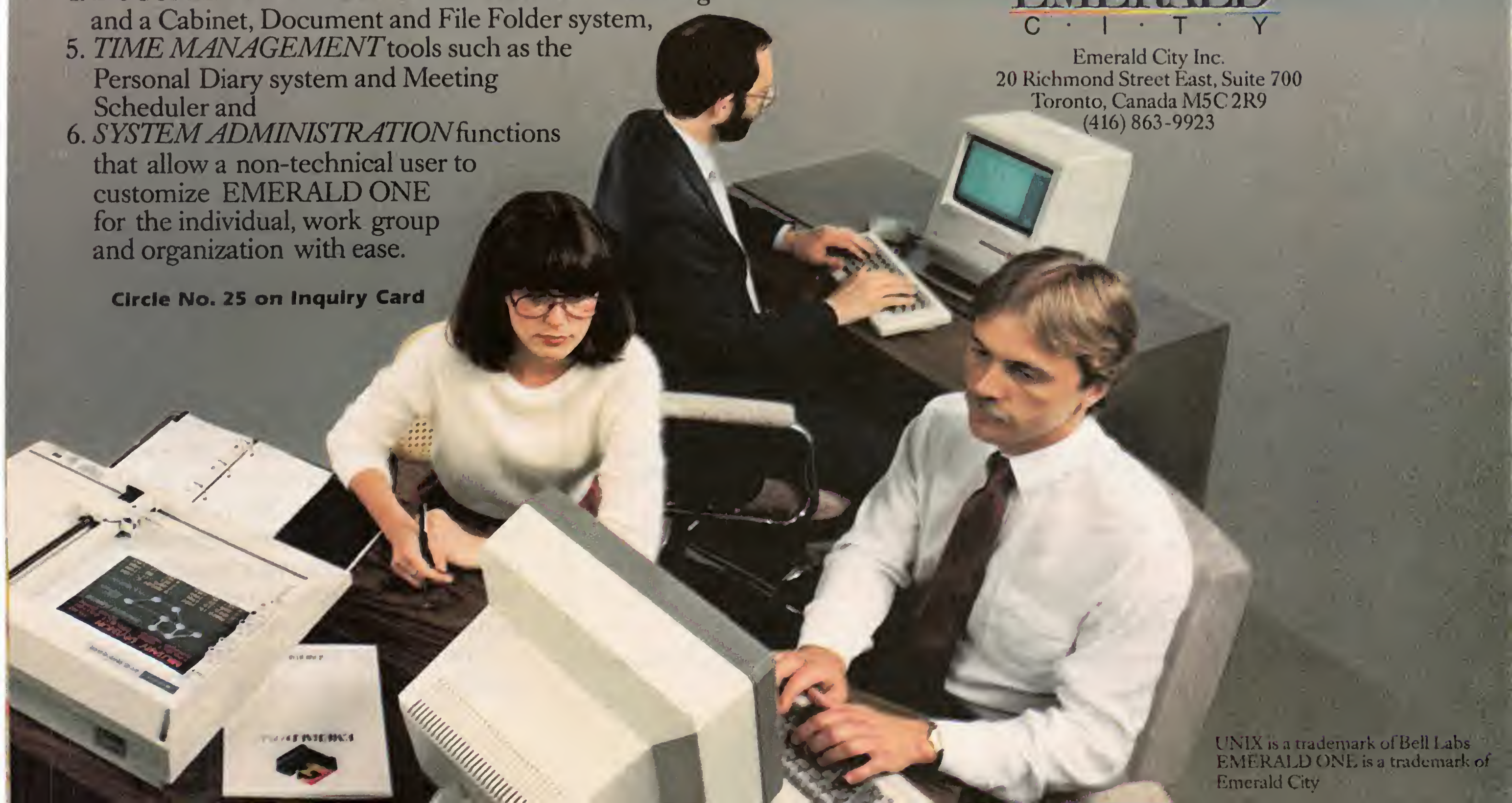
EMERALD ONE is the result of an intensive, multi-year research commitment by Emerald City and its sister company Trigon Systems Group, one of the most respected consulting companies in office integration.

Attractively priced for distribution by hardware manufacturers, system integrators and OEM's, EMERALD ONE is fully supported by an extensive marketing program designed to assist distributors in penetrating the integrated office market. Emerald City offers the reality of a complete business solution, not just technology.

Emerald City, the company with courage, brains and heart.

EMERALD C · I · T · Y

Emerald City Inc.
20 Richmond Street East, Suite 700
Toronto, Canada M5C 2R9
(416) 863-9923



UNIX is a trademark of Bell Labs
EMERALD ONE is a trademark of Emerald City

1250 connections.

The added functionality, of course, means that the Sytek system is also more expensive than Touchstone's. Approximate initial setup costs are $3500 for a Head End plus the cost of the cable. Each connection to the network (each terminal, computer port, and printer) adds about $400 to the cost of the system. Data transmission speeds, though, are still limited to the speed of an RS-232 port (9600 bits per second).

The cost may be right, but the Touchstone system is limited.

Up another rung on the network ladder, Excelan manufactures a *baseband* LAN—one that uses digital signals and a coaxial Ethernet cable. The system requires that a board be plugged into each computer on the network. In return, it eliminates direct connections between terminals and the network. Thus, users of the Excelan system can gain access to the network straight through the computer.

This is the type of LAN that truly can be transparent to the user. For example, with the proper software, it can automatically collect data from several machines when you run a report. Data transfer speed, moreover, is free of the limits imposed by RS-232 cable. The same 200K byte file that would take three and a half minutes to send via Touchstone's network takes only seconds to transfer via Excelan from machine to machine.

The drawbacks to this type of system stem from the requirement for network software and hardware to exist on each machine on the network. While their function is the same, the hardware and software required to run on a VAX will differ from that needed for a microcomputer.

Excelan manufactures boards for Multibus, VME, Q-bus, and Unibus. It also has a plan to assist OEMs in designing and manufacturing boards for other types of computers. The networking software is available for RSX-11 (PDP-11), VMS (VAX), and a variety of UNIX operating systems. Here again, Excelan has indicated a willingness to assist in developing necessary software for other operating systems. Nevertheless, if you have a system that Excelan does not already have a board and software for, it would be quite expensive to add it to your network.

The cost for an Excelan system ranges from about $4500 to

Information Technology Development Corporation

9952 Pebbleknoll Dr. • Cincinnati, Ohio 45247 • (513) 741-8968

At long last, ITDC - a leading supplier of UNIXtm and C software consulting, development and education services - is making its highly successful education courses available to the public. Taught in a classroom environment that offers over 50% hands-on training, our courses are taught by professionals for professionals.

1985 Summer, Fall and Winter Course Schedule

UNIX for Application Developers	Jun 17 - Jun 28 Sep 9 - Sep 20 Nov 11 - Nov 22
UNIX for End Users	Jul 15 - Jul 19 Aug 5 - Aug 9 Oct 14 - Oct 18 Dec 2 - Dec 6
C Programming Language	Jul 22 - Aug 2 Sep 30 - Oct 11
C-Shell Programming	Aug 26 - Aug 30 Nov 4 - Nov 8
BOURNE Shell Programming	Aug 19 - Aug 23 Oct 28 - Nov 1
UNIX Systems Administration	Sep 23 - Sep 27
INFORMIXtm Relational Data Base	Jul 8 - Jul 12 Aug 12 - Aug 16 Oct 21 - Oct 25

ITDC's courses are taught in Cincinnati, Ohio. Enrollment in any of the classes listed in the course schedule may be accomplished by contacting ITDC by phone or letter.

ITDC requests that you register well in advance of the course date, as ITDC classes are well attended and enrollment is limited. If a purchase order or payment has not been received at least two weeks before the class starts, your reservation will not be guaranteed.

Tuition is $500 per student for one week courses and $1000 per student for two week courses. Payment may be made my confirmed company purchase order, check or money order.

UNIX is a trademark of AT&T Bell Laboratories
INFORMIX is a trademark of Relational Database Systems

Circle No. 40 on Inquiry Card

$6000 per node (computer), plus the cost of the cable. But per-user costs can vary widely because Excelan allows each node to serve all of its users instead of imposing a one-user-per-node limitation. In the case of a VAX, the per-user cost can be quite low.

LAN PERSPECTIVE

To get an overview of the market and technology, I spoke with Dr. Robert M. Metcalfe, founder and chairman of 3Com Corporation and inventor of the Ethernet.

Dr. Metcalfe pointed out that networking is a varied market, embodying disjointed technology that constantly gives rise to confusion. "When looking at networks, people are always trying to compare apples and oranges," he explained. "An Ethernet is good for high-speed data transfer *within* a building. But it does not work with unintelligent peripherals." Metcalfe went on to explain that you would use another type of network to transfer data to a remote location. "There is a hierarchy of network use," he said. "Just as you drive a car to get to the airport to take a plane, you use a variety of networks to move your data."

When I asked Dr. Metcalfe what he saw in store for the future, he said he believed the price of networking hardware would approach zero. "You will no longer have *add-on* hardware for networking—it will be built in just the way AppleTalk is built into the Mac," he said. "You will see a transformation of perception; networking ability will be counted in as part of a system without anybody having to think about it. In the next few years, network ports will be taken for granted the way RS-232 ports are today."

But he went on to say, "I see networks of PCs as a major challenge to UNIX. They will implement some functions with greater

"When looking at networks, people are always trying to compare apples and oranges."

efficiency than minicomputers . . . I see the importance of the multiuser capability of UNIX taking a back seat to its multitasking ability. The LAN will replace the need for a multiuser system in many applications; multitasking will be more important than ever."

Dr. Metcalfe said that developing higher level standards is the greatest obstacle to the proliferation of networking. "This is a very complicated issue," he conceded. Much of the complication stems from the need for a network to connect different types of hardware, each running a different operating system.

"One scenario is that there will be two ways of doing business in the future," Metcalfe elaborated. "The first will be IBM's way, based on SNA. The second will be an international standard: I see protocols based on TCP/IP and XNS evolving toward that stan-

SOFTWARE DEVELOPERS!
YOU'VE JUST BEEN GIVEN THE BEST REASON YET TO GET OUR 68000/UNIX® DEVELOPMENT SYSTEM

THE VAR/68K® SERIES

VK-5XW20 (List price ~~$10,100.~~) **$5,000.**
Includes: Terminal, 20 Mb hard disk, floppy drive, 512K RAM, 8 ports and REGULUS®

VK-5XW20T20 (List price ~~$12,900.~~) **$6,500.**
Includes: all of above, plus 20 Mb tape streamer

RESELLERS!

Even more attractive specials are available to qualified resellers!

Smoke Signal has been designing, developing and manufacturing microcomputers based on the Motorola family of processors for the past six years. The VAR/68K is the most recent addition to our family of multi-user computers.

VAR/68K is a registered trademark of Smoke Signal
REGULUS is a registered trademark of Alcyon Corp.
UNIX is a registered trademark of Bell Laboratories

Due to the extremely low prices being offered, we can only accept cash or C.O.D. orders, and we must limit purchases to one per customer. Offer expires July 31st.

TO OBTAIN YOUR VAR/68K AT THESE LOW PRICES, CONTACT:
SMOKE SIGNAL
31336 VIA COLINAS
WESTLAKE VILLAGE, CA 91362
(818) 889-9340 / Telex 910-494-4965

Circle No. 39 on Inquiry Card

adax

X.25 FOR UNIX* Communications System

- Efficient, error-free data transmission to multiple hosts via international standard X.25, the only fully certified error-free public networking system used world-wide.
- User utilities
  - Remote user login
  - Remote mail service
  - Remote file transfer
- Compatible with widest number of host computers.
- Hardware available for VME, Multibus and others.
- Previously certified on TELENET, TYMNET and UNINET networks.
- Lowest cost per node.

Adax, Inc.

737 Dwight Way
Berkeley, CA 94710
(415) 548-7047

* UNIX is a trademark of Bell Laboratories.

Circle No. 38 on Inquiry Card

dard through the International Standards Organization (ISO)."

SPEAKING OF IBM . . .

With the announcement of the IX/370 operating system, IBM has finally thrown its weight behind UNIX System V, Release 2. The system AT&T considers standard will now run on the new IBM Sierra mainframes as a guest under the VM/SP operating system host. IBM says it will back the product with "full IBM support and service." This traditional sort of IBM support and service was never accorded to IBM's only other UNIX mainframe offering, VM/IX.

IX/370 customers will receive an enhanced System V. IBM has increased the block size to 4K bytes, and has given special treatment to short records. The new offering supports record locking, but does not include standard System V games, **crypt**, **cu**, or the online manuals. Perhaps most important, though, IX/370 *does* take advantage of the Sierra's virtual memory capability. Each process running under IX/370 will have access to up to 8 MB of virtual memory.

The significance of IBM's announcement is threefold. First, it is important that IBM has taken the step from System III (PC/IX and XENIX on the PC/AT) to System V. With both IBM and AT&T supporting System V, along with Microsoft's commitment to bring XENIX up to the System V standard, there can be little question but that a standard version of UNIX has been established.

Second, by offering IX/370 on the Sierra, IBM has added yet another piece of hardware to its growing arsenal of machines capable of running UNIX.

Third, IBM's announcement acknowledges that UNIX is an important market in which customers and potential customers of Big Blue have invested time and money. Part of the announcement reads, "IX/370 . . . enables customers to retain their UNIX-based program investment while taking advantage of System/370 power, function, and growth capabilities . . . "

A BRIGHT FUTURE

Along with a $1000 price cut, Morrow Designs has announced that a new type of display will be available for its portable Pivot machine. The Pivot, with two drives, 256K of RAM, and a new 16-line display, will sell for just under $2000.

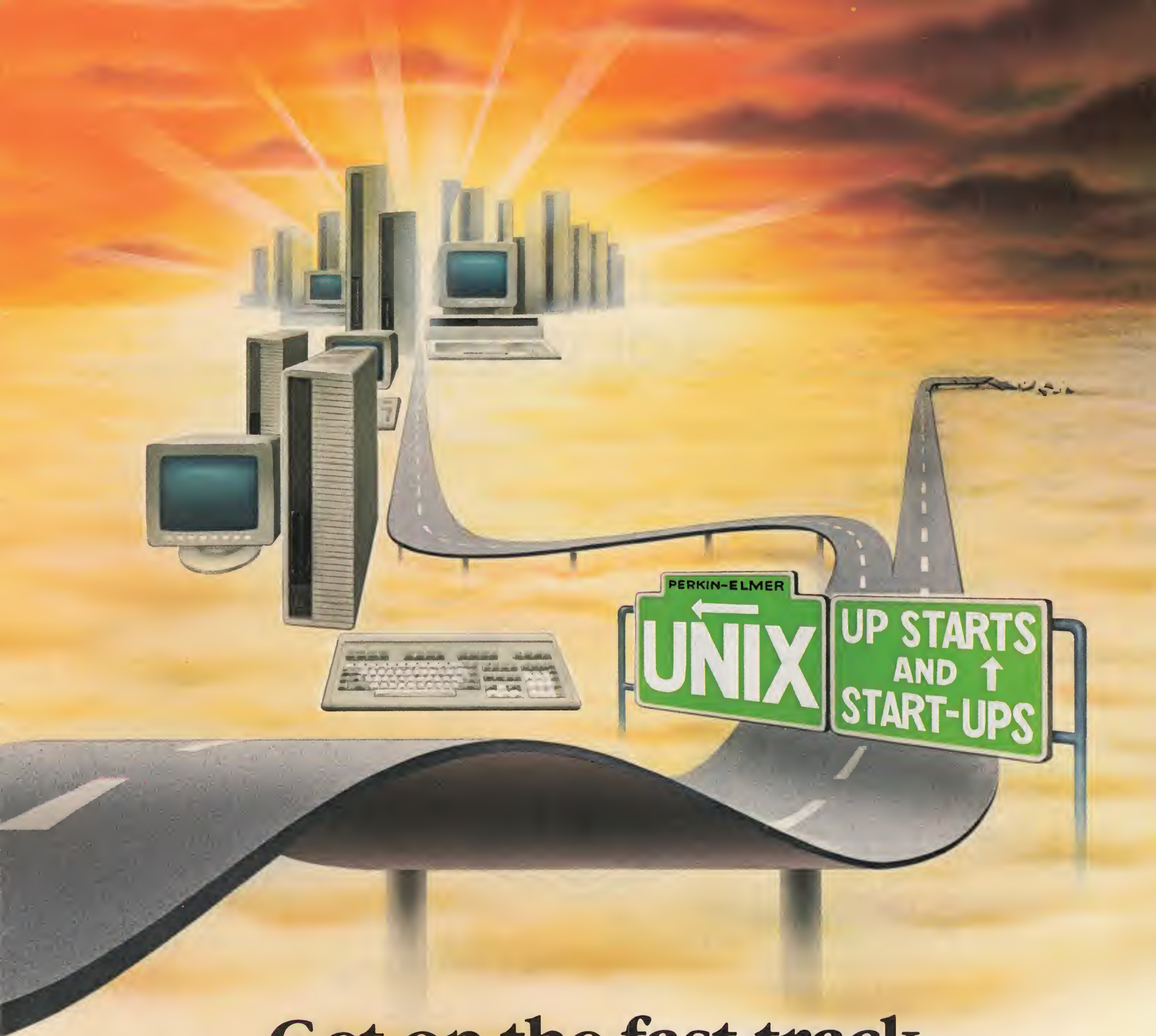
The new display uses a electroluminescent (EL) panel that's just 1/32nd of an inch thick behind a liquid crystal display (LCD) readout. Morrow claims the new display is easier to read in any environment than a conventional LCD display and that it consumes less power than characters generated by EL technology alone.

