

For the Advanced Computer User

Micro/Systems Journal™

Communicating at 9600 baud

see page 24



Also in this Issue

Experimenting with Protected Mode on the AT 34	
COMP Computing Standard MUMPS.....	38
Program Interfacing to Microsoft Windows	44
Creating MYLIB in Modula-2.....	57
Intercepting XIOS Calls.....	62

Complete Table of Contents on Page 3

**January/February 1987
Vol. 3 No. 1**

**\$3.95
U.S.A.**



MACROTECH—STILL THE S-100 PERFORMANCE PACESETTER

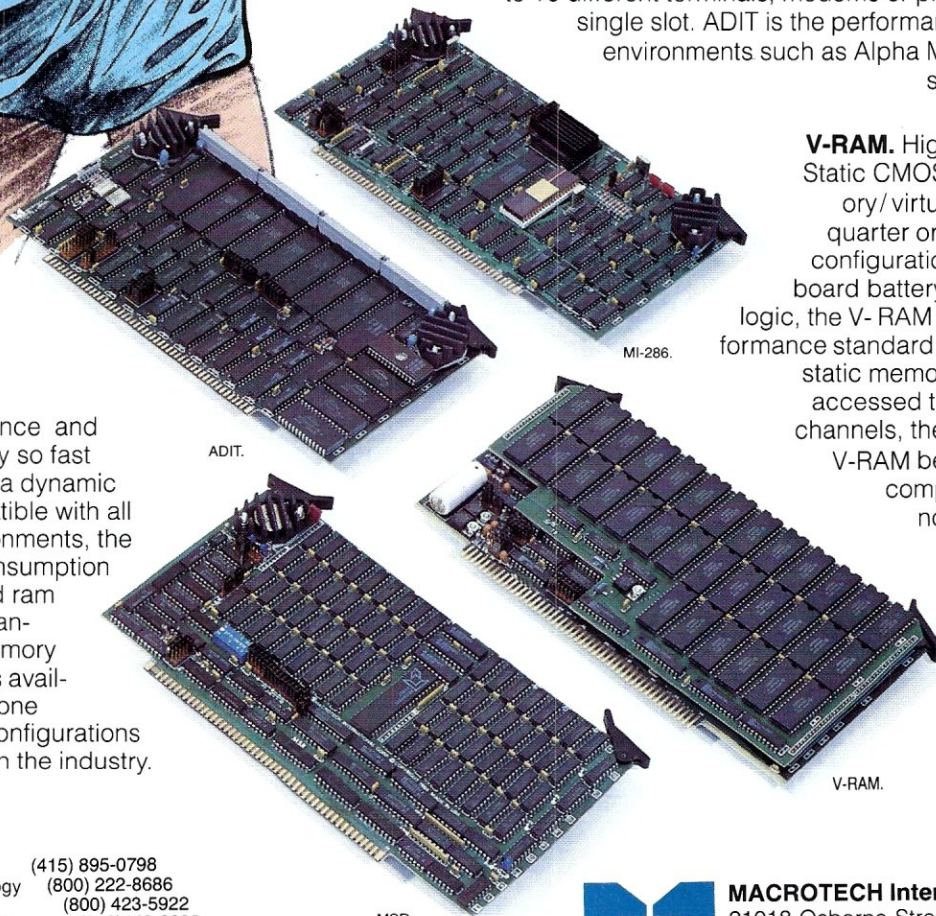


MI-286. Our 80286/Z80H Dual CPU Board is at least twice as fast as Compupro's 8085/88 and it's a direct replacement. The MI-286 has already become the standard by which other 80286 based systems are measured. Ask us for a complimentary Benchmark Report.

ADIT. There's nothing else like it on the market. It's an Intelligent I/O Board with its own real time firmware that lets you control up to 16 different terminals, modems or printers all from a single slot. ADIT is the performance standard in environments such as Alpha Micro where I/O speed is critical.

V-RAM. High performance Static CMOS system memory/virtual disk in either quarter or half megabyte configurations. With its on-board battery and power-fail logic, the V-RAM sets a new performance standard at conventional static memory prices. When accessed through I/O port channels, the half megabyte V-RAM becomes M Drive compatible with true non-volatile solid-state disk capability.

MSR. High performance and reliability in a memory so fast you won't believe it's a dynamic ram product. Compatible with all popular S-100 environments, the MSR's low power consumption and 120 nanosecond ram devices set a new standard for dynamic memory products. The MSR is available in quarter, half, one and two megabyte configurations at the lowest prices in the industry.



Dealers:
 Gifford Computer Systems (415) 895-0798
 Custom Computer Technology (800) 222-8686
 S-100 (800) 423-5922
 John D. Owens & Associates (212) 448-6298
 In England; Fulcrum (Europe) Ltd. (0621) 828763

Macrotech dealers also include most Compupro Systems Centers, Heathkit Electronic Centers and Alpha Micro Dealers.



MACROTECH International Corp.
 21018 Osborne Street
 Canoga Park, CA 91304
 (800) 824-3181 • in Calif. (818) 700-1501
 Telex: 9109970653

CompuPro Presents

The Swap Meet for the Rest of You

About three times a year, the gang at CompuPro cleans out the back room of stuff we can't sell as new and hauls it down to a traditional Silicon Valley event called Computer Swap America. In fact, there's one coming up on January 24th, and if you live in the area, you should attend. The bargains there are fantastic, and the highlight of the day is when Bill Godbout (our illustrious leader) gets up on a makeshift stage and holds a crazy auction. The deals are so great, that people have flown in from as far away as the East Coast, and more than covered the cost of their travel expense with the money they've saved.

This time, our back room walls are bulging—mainly because we changed 3rd party service organizations to Sperry CUSTOMCARE and

we got back all the service spares from the previous firm. So we decided that we'd bring some of the swap meet bargains to those of you that can't make it to the actual event. The items listed below are tested and functional, but may be discontinued models, returned service spares, used, cosmetic rejects, obsolete revisions, have wires, or anything else that prevents us from offering them as new or current. They are sold on an "as-is" basis. Quantities on these items are limited, subject to prior sale, and no rainchecks will be issued. Where possible, we will try to include a technical manual, but we make no guarantees as quantities are limited. **These products are for experienced hackers only! These items are not new and are not intended for use in commercial service!**

S-100 CPU Boards

CPU 286 (8 MHz 3 cycle)	\$249
CPU 8086 (8 MHz)	\$109
CPU 8086 (10 MHz)	\$139
CPU 8085/88 (6/8 MHz)	\$129
CPUZ (8 MHz)	\$119
CPU 68K (10 MHz)	\$139
68451-L8 MMU CHIP (8 MHz)	\$ 29
CPU 32016 (6 MHz w/ICU)	\$ 99
SPUZ 256K (8 MHz Z80 slave proc.)	\$179
SPUZ 64K (8 MHz Z80 slave proc.)	\$149

S-100 Static RAM and MDRIVE/H Boards

RAM 23 (64K-8/16 bit xfers)	\$109
RAM 21 (128K-8/16 bit xfers)	\$159
RAM 22 (256K-8/16 bit xfers)	\$269
MDRIVE/H (512K RAM Disk/Cache)	\$319
MDRIVE/H (2 Meg RAM Disk/Cache)	\$499

S-100 Disk Controller Boards

DISK 1 (8" floppy controller)	\$ 99
DISK 1A (5.25" and 8" floppies)	\$169
DISK 2/SELECTOR CHANNEL (SA4000)	\$129
DISK 3 (ST506)	\$199

S-100 I/O and Miscellaneous Boards

System Support 1 (w/battery)	\$139
PC VIDEO (with free SuperCalc 3!!)	\$249
NET PC (Arcnet for PCs, w/CPLINK)	\$189
NET 100A (S-100 Arcnet LAN)	\$189
INTERFACER 3 (8 serial ports)	\$169
INTERFACER 4 (3 serial, 1 Centr.)	\$139
CABLE 4	\$ 25
Special 50 pin to 3 DB-25 cable for I3 (2 reqd.) and I4 (1 reqd.)	

Enclosures and Motherboards

12 Slot Motherboard with Card Cage and DC cable	\$139
20 Slot S-100 Motherboard	\$ 99
6 Slot S-100 Motherboard	\$ 59
Enclosure 2-20 Slot MB w/PS	\$399

STD Bus Products

STD Bus Motherboard 16 slot	\$ 35
STD Bus 16K Static RAM	Set of 4/\$100

CP/M 80 2.2 or CP/M 816(86) no DRI docs. \$60.00 each or both for \$99.00

How to Order: These products may be ordered through participating CompuPro dealers, or you may order directly from us. Send check or money order (do not send cash) to: **CompuPro-Swap, 26538 Danti Ct., Hayward, CA 94545-3999**

Be sure to enclose the description and quantity of the items desired, your UPS shipping address (no P.O. Boxes, please!) and a daytime phone number. Also enclose \$5.00 shipping and handling, California residents enclose 6.5% sales tax. Remember, these items are sold on an as-is (no warranty) basis. All sales are final. All items are subject to prior sale and are first-come-first-served, so get your orders in quickly. If an item is out-of-stock, your money will be refunded, or we may substitute a functionally equivalent or better product. Please allow 6 to 8 weeks for delivery. **Note: Credit Card and Phone Orders will not be accepted. We cannot answer questions about these products on the phone either. If you don't know what they are or how to use them, you shouldn't buy them.**

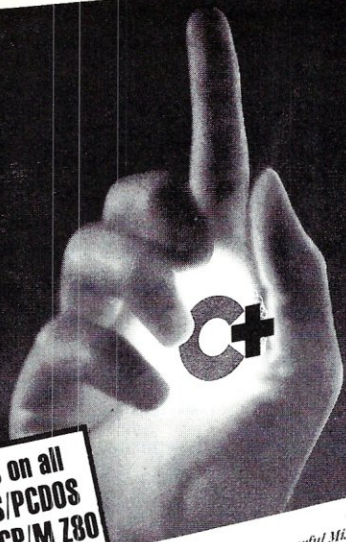
Subject to prior sale, these items and more may be available for purchase in person at Computer Swap America, January 24th and March 28th, 1987, Santa Clara County Fairgrounds, Tully Road, San Jose, California. Mark your calendars!

For an up-to-date list of what's still available and other special offers, join compupro.ad on BIX. Check any recent issue of BYTE Magazine for instructions on how to log onto BIX.

CompuPro™

MIX C COMPILER

Works on all
MSDOS/PCDOS
and all CP/M Z80
COMPUTERS!
(Not Copy
Protected!)



The Powerful Mix
C COMPILER
Harness the Power of the C Language
with this full featured compiler

Incredible Value
\$39.95
AT ONLY With 30
Day
Money-Back Guarantee

C FEVER *catch it!*

It's becoming an epidemic . . . everyone is switching to C! First there were a few hackers, then came the college students, next the major software houses, and now the rest of the programming world. Programmers everywhere are infected with the desire for SPEED, POWER, and PORTABILITY.

It's time to face the inevitable. You're going to catch the fever too! When you do, give us a call. We've got the best cure—an illustrated guide to the C language, plus a complete program development system. Everything you need to master the C programming language . . . all at a price that's less than the cost of a book!

But don't let this price fool you. Our system is powerful; it compiles twice as fast as the others, is completely standard, and it's very easy to use. Most C compilers are designed for wizards. We have designed ours for you!

- What do you get for a mere \$39.95?**
- A 450 Page book filled with sample programs, plus . . .
 - A fast, standard, full featured C compiler that supports all data types and the latest features like bit fields, enumerations, structure assignment, and passing/returning structures.
 - A fast linker that loads separately compiled files, searches libraries, and builds an executable program.
 - An extensive library of more than 170 functions (including the standard C functions and the computer specific functions that provide direct access to the operating system and BIOS).
 - Tools that allow you to optimize your programs for minimal space or maximum speed.

Operators are standing by . . . Please use this Number for ORDERS ONLY!
CALL TOLL FREE FOR RUSH ORDER DELIVERY!

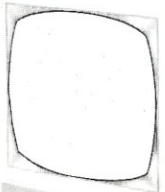
1-800-523-9520 IN TEXAS, PLEASE
CALL TOLL FREE
1-800-622-4070

For Technical Support Please call 1-214-783-6001
MIX Software, Inc. / 2116 E. Arapaho / Suite 363 / Richardson, Texas 75081
Or contact our Worldwide Distributors direct in:
Canada: Saraguay Software 1-416-923-1500 Switzerland: DMB Communication CH-1-825-53-29
Australia: Techflow 047-586924 France: Info/Tech 1-43-44-06-48

Split Screen Text Editor

an Incredible Value **\$29.95**
AT ONLY (Not Copy Protected!)
Works on all MSDOS
PCDOS and CP/M Z80
Computers

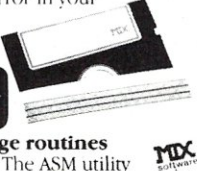
Our high powered editor is great for editing high level languages. It works just like Micropro's Wordstar, but macros allow you to create your own custom editor, and the split-screen feature lets you edit two files at the same time.



- The MSDOS/PCDOS version is loaded with special features:
- Execute any DOS command or RUN other programs from the editor.
 - Quickly edit files as large as 300,000 characters.
 - Compile MIX C programs directly from memory. The editor automatically positions the cursor to the first error in your program.

ASM UTILITY
an Incredible Value **\$10**
Value AT ONLY

Call assembly language routines from your C programs. The ASM utility works with Microsoft's MASM or M80 assembler. Macros make it easy! Works just as if you were calling a C function, and you can even call C functions from assembly language. Lots of useful assembly language functions are included as examples.



SPECIAL OFFER!

SAVE \$14.95 Off Our Regular List Price!

Limited Time Only
\$54.95
C Compiler & Text Editor

Please check method of payment:
 Check Money Order MasterCard/VISA
Your Card #: _____

Expires _____
Shipping Charges: (No charge for ASM Utility)
In the U.S.A.: Add \$5.00 per Order.
In CANADA: Add \$10.00 per Order.
OVERSEAS: Add \$10.00 per Text Editor. Add \$20.00 per C Compiler. Add \$30.00 for combined C Compiler and Text Editor.
Operating System: (Check one)
 CP/M Z80 MSDOS/PCDOS

Specify Your Computer Name _____
Specify Disk Format _____
NAME _____

Description	Quantity	PRICE	Total Order
Split-Screen Text Editor	_____	\$29.95	\$_____
C Compiler	_____	\$39.95	\$_____
C and Text Editor (Special)	_____	\$54.95	\$_____
ASM Utility	_____	\$10.00	\$_____
Texas Residents Add 6.125% Sales TAX		\$ _____	
Shipping Charges (See at Right)		\$ _____	
TOTAL OF YOUR ORDER:		\$ _____	

Telephone A/C (_____) _____
Street _____
City _____
State _____
Country _____ ZIP _____

MDX software 2116 East Arapaho
Suite 363
Richardson, Texas, 75081
Ask about our Volume Discounts!
Call 1-214-783-6001 **M**

For the Advanced Computer User

Micro/Systems Journal™

January/February 1987

Vol. 3 No. 1

Editorial

Founder and Editor Sol Libes
Technical Editor Don Libes
Associate Editors Lennie Libes
Susan Libes
Contributing Editors Robert A. Blum
A.G.W. Cameron
Ian F. Darwin
Stephen R. Davis
Hank Kee
Stephen M. Leon
Consulting Editor Michael Swaine

Editorial Coordinator Kobi Morgan

Production

Production Manager Bob Wynne
Production Kobi Morgan

Circulation

Director of Circulation Maureen Kaminski
Newsstand Sales Mgr. Stephanie Barber
Circulation Coordinator Kathleen Shay

Administration

Finance Manager Kate Wheat
Business Manager Betty Trickett
Accounts Payable Supv. Mayda Lopez-Quintana
Accts. Pay. Coordinator Kathy Robinson
Accounts Payable Asst. Patricia Albert
Accounts Receivable Mgr. Laura Di Lazzaro
Accts. Receivable Asst. Denise Giannini

Advertising

Advertising Director Robert Horton
Account Manager James Riley
(415) 366-3600
Advertising Coordinator Charles Shively

M & T Publishing, Inc.

Chairman of the Board Otmar Weber
Director C. F. von Quadt
President and Publisher Laird Foshay

Micro/Systems Journal (ISSN #8750-9482) is published bimonthly by M&T Publishing, Inc., 501 Galveston Dr., Redwood City, CA 94063; (415) 366-3600. Second-class postage paid at Redwood City and at additional entry points.

Article Submission: Please write the editors at *Micro/Systems Journal*, Box 1192, Mountainside, NJ 07092 or call (201) 522-9347 to see if there is interest in a particular subject. Please do not send the article unless we ask for it.

If you are interested in reviewing hardware or software please write telling us your interests, your background, and include a sample of your writing.

Send a stamped self-addressed business size envelope (to NJ address) for a copy of our Author's Guide.

Address Correction requested: POSTMASTER: send address changes to *Micro/Systems Journal*, 501 Galveston Dr., Redwood City, CA 94063. ISSN 8750-9482

Change of Address: Please send old label and new address to: *Micro/Systems Journal*, 501 Galveston Dr., Redwood City, CA 94063.

Correspondence: Editorial Office: Box 1192, Mountainside, NJ 07092; (201) 522-9347. The editor may also be contacted via MCI Mail (SLIBES). Business Office: 501 Galveston Dr., Redwood City, CA 94063; (415) 366-3600.

Advertising Rates: Available on request. Call (415) 366-3600 or write to: *Micro/Systems Journal*, 501 Galveston Dr., Redwood City, CA 94063.

Subscription rates: U.S.: \$20 for one year, \$35 for two years. Canada & Mexico: \$28 for one year, \$52 for two years. Europe & So. America: \$35 for one year, \$64 for two years. Asia, Africa, Australia & Near East: \$39 for one year, \$70

IN THIS ISSUE

FEATURE ARTICLES

- Communicating at 9600 baud** 24
Steve Bosak and John Sojak
- Experimenting with Protected Mode on the AT** 34
Marshall Brain
- COMP Computing Standard MUMPS** 38
Michael Guttman and Robert Engle
- Program Interfacing to Microsoft Windows** 44
William Wong
- Creating MYLIB in Modula-2** 57
Charlie Foster
- Intercepting XIOS Calls** 62
Alex K.H. Soya

PRODUCT REVIEWS

- B-Tree/ISAM File Handlers** 58
Andrew L. Bender, M.D.
- The DRC Terminal Kit** 71
Tony Dean
- Running CP/M Under MS-DOS: Part 2** 74
Robert J. Stek, Ph.D.

DEPARTMENTS

- Editor's Page** 4
- News, Views & Gossip** 6
Sol Libes
- Mail** 10
- Turbo Pascal Corner** 12
Stephen R. Davis
- The C Forum** 17
Don Libes
- The UNIX File** 78
Ian F. Darwin
- The Scientific Computer User** 80
A.G.W. Cameron
- PC/Blue Report** 82
Hank Kee
- In the SIG/M Public Domain** 84
Stephen M. Leon
- New Products** 86
- The Software Directory** 88
- Index** 89

Editor's Page

by Sol Libes



At the beginning of another year it is worthwhile to take a look at what we can expect to see in the way of the evolving marketplace. Will the dominance of the PC finally end, or are we in for another year of sameness?

THE PC STANDARD

There is no doubt that the last five years have seen the PC bus and PC/MS-DOS evolve into a standard. In some ways, this has been good; in other ways, not. The PC "standard" has made life easier for users who have to interface modems and printers and want to extend the capabilities of their systems. It has given software developers a known system configuration that has allowed them to develop powerful software. And it has generated economies of scale that have brought costs down to a level that has put substantial computing power within the reach of millions. On the other hand, it has stifled quite a bit of technological development. Many companies who tried to innovate rather than copy are out of business; companies such as Morrow Designs, Otrona, Gavalin, Mindset, and Osborne are gone. Many others who remain have chosen to conform rather than innovate.

Even IBM, who typically closed out product lines every 3-4 years, was forced, by sheer market momentum, to keep the product in production much longer than it would have liked, giving competitors an opportunity to compete by copying. The question now is whether we will see a new standard come to the fore this year.

There is no doubt that the PC/XT/AT has become a standard. Many companies are learning to innovate within the constraints of the standard, though. Hardware innovations have included speeded-up processing, expanded memory addressing, expanded storage capability,

improved graphics, and lower-cost networking systems. Software innovations include enhancements to the operating system, better user shells, and software that taxes the capabilities of the operating system. And, of course, the most important innovation is PC compatibility at significantly lower prices than IBM.

A Look At The Year Ahead

These enhancements have, to a great extent, put IBM in a catch-up position and hurt its competitive position in the marketplace. Although many still buy their basic machine from IBM, they end up installing non-IBM hardware and software on these systems. There is no doubt that 1987 will see this trend continue.

The basic PC/XT marketplace is now a commodity market. Purchasers look first for price, second for features, and third for support—just the opposite from the way IBM likes to sell to customers.

IBM has been making threats of withdrawing from the low-end of the PC/XT marketplace. I doubt that it can afford to do this, though. Rather, I expect IBM to introduce a newer, lower-cost PC and XT tailored for entry-level users (e.g., schools) and with limited capabilities (e.g., as just word-processing systems or as basic workstations).

THE AT STANDARD

The AT will almost certainly become the predominant standard in '87. Microsoft's new version (5) of PC/MS-DOS should become available this year. It will be designed specifically for AT-compatible machines and provide features and capability not available on PC/XT systems. Software developers can be expected to introduce software quickly for this new operating system and will abandon writing software for earlier versions. The PC/XT will soon become a dated system, much as CP/M is today, and its presence will gradually fade away.

The second reason for the ascendance of the AT to the predominant position is cost. The prices of AT-class systems have dropped substantially, as clone competition developed. The result is that the price differential between an XT and AT is no longer significant. For a little more money one gets a tremendous improvement in performance and capability.

Although 386-based systems will be widely available in '86, I think they will still not be a significant factor in the marketplace. They would need the support of IBM, software support, and price competition to be so. I do not expect this to happen until 1988 or 1989, at the earliest. The AT and DOS Version 5 should be the predominant system for 1987 and for much of 1988.

THE DOS STANDARD

DOS 5 will no doubt be the dominant operating system for the next year or two. XENIX and AT&T's 6300-Plus, running UNIX, has failed to gain significant acceptance. XENIX taxes the capabilities of the AT more than does DOS and leaves something to be desired in performance. Add to that the added cost of running XENIX and lack of software support for same, and it is likely that XENIX will never rise above its current status in the market.

Digital Research's Concurrent-DOS is without doubt superior to PC/MS-DOS, providing the multitasking features that Microsoft is just now getting around to incorporating in PC/MS-DOS. And it has been available for about two years. But DRI has been slow to provide sufficient software compatibility and support for Concurrent-DOS to software developers. Although a number of systems houses are using it to build low-cost multiuser systems, I do not expect that it will garner anything more than limited acceptance in the marketplace.

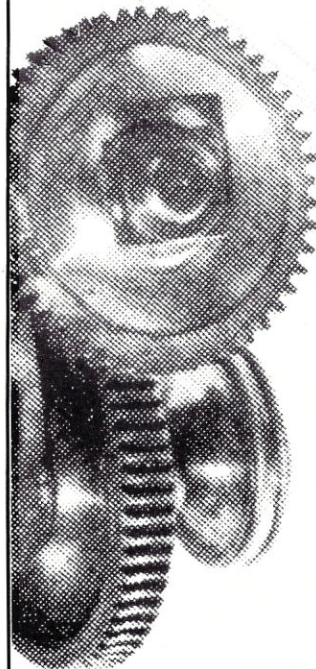
Also, DRI's GEM environment was available for more than a year before Microsoft's Windows and in many ways is still a better performer. But here again, DRI has not supported its product or promoted it. There is still a question as to whether these graphics environments will succeed unless they are built into the hardware, à la the Macintosh. Many software developers incorporate a graphics environment directly in their packages (e.g., pull-down menus and mouse support), so that the user doesn't need Windows or GEM.

GRAPHICS

One of the main areas of improvement in the next year on PCs will be in graphics. The graphics capabilities that are currently available on high-end workstations from Sun, Apollo, and DEC will soon be available on AT systems. The EGA-type display is gaining wide acceptance as more software supports it and price decreases make it attractive. Here again, innovators are learning how to work within the EGA standard to provide performance superior to that of the IBM-EGA. Early innovators provided features such as Hercules compatibility and easier switching between modes of operation. Now, innovators are introducing hardware and software improvements to the EGA standard to provide higher resolution and faster screen updating. The IBM version of the EGA, at this point, is an inferior performer at a higher price.

By mid-year we can expect to see the introduction of new graphics controller cards utilizing more powerful display-controller chips from TI and Intel. These controllers will relieve the main processor of much of its display control and will allow even faster and more powerful screen performance. Software that takes advantage of these capabilities will begin to appear by the year's end. CAD and page-make-up systems will be the biggest beneficiaries of these capabilities, and WYSIWYG (what you see is what you get) will be the standard for many applications.

I have touched on only a few of the improvements we can expect to see this year. There will also be improvements in memory, storage, printers, communications, and more. These improvements may not be as dramatic as in previous years, but they are something to look forward to. §



BD Software, Inc., maker of the original
CP/M-80 C Language Development
System, knows

Time is precious

So the compilation, linkage and execution speeds of BDS C are the fastest available, even (especially!) on floppy-based systems. Just ask any user! With 15,000+ packages sold since 1979, there are *lots* of users . . .

New! Ed Ream's RED text editor has been integrated into the package, making BDS C a truly complete, self-contained C development system.

Powerful original features: CDB symbolic source-level debugger, fully customizable library and run-time package (for convenient ROM-ing of code), XMODEM-compatible telecommunications package, and other sample applications.

National C User's Group provides direct access to the wealth of public-domain software written in BDS C, including text editors and formatters, BBS's, assemblers, C compilers, games and much more.

Complete package price: \$150.

All soft-sectored disk formats, plus Apple CP/M, available off-the-shelf. Shipping: free, by UPS, within USA for *prepaid* orders. Canada: \$5. Other: \$25. VISA, MC, COD, rush orders accepted.

BD Software, Inc.

BD Software, Inc.
P O Box 2368
Cambridge MA 02238
617 • 576 • 3828



News, Views & Gossip

by Sol Libes

RUMORS & GOSSIP

PC's Limited, the first company to ship a 12-Mhz AT clone is rumored to be readying 20- and 24-Mhz versions. These would probably run faster than the current crop of 386-based systems. The firm is also rumored to be readying a 386-based system running at more than 20-Mhz, while all current units run at 16-Mhz. Looks like PC's Limited aims to be the fastest kid on the block.

IBM has reportedly stopped producing its CGA (Color Graphics Adapter) card for the PC and is rumored to be planning to cease producing the EGA (Extended Graphics Adapter) card. Look for IBM to introduce an EGA-Plus card with higher pixel density and faster operation than the first EGA card.

There are also reports that IBM is beta-testing a PC/XT replacement based on the 8086. It takes up much less desk space than the PC/XT and uses 3½-inch floppy and hard disk drives. The motherboard has proprietary ICs and contains the display controller (a new high-resolution system), and network and mouse-interfacing circuitry. A new version of DOS 3.3 with mouse and network support is also being tested. Based on past experience, it is not wise to assume that a system being beta-tested by IBM will actually reach the market. The rumors do give some indication of future directions being seriously explored by IBM, however.

Hyundai (from South Korea) is reported about to start distribution of a \$699 (list) PC clone via mass merchandisers such as Caldor, Toys-R-Us, Federated, Target, and SaveMart. The unit will have one disk drive and 512K RAM. Options will include a second floppy and a 20Mb hard disk. This will probably cause companies such as Leading Edge and Epson to withdraw from the low end of the PC-compatible market.

AT&T is expected to show a 1986 pre-tax loss of \$700 to \$800 million in its computer business. Sales of AT&T's UNIX personal computer, made by Convergent Technologies, were particularly disappointing. It is estimated that AT&T sold fewer than 2,000 of these units last year—10 percent of what it had projected. Look

for AT&T to deemphasize its computer business in favor of its traditional phone business.

Intel, the last U.S. manufacturer of bubble-memory devices and systems, has finally given up and abandoned the business. This leaves the market, which is small and specialized, to foreign vendors such as Hitachi and Fujitsu. Intel thus follows in the footsteps of AT&T, Motorola, National Semiconductor, Rockwell International, and Texas Instrument.

Guess which computer maker spends the most money on advertising? As if you didn't already know—it's **IBM**. In the first six months of 1986, IBM reportedly spent \$15 million, of which \$5 million went for advertising PC products. That is more than most of IBM's competitors gross. Guess where they spent the largest share? Why, in the *Wall Street Journal*, of course—IBM knows where its customers are.

InfoCorp of Cupertino, California, a market researcher, reports that unit sales of the Apple Macintosh have surpassed those of IBM's AT (of course, that does not take into consideration all the AT clones being sold). InfoCorp also reported that IBM's and Compaq's market share decreased at the expense of Apple and the clone makers. IBM is still ranked first, Apple second, and Compaq third, with AT&T and Leading Edge tied for fourth place, and Kaypro and Epson tied for fifth.

Atari is rumored to be readying a laser printer with an under-\$1,000 price tag. There are already some low-cost desktop publishing software packages for the Atari ST system. This may round out the system package and make the ST more attractive to business users.

286-DOS EXPECTED SOON

In previous columns I discussed rumors of the new version of MS-DOS designed specifically for AT-compatible systems. The latest rumor is that this will be released in the second quarter by both Microsoft and IBM. 286-DOS, as it is now being called, has been in beta-test since last summer. As reported in an earlier column, it will provide true multitasking and be capable of addressing up to 16Mb of memory.

There are already more than one mil-

lion 286 systems installed, and predictions are that by midyear this figure will rise to more than 3.5 million. Further, sales of 286 systems are not expected to peak until late next year, when 386-based systems go into full production.

32-BIT BUS STANDARD

Approximately 50 manufacturers, including AT&T, Quadram, Emulex, and Chips & Technologies, have committed to supporting the **PCET bus** (PC Extended Technology) for 80386-based machines. The PCET bus was developed by **Phoenix Technologies Ltd.**, of Norwood, Massachusetts. The bus is an extension of the PC 8-/16-bit bus and hence is compatible with boards for those systems. As yet, no manufacturer has announced a system using this bus. Most of the systems that have been announced use the standard PC/AT 8-/16-bit bus system.

Compaq Computer, and others, use the AT bus and add a 32-bit bus slot capable of handling a high-speed memory board. This 32-bit bus is not compatible with the PCET bus. It is unlikely that IBM will adopt the PCET bus as part of its 386 systems.

386 MARKET REPORT

Several companies have released 80386-based systems. The current leader is **Compaq Computer**, with its Deskpro-386, and that is being shipped only in small numbers.

At this point, there is no software that takes advantage of the 386 (one could say the same for the 286, except for XENIX-286 and Concurrent DOS), so that users are running MS-DOS 3.x, letting most of the features of the 386 go unused. Microsoft is promising to release XENIX-386 by June, but as yet there is no word as to when it will release a 386 version of MS-DOS—it may not be this year.

The current potential market for 386-based systems thus appears to be limited to local area network servers and applications such as CAD, where the systems will function as faster versions of the AT.

The current 386 machines, are to a great extent, competing with a nonexistent IBM system. Many software developers and peripheral-board suppliers are waiting for IBM to release a 386 system. If this does not happen within the next six months, however, these companies may choose to release products. **Lotus Development** and **Ashton-Tate** are known to be working on 386-based products.

Also, most compatible system makers are holding off introducing systems, as they wait to see what **IBM** will do. They will only wait so long, though. The question is, how long can IBM keep the 386 market in check by delaying introduction of its 386 system? Everyone will be watching to see how well the Deskpro-386 does within the next six months. If it meets with

Uninterruptible Power Systems

Read what the experts say;

"Clary Corp's OnGuard 600VA has the makings of the perfect UPS: it's functionally invisible. Whether or not line voltage is available, it constantly supplies up to 600VA of nearly pure sine-wave power at an almost unvarying 120 volts. No UPS is more unobtrusive than the OnGuard."

-Winn L. Rosch,
PC Magazine,
SEPTEMBER 16, 1986

For heavy-duty use with large systems, our choice is Clary Corp.'s OnGuard 600VA. It has sufficient capacity to run a network server with external hard disk(s) long enough to bridge most outages or shut down the network in an orderly manner. Also, it's quiet enough for an office area and simple enough to use that it can be forgotten.

-PC Magazine,
Editor's Choice,
SEPTEMBER 16, 1986

DISCOVER THE CLARY SOLUTION

*\$1995.00



Available in 500VA*, 600VA,
800VA, 1000VA & 1200VA

- Capable of providing nearly twice its rated output capacity (with switching power supply loads).
- Designed for use with your IBM 5362 & 5364.
- On-line operation, static bypass switch and sinewave output.
- Fifth generation, high frequency PWM technology reducing size and weight by over 50%.
- Capable of running 135-watt PC-XT for 2½ hours.
- Meets FCC Part 15-Class A, NEC and IEEE 587 standards.
- Virtually silent.
- U.L. listed.