A 25-line version of the display is planned for release over the summer (the cost will be just under $3000). To maintain sales in the interim, Morrow has promised that when the larger displays become available, it will replace the 16-line displays with the larger ones for the difference in cost ($1000).

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

---

*Mark G. Sobell is the author of "A Practical Guide to the UNIX System" (Benjamin/Cummings, 1984) and "A Practical Guide to UNIX System V" (Benjamin/Cummings, available May, 1985). Mr. Sobell has been working with UNIX for five years and specializes in documentation consulting and **troff** typesetting. He also gives classes in advanced shell programming and **troff** macro development.* ■



Get on the fast track.

For low cost UNIX* applications turn to Perkin-Elmer's 7350A supermicro.

Perkin-Elmer gives you a head start in the growing world of UNIX system computing with the fastest supermicro in its class.** And we can keep you ahead because we're a Fortune 300 company with nearly a decade of experience with the UNIX operating system.

Move forward confidently with a fully supported, cost-effective solution; up to five users can share a single system. For the long haul, go the distance with the broadest range of compatible superminicomputers available today.

Why risk your future with start-ups or upstarts? Let Perkin-Elmer help put you on the path to success.

Choose the fast track: call 1-800-631-2154.

Or write to the Perkin-Elmer Corporation, Two Crescent Place, Oceanport, N.J. 07757. In N.J. call (201) 870-4712.

*UniPlus, UNIX, RM-COBOL and SIBOL are trademarks of UniSoft, AT&T Bell Laboratories, Ryan-McFarland and Software Ireland, respectively.

**In nine UNIX benchmark tests, the 7350A outperformed the published results of similar supermicrocomputers in its price range, while comparing favorably with many other larger and more costly systems. Prices and specifications subject to change without notice.

SPECIFICATIONS & OPTIONS

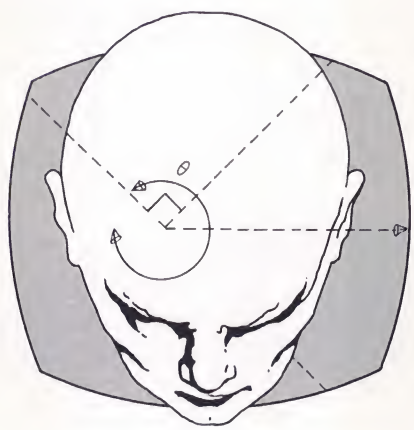
- 8MHz MC68000
- 512KB to 3MB RAM
- Floppy Disk
- 15MB or 40MB Winchester
- 12″ or 15″ Monochrome; 13″ Color Graphics Monitors
- 4 x RS-232C ports (Ethernet optional)
- UniPlus* port of UNIX System III
- MenuMaker/C Shell/AT&T Shell
- Word Processing, Spreadsheet, Relational DBMS, Graphics Libraries
- C, FORTRAN, BASIC-PLUS, RM-COBOL,* SIBOL*
- System prices start at $5,115 (quantity 100, U.S. only)

PERKIN-ELMER

PROBLEM SOLVER

Automated system administration

by Bob Toxen



These days, almost everything can be done faster, more reliably, and less expensively by computer—even the administration of computers themselves! It is amazing how much time a system administrator can save by automating daily chores. Among the tasks that can be automated are: system startups, checks for file system consistency, backups, and the addition of new users.

Of these, one of the most critical—and commonly automated—procedures is system startup. Smooth startups are vital to the integrity of file systems and the data they contain, so it's little wonder that many hardware vendors bundle automatic boot programs with the systems they sell.

The first step in bringing a system up is to press the system's boot button, sometimes called the *reset* button. (Even this can be automated on a VAX, but we will not cover that here.) After pressing the boot button, the system usually enters a state referred to as the *PROM monitor*. At this point, a command must typically be entered to start up UNIX. The command's content depends on the make and model of computer. On some systems, such as Silicon Graphics' IRIS, the system can be configured via DIP switches to boot UNIX automatically when the boot button is pressed, thus making the PROM monitor transparent to users. Such an approach is a good first step to automation.

Irrespective of how UNIX is booted, though, it will generally come up in *single-user mode*. At this point, an administrator will usually invoke **fsck** and remove daemon locks for **lpr**, **uucp**, or any other process—before bringing the system up in *multiuser mode*. This, too, can be automated. I usually make an entry in my /.cshrc file to determine whether I'm in single-user mode. If I am, the system starts a background process that brings on multiuser mode within 30 seconds. Thus, if there is a reason not to enter multiuser mode, I have plenty of time to kill the background process. The *.cshrc* sequence that provides this function for **csh** is:

```
if ( $$ < 5 ) then
        (sleep 30;multi)&
        echo 'Going Multi in 30 seconds
          (kill -9 $child to abort)'
endif
```

The alias you choose for *multi* depends on the version of UNIX you have. For example:

```
Version 7 and Berkeley 4.x:
      alias multi 'kill -9 $$'

System III:
      alias multi '/etc/init 2'

System V:
      alias multi '/etc/telinit 2'
```

This **alias** device generally cannot be used if one's single-user shell comes up in the Bourne shell because of a bug in **init** that fails to tell the shell that it is a login shell. The Bourne shell, moreover, looks at the *.profile* file rather than the *.cshrc* file for login shells.

For those interested in fixing the bug in **init**, note that the Bourne shell should be invoked as a login shell (with a name starting with a dash (–) in *argv[0]*). In System III, this can be done by linking */bin/sh* to */bin/-sh* and specifying the single-user shell as */bin/-sh*. This can be dangerous, though, since someone may think that *-sh* was accidentally placed in */bin* and so delete it. In that event, you would want to refer to my previous column on file system repair (October, 1984) since your system will subsequently not boot. In System V, the bug can be fixed by changing the line in **init** reading:

```
execlp(SU,SU,0);
```

to:

```
execlp(SU,SU,"-",0);
```

In other versions of UNIX, change the line looking something like:

```
execl("/bin/sh","sh",0);
```

to:

```
execl("/bin/sh","-sh",0);
```

A different shell may also be specified here if desired.

As soon as the system comes up, **fsck** must be run—before any disk writes occur, apart from the updating of a few inode access times. It is convenient to invoke **fsck** at the start of the */etc/rc* file, which is itself invoked as a shell script by **init** at the outset of multiuser mode. If one enters multiuser mode automatically, as discussed above, it is mandatory that **fsck** be run as the first item in */etc/rc*, unless it is done in the single-user *.cshrc* file. This is typically done as follows:

```
echo "Checking the file systems for consistency."
fsck
```

near the top of the */etc/rc* file. In System III and V, these lines should, of course, appear inside the **if** or **case** statement for multiuser mode (usually state 2). (System III and V offer many states besides single-user and multiuser modes. The */etc/rc* file is invoked when any of them are entered, and supplied with arguments that indicate what the new mode is, how many times it has been entered before, and what the previous mode was. System III even invokes */etc/rc* when single-user mode is entered. But

CEEGEN-GKS
GRAPHICS SOFTWARE
in C for Unix

- Full Implementation of Level 2B GKS
- Outputs, Inputs, Segments, Metafile
- Full Simulation for Linetypes, Linewidths, Fill Areas, Hatching
- Circles and Arcs, Ellipses and Elliptic Arcs, Bezier Curves
- Ports for Version 7, System III, System V
- Terminal, Plotter, Graphics Printer, Digitizer Drivers Available
- End-User, OEM, Distributor Discounts Available

Sales Reps and Distributor Inquires Invited.
Ceegen Corporation
20 S. Santa Cruz Ave., Suite 102
Los Gatos, CA 95030
(408) 354-8841
TWX: 9103506750 CEEGEN LG

CEEGEN is a trademark of Ceegen Corporation
Unix is a trademark of AT&T Bell Laboratories

Circle No. 41 on Inquiry Card

dyalog APL™

Version 3.0 Available Now!
The Reliable High Performance APL for UNIX* Systems

Dyalog APL is fast!
Version 3.0 is up to 10 times faster than previous versions!

Dyalog APL is functional!
Nested Arrays
Full Screen Editor
Full Screen Data Manager
Event Trapping
Interface to all UNIX* Facilities
Optional Graphics

Dylalog APL is reliable!
Dyalog APL has been in commercial use for over two years and is available NOW for most UNIX* Systems so call or write today.

MIPS Software Development, INC.
31555 W. 14 Mile Rd. #104
Farmington Hills, MI 48018
(313) 855-3552

¹ Improvements are a function of system and usage
* UNIX is a trademark of AT&T Bell Laboratories

Circle No. 42 on Inquiry Card

all of this is fodder for another column.) Of course, the */etc/checklist* file should contain a list of device filenames describing file systems to be checked by **fsck**, specifying the block device for the root file system, and defining raw (character) device files for other file systems.

In some implementations, */etc/rc*'s standard input, output, and error are not directed to the console. Symptomatic of this is that either the echo message does not appear on the console, you cannot get acknowledgement of your responses to **fsck**'s questions, or some of **fsck**'s messages are lost. To work around this condition, surround the body of commands in */etc/rc* with:

```
(
```

and:

```
) < /dev/console > &1 > /dev/console
```

You may also have to use **stty** to reset the console's modes.

Once all file system checks have been performed, the next step would typically be to mount the file systems that are most often used. This is particularly important since */usr*, commonly one of the file systems to be mounted, must be accessible for many of the commands that users will subsequently enter. The following sequence of file system mounts is typical for a system with two disks that are each split into three partitions:

```
# md0b is the swap device
echo "Mounting /dev/md0c as /usr";
  mount /dev/md0c /usr
echo "Mounting /dev/md1a as /mnt";
  mount /dev/md1a /mnt
echo "Mounting /dev/md1b as /scratch";
  mount /dev/md1b /scratch
echo "Mounting /dev/md1c as /usr/src";
  mount /dev/md1c /usr/src
```

Next, the administrator must perform "clean-up" operations to minimize any damage possibly caused by an earlier crash:

TOWER™ POWER

Give your Tower more Power!

up to 1.2 gigabytes of disk and tape storage:

- removable disk
- fixed media disk
- disk and tape

SHA Computers, Inc.
RD#3, Tait Rd., Box 51
Saratoga Springs, NY
12866
(518) 587-5886

Personal Secretary™ word processor
profit•maker™ integrated accounting system
DIBOLIX™ DIBOL™ for UNIX™

Trademarks Tower, NCR Corporation, DIBOLIX, C DIBOL Ventures, UNIX, AT&T DIBOL, Digital Equipment Corporation, Personal Secretary, Finished Software, profit maker, SHA Computers, Inc.

Circle No. 43 on Inquiry Card

ACUITY® business software is compatible with any budget, and all these systems:

UNIX-based Micros — VAX, VMS or UNIX
PRIME — Convergent
IBM-PC — Harris

With prices from $700 to $6500 for a fully supported package, any size company can afford our general accounting and specialized project cost software.

Packages available include project management, labor/ODC forecasting, work breakdown structure, customer order processing, bill of materials processing and inventory management.

Plus a complete set of accounting software including general ledger, payables, receivables, payroll and fixed assets.

Call (619) 474-2010 for details.

COMPUTER COGNITION

225 West 30th Street, National City, California 92050

Circle No. 44 on Inquiry Card

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

"UNIPLEX II"™

YOU'VE GOT IT!

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

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

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

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

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

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

OEM terms available. Mastercard and Visa accepted.

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

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

UniPress Software

Your Leading Source for UNIX™ Software.

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

Circle No. 18 on Inquiry Card

```
       echo "Preserving editor files"
/usr/lib/ex3.7preserve -
       echo "Clearing tmp dirs"
rm -rf /tmp /usr/tmp
mkdir /tmp /usr/tmp ; chmod 777 /tmp /usr/tmp
chgrp sys /tmp /usr/tmp ; chown root /tmp /usr/tmp
       echo "Removing locks"
rm -f /usr/adm/acct/nite/lock*
rm -f /usr/spool/uucp/LCK* /usr/spool/uucp/ST*
  /usr/spool/lpd/lock
       echo "Resetting logs"
cd /usr/adm ; cp sulog OLDsulog ;
  cp /dev/null sulog
cd /usr/adm ; cp cronlog OLDcronlog ;
  cp /dev/null cronlog
```

At this point, the administrator is ready to start up various daemons:

```
echo "Starting update"; /etc/update
echo "Starting cron";   /etc/cron
echo "Starting uucico"; /usr/lib/uucp/uucico -r1&
```

Some of the preceding commands are commonly incorporated into */etc/rc* files. An additional feature, though, that I've added to the */etc/rc* file on my M68010 workstation allows me to get the current time from a VAX by way of an Ethernet connection. To do this, I do a remote execution of **date** and use **sed** to change the output to a form suitable as a parameter to **date** on the local system. This allows the local system to set the date to within a few seconds of the VAX's date.

OTHER CANDIDATES FOR AUTOMATION

Incremental backups of disk file systems—that is, backups of all files that have been created or changed since the last full backup—are seldom done as often as they should due to the time and effort required. But incremental backups can be automated by making an entry in */usr/lib/crontab* that starts the process in the wee hours of the morning, when file systems are usually quiescent. (Even if there is some activity, the only files that won't be backed up are the ones that are actually changing at the time.) Backing up disk 0 onto a file on disk 1 and vice versa is usually safe.

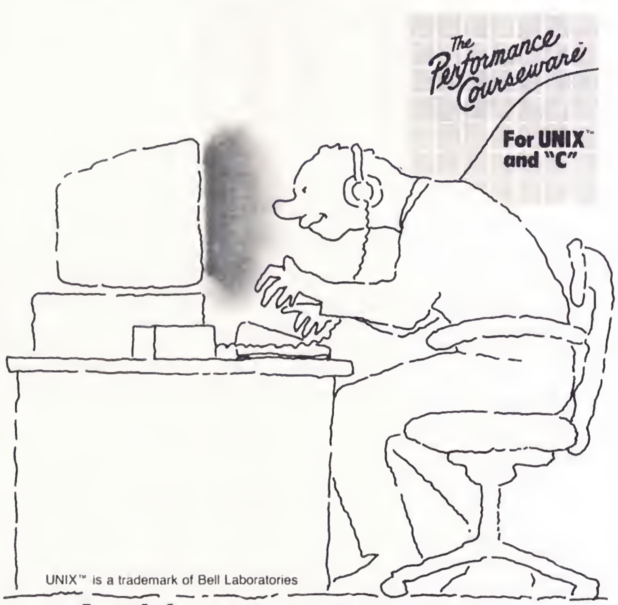
Alternatively, one could leave a tape in the tape drive, which is typically not used heavily, and backup onto it using a **crontab** entry. Of course, users should be warned not to leave write-enabled tapes in the tape drive. In one installation, I created a backup script that first checked to see if there was a tape in the drive. If so, it then read the first few blocks and would only perform the backup if it could ascertain that the mounted tape was, in fact, a backup tape.

Scripts for creating new user accounts offer another example of automated system administration. Such a script, run by root, can find the next unused **user-id** (perhaps kept in a file) and prompt for the account name, the person's name, and **group-id** to use. It could then add an entry to the */etc/passwd* file, create the home directory with the correct permissions and ownership, and copy in default *.profile*, *.cshrc*, *.login*, *.logout*, *.exrc*, *.mailrc*, and *.newsrc* files.

Many repetitive operations can be automated by creating a shell script or **csh** alias. Almost invariably, use of these tools results in less human effort and more efficient operation. Though I have discussed some of the most common automated procedures, you can probably think up many more that apply to your environment.

Bob Toxen is a member of the technical staff at Silicon Graphics, Inc. who has gained a reputation as a leading expert on **UUCP** *communications, file system repair, and UNIX utilities. He has also done ports of System III and System V to systems based on the Zilog 8000 and Motorola 68010 chips.* ■

The Best Learning Captivates



And it saves you money.

CALL NOW USER TRAINING CORPORATION
(408) 370-9710
591 W. Hamilton Ave. • Campbell, CA 95008

Circle No. 45 on Inquiry Card

TEXT EDITING

NEW RELEASE

UNIPRESS EMACS™

VERSION 2

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

Subject: Multi-window, full screen editor.