CLARY®

CLARY CORPORATION

320 West Clary Avenue
San Gabriel, CA 91776 (818) 287-6111 x620

ONGUARD™



any worthwhile acceptance, other companies may not wait for IBM to jump in. If they do, will they use the Compaq system as a standard? If not, we will have chaos, and IBM will have an opportunity to come in later and become the standard.

CONCURRENT DOS UPDATE

It is interesting to note that **Digital Research** has had Concurrent DOS out for close to two years, while Microsoft is only now starting to ship its concurrent version of MS-DOS (Version 4.0). 4.0 has only limited concurrent capability (programs running in the background must be specifically written for the purpose), while DRI's Concurrent DOS can run most standard packages in the background.

DRI released its latest version of Concurrent DOS (Concurrent PC DOS XM) in September. It has improved PC-DOS compatibility, full batch support, and supports up to 8Mb of RAM.

Quadram now bundles a single-user version of XM with its Quad EMS+ memory-expansion board. DRI sells a three-user version for the PC (\$395, or \$50 for owners of Version 4.1). HAAR Industries (2600 Virginia Ave. NW, Ste. 600, Washington, DC 20037; (202) 338-8550) and Goodall Computer Systems (3770 24th St., San Francisco, CA 94114; (415) 648-2174) sell versions that handle up to 16 users. CompuPro Systems, Nixdorf Computer, COMARK, and L/F Technologies are selling versions for systems they manufacture. The CompuPro and L/F Technologies systems are S-100-based. The COMARK system is Multibus-based.

A Concurrent User Group (CONUG, Box 734, Marina, CA 93933; (408) 384-6797(voice) and (408) 384-5575(modem)) publishes a monthly newsletter and maintains a software library of concurrent utilities and programs (membership is \$25/year). And, a book titled *Concurrent PC-DOS* (\$21.95) is available from Prentice-Hall (Englewood Cliffs, NJ 07632; (201) 592-2498).

WINDOWING STATUS REPORT

Microsoft and **Digital Research, Inc.**, have been aggressively pushing their **Windows** and **GEM** windowing systems for well over two years with little success. Only a handful of software houses have introduced packages that run under these front-end systems. By contrast, windowing is the rule rather than the exception on UNIX-based systems. No doubt this is because UNIX users do a great deal of multitasking, while this is rare for PC/MS-DOS users. This should change, however, when Microsoft releases its multitasking version of MS-DOS, expected later this year.

Software developers are faced with the problem that there is no standard for windowing. This is true in both the PC and UNIX environments. In the UNIX market, the leading contenders are NeWS (Net-

work Extensible Window System) from Sun Microsystems and X Windows, developed under an industry-financed project at MIT.

ANSI has formed a task group on display management to develop a windowing standard. At this point, there is no indication as to whether the group will pick NeWS, X Windows, Windows, or GEM. The likelihood is that separate standards will be adopted for the PC-compatible and UNIX marketplaces—and, that a standard will be set when DEC and IBM declare their endorsements or introduce their own systems.

WORTHWHILE NEW PRODUCTS

The "why-didn't-somebody-do-this-before-department": **The Torrington Co.** (59 Field St., Torrington, CT 06790; (203) 482-9511) has introduced a cordless PC mouse that does not require a tablet. It uses infrared technology and transmits to a 1 X 2-inch receiver. It has a rechargeable battery.

Maynard Electronics of Casselberry, Florida, has announced an Enhanced Run Length hard-disk controller that it claims is even better than standard RLL controllers. Standard RLL controllers increase hard-disk capacity by 50 percent. Maynard claims its ERLC controller will double the capacity of most hard-disk controllers and improve access time, since the heads do not have to move as far as they do on most of those other controllers.

Wyse Technology has introduced what is the fastest PC compatible on the market, an 8088-based system running at 9.54-Mhz, twice the speed of a standard PC. It is likely that we will soon see PC-compatibles running at well over 10-Mhz.

Quadram and **Paradise Systems**, following in Vega's footsteps, have announced EGA PC display-controller cards with greater resolution, faster operation, and more colors than earlier cards. These boards take advantage of the extended capabilities of multisync monitors such as the NEC and Sony.

SOFTWARE PRICES UP

While prices for systems tumble, prices for popular, name-brand software are going up or holding their own, despite predictions of industry sages. For example, **Microsoft**, when it released new versions of Word and Project, raised its prices from \$375 to \$450 and from \$250 to \$395, respectively. **Borland International**, the former champion of low prices, recently raised the price of Reflex from \$99 to \$149. And, **Lotus Development** raised the wholesale price of 1-2-3 three percent, forcing dealers to raise their discounted prices, while the list price remained constant.

The general rule seems to be that if a product is in demand, raise the price. If it is doing poorly, drop the price. Hence, Lotus, in response to poor sales of Jazz,

dropped its price from \$595 to \$395 and began offering a \$100 rebate.

Sales of popular software, such as 1-2-3 and dBASE, have not as yet been affected by low-priced software clones, look-alikes, or public-domain software. This is quite surprising, since some of these programs sell for as little as \$19.95.

ATARI VS. COMMODORE SUIT GOES TO TRIAL

Days before Jack Tramiel acquired **Atari** from Warner Communications, **Amiga Corp.** broke off negotiations to license a chip set it was developing for Atari. It returned a half-million-dollar advance to Warner and was soon acquired by Commodore, for \$25 million.

Atari subsequently filed suit, charging that Atari helped develop the Amiga chip set and that Amiga therefore had no right to break off negotiations arbitrarily.

Atari has reportedly already sold more than 100,000 ST systems, compared to less than 40,000 Amigas sold by Commodore. Both companies are currently turning a profit, and sales of the ST and Amiga system are expected to increase substantially this year. Atari is expected to announce a new version of the ST soon with enhanced graphics, faster speed, and larger memory size.

PC BUSINESS REBOUND?

The *Wall Street Journal* recently reported that the Personal Computer business, which had been in the doldrums since early 1984, is rebounding. They report that **IBM**, in September, "sold more PCs than in any month in its history." And, that **Intel** said that last year it sold twice as many 80286 chips as it had expected to sell, and that **IBM** had trouble meeting the demand for AT systems.

Many industry authorities are predicting a 20-30 percent industry growth for this year. Most of the increase is expected in sales of more powerful 286-based systems, networks, and desktop publishing systems. §

there is mail . . .

We welcome your letters, with their comments, compliments, criticism, and suggestions. We do not have the staff to answer all letters personally. All letters become the property of M/SJ and may be subject to editing. We do not print letters that do not include a name and address.

Please send your letters to: Micro/Systems Journal, Box 1192, Mountainside, NJ 07092.

MULTI-TASKING SYSTEM INFO WANTED

Dear M/SJ:

I welcome any and all information on Concurrent-DOS or Concurrent-CP/M. Alex Soya's articles and user's letters are invaluable. It's very hard to buy a concurrent bootable system, however. Aside from CompuPro ads, no leads are published.

Please publish a table of micros on which Concurrent runs. Include low-priced clones, 68000 systems, and versions for future release.

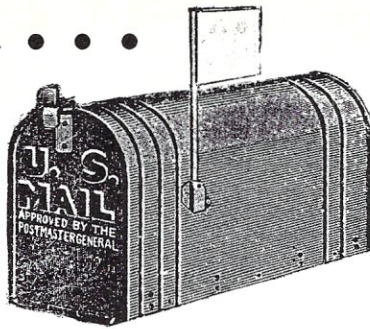
Nelson Richardson
New York, NY

Digital Research, Inc., has done a poor job of promoting Concurrent-DOS, an excellent multitasking system. DRI sells a bootable version for PC-based machines. Several readers who are running it on AT-compatible systems think it is terrific. Several companies are using Concurrent-DOS for multiuser/multitasking systems. The latest release is called Concurrent PC-DOS/XM and is available directly from DRI. To order, call DRI at (800) 443-4200. The price is \$395 and an update, for owners of Version 4.1, is only \$50. Haar Systems, Washington, DC ((202) 338-8550) sells an enhanced version of Concurrent PC-DOS/XM that will accommodate up to 16 users.

A list of companies selling bootable Concurrent-DOS systems will be included in Part III of Alex's tutorial, which is scheduled for the May/June issue.

Readers should note that DRI's Concurrent-DOS-286, Version 1.0, is currently being sold by IBM and is called the 4680 operating system. DRI is working on Version 2.0, which is expected to be offered for sale directly from DRI later this year.

For more information on Concurrent, you might log into Alex Soya's BBS ((305) 727-0331). There is also an excellent newsletter published for Concurrent users; write to CONUG (CONcurrent Users Group), Box 734, Marina, CA 93933. Membership is \$25/yr (\$35 outside the U.S.). CONUG also has a BBS ((408) 384-5575); log on as CONUG GUEST, with the password GUEST, for restricted access. And, you will also find quite a bit of interchange among Concurrent users on the



CompuServe DRI and GENie CP/M SIGs.

BACKUP POWER SUPPLY CONFUSION

Dear M/SJ:

Reader Carl Voelz (letters, pp. 12-15, Sept/Oct 1986 M/SJ) seems to be a bit confused about backup power supplies. His first point is incorrect (as you must have noticed) in two ways. First, in comparing square waves vs. sine waves, the peak voltages are chosen (with either waveform) to give "heating power" (integral under the wave) equal to the same nominal DC voltage (a single wave peak is 1.414 times the sine wave's nominal voltage). Thus, heating of a motor would be sensibly the same.

Second, if he's a fastidious designer, he's going to apply DC, not AC, sine or square, to a DC motor in a disk drive.

I don't understand his true vs. switchover types or the "mu" notation, but I do know that transient damage is caused by high amplitudes, not by (less than infinite in practice) rise times.

To depart from his letter, there are two ways to achieve short-term, uninterruptable power for a microcomputer. The easier (but more expensive) route is to use a DC-powered inverter to provide a (usually nonsynchronous) 115VAC square wave, into which the computer's normal AC supply is plugged. Two problems: the inverter is a current hog, particularly with light loads, and efficiency suffers further from multiple current conversions.

The more efficient route involves using the battery DC to power the computer directly through high-efficiency switching regulators. The battery (a nice, big filter capacitor, if you will) can then be float-charged from the AC mains. During a power outage, the battery takes over and continues operation.

David Mc Lanahan
Marlow, NH

PROPOSED FILE LENGTH STANDARD FOR CONCURRENT-CP/M AND CP/M PLUS

Dear M/SJ:

Both CP/M-Plus and CCP/M-86 offer token support for recording the exact (byte) length of a file in the directory. The method is crude, but workable. Using a

variant of BDOS function 30 (set file attributes), a program may set the S1 byte in the directory for a file to a value representing the number of bytes in the last record (sector) of the file. The byte length of the file is then:

$$(\text{sectors} - 1) * 128 + \text{lastSectorLength}$$

In more detail, the process is:

1. Set up the standard FCB containing the name of the file.
2. Set the F6 interface attribute bit (FCB[6]bit 7).
3. Set the CR field (FCB[32]) to the desired value.

Now a call on BDOS function 30 will copy the CR field into the S1 byte (FCB[13]) and record the contents in the directory. (The CCP/M manual refers to the S1 byte as the CS byte.)

The contents of the S1 byte can be retrieved from the DMA area on an OPEN, SEARCH-FIRST or SEARCH-NEXT BDOS function, each of which returns a small integer value (in the range 0-3) if the BDOS function is successful. The returned value is known as a directory code, and the S1 (CS) byte is found at:

$$\text{DMA}[\text{directoryCode} * 32 + 13].$$

The DRI documentation suggests that the S1 byte be set to hold the number of (valid) bytes in the last record (sector) of the file. It also says that no DRI utility sets or uses the file byte count.

To my knowledge, there is exactly one program that sets the S1 byte, and I wrote large portions of it. This seems the perfect opportunity to establish a standard.

There is one problem with using the S1 byte to store the number of valid bytes in the last sector. It is difficult to make the distinction between zero, meaning no valid bytes, and zero, meaning all bytes are valid. One could make the assumption that S1=0 means all bytes in the last sector are valid, while S1 in the range 1-127 means that only S1 bytes are valid. This would be consistent with existing programs that do not set the S1 byte. While workable, this logic is slightly complicated. I suggest the following standard:

When using BDOS function 30 to set the byte count for a file, the S1 byte should represent the number of *unused* bytes in the last sector of the file. While still consistent with existing programs that do not set the byte count, the logic for determining the length of a file is slightly simplified and is given by:

$$\text{sectors} * 128 - \text{unusedBytes}$$

Admittedly, the simplification appears minimal, but when one comes to write code sensitive to file byte counts, there is a substantial reduction in complexity with the proposed standard.

John Hastwell-Batten
Dural, Australia

§

DIGITAL RESEARCH COMPUTERS

(214) 225-2309

TEXT TO SPEECH BOARD!

PC/XT COMPATIBLE. MAKE YOUR COMPUTER TALK!

A VERY POWERFUL AND AMAZING SPEECH CARD. USES THE NEW GENERAL INSTRUMENTS SP0256-AL2 SPEECH CHIP AND THE CTS256A-AL2 TEXT TO SPEECH CONVERTER.

THIS BOARD USES ONE SLOT ON THE MOTHERBOARD AND REQUIRES A COM SERIAL PORT. BOARD MAY ALSO BE USED IN A STAND ALONE ENVIRONMENT WITH ALMOST ANY COMPUTER THAT HAS A RS232 SERIAL PORT. FEATURES ON BOARD AUDIO AMP OR MAY BE USED WITH EXTERNAL AMPS.

DEMONSTRATION SOFTWARE AND A LIBRARY BUILDING PROGRAM ARE INCLUDED ON A 5 1/4 INCH PC/XT DISKETTE. FULL DOCUMENTATION AND SCHEMATICS ARE ALSO INCLUDED.

NEW!

69.95

COMPLETE KIT

89.95

ASSEMBLED & TESTED

HARD DISK CONTROLLER CARD
THE FANTASTIC KONAN KDC-230. WITH MANUAL. SUPER EASY TO CONFIGURE AND USE.

HALF SIZE PCB. \$99⁰⁰ — ADD \$8.00 FOR CABLES

MEX-PC MODEM SOFTWARE PACKAGE **\$49.95**

A FANTASTIC COMMUNICATIONS PACKAGE WITH FEATURES TOO NUMEROUS TO LIST. SEE MAY/JUNE '86 MICRO SYSTEMS JOURNAL FOR FULL REVIEW OR CALL FOR BROCHURE. SUPPORTS COLOR, KERMIT PROTOCOL, Y MODEM BATCH, VT100 EMULATION, REMOTE OPERATION, ON LINE HELP, ETC. RUNS UNDER PC/MS DOS. 180 PAGE MANUAL & BINDER.

MEX-PACK LIST IS \$99.95 — **SPECIAL \$49.95**

HOT COMPUTER BOOKS!

GET THE MOST OUT OF THE INVESTMENT YOU MADE IN YOUR COMPUTER HARDWARE AND SOFTWARE WITH THE NEW WORDWARE SERIES OF COMPUTER REFERENCE AND TRAINING MANUALS. LEARN HOW TO MAKE ALL THAT EXPENSIVE SOFTWARE YOU BOUGHT REALLY COOK! **BUY ANY 3: TAKE AN EXTRA 10% DISCOUNT!**

- ILLUSTRATED TURBO PASCAL \$13.50
- ILLUSTRATED AUTOCAD \$18.50
- ILLUSTRATED WORDSTAR \$18.50
- ILLUSTRATED d BASE II \$13.95
- ILLUSTRATED d BASE III \$18.50
- ILLUSTRATED LOTUS 1-2-3 \$15.50
- ILLUSTRATED SYMPHONY \$17.50
- ILLUSTRATED C PROGRAMMING \$17.95
- ILLUSTRATED MS/PC DOS \$12.95
- ILLUSTRATED MULTIMATE \$17.50
- WORDSTAR CUSTOMIZING \$18.50
- COMMUNICATIONS HANDBOOK \$12.50
- ADVANCED TURBO PASCAL PROGRAMMING TECHNIQUES \$19.95

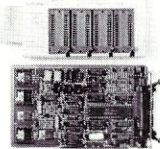
NEW!

\$100 BUSS LIVES!
WE STILL SELL \$100 PRODUCTS AND CP/M SINGLE BOARDS. FREE CATALOG.