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

New Features:

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

Features:

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

Price:

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

**Call for terms*

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

*OEM terms available.
Mastercard/Visa accepted.*

Trademarks of UniPress EMACS & MLISP, UniPress Software, Inc. UNIX & AT&T 3B Series, AT&T Bell Laboratories, VAX/VMS & Rainbow 100,+ Digital Equipment Corp. MS DOS, Microsoft Corp., WordStar, MicroPro

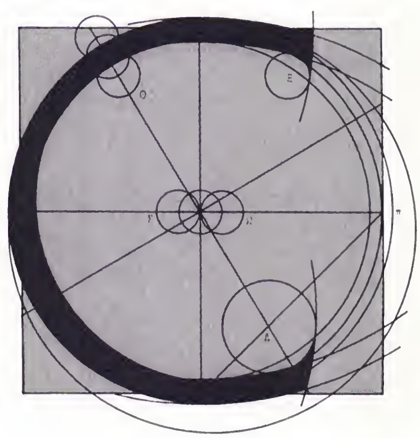
UniPress Software

Your Leading Source for UNIX Software.

C ADVISOR

IPC facilities in 4.2BSD

by Bill Tuthill



Inter-process communication (IPC) involves the transmission of data between two or more programs. Version 7, System III, and 4.1BSD provide only one reliable method for communication between processes—pipes. One drawback to pipes is that they can only facilitate communications between processes that are related through a common ancestor. That is, one process must be the parent of another, or both must be children of the same parent. In some applications, such as switching and database management, this is difficult if not impossible to arrange.

System V provides a substantial set of IPC facilities. Perhaps the most useful is memory sharing, which allows two unrelated but cooperating processes to address the same data space. Also available are semaphores (transmitted integers), message passing (transmitted strings), and named pipes (which share no common ancestor). An upcoming column will discuss these facilities. The System V IPC facilities work on a single machine, but can't move between two machines on a network.

The IPC facilities in 4.2BSD, on the other hand, were designed to work over a network. Processes can establish a *socket* to a process on another machine, through which data can travel in either direction. Sockets are more general than the System V IPC facilities; semaphores and message passing could be simulated by sockets. Unfortunately, memory sharing is not available in 4.2, probably because it was hard to implement in conjunction with virtual memory.

DOMAINS AND SOCKETS

The architecture of 4.2 provides an extensible set of communication *domains*, or standardized address formats. The two currently implemented domains are the UNIX system (AF_UNIX), for communication on a single machine, and the Internet (AF_INET), for communication between machines using protocols defined by DARPA (the Defense Advanced Research Project Agency). Other possible domains include Xerox Network System and IBM SNA.

In a domain, communication takes place between the endpoints we've already looked at known as *sockets*. Each can exchange information with other sockets within that same domain. In the UNIX domain, sockets have assigned pathnames, such as */dev/xtalk*. In the Internet domain, communication takes place using the Internet Protocol (IP), defined by DARPA.

Currently, there are three types of sockets: 1) the *stream* socket (SOCK_STREAM) provides bidirectional, reliable, sequenced, unduplicated data flow. In this type of socket, there are no record boundaries; 2) the *datagram* socket (SOCK_DGRAM) provides bidirectional data flow that might not be reliable, sequenced, or unduplicated. This type of socket allows for record boundaries; and 3) the *raw* socket (SOCK_RAW) provides access to underlying communication protocols, without specific semantics of its own. It is intended for programmers who must develop new communications protocols.

Each socket has an associated *protocol*. Within the Internet domain, for example, datagram sockets use the User Datagram Protocol (UDP), while stream sockets use the Transmission Control Protocol (TCP). Like many things originated by the Defense Department, TCP is slower and more complicated than it might have been. Sockets using the Xerox Network Systems (XNS) protocol would likely be more efficient and easier to use than those using TCP. In fact, a socket type (SOCK_SEQPACKET) has already been allocated for the XNS protocol.

Sockets may be either connected or unconnected. Connected sockets are used whenever data exchange must be reliable. Unconnected sockets are more useful than you might think, because there are primitives for exchanging information between them: **sendto()** and **recvfrom()**. Typically, datagrams, which are an unreliable form of data exchange, are transmitted between unconnected sockets. A descriptor for an unconnected socket may be obtained in the following manner:

```
int sock, domain, type, protocol;
 ...
sock = socket(domain, type, protocol);
```

For example, the following call creates a datagram socket in the Internet domain:

```
sock = socket(AF_INET, SOCK_DGRAM, 0);
```

And this call creates a stream socket in the UNIX domain:

```
sock = socket(AF_UNIX, SOCK_STREAM, 0);
```

In both cases, a zero parameter indicates the default protocol. The upper-case constants are defined in the include file *<sys/socket.h>*.

An unconnected socket descriptor may be converted into a connected socket descriptor in one of two ways: by actively connecting to another socket; or by being associated with a name in the communications domain, and then accepting a connection from another socket. Generally, server processes called daemons are associated with (or bound to) a specific name. They then listen until there is a connection to accept. Client processes actively seek out connections to these server processes when they require a particular service.

For a client, connecting a socket requires a program to get the host address of a server machine, and the port address for a known service. Then a socket must be created and connected. After the

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

COMPILERS

AMSTERDAM COMPILER KIT

Subject: A complete Kit of compilers, cross compilers and assemblers.

The Amsterdam Compiler Kit is the only C and Pascal UNIX package which includes a wide range of native and cross tools. The Kit is also easily modifiable to support custom targets.

Features:

- C and Pascal compilers (native and cross) for UNIX machines.
- Host and target machines include VAX™ 4.1/4.2 BSD, PDP™-11/V7, MC68000™ and 8086.™ Cross assemblers provided for 8080,™ Z80,™ Z8000,™ 8086,™ 6800,™ 6809,™ 68000,™ 6502 and PDP-11.
- The Kit contains complete sources* of all programs, plus comprehensive internals documentation on how to make modifications needed to add a new program language or new target machine.

*A source UNIX or C license is required from AT&T.

Price:

Full Source System	$9950
Educational Institutions	995

Selected binaries are available – contact us with your machine type.

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

OEM terms available.
Mastercard/Visa accepted.

UNIX is a trademark of AT&T Bell Laboratories. VAX & PDP 11 are trademarks of Digital Equipment Corp. MC68000, 6800 & 6809 are trademarks of Motorola Corp. 8080 & 8086 are trademarks of Intel Corp. Z80 & Z8000 are trademarks of Zilog Inc.

UniPress Software
Your Leading Source for UNIX™ Software.

Circle No. 46 on Inquiry Card

program finishes, the socket should be closed.

On the server side, things are a bit more complicated. Once a socket is created, it should be bound to a service name. Then the server should indicate how many outstanding connections may be queued to wait for acceptance. Finally, the server should go into an infinite loop, accepting, servicing, and closing connections requested by clients.

Normally, data can be transmitted between connected sockets with **read()** and **write()**. For unusual situations, such as out-of-band data, or looking at data without reading it, programs can use **send()** and **recv()** instead. The new **select()** system call allows programs to multiplex I/O requests among multiple sockets and file descriptors.

There is a new set of library routines for dealing with the Internet. All are structured like **getpwent()**, which reads the */etc/passwd* file. They include: **gethostent()**, which reads the */etc/hosts* file; **getnetent()**, which reads the */etc/networks* file; **getprotoent()**, which reads the */etc/protocols* file; and **getservent()**, which reads the */etc/services* file.

EXAMPLES

Figure 1 shows how a client process would connect to a login server running on a different machine. This code is a pared-down version of the **rlogin** command.

The **getservbyname()** call returns the entry for login via TCP in the */etc/services* file; the **gethostbyname()** call returns the entry for the specified machine from the */etc/hosts* file. The library routine **bzero()** initializes the Internet socket address structure, and then **bcopy()** copies the host address from

```
#include <stdio.h>
#include <netdb.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>

main(argc, argv)
int argc;
char *argv[];
{
        struct sockaddr_in sin;
        struct servent *sp;
        struct hostent *hp;
        int sd;

        if ((sp = getservbyname("login", "tcp")) == NULL) {
                fprintf(stderr, "rlogin: tcp: unknown service\n");
                exit(1);
        }
        if ((hp = gethostbyname(argv[1])) == NULL) {
                fprintf(stderr, "rlogin: %s: unknown host\n", argv[1]);
                exit(2);
        }
        bzero((char *)&sin, sizeof(sin));
        bcopy(hp->h_addr, (char *)&sin.sin_addr, hp->h_length);
        sin.sin_family = hp->h_addrtype;
        sin.sin_port = sp->s_port;
        ...
        if ((sd = socket(AF_INET, SOCK_STREAM, 0)) < 0) {
                perror("rlogin: socket");
                exit(3);
        }
        if (connect(sd, (char *)&sin, sizeof (sin)) < 0) {
                perror("rlogin: connect");
                exit(4);
        }
        ...
}
```

Figure 1 — *An example of how a client process might connect to a remote login server.*

the **hp** structure to the Internet socket address structure. After creating a socket, the program can connect to the server machine's socket, whose Internet socket address the program now knows.

Figure 2 shows how the server machine would set up the login daemon for connections initiated by client machines. This code is even sketchier than in the previous example.

Until the **socket()** call, this program is similar to the last one. The call to **bind()** ensures that the server listens at its expected location; **bind()** associates a name with a previously unnamed socket. The call to **listen()** establishes a queue for five simultaneous connection requests; if more requests are received, they'll be refused. Then, the server goes into a loop (UNIX hackers know that an infinite **for** loop executes faster than an infinite **while** loop). The **accept()** call blocks (waits) until a client requests service; when **accept()** returns, its return value is checked to verify that a connection has indeed taken place. The server then forks a child and invokes the routine **serve_login()**, not reproduced in the example, which performs processing for remote logins. The socket used by the parent for queueing connection requests is closed in the child, while the socket created by accepting the connection is closed in the parent. The client's Internet socket address structure is passed to the **serve_login()** routine, which requires it for authenticating clients.

In the 4.2 kernel, pipes have been implemented using sockets, which has made them faster than before. Internally, pipes are established with the **socreate()** routine, a packaged version of the **socket()** system call that passes the socket descriptor as the second parameter, like so:

```
pipe()
{
        ...
        socreate(AF_UNIX, &rso, SOCK_STREAM, 0);
        socreate(AF_UNIX, &wso, SOCK_STREAM, 0);
        ...
}
```


MULTI-USER
OPERATING SYSTEM

MAC XL™ UNIX

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

Subject: Full-multi-user UNIX for the MAC XL.

MAC XL UNIX, the UniPlus™+ Bell Labs UNIX System V, as ported by UniSoft Systems, transforms your MAC XL into a low-cost, high performance multi-user desktop workstation.

Features:

- *The full multi-user system includes powerful UNIX utilities, C compiler and development tools, text processing tools, along with vi, csh and termcap.* ***New lower price*** *for full system-**$1,350.***
- *Supports Apple 5 and 10-Mbyte drives. Increased disk space is available with hard drives which range from 16 to 92 Mbytes.*

Optional UNIX Applications Available:

Unify® Multi-user relational database.
Lex™ Word Processing.
Q-Calc Spreadsheet.
UniPress EMACS™ multi-window text editor system.

Programming Languages Available:

SVS Fortran
SVS Pascal
SVS Basic +
SMC Basic 4
RM Cobol
Irvine ADA

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

Dealer terms and demonstration systems are available. Mastercard/Visa accepted.

MAC XL is a trademark of Apple Computer; UNIX is a trademark of AT&T Bell Laboratories; UniPlus+ is a trademark of UniSoft Systems; UniPress EMACS is a trademark of UniPress Software, Inc., Unify is a trademark of Unify Corp.; Lex is a trademark of ACE Microsystems.

UniPress Software
Your Leading Source for UNIX™ Software.

Circle No. 47 on Inquiry Card

```
#include <stdio.h>
#include <netdb.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>

main(argc, argv)
int argc;
char **argv;
{
        struct sockaddr_in sin, from;
        struct servent *sp;
        int sd, asd;

        if ((sp = getservbyname("login", "tcp")) == NULL) {
                fprintf(stderr, "rlogind: tcp: unknown service\n");
                exit(1);
        }
        bzero((char *)&sin, sizeof(sin));
        sin.sin_port = sp->s_port;
        /*
         * missing code:
         * disassociate server from controlling terminal







         */
        ...
        if ((sd = socket(AF_INET, SOCK_STREAM, 0)) < 0) {
                perror("rlogind: socket");
                exit(2);
        }
        if (bind(sd, (caddr_t)&sin, sizeof(sin)) < 0) {
                perror("rlogind: bind");
                exit(3);
        }
        listen(sd, 5);
        for (;;) {
                if ((asd = accept(sd, &from, sizeof(from))) < 0) {
                        if (errno != EINTR)
                                perror("rlogind: accept");
                        continue;
                }
                if (fork() == 0) {
                        close(sd);
                        serve_rlogin(asd, &from);
                }
                close(asd);
        }
}
```

Figure 2 — *An example of how a server machine would set up the login daemon for connections initiated by client machines.*

The variable **&rso** is the read socket, while the variable **&wso** is the write socket. Sockets in the UNIX domain could probably be used to simulate named pipes on System V. But, to my knowledge, nobody has tried this yet. The implementation of UNIX domain sockets is buggy, and so, in fact, may have no use except for implementing pipes.

FUTURE DIRECTIONS

The IPC facilities in 4.2BSD are fairly complicated. Perhaps this is because the services they perform are inherently complex. As an abstraction, the **socket()** system call is clean and elegant. Because it works between machines, it has a clear advantage over the semaphores and message passing available on System V. On the other hand, **bind()** and **listen()** are heavily influenced by ARPA standards, and probably do not embody ideal network design.

The **select()** system call, for I/O multiplexing, is extremely useful when writing highly interactive applications. It has already been picked up by Version 8, the Bell Labs research system that supersedes Version 7. I will swallow a lizard if **select()** does not find its way into AT&T mainstream UNIX (System V and its successors) by 1990.

Some computer scientists maintain that the remote procedure call (RPC) model is superior to the **select** model. Certainly, either can be duplicated using the other. RPC is probably easier for the programmer to use, but **select** may be more efficient to implement. Next month's column will discuss one implementation of RPC in some detail.

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

UNICOMP
Technical Type
Typesetting Service

- UNIX*, Troff, Wizard
- Compugraphic 8400 Typesetter
- Quality appearance of books, proceedings, newsletters.
- Specializing in technical documents.
- Documents accepted via magnetic tape, phone lines, or paper.
- For information, samples, or estimates, call

505/662-EDIT (3348)

1580 Camino Redondo
Los Alamos, NM 87544

*Unix is a trademark of Bell Laboratories

Circle No. 53 on Inquiry Card

WE'RE BUYING

Have you developed vertical market business software for microcomputers? Is it the "best of the breed" in your field? Is the product tested, proven and ready to go? But have you found that the investment in packaging, marketing, distribution and support for national exposure is far greater than you can handle? If you can answer yes, we want to talk to you.

Accountants Microsystems Inc., AMI, is America's leading vertical software marketer. We're a large venture-backed firm with six regional offices, high level channels of distribution and a national support system.

We are looking for vertical packages written in "C" and operating under MS-DOS and XENIX/UNIX. Maybe that's your software. Call today, or send a letter to our manager of product acquisition.

ami
ACCOUNTANTS MICROSYSTEMS INC.
3633 136th Pl. S.E.
Bellevue, WA 98006
(206) 643-2050

MS-DOS and XENIX are trademarks of Microsoft Corportation
UNIX is a trademark of Bell Laboratories

NOW GET BERKELEY ENHANCEMENTS FOR YOUR 3B2

Z Systems is now offering its port of the Berkeley cshell for your AT&T 3B2. Through enrollment in the 3B Subscription Service, you will get the first software release which includes:
- Berkeley 4.2 compatible
- csh (less jobs)
- ls (links to lf,lr, . . .)
- more
- strings
- and others

Your annual subscription to 3BSS also includes newsletters and future software releases for a nominal media and handling charge.

The cost is just $200/yr.

To order call 303-526-1633 major credit cards accepted

Z SYSTEMS, INC.
602 Park Point Drive
Genesee Business Center
Golden, CO 80401

Call for information on other software and hardware products.

Circle No. 55 on Inquiry Card

THE UNIX GLOSSARY

Network vernacular

by Steve Rosenthal

Note: only those aspects of the terms that concern networking are discussed in this listing.

arbitration—the process of deciding which sender ought to proceed when the network detects that two or more stations are attempting to send at the same time. One common method has all stations back off for a small, random amount of time, allowing one to try again before the others. In other networks, a priority system based on device address determines the order.

baseband—a type of network that transmits data as pulses rather than impressions on a high-frequency carrier signal. Baseband networks are less expensive and easier to maintain than broadband networks, but they can't carry the multiple channels or mix of data and video the broadbands can. Most local area networks used to connect UNIX systems are of the baseband variety.

bridge—a connection between two or more similar networks. Unlike a gateway, which connects dissimilar nets and therefore must translate between protocols or character codes, a bridge primarily serves to electronically connect networks and synchronize their timing.

broadband—refers to networks that send information riding on carrier waves, rather than directly as pulses. Broadband equipment is expensive because it must modulate data carriers during transmission and extract data during reception. Most often, television-type coaxial cable, fittings, and amplifiers are used to provide the actual transmission path. The total capacity of a broadband network can be split into channels, some of which can be used for voice or video.

bus—a network topology (way of organizing connections) in which one channel runs to all nodes on the network. Each node must be able to recognize which messages are addressed to it. Similarly, because all stations on a bus use the same channel to transmit, a bus system needs some form of arbitration that can determine which station has priority when several nodes wish to send messages simultaneously. The Ethernet is a well-known example of the bus topology.

contention—an attempt by two or more senders to use a channel at the same time. UNIX networks either use systems to prevent contention (such as token passing networks, which assign each sender a turn to transmit) or allow contention, in which case collisions must be detected (as in CSMA/CD systems, where hardware detects an error state if two or more senders start transmitting simultaneously).