SCHEMA: Schematic Capture and Drawing Package

WORKS WITH PC'S AND COMPATIBLES. EASY TO USE WITH OFF THE SHELF PRINTERS, MICE, VIDEO CARDS, ETC. COMES COMPLETE WITH A HUGE COMPONENT DATABASE. SUPPORTS MONOCHROME, COLOR, OR EVEN EGA. FREE BROCHURE AND DEMO DISKETTE ON REQUEST. LIST \$500 — **SPECIAL \$396.15**

NEW! PC/XT EPROM PROGRAMMER **\$199**

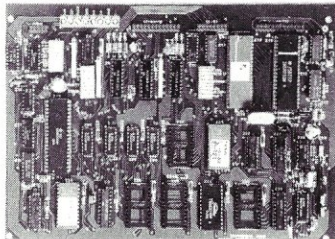


- * LATEST DESIGN * PROGRAMS UP TO 4 DEVICES AT ONE TIME * FEATURES EASY TO USE MENU DRIVEN SOFTWARE THAT RUNS UNDER PC OR MS-DOS.
- * USES AN INTELLIGENT PROGRAMMING ALGORITHM FOR SUPER FAST (8X) EPROM BURNING. * THIS PLUG-IN BOARD ATTACHES TO AN EXTERNAL MINI CHASSIS CONTAINING 4 TEXTOL Z.I.F. SOCKETS. * NO PERSONALITY MODULES REQUIRED * AUTOMATIC VPP SELECTION: 12.5V, 21V, OR 25V.
- * EPROM DATA CAN ALSO BE LOADED FROM OR SAVED TO A DISKETTE.
- * PROGRAMMING SOFTWARE SUPPORTS: 2716, 2732, 2732A, 2764, 2764A, 27128, 27128A, 27256, 27256A, 27512, AND 27512A. * ASSEMBLED AND TESTED, BURNED. IN WITH MANUAL. \$199 WITH SOFTWARE.

MICROTEK 2 M.B. EXPANDED MEMORY BOARD \$195 (O. K)
MEETS LOTUS/INTEL/MICROSOFT SPEC. FOR EXPANDED MEMORY. FOR USE AS EXPANDED MEMORY (EMS), ELECTRONIC DISK, OR PRINT SPOOLER. WITH A SUPERB MANUAL AND INSTALLATION SOFTWARE. FOR PC/XT OR MOST COMPATIBLES. FOR 2 MEGABYTES OF 150 NS RAM (72 DEVICES) INSTALLED AND BURNED IN ADD \$225.

ZRT-80 CRT TERMINAL BOARD! A LOW COST Z-80 BASED SINGLE BOARD THAT ONLY NEEDS AN ASCII KEYBOARD, POWER SUPPLY, AND VIDEO MONITOR TO MAKE A COMPLETE CRT TERMINAL. USE AS A COMPUTER CONSOLE, OR WITH A MODEM FOR USE WITH ANY OF THE PHONE-LINE COMPUTER SERVICES.

- FEATURES:
- * Uses a Z80A and 6845 CRT Controller for powerful video capabilities.
 - * RS232 at 16 BAUD Rates from 75 to 19,200.
 - * 24 x 80 standard format (60 Hz).
 - * Optional formats from 24 x 80 (50 Hz) to 64 lines x 96 characters (60 Hz).
 - * Higher density formats require up to 3 additional 2K x 8 6116 RAMS.
 - * Uses N.S. INS 8250 BAUD Rate Gen. and USART combo IC.
 - * 3 Terminal Emulation Modes which are Dip Switch selectable. These include the LSI-ADM3A, the Heath H-19, and the Beehive.
 - * Composite or Split Video.
 - * Any polarity of video or sync.
 - * Inverse Video Capability.
 - * Small Size: 6.5 x 9 inches.
 - * Upper & lower case with descenders.
 - * 7 x 9 Character Matrix.
 - * Requires Par. ASCII keyboard.



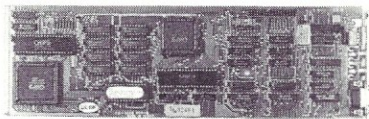
\$89.95 A&T ADD \$50
#ZRT-80
(COMPLETE KIT, 2K VIDEO RAM)

BLANK PCB WITH 2716 CHAR. ROM. 2732 MON. ROM
\$49.95

SOURCE DISKETTE - ADD \$10
SET OF 2 CRYSTALS - ADD \$7.50

FOR 8 IN. SOURCE DISK (CP/M COMPATIBLE) ADD \$10

PC/XT AT ENHANCED GRAPHICS ADAPTER



\$249

- * 3 CARDS IN ONE!
- * 100% IBM COMPATIBLE.
- * COMPATIBLE WITH IBM EGA, COLOR GRAPHICS ADAPTER, OR MONOCHROME ADAPTER.
- * 256K VIDEO RAM! (4 TIMES MORE THAN IBM!)
- * DUAL FREQUENCY OUTPUT FOR EITHER EGA OR STANDARD RGB COLOR MONITORS.
- * PERFECT MATE FOR NEC MULTISYNC COLOR MONITOR!
- * LIGHT PEN INPUT.
- * FULL 16 COLORS.
- * PARALLEL PRINTER PORT.
- * A SUPERIOR BOARD AT ABOUT 1/3 THE COST OF IBM EGA!
- * ASSEMBLED, TESTED, AND BURNED IN. WITH MANUAL.
- * USES CHIPS & TECH. VLSI 4 CHIP SET.
- * FIELD PROVEN BIOS.

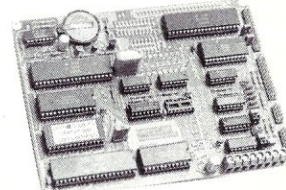
THE NEW 65/9028 VT ANSI VIDEO TERMINAL BOARD!

* FROM LINGER ENTERPRISES *

A second generation, low cost, high performance, mini sized, single board for making your own RS232 Video Terminal. This highly versatile board can be used as a stand alone video terminal, or without a keyboard, as a video console. VT100, VT52 Compatible.

MICRO SIZE!

- FEATURES:
- * Uses the new CRT9128 Video Controller driven by a 6502A CPU
 - * On-Screen Non-Volatile Configuration
 - * 10 Terminal Modes: ANSI, H19, ADM-5, WYSE 50, TVI-920, KT-7, HAZ-1500, ADDS 60, QUIME-101, and Datapoint 8200
 - * Supports IBM PC/XT, and Parallel ASCII Keyboards
 - * Supports standard 15.75 kHz (Horiz.)
 - * Composite or Split Video (50/60 Hz)
 - * 25 X 80 Format with Non-Scrolling User Row
 - * Jump or Smooth Scroll
 - * RS-232 at 16 Baud Rates from 50 to 19,200
 - * On Board Printer Port
 - * Wide and Thin Line Graphics
 - * Normal and Reverse Attributes
 - * Cumulative Character Attributes: De-Inten, Reverse, Underline and Blank
 - * 10 Programmable Function Keys and Answerback message
 - * 5 X 8 Character Matrix or 7 X 9 for IBM Monitors
 - * Mini Size: 6.5 X 5 inches
 - * Low Power: 5VDC @ .7A, ± 12VDC @ 20mA.



\$99.95

ADD \$40 FOR A&T
(Full Kit w/100 Page Manual)

SOURCE DISKETTE: PC/XT FORMAT 5 1/4 IN. \$15

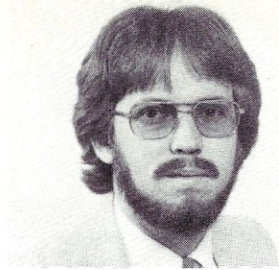
NEW OPTION
PC/XT keyboard ROM. Allows use of IBM style keyboard. \$15

Digital Research Computers

P.O. BOX 381450 • DUNCANVILLE, TX 75138 • (214) 225-2309

TERMS: Add \$3.00 postage. Orders under \$15 add 75¢ handling. No C.O.D. We accept Visa and MasterCard. Tex. Res. add 5-1/8% Tax. Foreign orders (except Canada) add 20% P & H. Orders over \$50 add 85¢ for insurance. Prices subject to change without notice.

Turbo Pascal Corner



by Stephen Randy Davis

This column features tips and techniques for using Turbo Pascal productively on MS/PC-DOS and CP/M microcomputer systems. It discusses typical problems and their solutions. Reader suggestions, comments, and questions are encouraged. Address them to Turbo Pascal Corner, Route 5, Box 107K, Greenville, TX 75401 or through MCI mail, 289-6124.

There have been several winners of the free public-domain software mentioned in the July/August column. For those readers unfortunate enough to have missed that column, I offered a disk of public-domain utilities to anyone submitting a problem or topic worthy of consideration in this column. The offer still stands, so keep those cards and letters coming! You can't win if you don't send it in.

Two commercial products have recently crossed my desk and I feel obliged to mention them. TurboPower Utilities has begun marketing a commercial version of David Baldwin's source-level Turbo debugger mentioned earlier under the name T-Debug Plus. This new version adds a few features to the public-domain TDebug, and I trust that TurboPower will continue to add more. This is a good buy if you do not have access to the public-domain version or if you need the customer support that TurboPower provides.

How many of you have run up against the 64K code limit of Turbo? Of course, there are always overlays, but for those of you to whom *overlay* is a bad word, Path-Finder Software at P.O. Box 43, Littleton, CO 80160, now offers TurboLink+. Not only does TurboLink+ support a large model for Turbo, but it also allows you to link Turbo modules with Turbo Prolog, C, Microsoft Pascal, assembler, and even interpretive and compiled BASIC programs. I cannot exactly recommend the technique used, but it does seem to work.

It looks like reviews of Turbo Pascal utilities are becoming a regular part of this column. I will review any such utilities that I receive.

TODAY'S TOPIC

One area with which not enough Turbo Pascal programmers concern themselves is

Some Worthwhile Turbo Utilities & Addressing in Pascal

that of addressing. Even an avowed nonassembler should have some understanding of how Turbo handles addresses.

For example, did you know that variables declared globally are more than twice as fast to access as variables declared locally to a procedure? (Globally declared variables are declared outside of any procedures.) A benchmark I wrote to test this effect took 0.7 seconds to run with global variables and 1.9 seconds with identically declared local variables. Global variables occupy specific memory locations, just like code, while local variables are stored on the stack, requiring more clock cycles for access on the 8088 or Z80. This effect is less pronounced on an AT because the 80286 processor is more efficient in accessing the stack. Knowing this, you can always speed up procedures by assigning arguments to global variables before manipulating them extensively and by using globally declared variables whenever reentrant code is not a consideration.

Another area in which addressing arises is that of argument passing. Did you ever wonder why sometimes arguments to procedures are declared with a "Var" preceding them and sometimes they are not? If so, try the following program:

```
Var
  a,b: Integer;

Procedure Test (a: Integer; Var b: integer);
Begin
  WriteLn ('a = ',a, ' b = ',b);
  a := 10; b:= 20;
  WriteLn ('a = ',a, ' b = ',b)
End;

Begin
  a := 1; b:= 2;
  Test (a,b);
  WriteLn ('in main routine a = ',a, ' b = ',b)
End.
```

Note that changing the Var variable *b* within the procedure also changes its value in the main routine, whereas changing the non-Var *a* has no effect in the main routine. This is because *a* is being passed by value. That is, the value 1 is passed to the procedure Test, which stores this into its own variable *a*. By adding the Var, you instruct Turbo to pass not the value of *b* but the address of *b*. Accesses of the variable *b* from within Test refer to the same *b* as from the main routine. Thus, changing *b* within the procedure changes its value everywhere.

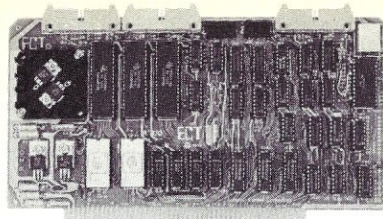
Var variables represent but one case where addresses are useful. "C" programmers become used to referring routinely to addresses of variables, arrays, and routines. Most become discouraged that Pascal apparently has no such ability. True, Chapter 15 of the Turbo manual introduces pointer variables, but it only describes them in terms of records.

Don't misunderstand me—linked lists of records via pointers to records is an important application of addresses. My point is that Pascal allows pointer variables to point to anything. For example, the following declarations are all quite legal:

```
Type
Ptr = ^Integer;
String = Array [1..15] of
Integer;
StrPtr = ^String;
PtrPtr = ^Ptr; {pointers to
pointers are even
allowed}
```

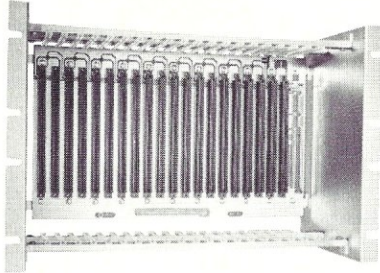
The list of possible applications is limited only by one's imagination. For example, pointers can be used to declare data structures larger than 64K. Suppose you

**CUSTOM
PRODUCTS**
DESIGN • LAYOUT
MANUFACTURING



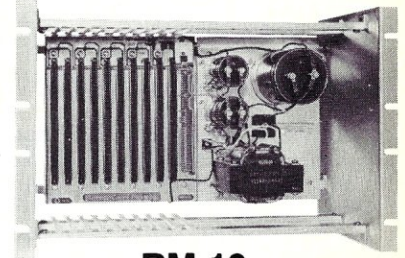
R 2/I/O
ROM/RAM & I/O

**S-100
PRODUCTS**

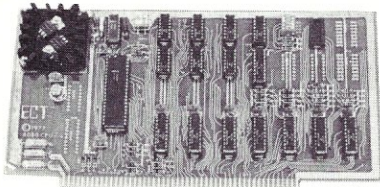


ECT-100-F
RACKMOUNT CARD CAGES

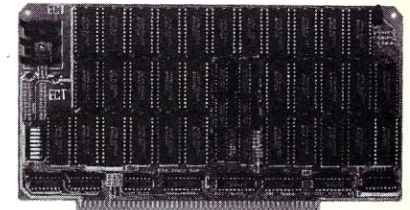
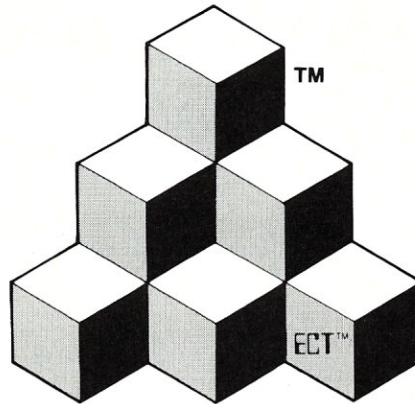
ECT™



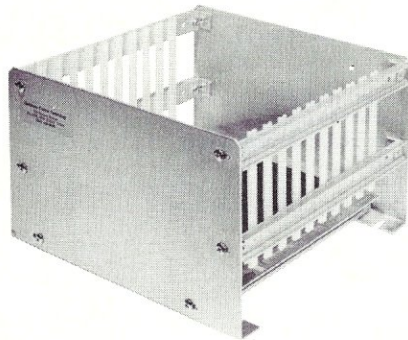
RM-10
CARD CAGE & POWER SUPPLY



8080 CPU
CENTRAL PROCESSING UNITS



64K RAM
FULLY STATIC MEMORY



CCMB-10-F MIN
6, 10 OR 20 SLOT CARD CAGES

**BUILDING BLOCKS
FOR
MICROCOMPUTER SYSTEMS,
DEDICATED CONTROLLERS
AND TEST EQUIPMENT**

CARD CAGES, POWER SUPPLIES
MAINFRAMES, CPU'S, MEMORY
I/O, OEM VARIATIONS

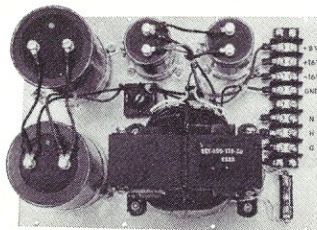


TT-10
TABLE TOP MAINFRAMES

ELECTRONIC CONTROL TECHNOLOGY, INC.

10 Cottage St., Berkeley Heights, NJ 07922 (201) 464-8086

**SPECIALIZING IN
QUALITY
MICRO COMPUTER
HARDWARE**



PS-30 A
POWER SUPPLIES

**MULTIBUS®
PRODUCTS**

MULTIBUS IS A TRADEMARK OF INTEL CORP.

wanted to save as many display screens as you had memory space for. Readers of my last column will remember that the display screen on a PC compatible can be viewed as a matrix of 80×25 integers. Listing 1 shows the portion of the program that allocates space for as many screens as possible given available memory. 64K is no limit here!

Have you ever attempted to pass a procedure to another procedure? By this I do not mean a call such as:

```
ProcTwo (ProcOne (arg))
```

for this is nothing more than passing the results of ProcOne to the second procedure, ProcTwo. I mean really passing ProcOne to ProcTwo. Wirth foresaw such declarations in his original definition of Pascal, but Borland did not provide for this in Turbo. With a little trickery, however, you can reintroduce the passing of procedures to procedures. Listing 2 demonstrates the calling of procedures indirectly.

Why would you ever want to call procedures indirectly? "C" programmers can tell you that it becomes possible to build a routine that can sort anything, if you can provide two routines: one to compare two things, and one to swap two things. You can then use the same sort routine to sort social security numbers, real numbers, latitude/longitude pairs, or whatever. It is just as easy to generate arrays of procedures to be called rapid fire from For loops, or depending upon a calculated integer index—doing it this way generates much smaller and faster code than a case statement.

Note that Z80 Turbo users can participate in the same trick. First, replace `Ofs()` with `Addr()`. Second, replace the 8086 assembler in `Indirect()` with something like the following (this should work, but I have no way of testing it):

```
POP HL
EX (SP),HL
JP (HL)
```

CONCLUSION

Sometimes it is advantageous to consider the assembler code being generated by the Turbo code you write. If something here needs further clarification, or if you have another problem that you would like to see covered, send it in and I will try to send you your copy of public-domain Turbo utilities. §

Stephen Randy Davis is a senior systems programmer for a defense contractor in Greenville, Texas; where he programs various microprocessors. He is also working on his Masters in physics.

{Listing #1 --

this program demonstrates that Turbo Pascal is not limited to arrays 14k bytes or less as is commonly felt. By using the heap, the only real limitation on the size of data arrays is system memory. The calls `New`, `MaxAvail`, and `SizeOf` are described in the Turbo manual.}

Type

```
Screen = Array [0..24] of Array [0..79] of Integer;
ScrnPtr = ^Screen;           {pointer to a screen}
```

Var

```
Images : Array [0..99] of ScrnPtr;
Max : Integer;
Memory : Real;
```

Begin

```
max := 0;
while maxavail > (sizeof(Screen) div 16) do
    {MaxAvail answers in paragraphs}
begin
    new (Images[max]);
    max := max + 1
end;
writeln ('number of blocks of ', sizeof(Screen), ' bytes
        allocated is ', max);
Memory := 1.0*max*sizeof(Screen);
writeln ('total allocated storage of ', Memory:6:0, ' bytes');
If Memory > 65535. Then
    writeln ('the magic 64k boundary has been breached!')
end.
```

{Listing #2 --

This program demonstrates the passing of one routine as an argument to another.}

Type

```
Address = Integer;
           {lets make up a type for pointers to routines}
```

Var

```
RoutineAdr : Address;
           {we will store the address of Target here}
```

{we use Indirect to effect the indirect call}

```
Procedure Indirect (Routine:Address;AnotherArg:Integer);
```

Begin

```
    Inline (
        {get the target address}
        $8B/$86/Routine/ {MOV AX,[BP+Routine]}
        {now pull the stack frame down}
        $8B/$E5/         {MOV SP,BP}
        $5D/             {POP BP}
        {and then jump to target routine}
        $FF/$E0)        {JMP AX}
End;
```

{target routine must have same arguments as Indirect}

```
Procedure Target (Dummy:Address;AnotherArg:Integer);
```

Begin

```
    writeln ('We made it to the target procedure
            (2nd arg =', AnotherArg:4, ')')
```

End;

Begin

```
    RoutineAdr := Ofs (Target); {put addr of Target into variable}
    writeln ('Here we go!');
    Indirect (RoutineAdr,2);     {pass address of routine as arg}
    writeln ('We made it back OK also') {show that we got back}
End.
```


Quality S-100 Products

Now Available! FULCRUM'S New...

256K STATIC RAM

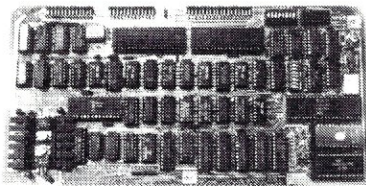
▶ 8/16 Bit Transfers
▶ 24 Bit Extended Addressing
▶ Battery Back-Up

\$495

Designed For Speed & Flexibility.

MPUZ CPU

This NEW MPUZ CPU utilizes the Z-80 8MHz uP as a basis for its 8MHz CPU for S-100 systems, and has been carefully designed to meet the requirements of the IEEE - 696 standard. The quality and performance this CPU provides is rarely found in S-100 products, and you can see why... **only \$299**

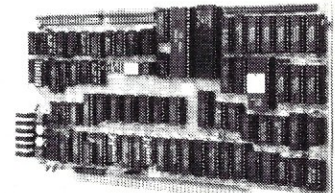


- ▶ 4 or 8MHz clock rate
- ▶ Two RS-232 serial ports
- ▶ Centronics printer ports
- ▶ Real time clock with battery back-up
- ▶ Vectored interrupts to any block location in memory
- ▶ Programmable timer
- ▶ ROM monitor
- ▶ Power on Jump
- ▶ On board wait states
- ▶ 2K of RAM space
- ▶ 24-bit extended addressing
- ▶ Latched Status
- ▶ Front panel compatibility
- ▶ MPM support

Best Value In Disk Controllers.

OMNIDISK

Now the FULCRUM OMNIDISK offers S-100 systems users a unique marriage of component compatibility and technological innovation. These together produce features not found in any conventional disk controllers made today. See for yourself what tomorrow looks like... **only \$299**



- ▶ Simultaneous support of both 5¼" and 8" floppy disks and hard disks
- ▶ Complete 24 bit DMA
- ▶ Power on boot for 5¼" and 8" floppy and hard disks
- ▶ Power on boot PROM
- ▶ On board de-blocking to save RAM space over BIOS
- ▶ Interfaces with the WD 1001* hard disk controller
- ▶ Supports 13 devices simultaneously
- ▶ Full track buffer allows controller to recall entire track
- ▶ DMA'S at 10 MHz
- ▶ Supports MS DOS
- ▶ 10K on board buffer saves two K of TPA

So before you buy another S-100 component, call or write for our FREE catalog. And see how your system can benefit from the FULCRUM difference.

*CP/M 2.2 configured for OMNIDISK \$60. *Trade mark of Digital Research. FREE U.P.S. ground shipping on prepaid orders. Shipping is added to VISA, M.C. and C.O.D. orders. CA residents, please add sales tax.

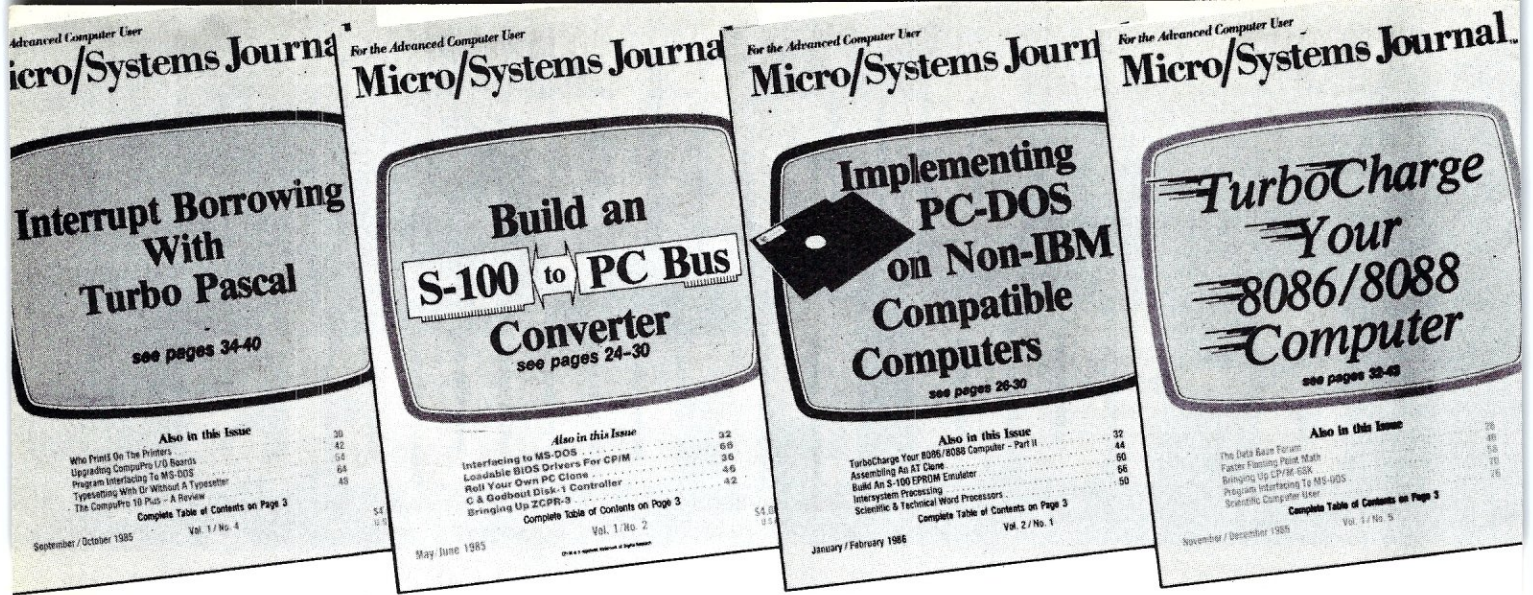


707/433-0202

**FULCRUM
COMPUTER PRODUCTS**

459 Allan Court, Healdsburg, CA 95448.

Also in FULCRUM'S Family: OMNIRAM 64K memory board, Serial I/O 2-2 & Video I/O interface boards, Relay board, I-8080, 8015 and 8035 main frames with 21 slot mother boards, CPA front panel and A/D board.



The only magazine by and for advanced micro users.

Micro/Systems Journal is the independent magazine for advanced software and hardware hackers who use MS/DOS, CP/M, MP/M, TurboDOS, Concurrent-DOS and Xenix. An important resource for programmers who use C, Turbo Pascal, Forth, Lisp and Assembler. It also covers hardware based on the S-100, PC and AT busses, as well as powerful single board and multi-user systems. Its name — MICRO/SYSTEMS JOURNAL!

Each issue is packed with practical articles...in depth tutorials, reviews, hints...the latest information on SIG/M, PC/Blue, C-User Group and PC-SIG Public Domain Software. Advanced topics such as 16-bit and 32-bit microprocessors, high performance data acquisition systems, interfacing to peripherals, patching application software packages for improved performance...and lots more! Strictly technical info...no fluff...stuff to keep every hacker up-to-date on the ever-changing micro technology.

Edited by Sol Libes, former editor of Microsystems, former columnist in Byte, PC Tech Journal, PC Week, Computers & Electronics and other magazines and author of 15 books. An authority in the field since the early seventies.

Here are examples of topics covered in recent issues.

- Build an S-100 to PC-Bus Interface
- Loadable Drivers for CP/M
- Structured Programming with Microsoft's M80 Assembler
- Review of Three 16-Bit Lisp & Prolog Packages
- Local Variables in Forth
- Installing ZCPR3
- Build a PC Clone for Under \$1,000
- Interfacing to MS-DOS
- Assembly Language Extensions For MS-Basic
- Reviews of Scientific Word Processors, Concurrent DOS and Turbo Pascal Version 3.0

Yes! I accept your special offer.

- Please send me 1 Year (6 issues) for \$20
- Increase my savings by sending me 2 Years (12 issues)

Name _____

Company _____

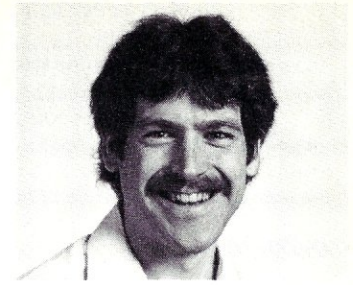
Address _____

City _____

- Payment enclosed Bill me later

Canada & Mexico add \$3 for surface mail; \$7 for airmail. All foreign subscriptions must be prepaid in U.S. dollars on a U.S. bank.

The C Forum



by Don Libes

cdecl—A Program to Explain C Declarations

This column features tips and techniques for using the C language productively. It discusses typical problems with using C and their solutions. Reader suggestions, comments, and questions are encouraged. Address them to "The C Forum," Micro/Systems Journal, Box 1192, Mountainside, NJ 07092.

cdecl is a program for encoding and decoding C-type declarations. It was written by Graham Ross (tektronix!tekmdp!grahamr). It is available from mod.sources on Usenet and is in the public domain. The entire source follows this article.

cdecl is easy to use. You simply run it and type in C declarations or English descriptions of C declarations. cdecl will convert either of those forms back and forth as necessary. cdecl also constructs casts. For example, if I type:

```
explain int (*foo)( )
```

cdecl responds:

```
declare foo as pointer to function returning int
```

If I type that in, cdecl prints the original declaration.

Some more examples: To declare an array of pointers to functions like malloc(), type:

```
declare fptab as array of pointer to function  
returning pointer to char
```

The result is:

```
char *(*fptab[])( )
```

The proper declaration for UNIX's signal() cannot be described in cdecl's language (it can't be described in C either). An adequate declaration for most purposes is given by:

```
declare signal as function returning pointer to  
function returning int
```

```
int (*signal( ))( )
```

The function declaration that results has two sets of empty parentheses. The author of such a function might wonder where the parameters go.

```
declare signal as function (sig_args) returning  
pointer to function (other_args) returning int
```

provides the solution:

```
int (*signal(sig_args))(other_args)
```

To cast something into signal's type (it must be a pointer to it):

```
cast foo into pointer to function returning pointer  
to function returning int
```

```
(int (*(*)())())foo
```

So the next time you have to figure out how to:

```
declare foo as function returning pointer to  
function returning pointer to array 5 of pointer to  
pointer to pointer to pointer to function returning  
pointer to function returning pointer to function  
returning float
```

you will know to write it as:

```
float (*(**(**(*(*foo())())[5])())())()
```

COMMAND LANGUAGE

There are four statements in the language. The declare statement composes a C-type declaration from a verbose description. The cast statement composes a C-type cast as it might appear in an expression. The explain statement decodes a C-type declaration, producing a verbose description. The help statement describes the others.

The following grammar describes the language. In the grammar, words in < > are nonterminals, and bare lowercase words are terminals that stand for themselves. Bare uppercase words are other lexical tokens: NOTHING means the empty string; NAME means a C identifier; NUMBER means a string of decimal digits; and NL means the new line character.

```
<program> ::= NOTHING  
           |<program> <stat> NL  
<stat>    ::= NOTHING  
           |declare NAME as <decl>  
           |cast NAME into <decl>  
           |explain <cdecl>  
           |help  
<decl>    ::= array of <decl>  
           |array NUMBER of <decl>  
           |function returning <decl>  
           |function ( NAME ) returning <decl>  
           |pointer to <decl>  
           |<type>  
<cdecl>   ::= <cdecl1>  
           |* <cdecl>  
<cdecl1> ::= <cdecl1> ( )  
           |<cdecl1> [ ]
```



```

|<cdecl1| [ NUMBER ]
| ( <cdecl | )
| NAME
<type> ::= <typename> | <modlist>
| <modlist> <typename>
| struct NAME | union NAME | enum NAME
<typename> ::= int | char | double | float
<modlist> ::= <modifier> | <modlist> <modifier>
<modifier> ::= short | long | unsigned

```

WARNINGS

cdecl is somewhat lacking in error messages. It swallows syntactically incorrect statements without complaint, although the declare statement tries to point out constructions that are not supported in C. Also, certain nonportable constructs are flagged.

Lastly, cdecl won't help you figure out storage classes or initializations.

SOURCES

Following this article are the sources to cdecl. Included are a grammar file that should be run through yacc, a tokenizer that should be run through lex, and some utility routines. Compile them together and you will have cdecl. (Austin Code Works sells yacc and lex if you don't already have them.)