CSMA/CA—an abbreviation for the phrase "carrier sense multiple access with collision avoidance". This mouthful refers to a method of sending data on a network. With CSMA/CA, every station on the network refrains from transmitting when they sense that another station is using the channel (hence the reference to "carrier sense"). But unlike collision detect systems, collision avoidance systems leave it up to higher levels of network software to detect when two or more stations start at the same time, and thus create a collision and corrupt the data.

CSMA/CD—an abbreviation for the phrase "carrier sense multiple access with collision detec-

A NEW DIMENSION IN DATABASE MANAGEMENT

MISTRESS is the fully relational database management system (RDBMS) for UNIX.* It features the Structured Query Language (SQL*) for the end user as well as standard programming interfaces to the C language for the DP professional. Advanced concepts include variable-length character fields, dynamic storage allocation, and B+ Tree indexing. MISTRESS has been designed exclusively for the UNIX environment and is totally written in C.

MISTRESS/32 is the advanced relational database management system for extended addressing UNIX products. MISTRESS/32 features enhanced capabilities for security, recovery and data integrity, as well as a fully integrated report writer and screen interface. MISTRESS/32 is the recommended system for more demanding applications.

*UNIX is a trademark of Bell Labs. IBM and SQL are trademarks of International Business Machines.

RHODNIUS Incorporated

10 St. Mary Street, Toronto, Ontario, Canada M4Y 1P9 (416) 922-1743 Telex: 06-986766 TOR.

Circle No. 20 on Inquiry Card

tion". With CSMA/CD, stations on the network refrain from transmitting when they sense that another station is using the channel (this is known as "carrier sense"). If two or more stations start up at once, the lower levels of hardware and software detect the error, abort the transmission, and try again. CSMA/CD needs no master control station, but it offers no guarantee that a message will be passed within a specified time interval.

datagram—a method of sending messages across networks that breaks up each message into packets that are sent independently (possibly along different routes) across the network before being reassembled at the destination. Because the datagram method allows messages to bypass bottlenecks, it offers higher throughput at the cost of greater complexity, caused by the need to reassemble messages and request corrections for erroneous packets.

E-mail—the generic term for electronic mail (and also claimed as a trademark by some electronic message services and programs). Most UNIX systems include a **mail** facility, as do most commercial networks, though the mail programs offered by the networks are typically more sophisticated.

Ethernet—a local data network system developed by Xerox, Intel, and Digital Equipment Corporation. Ethernet is a baseband system (one that transmits data in the form of pulses) that uses coaxial cable. Traffic flow is regulated using a CSMA/CD protocol (which stops transmission and resends data if two or more stations start sending at the same time). Under Ethernet, CSMA/CD runs at speeds up to 10 megabits/sec. In the seven-layer ISO model for networks, Ethernet covers steps one and two.

FDM—an abbreviation for "frequency-division multiplexing", a method of putting more than one signal on each communications pathway by dividing the possible spectrum of signals into frequency bands. FDM is used in some of the more sophisticated local area networks based on broadband technology (in which data rides on carrier signals).

file locking—on a multitasking computer system or network, "file locking" describes a software procedure that prevents a second process or user from accessing a file while it is already in use. AT&T has promised to add file locking to System V, and the feature already is included in other UNIX implementations.

gateway—a connection between two dissimilar networks. Typically, a gateway consists of a box or card that receives cables from two nets. Logically, the gateway takes messages, strips each transmission down to a level common to the two systems, and then builds messages in the form required by the destination system. A typical UNIX gateway might connect DECnet to Arpanet or ResNet. A bridge, by contrast, describes a connection between two or more similar networks.

IEEE 802—a set of standards developed by the IEEE (Institute of Electrical and Electronic Engineers) to describe the physical

Leading Edge Image Processing & Communication Technologies

Teknekron, based in the San Francisco Bay area, offers a diverse range of technical challenges in the areas of hardware and software development.

As a rapidly growing private company, we are successfully involved with the development of leading edge image processing and communication technologies. We are seeking **Project Leader** level and **Software Development Engineers** with a BSEE/CS or MSEE/CS and a minimum of 3 years' experience in the design and implementation of software in the following areas:

- Image processing
- Computer networking
- UNIX*/C
- Systems programming
- Operating systems & drivers
- Data communications
- HDLC
- AX
- 68000

For confidential consideration, please forward a resume to: **Director of Corporate Development, Teknekron, 2121 Allston Way, Berkeley, CA 94704.** An equal opportunity employer. PRINCIPALS ONLY, PLEASE.

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

Teknekron, Inc.

Circle No. 59 on Inquiry Card

and electrical connections of local area networks (LANs). There are actually several different groups of IEEE 802 standards that consider networks classified according to topologies or protocols.

interoperability—the ability to request through one machine on a network that a program be run on another machine. This feature is not supported in standard UNIX, but it is included in some networks.

ISO—short for "International Standards Organization", an international group whose seven-layer model for networks has been very influential in the design and characterization of local area networks.

LAN—an abbreviation for "Local Area Network", a connection between computers or computerized systems within a building or immediate neighborhood. Most LANs are limited to a range of several hundred meters to a few kilometers.

OSI—short for the "Open System Interconnection" Reference Model, the conceptual design for layered networks that was developed by the International Standards Organization (ISO). The OSI model separates the network into functional layers, allowing the internal workings of any layer or group of layers to be changed or improved without effect on the other layers—apart from those modifications that are necessary in the interface. Most UNIX local area networks are designed as layers, but they do not always strictly follow the OSI partitions.

overhead—the ratio of a network's addressing, data checking, and system control information to the useful data it sends.

packet—a grouping of data created by chopping a message into small blocks and adding routing and control information. The packets can then be sent independently and reassembled at the destination into a complete message. Most UNIX networks send data as packets.

record-locking—the exclusion of users from accessing (or sometimes simply writing to) a record in a file while another user is accessing that record. This prevents a corruption of data that might otherwise occur if each user made changes without accounting for information that others might be providing. Record-locking only affects the records in use, allowing other users to access the rest of the file. AT&T has announced its intention to add record-locking to System V, but as of this writing, the feature is only offered in non-AT&T UNIX dialects.

ring—a network topology (way of connecting stations) that routes messages through each station in turn on the network. Most ring networks use a token-passing protocol that allows a station to put a message on the network when it receives a special bit pattern. Many networks that are logical rings actually connect all wires at a central hub, thus taking on the physical appearance of a star.

server—a station on a network that handles special chores, such as disk storage, printing, or communications. A *dedicated server* handles only its particular task, while a *shared server* can take care of a number of tasks at the same time. When used by itself, the term "server" usually applies to a disk or file server, a device that provides high-speed disk storage for the network.

star—a network topology (way of making connections) that brings all links to a central node. It is used more often for switched circuits such as telephone ex-

UNIX* and C TRAINING
in
TEXAS
Regularly Scheduled Classes
in
AUSTIN
Dallas, Houston
and
ON-SITE
Classes Available

Call or write to register
or to get our schedule

TELOS CONSULTING SERVICES
815 Brazos, Suite 504
Austin, TX 78701
512/472-7244

*UNIX is a trademark of Bell Laboratories

Circle No. 57 on Inquiry Card

"C" UNIX*

CGS Computer Assoc., Inc., is presently providing applications & systems dev'lt to a wide variety of Fortune clients. Our continued expansion has created several career opptys in the Tri-State area. We req Sr. Analysts & Programmers with at least 2 yrs exp in Applications or Systems Dev'lt, using the following: "C" Language, UNIX, PASCAL, or PL/I. Familiarity with Systems, Internals, is highly desired. Working knowledge of network protocols, multi processor systems, along with hands-on Batch or Mini is helpful. Excellent compensation.

MR. GUS GUIDA
(201) 750-4141
CGS Computer Assoc., Inc.
1460 Rt. 9 North, Suite 201
Woodbridge, N.J. 07095

*Unix is a trademark of Bell Labs

Circle No. 58 on Inquiry Card

changes and PBXs than for local networks. Even non-star networks sometimes use a star shape, though, running all connections to a central hub. But at the hub of these non-star networks, the links are actually connected in a bus or ring, meaning that the network logically acts as if it had that topology. In these instances, the star shape merely serves as a convenience for running cables.

TCP—an abbreviation for "transmission control protocol", the method of regulating how computer systems talk to networks favored by the Arpanet. This software interface is frequently used to connect UNIX systems to larger networks.

TDM—an abbreviation for "time-division multiplexing", the sharing of a communications channel or other resource by assigning it to each user in turn for very short intervals of time. Although most local area networks share a bus or channel over time, TDM usually refers to the more structured timesharing used by large networks. The more demand-based allocation of resources typically employed by local area networks is termed "contention".

token ring—a type of network that uses a ring topology (where messages are passed from one machine to the next in a circle, with the station that recognizes the message address grabbing it as it goes by) and a token protocol (where each station waits to transmit until it has received a special bit pattern that circulates to each station in the network in turn). Token rings are popular in real-time and control networks because each station is guaranteed access to the network at regular intervals.

topology—the arrangement of pathways in a network. The most common are the *rings* topology (where messages pass through each station in turn), the *stars* topology (where messages pass through a central node), and the *bus* topology (where each message is presented to all nodes).

twisted pair—a connection made with two insulated wires wrapped around a common axis. Twisted pair wiring, which is also used for telephone circuits, is very inexpensive, relatively easy to install, and compact, but it can only carry data at moderate rates (generally under one megabit per second). It is used by many lower-speed local area networks as the connection medium.

virtual circuit—a connection established for the duration of a session. By maintaining the connection rather than routing each packet independently, the network saves the time and effort needed to add addressing information to each individual data packet. Virtual circuits are more popular on large networks, while local area networks more often use some variation on datagram methods.

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

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

WHEN SERIOUS PROGRAMMING IS YOUR BUSINESS...

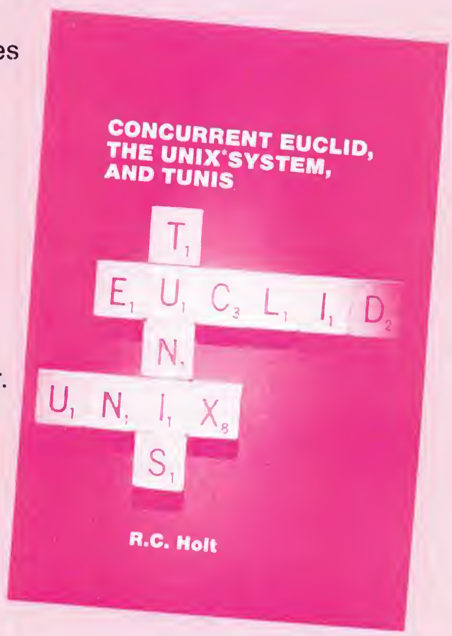
The Concurrent Euclid language for systems programming provides the best in efficiency, portability, reliability, and maintainability

Compilers running on UNIX/VAX, UNIX/11, VMS/VAX, with code generated for MC68000, MC6809, NS32000, 8086/8088 PDP-11, and soon running on IBM-PC

CONCURRENT EUCLID

Compiler: CSRI Distribution Mgr.
Sandford Fleming Bldg 2002
10 King's College Road
Toronto, Canada M5S 1A4
Tel: (416) 978-6985

Book:
CONCURRENT EUCLID,
THE UNIX SYSTEM AND TUNIS
Available from:
Addison-Wesley Publishing
Company, Reading, MA. 01867
Tel: (617) 944-3700



CONCURRENT
EUCLID

Circle No. 56 on Inquiry Card

Circle No. 63 on Inquiry Card ►

10TH ANNIVERSARY

Portland oregon

USENIX Conference and Vendor Exhibition
Marriott Hotel

June 11-14, 1985

USENIX

THE SPONSOR

For the latest in UNIX* applications and research, people look to USENIX, a not-for-profit association of individuals and institutions dedicated to fostering the development and communication and UNIX and UNIX-like systems and C Programming language. USENIX sponsors technical conferences and vendor exhibitions, produces and distributes a newsletter and serves as coordinator of a software exchange for appropriately licensed members.

THE TECHNICAL SESSIONS

The technical sessions emphasize issues of interest to the UNIX community. Suggested topics for formal papers, panel discussions and informal discussions will include the following:

▲ Text Processing ▲ Graphics
▲ Compilers and Software Distribution
▲ Network File Systems
▲ Programming Environments
▲ Standards Activities
▲ Multiprocessing ▲ Window Managers

TUTORIALS

In-depth tutorial seminars will be conducted all day by leading UNIX technical experts. Topics include:

- ▲ UNIX internals
- ▲ Networking
- ▲ Communications
- ▲ Applications
- ▲ C Programming

VENDOR EXHIBITION

Vendors are invited to display advanced technology relevant to the UNIX technical community.

For Complete Conference Information, Call:

(213) 592-3243 or (213) 592-1381
or Write:
USENIX Conference Office
P.O. Box 385
Sunset Beach, CA 90742

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

RECENT RELEASES

CIES OFFERS UNIX FOR IBM PCS

Two UNIX-compatible business systems have been introduced by CIE Systems, Inc. The CIES 680/100 and 680/200 business systems are designed for small to medium-sized businesses with first-time data processing users sharing computer resources and information, and for users with IBM PCs seeking entry into the UNIX world.

Depending upon each system configuration, the 4-12 user 680/100 is priced between $14,995 and $30,000. The 40-user 680/200 is priced at $29,995 and up. The systems are available 30 days ARO.

The CIES 680 family now consists of the 680/100 and 680/200 and the 1-4 user 680/20. Available with the systems is a range of existing CIES software, including word processing, accounting (with payroll) and spreadsheet applications packages; the applications processor PRO-IV; and PCworks, a networking application that links IBM PCs and PC-compatibles to the CIES 680 computers.

CIE Systems is located at 2515 McCabe Way, Irvine, CA 92713; 714/660-1800.

Circle No. 67 on Inquiry Card

SOFTWARE DEVELOPMENT PACKAGE

REX-TOOLS is a C programming package that converts most M68000-based host computers operating under System III, V, or Berkeley UNIX into a multiprogrammer, microprocessor software development system, has been introduced by Systems and Software, Inc.

The new product, which was designed for developing, testing, and debugging software targeted at Intel 8086, 8087, 8088, and 80186 microprocessors, integrates four separate tools: SoftProbe™ Simulator, C Cross Compiler, REX-SMA Assembler, and REX Real-Time Executive.

REX-TOOLS's SoftProbe Simulator is an interactive symbolic simulator that allows module program testing for the Intel 8086 and 80186 environments, supports debugging and logic analysis in both assembly and high-level language.

The simulator can execute all target chip CPU operations at the bus-cycle level, simulate I/O and external interrupts as well as CPU functions that operate on registers, instruction pointers, and status flags, and accepts the same absolute-code module format as an Intel in-circuit emulator.

The new product's C Cross Compiler package, which supports the full C language, includes an assembler, linker, and object code locater to allow development in both C and assembly language. The compiler generates ROMable object code aimed at real-time embedded systems applications such as automation devices, robotics equipment, and real-time controllers.

REX/SMA Assembler consists of four utility programs—a structured macro assembler (SMA), an object code liner, an absolute code locater and object librarian—that are compatible with the Intel ASM86/87/88 and 186 at source code, relocatable code, and absolute code level.

The Assembler's four programs ease transition between the Intel and REX language environments, permitting the software developer to write or edit directly in assembly language for the target processor.

The REX Real-Time Executive, the remaining tool in REX-TOOLS, is an event-driven real-time program that provides a number of functions including intertask synchronization, time-based synchronization, asynchronous event coordination, interrupt handling, memory management, and 8087 co-processor synchronization.

A 30-minute online demonstration of REX-TOOLS is available by telephone for those having a VT100 compatible terminal and a 1200 baud modem. The demonstration is available Monday through Friday, 9 am to 5 pm, PDT, by calling 714/241-8041. Both the user name and password for system log on are DEMO.

Available for immediate delivery, REX-TOOLS's starting price is $5500 (for the C Cross Compiler); quantity prices are available for hardware vendors and system integrators.

More information is available from Systems and Software, Inc., 3303 Harbor Boulevard, Suite C-11, Costa Mesa, CA 92626; 714/241-8650.

Circle No. 68 on Inquiry Card

Try listening to UNIX

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

UNIX™ REVIEW

THE MAGAZINE FOR THE UNIX COMMUNITY

UNIX is the trademark of Bell Laboratories, Inc.
UNIX REVIEW is not affiliated with or sponsored by Bell Laboratories.

MASSCOMP OFFERS MASS STORAGE SUBSYSTEMS DUO

Two mass storage subsystems have been introduced by Masscomp: a removable media disk and a 6250 bits-per-inch (bpi) half-inch mag tape.

Both system controllers can handle two drives and are field-installable. The media disk, with controller and one drive, is priced at $21,000; an additional drive is $15,000. The mag tape, with controller and one drive, is priced at $26,000; the second drive is $22,000. OEM discounts are available.

The disk pack, with MR-608 80 MB, 9-inch, totally-removable media, offers high data storage security. Front-loading, it has 67.4 MB of usable data storage, making it convenient for data backup.

The disk pack uses a standard SMD interface and measures 10½ inches high by 9½ inches wide. With a cabinet, it is rack-mountable. The optional second drive can be mounted side by side.

The mag tape offers a high system data throughput of 400 kilobytes-per-second. A high data storage density of 180 megabytes-per-reel makes it convenient for data backup. The mag tape is fully compatible and completely interchangeable with other computer systems supporting the ANSI tape standard.

Because it is a dual-density system, the mag tape can run at 6250 bpi for group code recording (GCR) data format or 1600 bpi for phase encoded (PE) data format. Both streaming and start/stop operation are possible.

For more information, contact Masscomp, One Technology Park, Westford, MA; 617/692-6200.

Circle No. 69 on Inquiry Card

INTERLAN EXPANDS NET/PLUS SERVICES

Interlan, Inc., has expanded the services of its NET/PLUS Systems Product Line with the introduction of a Network Communications Server/Internet Router and Network Management Utilities that run on the IBM Personal Computer. The internet router (NCS/IR) provides transparent

New from **Image Network**!

Documenter's Workbench®

for laserprinters and typesetters.

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

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

- *System V*
- *Berkeley 4.2*
- *VAX/Ultrix*
- *IBM/PC MS/DOS*
- *Eunice*
- *UniPlus+*
- *System III*
- *V 7*
- *VAX/VMS*
- *Amdahl/UTS*
- *Xenix*
- *UNOS*

- *DEC LN01s, LN03*
- *APS–5 typesetter*
- *Compugraphic 8400*
- *Xerox 2700, 3700*
- *Xerox 8700, 9700*

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

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

Image Network, (408)746-3754,
424 Palmetto Drive, Sunnyvale, CA, 94086-6760

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

Circle No. 48 on Inquiry Card

Q-CALC

A superior spreadsheet on UNIX*

As powerful as Lotus 1-2-3*

- large spreadsheet
- many business functions
- complete GRAPHICS package
- translates 1-2-3 models into Q-CALC
- already ported to: VAX, Callan, Fortune, Nixdorf, Cyb, Plexus, Codata, Cadmus, Masscomp, SUN, etc.

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

*Lotus 1-2-3 is a trademark of Lotus Development Corp. UNIX is a trademark of Bell Labs.

Circle No. 49 on Inquiry Card

NEVER BEFORE AVAILABLE!

- Passage, Uniq's TCP/IP Solution to UNIX SYSTEM V. 2 Networking.
- Ready for immediate delivery on Digital's VAX Processors.
- Local Area Network and DDN ARPANET Support.
- Uniq is committed to UNIX... UNIX Networking...UNIX Applications ...UNIX Consulting...We make it Work.

uniq

312/879-1008
DIGITAL TECHNOLOGIES
28 S. Water Street, Batavia, IL 60510

Los Angeles	Washington, D.C.	Boston
213/277-6288	703/448-8508	603/883-4860

*Unix and System V are trademarks of AT&T Bell Laboratories.

Circle No. 23 on Inquiry Card

data transmission between two physically separate Ethernets at speeds up to 224 Kbps. Packets are routed across one or two dedicated synchronous data links using the Xerox Network Systems (XNS) Internet Protocols (ITP). The Network Management Utilities run on the IBM PC, PC/XT, PC/AT, or COMPAQ Personal Computers.

The NCS/IR's use of the XNS ITP protocol guarantees full capability with Interlan's other system products.

The NCS/IR supports several forms of communication lines. Two 9600 bps links may be connected via an RS-232 interface. Alternatively, a V.35 or RS-449/RS-422 communications circuit may be connected at speeds up to 224 Kbps.

The Network Management Utilities provide a network operator with tools for controlling a large collection of services on an Ethernet. The utilities allow the PC management station to service boot requests from any server on the network. The ability to maintain load images for different types of network services allows the PC to function as a single primary boot server.

The Network Management Utilities include a single-user NTS program that gives the network manager full access to NTS user commands.

Network events such as requests for boot service may optionally be logged by the Network Management Utilities.

The Network Management Utilities also provide an integrated VT100 Terminal Emulator which allows the PC to interface to other Network Terminal Servers (NTS) on the Ethernet. The Network Management Utilities include a PC-compatible Ethernet controller board and diagnostic software. The package is priced at $1000.

Interlan is at 3 Lyberty Way, Westford MA 01886; 617/692-3900.

Circle No. 72 on Inquiry Card

PACT CONNECTS IBM MAINFRAMES WITH PCs AND WORKSTATIONS FROM OTHER VENDORS

Network Research Corporation (NCR) and Spartacus, Inc., have entered into a joint marketing agreement to make available a connection between IBM mainframes, IBM and compatible personal computers, and non-IBM workstations.

The combination of NRC's Fusion LAN software and the KNET software from Spartacus will allow several hundred personal computers and workstations to be connected, via the Spartacus M-200 channel-attached Ethernet controller, to IBM and compatible mainframes using the VM operating system.

The Fusion-KNET combination provides file transfer (FTP), terminal emulation (Telnet), and interprocess communication between applications in both IBM and non-IBM computers. The performance of Fusion and KNET allows throughput of over one megabit to be achieved in many applications.

The agreement opens the door to the IBM mainframe environment for Fusion customers, and

UNIX*
JOBS
REGISTRY

National registry of candidates and jobs in the Unix field. Please give us a call; send a resume; or request a free Resume Workbook & Career Planner. We are a professional employment firm managed by graduate engineers.

800-231-5920
P. O. Box 19949, Dept. UR
Houston, TX 77224
713-496-6100

Scientific Placement, Inc.

*Unix is a trademark of Bell Labs

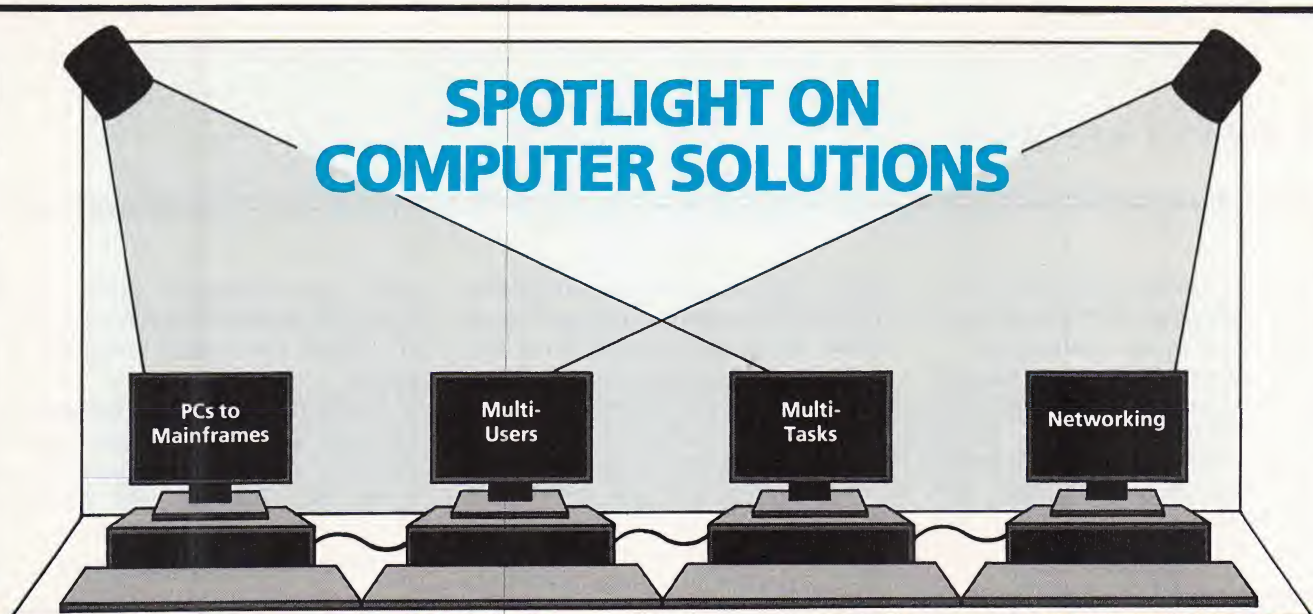
Circle No. 70 on Inquiry Card

FRANZ
THE FIRST
NAME IN
LISP

Franz Lisp from Franz Inc. is currently available on a wide range of machines under UNIX and VMS. Franz sets the standard for Lisp. Call or write for more information.

Franz Inc.
2920 Domingo, Suite 203
Berkeley, California 94705
(415) 540-1224

Circle No. 71 on Inquiry Card



UNIX* SYSTEMS EXPO/85

Spring

BE THERE!
For Computer Solutions To

- Office/Factory Automation
- Engineering/Scientific Applications
- Text Processing
- Database Access

BE THERE!
To Learn About

- Linking PCs to MainFrames
- Multi-User/Tasking Operations
- Networking
- Clusters/Device Sharing

April 24–26
San Francisco Moscone Center

See exhibits from the leading and most innovative hardware and software companies. Attend the full conference program with over 40 user/marketing oriented sessions.

Don't Miss UNIX Systems Expo/85 for:

- Engineers
- Scientists
- Corporate Users
- VARs
- Software Developers
- Vertical Marketers
- Systems Integrators
- Distributors
- Dealers
- Retailers

An exclusive production of
Computer Faire, Inc./
A Prentice-Hall Company
617/965-8350
415/364-4294

UNIX* SYSTEMS EXPO/85 *Spring*

*UNIX is a trademark of AT&T Bell Laboratories

Circle No. 22 on Inquiry Card

makes the range of machines currently supported by Fusion available to Spartacus customers.

For more information, contact Network Research Corporation at 1101 Colorado Avenue, Santa Monica, CA 90401; 213/394-7200. Also: Spartacus, Inc., at 5 Oak Park Drive, Bedford, MA 01730; 617/275-4220.

Circle No. 77 on Inquiry Card

NEW INFORMIX

Relational Database Systems, Inc., has announced the availability of Informix 3.3, the latest version of its relational database management system (DBMS).

Informix 3.3 offers users greater control over screen manipulation while using Perform, the form generation facility of Informix. The user has the advantage of additional procedural syntax to allow more specific control of the data maintenance process. Enhancements to Perform allow for calculations, cursor control, and conditional processing.

Informix is designed for the DP professional and systems integrator. In addition to the advanced Perform screen builder, Informix includes such features as a report writer, a query language, menu creation facilities, and an extensive C language interface.

RDS offers Informix on UNIX and UNIX-compatible systems. It and other RDS products are available on over 60 different machines.

RDS is located at 2471 E. Bayshore Road, Suite 600, Palo Alto, CA 94303; 415/424-1300.

Circle No. 76 on Inquiry Card

UNIX/C Professionals

Rewarding opportunities exist for **UNIX/C** professionals to work as consultants in the New York-New Jersey area. Knowledge of "**C**" (required), kernel, and communications are advantageous. **UNIX** guru's, wizards, and former "root's" especially welcome. Experienced people only.

The advantages of consulting such as job mobility, independence, and high pay are well known. While MAB Systems place people with various specialties, we have a special emphasis in the areas of **UNIX** and **C**. Contact us even if you aren't looking for a new position and we will send you our free "MAB Systems Guide to Consulting".

MAB SYSTEMS INC.
878 West End Ave. Suite #11D
New York, NY 10025
(212) 678-7881

UNIX is a trademark of AT&T Bell Labs

Circle No. 73 on Inquiry Card

RATIONAL INTERPRETER FOR C

Rational Systems, Inc., has released Version 1.00 of Instant-C, a C language interpreter for the full C language on microcomputers that combines interpreter ease-of-use and the speed of compiled code.

Instant-C executes programs 20 to 50 times faster than interpreted BASIC, and works several times faster than FORTH of **pcode**-based compilers.

Instant-C is available for the IBM PC and compatibles, and implements standard C, including **float**s and **long**s, so programs developed with Instant-C can be moved to other systems or compiled with an optimizing compiler.

The built-in full-screen editor displays source code errors with the cursor set to the trouble spot, and can list over 200 diagnostic messages.

Instant-C's debugger works in C language, not at the machine code level. Programs can be developed incrementally since any function or statement can be tested before the whole program is coded.

Instant-C is retailed for $500. To order or obtain further information, contact Rational Systems, Inc., PO Box 480, Natick, MA 01760; 617/653-6194.

Circle No. 75 on Inquiry Card

SIR/DBMS RUNS UNDER FOUR UNIX VERSIONS

SIR, Inc., has made SIR/DBMS, its relational database management system, available for four UNIX-based operating systems. SIR/DBMS now operates under 4.2BSD, Hewlett-Packard's HP-UX, Data General's DG MV/UX, and Apollo's AUX.

Included in the SIR/DBMS package is SQL+, an expansion of IBM's Structured Query Language. SQL+ is an interactive relational query system that lets users of SIR/DBMS interrogate their databases using English language commands, automatically displaying the information requested.

Special features of SIR/DBMS include: an active data dictionary; direct interfaces with BMDP, SAS, and SPSS statistical software packages; and flexible report generation including publication-quality tabular display.

In addition to DBMS and SQL+, SIR/DBMS includes the following integrated components: *Forms*, for interactive screen-oriented data entry and query-by-forms; *Host*, a language interface for access to one or more SIR/DBMS databases; *Help*, for full online documentation and user assistance; and *Graph*, for production of scientific and business graphics.

For more information, contact SIR at 820 Davis St., Evanston, IL 60201; 312/475-2314. ■

Circle No. 74 on Inquiry Card

BREAK THROUGH THE SOFTWARE BOTTLENECK

UNIX™ APPLICATION DEVELOPMENT

TODAY is far more than the awkward collection of tricks and tools that are often labelled "4GL". TODAY provides a **COMPLETE application development environment** that will revolutionize the way you develop and maintain applications. **No UNIX knowledge is necessary.**

Let's put it frankly : developing an application is a costly proposition. You'll need a highly skilled team of designers, analysts and programmers, and several man-years to get things off the ground. And that's not to mention the on-going costs of documentation, customization and maintenance!

TODAY tackles these problems through a new methodology with high performance architecture and a comprehensive range of features. It's so quick and easy to use that TODAY developers can do the whole job — design, analysis, development and documentation.

TODAY provides a comprehensive range of features that keep application building easy while optimizing development resources :

- **Powerful recursive logic and Decision Tables**
- **Synonyms, Menus, Prompts, Helps and Defaults for streamlined definitions**
- **Screen Painter**
- **A Report Generator which includes a Painter**
- **Push-button Self-documentation**
- **Audit Trails**
- **Source-code security through run-time only configurations**
- **Developed Applications instantly portable across UNIX-based machines**

Because definitions are **Dictionary-based,** any changes are easily made in one central location. A key feature, **"tailoring"** lets you alter an application — perhaps to customize it for a particular site or user — without affecting the original version. If required, applications can be set up as Models (Prototypes) and later enhanced to grow and change with the business. Tailoring versions is the perfect solution for quickly generating multiple applications based on one Model.

TODAY runs under UNIX or UNIX-compatible operating systems on **super-mini down to micro business computers** using any of a range of databases. And if that's not enough, TODAY is backed by 14 man-years of research and development and the confidence of users who are breaking time zones in software development. Watch for our stand at the UNIX Expo in San Francisco, April 24-26, or contact us at :

bbj Computer Services Inc.
2946 Scott Bvde
Santa Clara, CA 95054
Telephone : (408) 727-4464

Circle No. 79 on Inquiry Card

Unix is a trademark of A.T.& T Bell Laboratories Inc. TODAY. Copyright © bbj Computer Services Pty Ltd. Melbourne, Australia, July 1983

HARDWARE

Continued from Page 45

ery in the case of damaged packets or host failures much more complex than CSMA/CD. Therefore, CSMA/CD can be a good choice for unreliable local networks that cover five kilometers or less. On longer reliable networks, token passing is preferred.