Note that some C libraries use strchr() instead of index(). In that case, use the appropriate #define. §

Don Libes is a computer scientist working in the Washington, D.C. area. He works on artificial intelligence in robot control systems.

```

/* cdecl.y - yacc grammar for cdecl */
%{
#include <stdio.h>

#define MB_SHORT0001
#define MB_LONG0002
#define MB_UNSIGNED0004
#define MB_INT0010
#define MB_CHAR0020
#define MB_FLOAT0040
#define MB_DOUBLE0100

int modbits = 0;
int arbdims = 1;
char *savedtype;
char *savedname;
char *ds(), *cat();
char *index(), *malloc();
char prev;
%}

%union {
char *dynstr;
struct {
char *left;
char *right;
} halves;
}

%token DECLARE CAST INTO AS HELP EXPLAIN
%token FUNCTION RETURNING POINTER TO ARRAY OF
%token <dynstr> NAME NUMBER STRUCTUNION UNSIGNED LONG SHORT
%token <dynstr> INT CHAR FLOAT DOUBLE
%type <dynstr> mod_list tname type modifier
%type <dynstr> cdecl cdecl1 cdims adims c_type
%type <halves> adecl

%start prog

%%
prog: /* empty */
| prog stat
;

stat: HELP '\n'
|
help();
}
| DECLARE NAME AS adecl '\n'
{

```

```

printf("%s %s%s\n", savedtype, $4.left, $2, $4.right);
free($4.left);
free($4.right);
free($2);
}
| CAST NAME INTO adecl '\n'
{
if (prev == 'f')
unsupp("Cast into function");
else if (prev=='A' || prev=='a')
unsupp("Cast into array");
printf("%s", savedtype);
if (strlen($4.left)+strlen($4.right))
printf(" %s", $4.left, $4.right);
printf("%s\n", $2);
free($4.left);
free($4.right);
free($2);
}
| EXPLAIN type cdecl '\n'
{ printf("declare %s as %s%s\n", savedname, $3, $2); }
| '\n'
| error '\n'
}
yyerrorok;
}
;

cdecl cdecl1
| '*' cdecl
{ $$ = cat($2, ds("pointer to "), NULL); }
;

cdecl1: cdecl1 '(' ' '
{ $$ = cat($1, ds("function returning "), NULL); }
| cdecl1 cdims
{ $$ = cat($1, ds("array "), $2); }
| '(' cdecl ')'
{ $$ = $2; }
| NAME
{
savename($1);
$$ = ds("");
}
;

cdims: '[' ']'
{ $$ = ds("of "); }
| '[' NUMBER ']'
{ $$ = cat($2, ds(" of "), NULL); }
;

adecl: FUNCTION RETURNING adecl
{
if (prev == 'f')
unsupp("Function returning function");
else if (prev=='A' || prev=='a')
unsupp("Function returning array");
$$ .left = $3.left;
$$ .right = cat(ds("("), $3.right, NULL);
prev = 'f';
}
| FUNCTION '(' NAME ')' RETURNING adecl
{
if (prev == 'f')
unsupp("Function returning function");
else if (prev=='A' || prev=='a')
unsupp("Function returning array");
$$ .left = $6.left;
$$ .right = cat(ds("("), $3, ds(")"));
$$ .right = cat($$.right, $6.right, NULL);
prev = 'f';
}
| ARRAY adims OF adecl
{
if (prev == 'f')
unsupp("Array of function");
else if (prev == 'a')
unsupp("Inner array of unspecified size");
if (arbdims)
prev = 'a';
else
prev = 'A';
$$ .left = $4.left;
$$ .right = cat($2, $4.right, NULL);
}
| POINTER TO adecl
{
if (prev == 'a')

```



```

unSUPP("Pointer to array of unspecified dimension");
if (prev=='a' || prev=='A' || prev=='f') {
$$left = cat($3.left,ds("(*)",NULL));
$$right = cat(ds(""),$3.right,NULL);
} else {
$$left = cat($3.left,ds("(*)",NULL);
$$right = $3.right;
}
prev = 'p';
}
| type
{
savetype($1);
$$left = ds("");
$$right = ds("");
prev = 't';
}
;

adims: /* empty */
{
arbdims = 1;
$$ = ds("");
}
| NUMBER
{
arbdims = 0;
$$ = cat(ds("[", $1, ds("]"));
}
;

type: tinit c_type
{ mbccheck(); $$ = $2; }
;

tinit: /* empty */
{ modbits = 0; }
;

c_type: mod_list
{ $$ = $1;
| tname
{ $$ = $1; }
| mod_list tname
{ $$ = cat($1,ds(" "),$2); }
| STRUCTUNION NAME
{ $$ = cat($1,ds(" "),$2); }
}
;

tname: INT
{ modbits |= MB_INT; $$ = $1; }
| CHAR
{ modbits |= MB_CHAR; $$ = $1; }
| FLOAT
{ modbits |= MB_FLOAT; $$ = $1; }
| DOUBLE
{ modbits |= MB_DOUBLE; $$ = $1; }
;

mod_list: modifier
{ $$ = $1; }
| mod_list modifier
{ $$ = cat($1,ds(" "),$2); }
;

modifier: UNSIGNED
{ modbits |= MB_UNSIGNED; $$ = $1; }
| LONG
{ modbits |= MB_LONG; $$ = $1; }
| SHORT
{ modbits |= MB_SHORT; $$ = $1; }
;
%%
#include "cdlex.c"

#define LORS(MB_LONG|MB_SHORT)
#define UORL(MB_UNSIGNED|MB_LONG)
#define UORS(MB_UNSIGNED|MB_SHORT)
#define CORL(MB_CHAR|MB_LONG)
#define CORS(MB_CHAR|MB_SHORT)
#define CORU(MB_CHAR|MB_UNSIGNED)

mbccheck ()
{
if ((modbits&LORS) == LORS)
unSUPP("conflicting 'short' and 'long'");
if ((modbits&UORL) == UORL)
unPORT("unsigned with long");
if ((modbits&UORS) == UORS)

```

Modula-2

IBM PC/DOS

Native Code Compiler

This is a full implementation of Niklaus Wirth's Modula-2 language. Our product is not an interpreter, but a true 8086 compiler, using state-of-the-art techniques. A Unix-like "make" utility is included which provides automatic recompilation of modified source programs.

The code generator produces object module input for the DOS link utility. You may combine your Modula-2 programs with code from other languages such as Assembler. The software also operates on PC compatibles using MS/DOS. All the run time source code is included. None of the software is copy protected, and is fully supported and maintained by farware. No royalties are charged for the use of the run time object. A complete and comprehensive reference manual is included in the purchase price. The manual is available separately for \$25.00.

Site licenses and quantity discounts are available.

\$89.95 Complete

farware
1329 Gregory
Wilmette, IL 60091
(312) 251-5310

Master Card and Visa Accepted

**THE PROVEN
MICRO FAMILY**
for all the
**INDUSTRY
APPLICATIONS**



SLICER®

*Specials

New Low Prices!

*Multi-User Slicer Combo	\$950.00
Slicer SBC A&T (No RAM)	
Slicer 1 MEG Board A&T (Complete)	
CCP/M by Digital Research Inc.	
Slicer SBC A&T 256K	\$445.00
*Slicer SBC A&T 128K	\$350.00
with double deck sockets for additional 128K	
*Slicer SBC A&T (No RAM)	\$295.00
*Slicer 1 MEG Board A&T	\$450.00
*Slicer 1 MEG Board (Full Kit)	\$295.00
Slicer PC Expansion Board A&T	\$495.00
CCP/M (Digital Research Inc.)	\$250.00
MS DOS (Micro-Soft Corp.)	\$175.00
*New Slicer Bios for MS DOS	Call

Other kit forms available - Enclosures & other support hardware in stock - Call or write for latest information & prices!

SLICER COMPUTERS INC.

2543 Marshall Street N.E.
Minneapolis, MN 55418

(612) 788-9481
BBS (612) 788-5909

BOBCAT

The finest disk catalog program available
for PC-DOS or CP/M

- * Keeps track of files on floppies or hard disks
- * Compatible with PC-DOS 2.0 or 3.0; CP/M 2.2 or 3.0
- * Contains the most advanced features:
 - 65 character individual file comment line
 - 45 character individual disk title
 - ASCII BOBFILE (disk number file) storing the disk title and file comment lines. Great for quick finds or looks in each disk or directory
 - full directory and subdirectories compatibility
- * File comment lines for both text and binary files
- * Unprotected for ease of use
- * A proven utility sold since 1981

Price: US Residents \$49.95 US funds
 Canadian Residents \$49.95 Canadian funds
 Other Countries \$54.95 US funds
 plus \$3.00 postage and handling
 Ontario Residents please add 7% sales tax
 Extra DOS or licenses - \$15.00 each to \$75.00 maximum

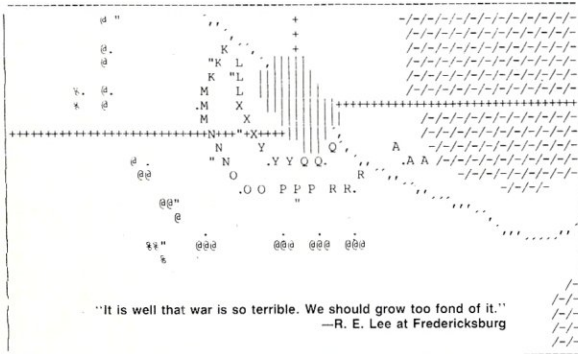


MASTERCARD, VISA, BANK DRAFTS, MONEY ORDERS
 COMPANY CHECKS, PURCHASE ORDERS
 Please allow 10 days clearance for PERSONAL CHECKS

R&i. MicroServices Inc.
 Box 15955, Station F
 Oshawa, Ontario, Canada
 K2C 3S8 (613)225-7904



The home of the BOBCAT



Robert E. Lee would have been very fond of **GENERAL**. **GENERAL** simulates the anxiety, uncertainty, and despair felt by the commanders of armies in an imaginary war taking place a century or so ago. **GENERAL** is . . .

. . . a **TWO-PLAYER** game. It requires TWO CP/M-80 or IBM PC computers communicating via modem or cable at 300 baud. The computers are non-playing "umpires"; they produce statistical reports and animated displays of the battle situation, and relay the players' commands between the machines.

. . . a **REAL-TIME** game. It is always "your turn". The video display changes continually as time passes.

. . . a **COMPLEX** game. Each player controls an army of twenty-six units. There can be action in several places at once, and the decision which to attend to first can be critical.

. . . a **SERIOUS** game. It has had over five years of development and testing. A typical game lasts about two hours and requires constant, intense concentration. The game has an immense variety of possible situations, and does not grow stale with repeated playing.

GENERAL is available from: Tee-Kay Software
 P.O. Box 23771
 Columbus, Ohio 43223

Diskette and User's Manual: \$50.00
 User's Manual only: \$10.00
 Serial I/O Software Listings: \$ 2.00

A variety of 5" CP/M disk formats is available (including Apple), as well as standard 8" SSSD. Specify desired format when ordering. NOTE: The CP/M version may require a small amount of customization of screen and modem control software. Sample ASM sources (8080 code) are included on the **GENERAL** diskette. Listings are available on request.

```
unport ("unsigned with short");
if ((modbits&CORL) == CORL)
unport ("long char");
if ((modbits&CORS) == CORS)
unport ("short char");
if ((modbits&CORU) == CORU)
unport ("unsigned char");
}
```

```
savetype(s)
char *s;
{
savedtype = s;
}
```

```
savename(s)
char *s;
{
savedname = s;
}
```

```
/* cdecl.l - tokenizer for cdecl */
%{
#include <ctype.h>
```

```
char *visible();
%}
N[0-9]
A[A-Z_a-z]
AN[0-9A-Z_a-z]
%%
arrayreturn ARRAY;
asreturn AS;
castreturn CAST;
declarereturn DECLARE;
explainereturn EXPLAIN;
functionreturn FUNCTION;
helpreturn HELP;
intoreturn INTO;
ofreturn OF;
pointerreturn POINTER;
returningreturn RETURNING;
toreturn TO;
```

```
char{ yylval.dynstr = ds(yytext); return CHAR; }
double{ yylval.dynstr = ds(yytext); return DOUBLE; }
enum{ yylval.dynstr = ds(yytext); return STRUCTUNION; }
float{ yylval.dynstr = ds(yytext); return FLOAT; }
int{ yylval.dynstr = ds(yytext); return INT; }
long{ yylval.dynstr = ds(yytext); return LONG; }
short{ yylval.dynstr = ds(yytext); return SHORT; }
struct{ yylval.dynstr = ds(yytext); return STRUCTUNION; }
union{ yylval.dynstr = ds(yytext); return STRUCTUNION; }
unsigned{ yylval.dynstr = ds(yytext); return UNSIGNED; }
```

```
{A}{AN}*{ yylval.dynstr = ds(yytext); return NAME; }
{N}{+}{ yylval.dynstr = ds(yytext); return NUMBER; }
```

```
[\t ];
[*\{}()\n]return *yytext;
{
printf("bad character '%s'\n", visible(*yytext));
```

```
return *yytext;
}
%%
char *
visible(c)
{
static char buf[5];
```

```
c &= 0377;
if (isprint(c)) {
buf[0] = c;
buf[1] = '\0';
} else
sprintf(buf, "\\%02o", c);
return buf;
}
```

```
/* cdsupp.c - support routines for cdecl */
#include <stdio.h>
char *malloc();
```

```
main()
{
yyparse();
}
```


– A GENERAL-PURPOSE CATALOGUE SYSTEM WITH FAST, MULTIKEY BOOLEAN SEARCH –

- User abstracts **keywords** from source materials:

REAL ESTATE LISTINGS
LEGAL TRANSCRIPTS AND DEPOSITIONS
COMPANY AND CLIENT PROFILES
PERSONNEL PROFILES

RESUME'S
PATIENT CHARTS
ARTICLES, BOOKS, CLIPPINGS
WRITER'S OR RESEARCHER'S NOTES

HOLIDAY TOURS AND TRIPS

- Stores the abstracts in a simple universal file structure.
- Searches the "library" for relevant abstracts using **general multikey boolean forms** composed of **lists of keywords**.

- List syntax for boolean forms is simple; lists are stored to file for easy editing.

• **Search algorithm is FAST.**

• **Every application handled in the same way.**

- The all-purpose file structure has just two fields: abstract id number, and keyword. One record for every keyword. User sets the widths of the fields.
- **Categorization of keywords** is accomplished by a variable one-character prefix on the keyword (any printable ASCII character).
- Search lists come in four types: **MATCH**, **NONMATCH**, **GRTR/=** and **LESS/=**.
- Multilist queries come in two types: **ALL** and **ANY**.
 An item satisfies an **ALL** query if it satisfies all the lists (**AND** logic).
 An item satisfies an **ANY** query if it satisfies any list (**OR** logic).
- A list may have any number of keywords, and any mix of categories.
- **Search output is a list of "hits"**, the id numbers of the abstracts which satisfy the query, stored to a file.

Resume Example																								
Categories are: T = title, F = function, G = year graduated, P = product area, \$ = salary, D = degree, M = major, S = school																								
Sample Abstract #43	Sample Queries																							
<table border="0"> <tr><td>id</td><td>keyword</td></tr> <tr><td>43</td><td>T_Asst. Director</td></tr> <tr><td>43</td><td>F_Sales</td></tr> <tr><td>43</td><td>G_73</td></tr> <tr><td>43</td><td>P_Fabrics</td></tr> <tr><td>43</td><td>P_Yarns</td></tr> <tr><td>43</td><td>P_Clothing</td></tr> <tr><td>43</td><td>\$_37000</td></tr> <tr><td>43</td><td>D_MBA</td></tr> <tr><td>43</td><td>M_Marketing</td></tr> <tr><td>43</td><td>S_Columbia U.</td></tr> </table>	id	keyword	43	T_Asst. Director	43	F_Sales	43	G_73	43	P_Fabrics	43	P_Yarns	43	P_Clothing	43	\$_37000	43	D_MBA	43	M_Marketing	43	S_Columbia U.	<p>(a) experience in Sales and Marketing; not a VP or Director; 10-15 years work experience; current salary not higher than \$50K.</p> <p style="text-align: center;">ALL</p> <p style="text-align: center;">MATCH F_Sales</p> <p style="text-align: center;">MATCH F_Marketing</p> <p style="text-align: center;">NONMATCH T_Director T_VP</p> <p style="text-align: center;">GRTR/= G_71</p> <p style="text-align: center;">LESS/= G_76</p> <p style="text-align: center;">LESS/= \$_50000</p>	<p>(b) not a steel industry executive (Director, VP or C.E.O.)</p> <p style="text-align: center;">ANY</p> <p style="text-align: center;">NONMATCH T_Director T_VP T_C.E.O.</p> <p style="text-align: center;">NONMATCH P_Steel</p>
id	keyword																							
43	T_Asst. Director																							
43	F_Sales																							
43	G_73																							
43	P_Fabrics																							
43	P_Yarns																							
43	P_Clothing																							
43	\$_37000																							
43	D_MBA																							
43	M_Marketing																							
43	S_Columbia U.																							

- Searching is fast because all entries across all categories are indexed simultaneously — all "search keys" are inverted at once.
- The system is menu-driven except for file creation and editing.
- Users should know dBASE III, but don't need to be programmers.
- **dBASE III source code is included**, so if you are a programmer you can adapt the system to taste.
- The system provides an option for **automatic encoding of keywords and phrases into 3 or fewer bytes**, leading to dramatic disk savings. (An application of this type is included with every order: abstracts of dBASE III material from TechNotes.)

To order, please send a check for \$49 + \$3.50 (S&H) to

Thinker's Apprentice / 392 Central Park West, Apt. 12X / New York, NY 10025
 (212) 222-5050

(Residents of New York State, please add state and local sales tax.)

dBASE III is a registered trademark, and TechNotes is a copyright publication of Ashton-Tate.


```

unport(s)
char *s;
{
printf("Warning: Unsupported in C -- %s\n",s);
}

unport(s)
ächar *s;
{
printf("Warning: Non-portable construction -- %s\n",s);
}

yyerror(s)
char *s;
{
printf("%s\n",s);
}

yywrap()
{
return 1;
}

/*
* Support for dynamic strings:
* cat creates a string from three input strings.
* Input strings are free'd by cat (so they better have
  been malloc'd).
* ds makes a malloc'd string from one that's not.
*/

char *
cat(s1,s2,s3)
char *s1,*s2,*s3;
{
register char *newstr;
register unsigned len = 0;

if (s1 != NULL) len = strlen(s1) + 1;
if (s2 != NULL) len += strlen(s2);
if (s3 != NULL) len += strlen(s3);
newstr = malloc(len);
if (s1 != NULL) {
strcpy(newstr,s1);
free(s1);
}
if (s2 != NULL) {
strcat(newstr,s2);
free(s2);
}
if (s3 != NULL) {
strcat(newstr,s3);
free(s3);
}
return newstr;
}

char *
ds(s)
char *s;
{
register char *p;

p = malloc((unsigned)(strlen(s)+1));
strcpy(p,s);
return p;
}

static char *helptext[] = {
"[] means optional; {} means 1 or more; <> means defined
elsewhere\n",
"command:\n",
" declare <name> as <english>\n",
" cast <name> into <english>\n",
" explain <gibberish>\n",
"english:\n",
" function [( <name>)] returning <english>\n",
" array [<number>] of <english>\n",
" pointer to <english>\n",
" <type>\n",
"type:\n",
" [[<modifier>]] <C-type>\n",
" {<modifier>} [<C-type>]\n",
" <sue> <name>\n",
"name is a C identifier\n",
"gibberish is a C declaration\n",
"C-type is int, char, double or float\n",
"modifier is short, long or unsigned\n",
"sue is struct, union or enum\n",

```

```

NULL
};

help()
{
register char **p;

for (p=helptext; *p!=NULL; p++)
printf("\t%s",*p);
}

```

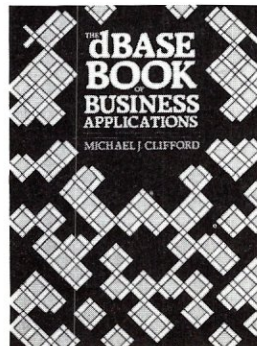
The dBASE Book of Business Applications

The dBASE Book of Business Applications is a tutorial and programming guide for anyone committed to automating office systems and maximizing productivity with dBASE II. The task-by-task approach illustrates the flexibility of dBASE II in adapting to **your** daily business routines.

The book includes the source code for several working application programs including a mailing list manager, a portable computer application, an inventory system, and an invoicing and sales reporting system. You'll find:

- Money and Material Management
- Useful dBASE II Utilities
- Communicating with Clients
- dbasics of Business
- Expense on the Go: A Portable Computer Application

At only \$19.95, this is an indispensable guide to office automation and increased productivity!



**TO ORDER:
CALL TOLL FREE 800-533-4372**

(Mon-Fri, 8-5 Pacific Time)
In Calif: 800-356-2002

Or, return your order and payment to:
M&T Publishing, 501 Galveston Dr.,
Redwood City, CA 94063

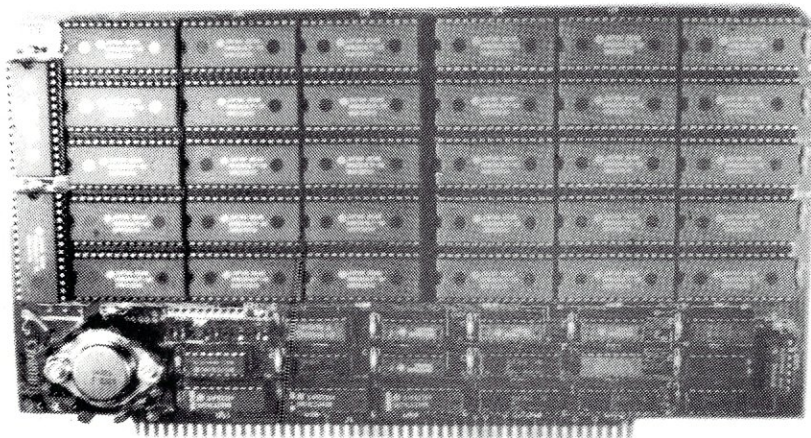
256K BYTES OF 100ns STATIC MEMORY

\$395

NO HIDDEN CHARGES!

Performics prices include dynamic burn in and UPS 2nd day air delivery.

* 128K byte version (depopulated 256K) \$345



S-100

STANDARD FEATURES

- 100ns memory chips
- Operates in excess of 12 MHz
- Certified system ready (industrial grade dynamic burn in)
- Supports 8 and 16 bit data transfers
- IEEE 696/S-100 compatible
- 24 bit addressing
- Address strappable to any 128K block within the 16 meg address range
- Extremely low power consumption
- Single +5 volt operation
- One year warranty

AVAILABLE SOON! High speed dynamic ram boards. 1 & 2 megabyte densities at Performics realistic prices.

PERFORMICS

INC.

Delivering high performance products and excellent service at realistic prices.

(603) 881-8334 ORDER NOW!

TERMS: Check, Visa/Mastercard
Performics guarantees our boards will work in your system or return for full refund.

11 Morning Dove Road ● Kingston, NH 03848
● Hudson, NH 03051



Communicating at 9600 baud

by Steve Bosak and John Sojak

Several factors have made the ability to perform high-speed data transfer over standard analog phone lines increasingly important. While leased lines have been the preferred method, their cost and availability limit the potential base of users. There are simply many more people who can benefit from high-speed data transfer than leased lines can accommodate. Researchers and scientists who wish to exchange massive amounts of data over long distances have few choices when confronted with time-critical computations. Facsimile and CAD programs that require quick screen refreshes must currently put up with slow dial-up modem speeds to and from the remote.

The next few years will see a change. A few 9600-bps dial-up modems are available now, and more will be released in the coming months. The technologies used in these speed demons are impressive; the manufacturers have surmounted formidable communications obstacles to bring these products to market.

This two-part article will discuss the protocols involved in high-speed communications, in terms of both theory and implementation. This first installment covers representative modems from the CCITT Version 29 family, as well as Telebit's proprietary protocol used in its TrailBlazer modem. All modems were tested over a period of weeks on standard dial-up long-distance lines. A description of V.32 follows, as a comparison to the V.29 recommendations. V.32 modem evaluation and a more detailed discussion of V.32 recommendations will be presented in part two.

The main drawback to communication over standard phone lines has always been the inconsistency of line quality. Even short haul lines vary widely. Operating in the telephone bandwidth of 200-3200 Hz, only middle bands are relatively reliable in quality. As you move to the edges of the available channel, distortion and noise increases. Low-speed modems have little problem confining themselves to the main part of the bandwidth. Generally speaking, the easiest way to attain higher speeds is to use a greater chunk of the available band or pack more data into the stable portions of the channel. Very high-speed modem transmission requires both methods.

A Review of Three High Speed Dial-up Modems—Part 1

V.32 OPERATING PRINCIPLES

The new 9600-bps modems fall into one of two categories: asymmetrical or symmetrical (half or full duplex). In conventional, full-duplex 212A and 103 Bell modems, the bandwidth is roughly split between two symmetrical channels, Answer and Originate. Both modems have approximately equal chunks of the bandwidth to operate on. Continuing in this tradition, modems adhering to the Consultative Committee for Telegraph and Telephone (CCITT) V.32 recommendation for data communication have allocated symmetrical channels to the answering and originating modems. As these channels are greedy for space to attain 9600-bps transmission rates, however, the two channels completely overlap in the bandwidth. To assure the integrity of the data each modem must ignore its own transmission—in effect, it tunes itself out. This echo cancellation can be accomplished with a great deal of accuracy over short distances or when the conditions of the line remain constant. Long-haul communications compound the problem of echo cancellation in that line quality and line-switching paths often change, causing erratic echo time rates. Therefore, modems employing cancellation must be able to determine the echo dynamically and filter it out of the stream without loss of the incoming data.

While expanding the use of the available bandwidth poses certain obstacles in design, the method of packing data into those channels is also tricky. The Bell 103 modem offers the simplest example of binary data transmission, one frequency or tone is transmitted for a one and another tone for a zero. To pack the data more densely, you must use further modulation.

Moving from a pure frequency modulation scheme, high-speed modem manufacturers have begun to encode the data by modulating the signal's phase and amplitude. The effect is that each combination of signals then represents more bits of information. Rather than having the on/off representation of frequency modulation, modems employing phase and amplitude modulation have more possible data states. This scheme, known as Quadrature Amplitude Modulation, or QAM, is already in use on 2400-bps modems. To further pack the data and assure its integrity even beyond QAM, some V.32 9600-bps modems use an additional modulation known as Trellis Coded Modulation (TCM). In part two of this article, we will examine the use of both TCM and QAM in V.32 modems.

The complexity of designing a full-duplex V.32 modem is enormous, and the V.32 modems that have been produced so far have price tags that reflect the sophisticated hardware required. British Telecom and Concord Data have introduced V.32s at around \$3500. The V.32 field is about to expand, however, with NEC America, Codex, and U.S. Robotics planning to introduce their own V.32s in mid to late summer. Representative modems from the V.32 family will be evaluated in part two of this article.

V.29 OPERATING PRINCIPLES

The V.29 recommendation allows asymmetrical or half-duplex transmission at speeds of 9600-bps or higher. In an asymmetrical scheme, the bandwidth is carved out on demand. Most data communications involve the transmission of large blocks of information in one direction or another—not both. Since the data and its speedy arrival is of importance, why not allocate most of the available bandwidth to whichever end needs to transmit that data? Essentially, that is what V.29 modems do. When the modem determines that data is being sent in one direction, the majority of the bandwidth is turned over to the transmission, while a narrow portion is reserved for the receiver for house-keeping chores such as acknowledgement of received packets.

There the similarities among V.29 mo-



Dr. Dobb's Stands Apart

Take a good, hard look at the crowded computer magazine market. If you're a serious microcomputerist, one magazine stands apart. **Dr. Dobb's Journal**.

Dr. Dobb's is not for everyone. It is written by and for expert programmers, and it's the oldest and most technically sophisticated microcomputer publication available. Since 1976, **Dr. Dobb's Journal** has been the unchallenged leading source of software tools for advanced programmers.

With the industry moving forward faster than ever, you need to stay a step ahead. **Dr. Dobb's** sets the pace with:

- regular columns on C, Unix, MS-DOS and 16-bit software;
- algorithms and problem solving;
- lively discussions of fundamental software issues;
- inside information on commercial languages and operating systems;
- tips on advanced programming topics.

The information and valuable code contained in **Dr. Dobb's** makes each issue indispensable for serious computing professionals and enthusiasts. Don't miss a single issue of this valuable resource. Subscribe today. If you aren't fully satisfied, cancel your subscription and keep the first issue as a free sample.

To start your subscription, fill out this coupon and return it to:

Dr. Dobb's Journal
P.O. Box 27809, San Diego, CA 92128

.....
 • **Yes!** Please enter my subscription for 12 issues of:
 • **Dr. Dobb's Journal** for \$25.

• If I am not fully satisfied I will write "cancel" on my subscription invoice and keep the first issue as a free sample.

• Name _____

• Address _____

• City _____

• State _____ Zip _____

• Bill me later I've enclosed \$25 Charge my VISA M/C

• Card No. _____

• Expiration _____

• Signature _____

• Offer good in U.S. only Allow 6 to 12 weeks for delivery. 3046

.....

Dr. Dobb's Journal of
Software Tools
 FOR THE PROFESSIONAL PROGRAMMER

dems end. Although most employ QAM of one sort or another, manufacturers have superimposed their own error detection and correction (EDC) and, in some cases, added further compression schemes. As V.29 does not call for an exact type of QAM EDC, it has been left to the manufacturers to determine the best way to assure data integrity. To give you an idea of the different varieties of QAM available, there are at least four versions of MNP, a QAM that also employs data compression, and one QAM based on CRC detection.

Rather than reinventing the wheel at each company, many V.29 9600-bps modem manufacturers rely on a modem chip-set produced by Rockwell International called the R96FAX. There are now a dozen or so companies that have taken advantage of the R96FAX board, because the chips implement V.29 with virtually no

hardware overhead other than the board itself. By doing so, they can offer a V.2X modem at a reasonable price.

The R96FAX communications engine designed by Rockwell was intended for facsimile document transmission applications. The most limiting factor with any type of facsimile document transmission is simply the time it takes to move massive amounts of digital document pixel information from one machine to another over conventional 3002 dial-up telephone lines. The characteristics of fax-type data are not unlike that of a typical information file. The silicon doesn't know the difference between bits produced by an optical scanner and the bits within a file. The Rockwell chip-set and board provided a cost-effective method of transferring information while conforming to the CCITT transmission format.

The R96FAX contains impressive silicon. The board is about $4 \times 2\frac{1}{2}$ inches and contains two 64-pin QIUP (quad inline), one 28-pin, and one 8-pin dual inline IC packages on it. The silicon implementations are a digital signal processor, digital switched capacitor filter elements, and a proprietary analog processor chip. The more interesting features are a switched capacitor array used for a digitally tunable active filter, on-chip active cable equalization for both the transmit and receive data paths, and an automatic adaptive equalizer in the receive path that adjusts itself on the fly to compensate for poor line quality when in V.29 modes. The silicon itself will also generate and detect several types of discrete tones and has diagnostic capability. The off-board support required is virtually any type of processor for error detection, data buffering, and basic RS-232 DCE control.

The R9600FAX modulation is a bit data stream phase shifted and amplitude modulated on the carrier frequency. At 9600 baud, there are 4-bit groups (QAM) representing a 16-point "eye" if one were to view the signal on the line in the time domain. With the V.29 protocol at 9600, the data stream is divided into four groups of two bits each impressed on the carrier frequency. At 7200, there are 3-bit groups (tribits) forming a 16-point structure. At 4800, there are 2-bit groups (dibits) forming an 8-point structure that comprise the encoding. The board set is capable of CCITT V.29 (9600, 7200, 4800, 2400, and 300 baud). The modem transmitter itself, exclusive of any support circuitry, is capable of driving +5 dBm into a 600-ohm load and the receiver dynamic range is from 0-47 dBm. Typical bit-error rate over voice-grade dial-up lines at a 23 dB signal-to-noise ratio and 9600 baud is 10^{-6} in the V.29 mode.

THE ELECTRONIC VAULTS UPTA 96

The Electronic Vaults UPTA 96 utilizes the Rockwell R96FAX board as the data-communications engine controlled by an Intel 8051 family microcomputer. On board is 8K of static RAM and 8K of "program once" PROM for the operating firmware. The modem retails for \$895.

The design objectives of furnishing a functional V.29 data pump in a cost-effective manner were met well. The command set is Hayes-compatible, with extensions to handle the V.29 protocol and some quirks inherent with 9600-bps asynchronous communications. The data block packing method is a header, then a block with an appended CRC16 for ARQ error detection on the receiver end. The modem will do dynamic rate adjustments of 9600, 7200, 4800, 2400, 1200, and 300 baud depending on line quality. The heart of the UPTA is the Rockwell R96FAX modem, which is a powerful V.2X data pump for a dial-up line in its own right.

PC-SPRINT

"PC-Sprint is the most cost effective PC Speedup product on the market."

— Computer Shopper Magazine

- Run your PC, XT or clone at 7.38 mhz.
- 280% Speedup (Norton SI rating)
- Speeds up all software — you can see the difference
- External speed switch
- External reset button
- Change speed "on the fly"
- Compatible with 8087
- Works with all color or mono displays
- "Slotless" plug-in on most PCs
- Includes: Selectable top speed, instructions, warranty, tool, remote mount switch, free BBS subscription

\$89⁹⁵

V20 add \$10. Call for information on other products

Exec-PC, Inc.