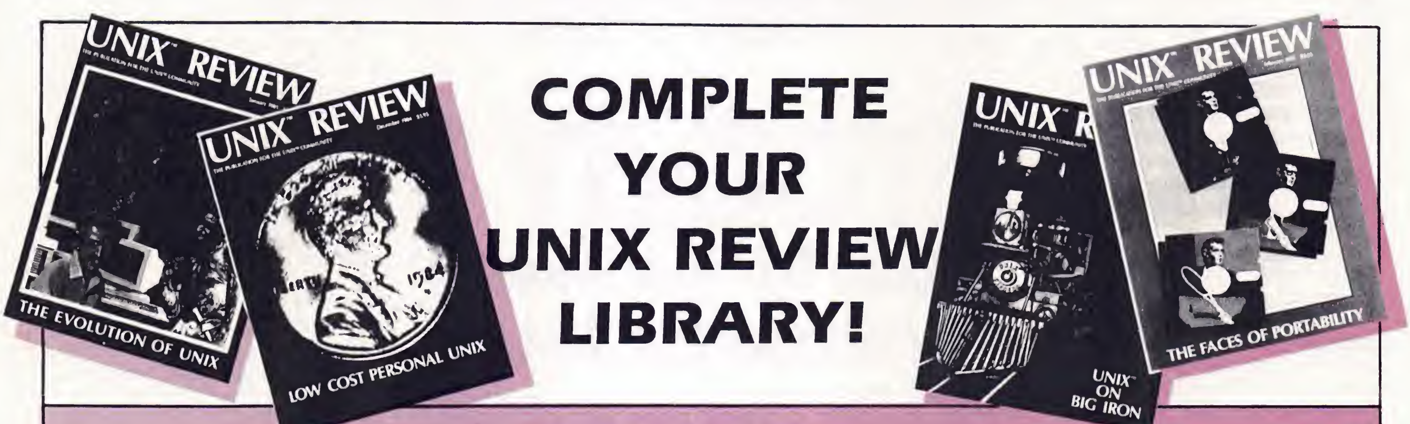
Note that CSMA/CD can be used *only* on relatively short networks, because both carrier sensing and collision detection require that the end-to-end round trip time of the packet be much less than the total time taken to send the packet. At 10 mbps, for instance, a minimum Ethernet packet of 72 bytes (576 bits) takes 57.6 microseconds to transmit. The Ethernet specification requires that the single bit end-to-end round-trip delay be less than 51.2 microseconds. At the physical transmission speed of a coaxial Ethernet, 51.2 microseconds translates into nearly 12 kilometers of cable, but other propagation delays and "repeaters" bring the maximum cable length between any two stations to 1.5 kilometers—with only a few microseconds to spare! Therefore, as signaling rates increase, the distances that CSMA/CD implementations can effectively service decrease, making it a useless protocol for very high speed media.

CONCLUSIONS

It is important to match hardware, software, and architecture to your network needs. If you don't need distributed computing bandwidth, don't pay for it. Also, if you are going to invest in a network today, plan on it being obsolete in five years.

The outcome of the race between competing network medias and topologies is simply not clear at this point. Should anyone ask, you can say that fiber-optics and rings appear to be favored . . . but don't bet on it.

Prior to taking his current post as Engineering Manager at Silicon Graphics, Inc., Bruce Borden developed the IP/TCP package for Excelan's Ethernet board and helped found 3Com Corporation. His first UNIX experience came 11 years ago when he set up Version 4 on Harvard University's undergraduate computing system. ■

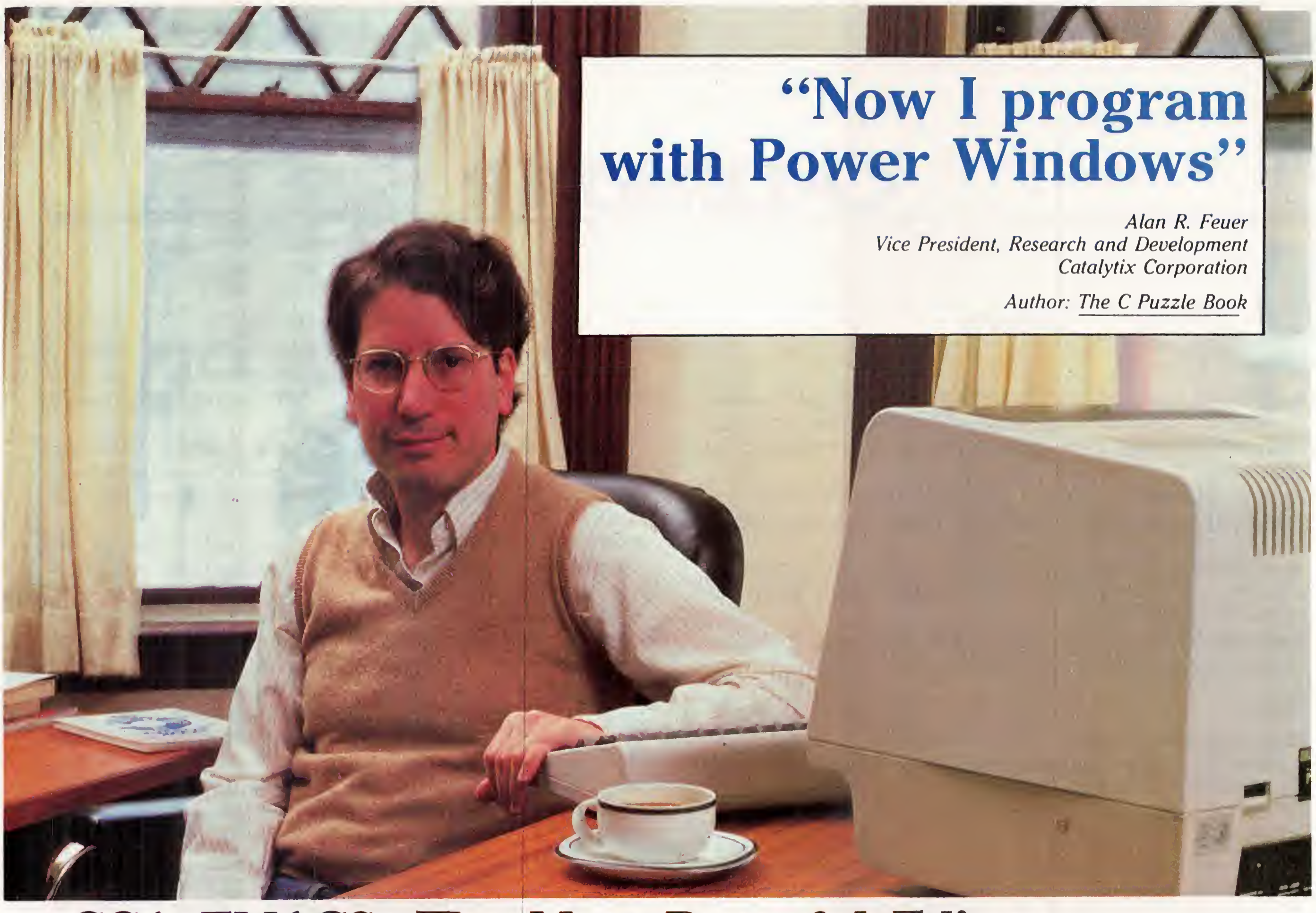


COMPLETE YOUR UNIX REVIEW LIBRARY!

June/July 1983—UNIX on the IBM/PC . . . . ☐
August/September 1983—Sritek and Venix . ☐
October/November 1983—UNIX Typesetting ☐
December/January 1984—Vi and Emacs . . . ☐
February/March 1984—UNIX Databases . . . ☐
April/May 1984—Menu-based User Interfaces ☐
June 1984—Big Blue UNIX . . . . ☐
July 1984—The AT&T Family . . . . ☐
August 1984—Documentation . . . . ☐
September 1984—System Administration . . . ☐
October 1984—UNIX on Big Iron . . . . ☐
November 1984—User Friendly UNIX . . . . . ☐
December 1984—Low Cost UNIX . . . . ☐
January 1985—Evolution of UNIX . . . . ☐
February 1985—UNIX Portability . . . . ☐

Back issues are $4.95 each including postage. Payment in advance is required. Send this order form with check (US funds payable at US bank only) or credit card information to: REVIEW Publications, 901 S. 3rd St., Renton, WA 98055.
Additional $1.00/issue for foreign mail.

Name ______________
Company ______________
Address ______________
City ________ State__Zip ______
M/C or VISA # ______________
Exp. Date ______________



CCA EMACS...The Most Powerful Editor Environment Available for Unix and VAX/VMS

"Programming with CCA EMACS, I can look at two or more files at once in different windows and then move text between them."

Alan Feuer is just one of many demanding programmers who have discovered that CCA EMACS™ makes program editing and system development much easier and faster. And "power windows" are only part of the reason Alan Feuer uses CCA EMACS...

Unprecedented power, speed, functionality, extensibility, pliability, and consistency across systems and on *any* terminal are others. CCA EMACS includes close to 400 built-in commands which let you do any job with only a few keystrokes, even the kinds of things that are difficult or impossible with other editors. And with our full Common Lisp-based extension language, Elisp™, you can customize CCA EMACS to meet all your specific program needs.

CCA EMACS has two extensive recovery facilities to protect against system failures. Supported by a full online documentation package, including tutorial, the system can be used by beginners and experts alike.

This complete kit of editing tools runs under Berkeley Unix™ (4.1BSD and 4.2BSD), Bell Unix (Systems III and V), Xenix™, and VAX/VMS™.
Binary prices range from $380 to $850 for Unix to $1900 for VMS.

CCA Uniworks, Inc.
Productivity Tools for Programmers
20 William Street, Wellesley MA 02181

Circle No. 16 on Inquiry Card

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

800-222-0214

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

Please send me information on:
- ☐ CCA EMACS
- ☐ The Safe C Development Tools
- ☐ AI Development Tools
- ☐ Your complete line of state-of--the-art programming tools

☐ Please send license forms

Name ______________________
Title ______________________
Company ______________________
Address ______________________
City, State, Zip ______________________
Phone (______) ______________________

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

A Crowntek Company

Unix, VAX and VMS and Xenix are trademarks of Bell Laboratories, Digital Equipment Corporation, and Microsoft Corporation, respectively. Safe C is a trademark of Catalytix Corporation. CCA EMACS and Elisp are trademarks of CCA Uniworks, Inc.   UR385

CALENDAR

EVENTS

APRIL

April 14-18 Association of Computing Machinery, San Francisco: "Human Factor in Computing Systems". Contact: ACM Conference Coordinator, 11 W. 42nd St., New York, NY 10036. 212/869-7440.
April 22-23 Uni-Ops, Walnut Creek, CA: "Post-Structural Programming Symposium". Contact: Uni-Ops, P.O. Box 27097, Concord, CA 94527. 415/945-0448.
April 24-26 UNIX Systems EXPO, San Francisco: "EXPO/85". Contact: Computer Faire Inc., 181 Wells Ave., Newton, MA 02159. 617/965-8350.
April 30 SVNet, Palo Alto, CA: "UUCP and Usenet". Contact: SVNet, P.O. Box 700251, San Jose, CA 95170-0251.

MAY

May 1 Yates Ventures, San Jose, CA: "AT&T, IBM, and UNIX: Desktop Directions". Contact: Glenn Chase, 3350 W. Bayshore Rd., Suite 201, Palo Alto, CA 94303. 415/424-8844.

JUNE

June 11-14 Usenix Association, Portland: "Usenix Conference and Vendor Exhibition". Contact: Usenix Conference Office, P.O. Box 385, Sunset Beach, CA 90742. 213/592-3243.

SEPTEMBER

September 18-20 National Expositions Inc., New York. "UNIX EXPO". Contact: Don Berey, 14 W. 40th St., New York, NY 10018. 212/391-9111.
September 26-28 8th Northeast Computer Faire, Boston. Contact: Computer Faire, Inc., 181 Wells Ave., Newton, MA 02159. 617/965-8350.

OCTOBER

October 2-5 UNIX Systems EXPO, Boston: "EXPO/85". Contact: Computer Faire Inc., 181 Wells Ave., Newton, MA 02159. 617/965-8350.

TRAINING

APRIL

April 1-2 Intelligent Solution, Washington, D.C.: "UNIX Concepts". Contact: Intelligent Solution, 849 22nd St., Santa Monica, CA 90403. 213/207-5356 or 800/367-0948.
April 3-4 Intelligent Solution, Washington, D.C.: "Programming in C". Contact: Intelligent Solution, 849 22nd St., Santa Monica, CA 90403. 213/207-5356 or 800/367-0948.
April 5 Intelligent Solution, Washington, D.C.: "UNIX Overview". Contact: Intelligent Solution, 849 22nd St., Santa Monica, CA 90403. 213/207-5356 or 800/367-0948.
April 8-12 Structured Methods Inc., New York: "UNIX System Workshop". Contact: SMI, 7 West 18th St., New York, NY 10011. 212/741-7720 or 800/221-8274.
April 8-19 Information Technology Development Corp., Cincinatti: "The UNIX System". Contact: ITD, 9952 Pebbleknoll Dr., Cincinatti, OH 45247. 513/741-8968.
April 9 Computer Technology Group, London: "UNIX Overview". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, Ill. 60604. 800/323-UNIX.
April 9-11 Bunker Ramo Information Systems, Trumbull, CT: "Diagnostic UNIX". Contact: Bunker Ramo, Trumbull Industrial Park, Trumbull, CT 06611. 203/386-2223.
April 9-12 Integrated Computer Systems, Boston: "UNIX: A Hands-on Introduction". Contact: Integrated Computer Systems, 6305 Arizona Pl., P.O. Box 45405, Los Angeles, CA 90045. 213/417-8888.
April 10-12 Computer Technology Group, London: "UNIX Fundamentals for Non-Programmers". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, Ill. 60604. 800/323-UNIX.
April 10-12 Specialized Systems Consultants, Bellevue, WA: "Hands-On UNIX for Programmers". Contact: SSC, P.O. Box 7, Northgate Station, Seattle, WA 98125. 206/367-8649.
April 15 NCR Corp., Dayton, OH: "UNIX Operating System". Contact: NCR Corp., CASE-Special Orders, 101 W. Schantz Ave., Dayton, OH 45479. 800/845-2273 or 800/841-2273.
April 15-16 Computer Technology Group, Dallas: "Advanced C Programming Workshop". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, Ill. 60604. 800/323-UNIX.
April 15-16 Computer Technology Group, San Francisco: "Advanced C Programming Workshop". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, Ill. 60604. 800/323-UNIX.
April 15-16 Bunker Ramo Information Systems, Trumbull, CT: "UNIX/C Applications". Contact: Bunker Ramo, Trumbull Industrial Park, Trumbull, CT 06611. 203/386-2223.
April 15-17 Computer Technology Group, London: "UNIX Fundamentals for Programmers". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, Ill. 60604. 800/323-UNIX.
April 15-19 Computer Technology Group, Boston: "C Language Programming". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, Ill. 60604. 800/323-UNIX.
April 15-19 Computer Technology Group, Washington, D.C.: "C Language Programming". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, Ill. 60604. 800/323-UNIX.
April 15-19 Structured Methods Inc., Washington, D.C.: "C Language Workshop". Contact: SMI, 7 West 18th St., New York, NY 10011. 212/741-7720 or 800/221-8274.
April 16-19 Integrated Computer Systems, Toronto: "UNIX: A Hands-on Introduction". Contact: Integrated Computer Systems, 6305 Arizona Pl., P.O. Box 45405, Los Angeles, CA 90045. 213/417-8888.
April 17-19 Computer Technology Group, San Francisco: "Advanced C Programming Under UNIX". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, Ill. 60604. 800/323-UNIX.
April 17-19 Computer Technology Group, Dallas: "Advanced C Programming Under UNIX". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, Ill. 60604. 800/323-UNIX.
April 17-19 Digital Equipment Corp., Chicago: "Comprehensive Overview of the UNIX Operating System". Contact: Digital Education Resources, 12 Crosby Drive, Bedford, MA 01730. 617/276-4949.
April 18-19 Computer Technology Group, London: "Shell as a Command Language". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, Ill. 60604. 800/323-UNIX.
April 18-19 Structured Methods Inc., New York: "Using **Lex** and **yacc**". Contact: SMI, 7 West 18th St., New York, NY 10011. 212/741-7720 or 800/221-8274.
April 18-22 Bunker Ramo Information Systems, Trumbull, CT:


UNIX EXPO
UNIX EXPO
The UNIX Operating System
Exposition & Conference
September 18–20, 1985
New York City, NY
UNIX EXPO will once again serve as the proven business and learning opportunity for professionals seeking to keep pace with the expanding UNIX universe. Join with thousands of your colleagues from around the world who will do business at UNIX EXPO. Take advantage of the best UNIX has to offer:
• An exhibition featuring over 200 of the leading suppliers of UNIX-based hardware, software and services.
• A multi-dimensional tutorial program addressing current technical topics and taught by the finest instructors in the industry.
• An expanded conference program focusing on issues vital to the computer professional.
"THE PROVEN UNIX MARKETPLACE"
Return the coupon for the details on all aspects of UNIX EXPO.
☐ I am interested in attending UNIX EXPO
☐ I am interested in exhibiting at UNIX EXPO
Name ______ Title ______
Company ______
Address ______
City ______ State ______ Zip ______
Return to: National Expositions Co., Inc.
14 West 40th Street, New York, N.Y. 10018
(212) 391-9111 • Telex 135401 DIMCOMM
*UNIX™ IS A REGISTERED TRADEMARK OF BELL LABS UNIX EXPO IS NOT AFFILIATED WITH BELL LABS

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

April 22 NCR Corp., Dayton, OH: "C Programming". Contact: NCR Corp., CASE-Special Orders, 101 W. Schantz Ave., Dayton, OH 45479. 800/845-2273 or 800/841-2273.

April 22-23 Computer Technology Group, Boston: "Shell Programming". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, Ill. 60604. 800/323-UNIX.

April 22-23 Computer Technology Group, Washington, D.C.: "Shell Programming". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, Ill. 60604. 800/323-UNIX.

April 22-23 Specialized Systems Consultants, Bellevue, WA: "Hands-On UNIX for Non-Technical People". Contact: SSC, P.O. Box 7, Northgate Station, Seattle, WA 98125. 206/367-8649.

April 22-26 Computer Technology Group, Dallas: "Berkeley Fundamentals and 'csh' Shell". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, Ill. 60604. 800/323-UNIX.

April 22-26 Computer Technology Group, San Francisco: "Berkeley Fundamentals and 'csh' Shell". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, Ill. 60604. 800/323-UNIX.

April 22-26 Computer Technology Group, London: "C Programming Language". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, Ill. 60604. 800/323-UNIX.

April 22-May 3 Information Technology Development Corp., Cincinatti: "UNIX System Administration". Contact: ITD, 9952 Pebbleknoll Dr., Cincinatti, OH 45247. 513/741-8968.

April 22-May 3 Information Technology Development Corp., Cincinatti: "Advanced C Programming". Contact: ITD, 9952 Pebbleknoll Dr., Cincinatti, OH 45247. 513/741-8968.

April 23-26 Integrated Computer Systems, Palo Alto: "UNIX: A Hands-on Introduction". Contact: Integrated Computer Systems, 6305 Arizona Pl., P.O Box 45405, Los Angeles, CA 90045. 213/417-8888.

April 23-27 Plum Hall, Somers Point, NJ: "Advanced C Topics". Contact: Plum Hall Seminars, 1 Spruce St., Cardiff, NJ 08232. 609/927-3770.

April 24 Bunker Ramo Information Systems, Trumbull, CT: "UNIX Marketing". Contact: Bunker Ramo, Trumbull Industrial Park, Trumbull, CT 06611. 203/386-2223.

April 24-26 Computer Technology Group, Washington, D.C.: "Using Advanced UNIX Commands". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, Ill. 60604. 800/323-UNIX.

April 24-26 Computer Technology Group, Boston: "Using Advanced UNIX Commands". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, Ill. 60604. 800/323-UNIX.

April 25-29 Bunker Ramo Information Systems, Trumbull, CT: "Advanced UNIX". Contact: Bunker Ramo, Trumbull Industrial Park, Trumbull, CT 06611. 203/386-2223.

April 29-30 Computer Technology Group, London: "Shell Programming". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, Ill. 60604. 800/323-UNIX.

April 29-30 Structured Methods Inc., Atlanta: "Introduction to UNIX". Contact: SMI, 7 West 18th St., New York, NY 10011. 212/741-7720 or 800/221-8274.

April 29-May 3 Bunker Ramo Information Systems, Trumbull, CT: "C Programming". Contact: Bunker Ramo, Trumbull Industrial Park, Trumbull, CT 06611. 203/386-2223.

April 29-May 3 Computer Technology Group, Boston: "UNIX Internals". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, Ill. 60604. 800/323-UNIX.

Let us prove how Cromemco systems can increase your satisfaction with UNIX System V.

Call, or visit, one of our Official System Centers today:

USA

In Arizona:
Artemis Computer 602/957-0469
Systems Solutions, Inc. 602/224-0026
Professional Data Systems, Inc. 602/265-6656

In California:
Quintec 818/889-4819
American Computer & Communications 415/348-1956
MCM Enterprises 415/327-8080
American Computers & Engineers 213/477-6751
Kierulff Electronics 213/725-0325
Accountability Systems 714/639-4570
Excalibur 916/972-9252
Kierulff Electronics 714/278-2112
Kierulff Electronics 408/971-2600
Cromemco, Inc. 818/346-6690

In Connecticut:
Datacraft, Inc. 203/673-6952

In Florida:
Automated Computer Systems 305/594-3819
Computer Centre 813/484-1028
Royal Data, Inc. 305/267-1960

In Georgia:
Cromemco, Inc. 404/391-9433
Systems Atlanta 404/928-0240
Kierulff Electronics 404/447-5252
Southern Exchange, Inc. 404/921-2662

In Illinois:
Commercial Data Systems 309/797-9401
Computerland 312/967-1714
Southern Exchange, Inc. 404/921-2662
Alpine Computer Center, Inc. 815/229-0200
Cromemco, Inc. 312/934-0650
Alternate Computer Services 312/893-1992

In Indiana:
Memory Bank, Inc. 312/891-1064 219/931-0203
Harbourtown Sales 317/877-4900
Microcomputer Specialists, Inc. 219/762-8541

In Kansas:
Tradewind Systems 316/624-8111

In Louisiana:
Muse Data Technologies 504/293-0320
Standard Systems, Inc. 318/625-8613

In Maryland:
Dynamic Data Processing 301/657-1211

In Massachusetts:
Kierulff Electronics 617/667-8331
Cromemco, Inc. 617/938-7010

In Michigan:
United Microsystems Corporation 313/668-6806
Jepson Group, Inc. 616/698-8700
Automated Business Consultants 313/478-0557

In New Jersey:
Kierulff Electronics 201/575-6750

In New Mexico:
South West Computer Stores, Inc. 505/292-6568

In New York:
Custom Computer Specialists 516/231-1155
C.C.S., Inc. 212/986-7520
Trexis, Inc. 914/268-5161

In Ohio:
Lucas Office Equipment & Service, Inc. 513/433-8484
Odyssey Systems, Inc. 216/526-9933
(ISIS) Innovative Systems/Integrated Software 419/531-0220

In Pennsylvania:
Modular System Design 412/521-6700
Marketline Systems 215/355-5400

In Texas:
Kierulff Electronics 214/343-2400
Gunn Enterprises 713/781-6911
Procomp 713/266-3648
Computer Crossroads of America 214/231-6108

In Virginia:
SMS Data Products 703/827-0640
Business Communications Systems 703/344-5563
VCA Corporation 703/281-4666

In Washington:
Kierulff Electronics 206/575-4420

In West Virginia:
Systems Support 304/766-7762

In Wisconsin:
Computer World 414/733-9547
Bay Tech of Wisconsin, Inc. 414/846-3038
Computer World 414/499-9983

INTERNATIONAL

In Austria:
Ing Stahe GmbH 222/575-6950

In Australia:
Minicomp Software & Education 61-2/957-6800

INTERNATIONAL cont.

In Australia cont.:
Micro Data PTY LTD. 61-9/328-1179
Insystems P/L 61-3/690-2899

In Canada:
Cro-Corp Computer Solutions 403/286-8459
D.E. Systems 613/729-5164
Future Electronics 610/421-3251

In Costa Rica:
Control Electronica 506/24-44-44

In Denmark:
Merkur Data Services A/S 45-2/63-01-55

In England:
Jarogate Ltd. 44-1/671-6321

In France:
Penaranda Informatique 56-979618

In Greece:
Algorithm Ltd. 370-1/933-0551/5858

In Hong Kong:
Vanda Computer & Equipment Co. 852-3/348702-5

In Israel:
Information Systems Ltd. 03-775111

In Italy:
CO.N.I.A. 39-51/375001

In Japan:
Asahi Glass 81-3/218-5848

In Mexico:
Micromex, S.A. de C.V. 905/687-8886/8913 905/536-5503

In Mid-East:
Multi Media Video, Inc. CA USA 408/727-1733
National Computer System Pakistan 5156644
Computer System Marketing Center Saudi Arabia 966/2-651-7707 966/2-653-0580

In The Netherlands:
Rocomp B.V. 31-40/524055

In Norway:
Micro Systems A/S 47-2/41-69-76

In Scotland:
Micro Centre Complete Micro 44-31/556-7354

In Sweden:
Datroisering Konsult AB 46-8/753-3090

In West Germany:
Cromemco GmbH 49-6196/481606
Digitronic Computer-systeme GmbH 4103/88672/73
Casy-X Computer Systeme GmbH 2173/52071/72
Comicro Deutschland 49-2151/795577
CE Computer GmbH 02151-22121
Anitex Computer Systems GmbH 08142-13091

CROMEMCO · OFFICIAL UNIX SYSTEM V SYSTEMS CENTER

Cromemco®

CROMEMCO COMPUTERS: DESIGNED TO MAKE UNIX SYSTEM V EVEN BETTER...

UNIX System V, the new standard in multi-user microcomputer operating systems, gives you high performance features along with the portability and flexibility of a standard.

Cromemco computers can make UNIX System V even better. Because our systems are designed with UNIX in mind. First of all, we offer UNIX System V with Berkeley enhancements. Then, our hardware uses advanced features like 64K of on-board cache memory and our high speed STDC controller to speed up disk operations–very important with UNIX.

More capability and expandability

We have a high-speed, 68000-based CPU that runs at 10 MHz, coupled with a memory manager that uses demand-paging and scatter loading to work *with* UNIX, not for it.

We provide room for expanding RAM to 16 megabytes–with error detection and correction–for running even the most sophisticated and advanced microcomputer programs. And the power to accommodate up to 16 users–all with plenty of memory.

But we give you even more.

A complete solution

We give you a choice in systems: the System 100 series, expandable up to 4 megabytes of RAM, and the System 300 series, expandable to 16 megabytes. A high speed 50 megabyte hard disk drive is standard on the systems. And you can expand the hard disk capacity up to 1200 megabytes using standard SMD drives. You can add floating point processing. High resolution graphics. Video digitizing and imaging. Communications through standard protocols. Mainframe interface.

And software support is here to meet your needs. We offer major programming languages, database management systems, communications software, including SNA architecture, X.25 protocol, and Ethernet; even a program to interface to an IBM PC if you need to. And, of course, access to the broad range of standard UNIX applications programs that is growing dramatically every day.

Easy to use.

We also make our systems easier to use, because we install the operating system before we ship your computer. No complicated installation procedures. And the Berkeley enhancements give you the standard UNIX System V operating system, but with the added convenience of these widely acclaimed improvements.

Cromemco's System 100 and System 300 computers: designed to be the highest performance UNIX systems available anywhere.

Just call or visit one of our UNIX System V Official System Centers to see for yourself. They'll also give you a copy of our new publication, "What you should know before you buy a UNIX system."

Or contact us directly.

We'll be glad to show you how to get a better UNIX system.

Corporate Headquarters: Cromemco, Inc., 280 Bernardo Avenue, P.O. Box 7400, Mountain View, CA 94039. (415) 969-4710. In Europe: Cromemco GmbH, 6236 Eschborn 1, Frankfurter Str. 33-35, P.O. 5267, Frankfurt Main, Germany.

Circle No. 26 on Inquiry Card

Cromemco®

UNIX is a trademark of Bell Laboratories.
IBM is a trademark of International Business Machines Corp.

April 29-May 3 Computer Technology Group, Washington, D.C.: "UNIX Internals". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, Ill. 60604. 800/323-UNIX.
April 29-May 3 Computer Technology Group, Chicago: "C Language Programming". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, Ill. 60604. 800/323-UNIX.
April 29-May 3 Computer Technology Group, Los Angeles: "C Language Programming". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, Ill. 60604. 800/323-UNIX.
April 29-May 10 Information Technology Development Corp., Cincinatti: "Advanced C Programming". Contact: ITD, 9952 Pebbleknoll Dr., Cincinatti, OH 45247. 513/741-8968.
April 29-May 10 Information Technology Development Corp., Cincinatti: "UNIX System Administration". Contact: ITD, 9952 Pebbleknoll Dr., Cincinatti, OH 45247. 513/741-8968.

MAY

May 1-3 Computer Technology Group, London: "Using Advanced UNIX Commands". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, Ill. 60604. 800/323-UNIX.
May 3 Specialized Systems Consultants, Bellevue, WA: "UNIX for Managers". Contact: SSC, P.O. Box 7, Northgate Station, Seattle, WA 98125. 206/367-8649.
May 6 NCR Corp., New York: "UNIX Operating System". Contact: NCR Corp., CASE-Special Orders, 101 W. Schantz Ave., Dayton, OH 45479. 800/845-2273 or 800/841-2273.
May 6-7 Computer Technology Group, Chicago: "Shell Programming". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, Ill. 60604. 800/323-UNIX.
May 6-7 Computer Technology Group, Los Angeles: "Shell Programming". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, Ill. 60604. 800/323-UNIX.
May 6-10 Bunker Ramo Information Systems, Trumbull, CT: "Advanced C". Contact: Bunker Ramo, Trumbull Industrial Park, Trumbull, CT 06611. 203/386-2223.
May 7-9 Computer Technology Group, Boston: "UNIX Administration". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, Ill. 60604. 800/323-UNIX.
May 7-9 Computer Technology Group, Washington, D.C.: "UNIX Administration". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, Ill. 60604. 800/323-UNIX.
May 7-10 Integrated Computer Systems, Washington, D.C.: "UNIX: A Hands-on Introduction". Contact: Integrated Computer Systems, 6305 Arizona Pl., P.O Box 45405, Los Angeles, CA 90045. 213/417-8888.
May 7-10 Computer Technology Group, London: "UNIX Internals". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, Ill. 60604. 800/323-UNIX.
May 8-10 Computer Technology Group, Los Angeles: "Using Advanced UNIX Commands". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, Ill. 60604. 800/323-UNIX.
May 8-10 Computer Technology Group, Chicago: "Using Advanced UNIX Commands". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, Ill. 60604. 800/323-UNIX.
May 13 NCR Corp., Los Angeles: "UNIX Operating System". Contact: NCR Corp., CASE-Special Orders, 101 W. Schantz Ave., Dayton, OH 45479. 800/845-2273 or 800/841-2273.
May 13-14 Computer Technology Group, Boston: "Advanced C Programming Workshop". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, Ill. 60604. 800/323-UNIX.
May 13-14 Computer Technology Group, Washington, D.C.: "Advanced C Programming Workshop". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, Ill. 60604. 800/323-UNIX.
May 13-17 Computer Technology Group, Chicago: "UNIX Internals". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, Ill. 60604. 800/323-UNIX.
May 13-17 Computer Technology Group, Los Angeles: "UNIX Internals". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, Ill. 60604. 800/323-UNIX.
May 13-17 Bunker Ramo Information Systems, Trumbull, CT: "Intro to UNIX". Contact: Bunker Ramo, Trumbull Industrial Park, Trumbull, CT 06611. 203/386-2223.
May 13-24 Information Technology Development Corp., Cincinatti: "The UNIX System" and "The C Programming Language". Contact: ITD, 9952 Pebbleknoll Dr., Cincinatti, OH 45247. 513/741-8968.
May 15-17 Computer Technology Group, Boston: "Advanced C Programming Under UNIX". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, Ill. 60604. 800/323-UNIX.
May 15-17 Computer Technology Group, Washington, D.C.: "Advanced C Programming Under UNIX". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, Ill. 60604. 800/323-UNIX.
May 15-17 Computer Technology Group, London: "UNIX Administration". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, Ill. 60604. 800/323-UNIX.
May 15-17 Digital Equipment Corp., Denver: "Comprehensive Overview of the UNIX Operating System". Contact: Digital Education Resources, 12 Crosby Drive, Bedford, MA 01730. 617/276-4949.
May 20 NCR Corp., Los Angeles: "UNIX System Administration". Contact: NCR Corp., CASE-Special Orders, 101 W. Schantz Ave., Dayton, OH 45479. 800/845-2273 or 800/841-2273.
May 20-21 Computer Technology Group, London: "Advanced C Programming Workshop". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, Ill. 60604. 800/323-UNIX.
May 20-24 Computer Technology Group, Boston: "Berkeley Fundamentals and 'csh' Shell". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, Ill. 60604. 800/323-UNIX.
May 20-24 Computer Technology Group, Washington, D.C.: "Berkeley Fundamentals and 'csh' Shell". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, Ill. 60604. 800/323-UNIX.
May 20-24 Bunker Ramo Information Systems, Trumbull, CT: "Advanced UNIX". Contact: Bunker Ramo, Trumbull Industrial Park, Trumbull, CT 06611. 203/386-2223.
May 20-24 Specialized Systems Consultants, Bellevue, WA: "C Programming Workshop". Contact: SSC, P.O. Box 7, Northgate Station, Seattle, WA 98125. 206/367-8649.
May 21-23 Computer Technology Group, Chicago: "UNIX Administration". Contact: Computer Technology Group, 310 S. Michigan Ave., Chicago, Ill. 60604. 800/323-UNIX.

Please send announcements about training or events of interest to: UNIX Review Calendar, 500 Howard Street, San Francisco, CA 94105. Please include the sponsor, date, and location of event, address of contact, and relevant background information.

SHACC UP WITH CONCENTRIC ASSOCIATES, INC.

Concentric Associates, Inc. announces a new software product:

Shacc—the **sh**ell **acc**elerator—is a compiler for the Bourne shell. It translates Bourne shell programs into C and then invokes the C compiler to produce an "a.out" file. The C code that is generated is well-structured and very readable, so it can be further optimized by hand if you like.

Shacc'd code:

- Is faster than interpreted shell code.
- Is more resistant to piracy.
- Is more space and time efficient—you can make good use of shared text and the sticky bit.
- Will work properly in setuid files.

Shacc allows you to write production code in the Bourne shell: Do the fast prototyping in shell, make it right, and then **shacc** it and ship it.

If your shell programming's not up to speed, call Concentric Associates. Also ask about our UNIX™ teaching and consulting services.

shacc
By Paul Ruel
Concentric Associates, Inc.

For further details, call or write:
Jim Butz/Concentric Associates, Inc./One Harmon Plaza/Secaucus, NJ 07094/201•866-2880

UNIX™ is a trademark of AT&T Bell Laboratories

See us at UNIFORUM, Booth 2218.

Circle No. 27 on Inquiry Card

THE LAST WORD

Letters to the editor

Dear UNIX REVIEW,

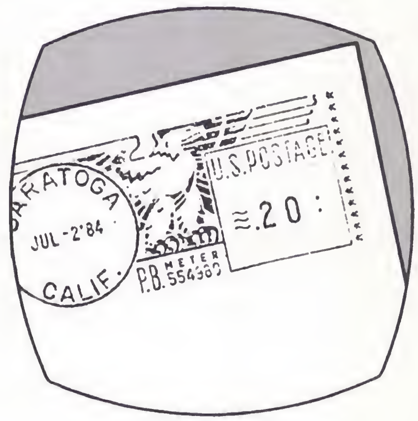
First off, congratulations on the magazine. The quality is up a whole lot. The thrust and balance is excellent. In fact, there is now so much fascinating material that I have to do a lot of reading and am now 1½ months behind.

While there has clearly been a much increased effort in technical editing, the millenium has not quite arrived. In the September issue, the shell in figure 8, page 37, was clobbered—probably by **nroff**. This sort of thing happened much more often under the previous watch. So...a request: could a supplement of corrected programs be compiled from the back issues by one of your hired gurus? And . . . a tip: backslashes get eaten by **troff**, unless (da dum!) one uses the apparently little known **nroff** escape sequence: "\e". This is the principle culprit in the **sed** command towards the top of the shell, as can be seen by examining the McIlroy shell referenced in the header.

I've also been able to make it through October (great issue!) and have started in on November. Almost immediately, I ran into the program "blues" problem again. On page 24, the application:

```
$ pick * | rm
```

does not seem to work (nothing catastrophic, but even if you respond 'y', the file is not removed). In case it matters, I'd better mention that the problem seems to be



with **rm** (or at least the 4.2BSD version thereof) rather than with **pick.**

Notice that Kernighan and Pike do *not* cite this usage of **pick**. They favor:

```
$ rm 'pick *'
```

which *does* work.

Sincerely,

Jack K. Cohen
Professor of Mathematics
Center for Wave Phenomena
Colorado School of Mines
Golden, CO

Thank you for your comments. Though we abhor technical errors, they do occur on occasion—and must be corrected. The partial shell script you cited from the September issue was corrected in the December issue. Your comment about the command in the November issue will hopefully help readers sidestep that error.

We're pleased that readers such as yourself take the time to set us straight when we stray. We'll also take your suggestion to heart about making use of a referee "guru".

Editor

Dear UNIX REVIEW,

I have been searching for some time for a glossary of UNIX terms and just happened to see your column.

The glossary is very easy to read and seems to be fairly comprehensive. I would like to purchase a copy of the book if it exists.

I have checked at a few computer book stores and not one had any type of UNIX glossary or even knew if any such books existed.

A copy of your glossary would make life a lot easier for many of us here at ADR.

Regards,

Mark Salerno
Documentation Specialist
Applied Data Research, Inc.
Dallas, TX.

As of this writing, Steve Rosenthal, the author of our glossary, has not published his definitions in book form. Like you, the editors feel there is need for his work to be bound in a single volume and are encouraging him to find a publisher.

Editor

Letters to the editor should be addressed to: Mark Compton, UNIX REVIEW, 500 Howard Street, San Francisco, CA 94105.

TANDY... Clearly Superior™

The New Tandy 6000: America's leading multi-user system now has twice the power for $1000 less.

For many companies, it's hard to justify the cost of a separate computer for each employee. That's why we designed the Model 16 multi-user business computer. With its XENIX operating system, several people throughout your office can perform completely different tasks at the same time using just one computer and low-cost display terminals. It's efficient and cost effective.

Now We've Improved on a Proven Performer

Performance-wise, the new Tandy 6000 is faster than the 16B. It has twice the amount of memory. And our latest multi-user operating system comes with the system at no extra charge. And our 15-megabyte hard disk Tandy 6000 (26-6022, $5499) is one thousand dollars *less* than last year's Model 16B.

The World's Best Value, Made in America

With the Tandy 6000, your total investment is far lower. Three (or even six) employees can use a single 6000. All you need are two (or more) terminals, starting at $599 each. Everyone in the system can share the 6000's optional printer, modem and other accessories. And you'll only need to buy one copy of the programs you need.

See the Tandy 6000 today.

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

Radio Shack®
COMPUTER CENTERS
A DIVISION OF TANDY CORPORATION

Prices apply at Radio Shack Computer Centers and at participating stores and dealers. XENIX/TM Microsoft Corp.

Send Me a New 1985 Computer Catalog!
Mail To: Radio Shack, Dept. 85-A-543, 300 One Tandy Center, Fort Worth, Texas 76102

NAME
COMPANY
ADDRESS
CITY
STATE ZIP
PHONE

Circle No. 66 on Inquiry Card

ADVERTISER'S INDEX

COMING UP IN MAY

Distributed Resource Sharing

- **Architectural Issues.**
- **Why Has It Taken So Long?**
- **Remote Procedure Calls.**
- **Distributed File Systems.**
- **Linking Into IBM Mainframes.**

MOVING?

SUBSCRIBER SERVICE

Don't miss an issue of UNIX Review. Please enclose your current address label, or copy all information from the label and enclose it with your new address.

Write in new address below:

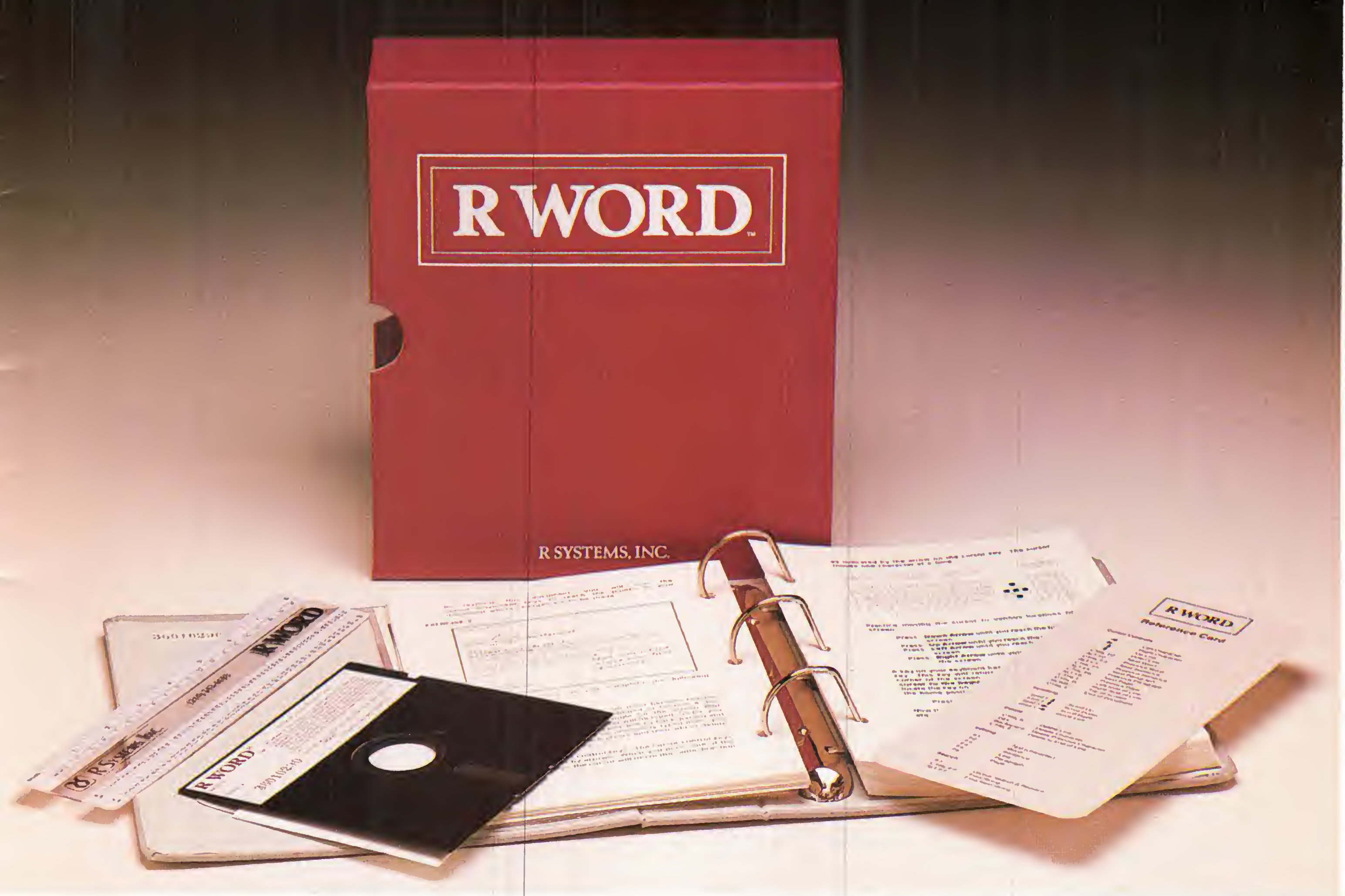
Name ______________________

Company ______________________

Address ______________________

City/State/Zip ______________________

Send all subscription correspondence to:
Review Publications, P.O. Box 7439
San Francisco, CA 94120-9864



"R WORD is an excellent program. It is one of the easiest, yet full-featured, word processors I have ever used."

Bob Woodard of the University of Iowa continues in his letter to us about R WORD, "It beats the heck out of the word processing package available with our machine. R WORD is definitely the favorite of the people who have tried it."

We've received many letters like Bob Woodard's, praising the R Family of Office Automation Software. Now, we'd like **you** to meet R Family — three programs which are powerful, easy-to-use, well-documented, cost-effective, and supportable. And they're designed for single-users, multi-users, and multi-computer users.

R Office Manager

Suggested List $295

R Office Manager is the most power-

R WORD II

Suggested List $895

R WORD II is designed for a production word processing environment which requires full-featured word processing, yet R WORD II offers amazing simplicity of use. Features include Help Menus, Cursor Controls, Editing Functions, Formatting, Three-Level File Structure, Printing Features, Mail Merge, Default Set Up, and Utility Functions.

The program also offers Global Search and Replace, Directories, Spelling Checker, Table Math, Headers and Footers, Automatic Table of Contents, Automatic Paragraph Numbering, Document Assembly, and much more.

R WORD III

Suggested List $1295

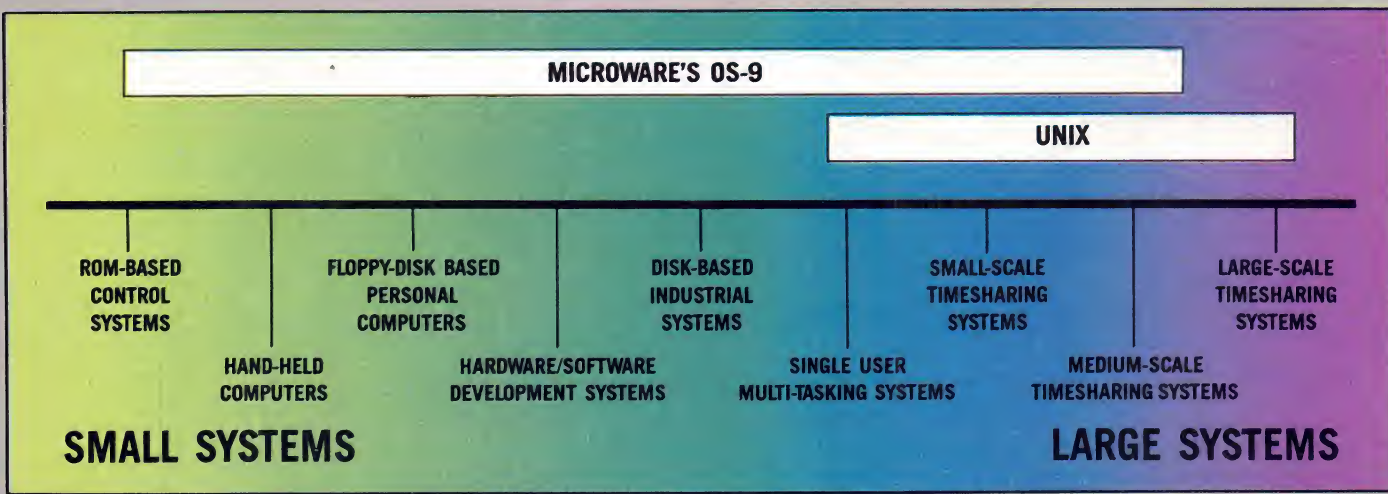
1-(800) 527-7610

Call us toll-free for more information about R Family. We are currently ported to and shipping R Family programs on the Tandy Model 16, NCR Tower + XP, Apple Lisa, Pixel, Plexus, and Altos 68000. Additional UNIX and XENIX ports are occurring weekly. We are also shipping for all MS-DOS and PC-DOS operating systems. Manufacture, distributor, dealer, national account, and porting inquiries are invited.

When you select word processing, office management, and office automation software, consider the benefits of R Family. This one set of programs can satisfy single-user, multi-user, and multi-computer requirements.

R Family offers you consistency, versatility, and supportability, as

Only Microware's OS-9 Operating System Covers the Entire 68000 Spectrum



Is complicated software and expensive hardware keeping you back from Unix? Look into OS-9, the operating system from Microware that gives 68000 systems a Unix-style environment with much less overhead and complexity.

OS-9 is versatile, inexpensive, and delivers outstanding performance on any size system. The OS-9 executive is much smaller and far more efficient than Unix because it's written in fast, compact assembly language, making it ideal for critical real-time applications. OS-9 can run on a broad range of 8 to 32 bit systems based on the 68000 or 6809 family MPUs from ROM-based industrial controllers up to large multiuser systems.

OS-9'S OUTSTANDING C COMPILER IS YOUR BRIDGE TO UNIX

Microware's C compiler technology is another OS-9 advantage. The compiler produces extremely fast, compact, and ROMable code. You can easily develop and port system or application software back and forth to standard Unix systems. Cross-compiler versions for VAX and PDP-11 make coordinated Unix/OS-9 software development a pleasure.

SUPPORT FOR MODULAR SOFTWARE — AN OS-9 EXCLUSIVE

Comprehensive support for modular software puts OS-9 a generation ahead of other operating systems. It multiplies programmer productivity and memory efficiency. Application software can be built from individually testable software modules including standard "library" modules. The modular structure lets you customize and reconfigure OS-9 for specific hardware easily and quickly.

A SYSTEM WITH A PROVEN TRACK RECORD

Once an underground classic, OS-9 is now a solid hit. Since 1980 OS-9 has been ported to over a hundred 6809 and 68000 systems under license to some of the biggest names in the business. OS-9 has been imbedded in numerous consumer, industrial, and OEM products, and is supported by many independent software suppliers.

Key OS-9 Features At A Glance

- Compact (16K) ROMable executive written in assembly language
- User "shell" and complete utility set written in C
- C-source code level compatibility with Unix
- Full Multitasking/multiuser capabilities
- Modular design - extremely easy to adapt, modify, or expand
- Unix-type tree structured file system
- Rugged "crash-proof" file structure with record locking
- Works well with floppy disk or ROM-based systems
- Uses hardware or software memory management
- High performance C, Pascal, Basic and Cobol compilers

microware®

OS-9™

MICROWARE SYSTEMS CORPORATION
1866 NW 114th Street
Des Moines, Iowa 50322
Phone 515-224-1929

Microware Japan, Ltd
3-8-9 Baraki, Ichikawa City
Chiba 272-01, Japan
Phone 0473(28)4493