P.O. Box 11268 Shorewood, WI 53211

(414) 242-2173



There is little need for much else in order to achieve a good V.2X 9600-baud data transfer. There are eight front-panel LED status indicators for TX and RX data, ring, carrier off hook, and so forth. The rear panel contains connectors for the usual DCE RS232 EIA, a modular RJ11 telephone connection for the phone line, an on/off switch, and DIN style connector for the external power supply. The motherboard is a conventional, two-sided G10 glass epoxy board with good decoupling and bypassing. The RFAX board is piggybacked on the motherboard with standoff spacers.

THE RACAL VADIC 9600VP

The Racal-Vadic 9600VP, with a retail price of \$1495, is also based on the Rockwell R96FAX modem. The 9600VP is controlled by a Z80B 8-bit microprocessor family chip-set, consisting of an SIO/2, CTC, and, of course, the Z80B processor itself. On board is 16K of static RAM and 2K of battery backup RAM for presets and library storage. The operating firmware is contained in a 32K EPROM. The Z80 takes care of housekeeping, data compression, error detection, and data correction, while the actual task of line transfer is taken care of by the Rockwell board. At 6 MHz, the combination of the Z80B and the SIO is an excellent choice for modem control.

Some of the unique aspects of the 9600VP are transparent data compression and error correction. The 9600VP typically will reduce the physical size of an ASCII text file from 20 to 60 percent, making a typical packet sent by the modem smaller, bettering the effective throughput over a straight byte-for-byte transfer. The internal transparent error correction Racal employs, called MNP, will minimize the number of retransmitted blocks. Not all errors are correctable, but if several bits can be corrected locally, the overhead of retransmitting the entire block will be saved. MNP, in effect, can correct mistransmitted blocks by reevaluating the block sent against the coded QAM. If correction is impossible locally, the receiving modem requests a retransmission. The 9600VP, like most other RFAX-based 9600 modems, will also adjust its communications rate dynamically to compensate for phone-line quality in a fall-back and fall-ahead fashion. The modem monitors call progress tones for no answer, busy, reorder, and so forth.

There is a complete complement of front-panel LED status indicators, à la Hayes, indicating data and control status. In addition, there is a membrane keyboard on the front panel to control the voice/data, loopback, and dialing functions, in conjunction with the usual AT commands. Most of these key switches have status LEDs embedded under a translucent viewing area.

The rear panel contains the standard DCE configuration EIA connector and two RJ11 modular telephone connectors—one for the line, the other for a phone. Power is supplied via a DIN-style connector and an external power supply. The motherboard is multilayered. The RFAX board is piggybacked on the motherboard with standoff spacers. The mechanical assembly is superb, with machine-tool sockets on the critical components. These are high-quality sockets with superior extraction resistance. The modem is capable of Bell 103 and 212A transmission, with automatic detection of either.

THE TELEBIT TRAILBLAZER RA12E-T1

If V.29 and V.32 were not enough to crowd the possibilities of moving data at 9600-bps, Telebit manufactures a modem with its own proprietary protocol called PEP

(Packetized Ensemble Protocol). Although it too is an asymmetrically designed modem, it does not follow CCITT recommendations. Telebit TrailBlazer RA12E-T1 Packetized Ensemble modem is a complete departure from the R96FAX-type devices. It is a complete departure from V.2X protocol altogether. The modem represents a good cross section of state-of-the-art silicon devices that can be put together to do upper-middle to lower-high-end digital signal processing in real time. On board is a 16-bit Motorola 68000 processor and a 16-bit Texas Instruments TMS32010 Digital Signal Processor. There are also two hybrid, custom, surface-mounted 8K × 16 RAM modules. It looks as if one is for the 68000, the other for the 32010.

Firmware for the 68000 is contained in two 32K × 8 PROMs of space for the operating system. The signal-processor firm-

• Z Best Sellers •

Z-COM (7 disks) \$119.00

Easy auto-installation complete Z-System for virtually any Z80 computer presently running CP/M 2.2. In minutes you can be running ZCPR3 and ZRDOS on your machine, enjoying the vast benefits. Includes 70+ utility programs and **ZCPR3: The Manual**.

Z-Tools (4 disks) \$150.00

A bundle of software tools individually priced at \$260 total. Includes the ZAS Macro Assembler, ZDM debuggers, REVAS4 disassembler, and ITOZ/ZTOI source code converters. HD64180 support.

PUBLIC ZRDOS (1 disk) \$59.50*

If you have acquired ZCPR3 for your Z80-compatible system and want to upgrade to full Z-System, all you need is ZRDOS. ZRDOS features elimination of control-C after disk change, public directories, faster execution than CP/M, archive status for easy backup, and more!

TERM III (6 disks) \$99.00*

TERM III has been called "a cosmos". Why? Because it's 29 programs, featuring T3MASTER and T3SERVER, are galactic in scope. KERMIT, XMODEM, 1k XMODEM, and batch YMODEM protocols supported.

DSD (1 disk) \$129.95

The premier debugger for your 8080, Z80, or HD64180 systems. Full screen, with windows for RAM, code listing, registers, and stack. We feature ZCPR3 versions of this professional debugger.

Quick Task (3 disks) \$249.00

Z80/HD64180 multitasking realtime executive for embedded computer applications. Full source code, no run time fees, site license for development. Comparable to systems from \$2000 to \$40,000! Request our free Q-T Demonstration Program.

*ZCPR3 required.

Z-System OEM inquiries invited.
 Visa/Mastercard accepted. Add \$4.00 shipping/handling in North America, actual cost elsewhere.
 Specify disk format.



Echelon, Inc.

885 N. San Antonio Road • Los Altos, CA 94022
 415/948-3820 (Order line and tech support)

ware is contained in two of what appear to be $4K \times 8$ devices, making a total of $4K$ available for the digital signal processor firmware. There is an incredible amount of processing horsepower between these two devices. Telebit has defined a new approach to high-speed communications. What happens here is that the DSP (digital signal processor) implements a Fast Fourier Transform on the phone line, breaking it into 512 channels of 7.8 Hz segments and determining what the signal-to-noise ratio and loss for each segment is. Each channel is then either used for data or, if it is determined to be a troublemaker, digitally filtered out on the receiver. Typically, 400 or so channels are active during any session.

Three types of modulation techniques are used: 2-, 4-, and 6-bit QAM. An effective communication rate is determined for each channel, and it is either filtered out altogether, or given 2-bit QAM, 4-bit QAM, or 6-bit QAM. For idle and, for example, console-to-console conversations, the modem will send 23-millisecond duration packets. If the 23-ms packet begins to fill up, the modem will switch to 136-ms packets. There is no error correction at all, but a selective retransmission of packets with errors. If multiple errors are determined, the firmware will recalibrate all over again to determine which channels are now good and which have gone sour. This takes just a fraction of a second. The TrailBlazer monitors call progress tones and has a speaker for connection progress. The single PC board is peppered with discrete "glue" chips, about 30 RAM and PROM, exclusive of the processors. The power supply is built to handle the requirements of the modem, with 3 watts being generated as heat.

There is an unusual but necessary fan in the rear corner to provide forced air circulation. It is quiet and would not pose any type of problem in a typical office environment. There has been substantial attention paid to detail in the mechanical assembly. For example, there are two ferrite-core, wirewound, torrid filters in the power supply that have been placed in a small plastic box mounted to the PC board. Most filters of this type are glued to the board with silicone rubber adhesive. The front panel has the usual status LEDs, and there is a power switch on the right-hand side toward the front. The rear panel has the DCE configuration EIA connector, two RJ11 telephone type connectors, and an interlocked post-power connector for the remote supply. TrailBlazer also incorporates Bell 212A and 103, V.22 and V.22-bis emulation.

TESTING THE MODEMS

In testing the performance of the 9600VP, UPTA 96, and TrailBlazer, we transmitted straight ASCII text files of 196,096 bytes over a typical long-distance phone line. The text file was checked for integrity at

sending and receiving ends by CRC file checks and with DOS utility file compares. Two methods of software transfer were employed. The initial tests were straight, nonprotocol ASCII dumps with the MEX-PC communications program. More exhaustive testing was performed with Crosstalk XVI Fast Version 1.6 software, set to the hardware manufacturers' specifications. Bit-transfer rates were determined by $196,096 \text{ bytes} \times 10$ to determine bits, divided by the transfer time in seconds. Hard disk access time was clocked and subtracted from the transfer time, giving the essential throughput, including handshaking, EDC, and turnaround time. Retransmission of packets was included in the timed tests. The results are shown in Tables 1-4 in the sidebar.

It should be noted that these modems are not designed for interactive or conversational modes. If terminal-to-terminal communication is desired, the operators would do well to switch to the modem's more conventional 103 or 212A speeds. The high speeds are primarily for facsimile and data transfer. Slow character echo will have to be tolerated at the higher speeds when typing back and forth between terminals. The lower speeds in the Racal-Vadic include 300 and 1200-bps, using standard 103 and 212A hardware chips; we found both to operate smoothly and without trouble. The Telebit's lower speeds—1200 and 2400—are emulated by the existing hardware. This emulation was tricky, at best, to run successfully. We had severe problems getting the TrailBlazer to talk to either a Hayes 2400 or USR Courier 2400 in the TrailBlazer Version 22 or Version 22 bis modes. The UPTA 96 we tested had no low-speed transmission; however, the latest version of the 96 can include 300-, 1200-, and 2400-bps rates via snap-on chip-sets to the 96's current board. Like the Racal-Vadic, these are true 103, 212A, and V.22-bis modems utilizing industry standard Rockwell chip-sets.

SETTING UP AND USING THE MODEMS

By far the most difficult modem to get up and running for tests was the Racal 9600VP. The number of registers that had to be set for software purposes was a tad bewildering. To be fair, this register set is extensive, and provides the user with remarkable flexibility in configuration. Synchronous and the 212A asynchronous modes in one modem would almost dictate that some registers be user-selectable. Averaging our test results on the 9600VP, the throughput of the Racal was 5.609k-bps, indicating the modem had fallen back to the 7200-bps mode. Both 16- and 255-block-size transfers were used.

The UPTA 96 was the easiest of the modems to configure and begin transfers with. Even with two and three retransmissions of faulty blocks in two of the

transfers, the UPTA turned in a respectable 6.167k-bps, using the manufacturer's recommended block size of 16. The TrailBlazer did indeed live up to its moniker—with block sizes of 255, the modem screamed in over the line at 7.971k-bps.

To be sure, the effective rates are determined by line quality in real time. Line analysis was available, but we deemed it inconclusive—quality can change dramatically at any given point, bursts and line hits can occur at any time during a transmission. Therefore, care was taken to evaluate the modems over many days and during various times of the day. The discrepancy between the UPTA 96 and the Racal-Vadic 9600VP can probably be accounted for in terms of variants in the line noise. The significant speed difference between the UPTA 96, the 9600VP, and the TrailBlazer can only be accounted for by modem design, however. That kind of a speed gap must indicate a superior design and transfer method.

SOFTWARE

Theoretically, any software package that allows nonprotocol file transfers and serial port settings to at least 9600-bps should be able to handle these modems—or should we say, the modems will handle the software. The less interference with the modem's transmission, the better. We found that a straight ASCII file transfer, no handshaking, ran the smoothest. In MEX-PC we merely called for a straight ASCII dump from the terminal mode. Blanking or quieting the screen writes with MEX's QUIET command gave a marginally faster throughput than an ASCII transfer in which the data was painted to the screen in addition to a capture buffer.

All the modems ran with a minimum of effort in these straight ASCII dumps, and any software that will allow a hands-off transmission will no doubt work. Due to the modems' internal error detection and correction, straight dumps of binary files should pose no problems. We experienced some difficulty with the Crosstalk-Fast software, particularly with the Racal-Vadic and TrailBlazer. To be fair, Crosstalk was not a recommended package for the 9600VP. In the case of the Racal-Vadic, we had to find an acceptable block size. The TrailBlazer quieted down once we set the serial port speed to 19.2k and the modem was allowed to wail at full speed. Again, the less active the software during transfers, the better.

We did marginal tests with QMODEM and Procomm 2.3. Throughput with the various software was similar for each modem tested; for instance, an ASCII dump via MEX and a Crosstalk transfer with the TrailBlazer had only minor bps speed differences—not enough to call statistically significant. Again, this should come as no surprise, as the modem is handling all error detection and correction; the software merely facilitates dial up and assists in a



My department took all the blame They said that putting data processing in control of PC software was too much trouble. Then people started copying software. When we were sued, guess where the fingers pointed.

Software piracy is a Federal crime. At ADAPSO, the computer software and services industry association, we're working to help data processing managers like you prevent the problem. And to help your company avoid substantial fines and legal fees.

You can help top management protect your company by actively discouraging software piracy. ADAPSO can provide you with sample policy statements and free booklets for company employees.

Just return the coupon. It could put you back in control. And it might keep your department off the hook.

Send this coupon or your business card for more information.
Or call us at (703) 522-5055 and ask for Muriel.
ADAPSO, 1300 North Seventeenth Street, Arlington,
Virginia 22209

NAME _____ TITLE _____

COMPANY _____

ADDRESS _____

CITY _____ STATE _____ ZIP _____



Software Piracy is a Federal Crime

PRINTERS & BUFFERS

brother

- M-1109 100-25 CPS, P&S TRACTOR \$189
- M-1509 180/45 CPS, P&S, W/TRAC CALL
- M-1409 AS ABOVE BW 110 CO.,
PARA CALL
- M-1709 240/50 CPS, "PAPER PARK"
FX-286 & IBM PROPRINTER COMPAT, ... CALL
- HR-35 36CPS DAISY, WIDE CARR \$729
- HR-20 20 CPS \$349
- HR-15XL 17 CPS DAISY WHEEL \$295
- CITIZEN MSP-10 .. \$309 MPS-15 .. \$429
- CITIZEN MSP-20 .. \$349 MPS-25 .. \$499
- CORDATA LP-300 LASER WITH FORM EASY
300 x 300 RES 1/2 PG GRAPHICS \$2,089
- CORDATA LP-300 XL LASER
300 x 300 RES 1/2 PG. GRAPHICS
W/FORM-EASY \$2,719
- HANSON BUFFER 64K-256K S-S, S-P, P-s,
P-P \$265
- HANSON LP3000+ LASER W/ HP GL
PLOTTER COMMANDS CALL
- STAR MICRONICS SR-15 \$595
- TOSHIBA P321 PARA & SERIAL \$589
- TOSHIBA P341 \$899
- TOSHIBA P351 PARA & SERIAL \$1,199
- OKIDATA CALL

XEROX Diablo

- D25 \$475 635 \$815
- D80-IF \$1,389 34LQ \$939
- 4045 LASER W/COPIER, 512K RAMS \$4,798

Vohrax

- PERSONAL SPEECH SYSTEM \$295

POWER SYSTEMS & ACCESS

COMPUTER ACCESSORIES

- ALL CABLES Call for Lg Discount!
- S4 4 CIRCUIT SURGE SUPPRESSOR \$19
- S10 6 CIRCUIT SURGE SUPPRESSOR \$29
- P15 5 CIRCUIT MONITOR BASE
WITH MODEM PROTECTION \$98
- U1000 POWER SAVER, (TOPAZ)
W/TRUE "SINE" WAVE CALL



- BC-425 WATT S.P.S. 30 MIN \$419
- BC-675 WATT S.P.S. 70 MIN \$6 25
- BC-1000 WATT 40 MIN \$939
- SAFE (SAFT) SPS1000 WATT
SINE WAVE \$949

MODEMS



- PROMODEM 1200 \$259 2400 CALL
- PROMODEM 1200G \$179; 2400G \$339
- PROMODEM 300C APPLE IIC \$65



- MT224EH MNP ERROR CORREC \$529
- MT224ER RACK MOUNT W/EDC \$495

- CTS 224ADH HAYES COMPATIBLE \$295

- HAYES-ALL MODEMS CALL
- U.S. ROBOTICS BLOW-OUT CALL

MONITORS & TERMINALS

- NEC MULTISYNC CALL
- TECMAR RGB HI-RES ZVM-1360 \$498
- ZENITH ZVM-1240 (IBM) FLAT SCRIN \$149
- PRINCETON GRAPHICS CALL



- CM-1360 6450x200 RGB, 13", G/A SW \$339
- CM-1380F EGA 640x350 RES 256K \$450
- TEGA-22 256K EGA 1/2 CARD \$245
- 1422A DUAL FREQ. TILT/SWIVEL,
14", GREEN/AMBER \$139



- LINK 125 EMUL WY-50 ETC., HI-RES
14" GREEN, 6 SCROLL RATES, IBM SELEC,
K.B. 1 YEAR WARRANTY \$369
- PC-TERM EMUL WY-50, TVI925, IBM-AT
KB, GOOD FOR PC-SLV BDS \$419

- WYSE 60 \$475

- LIBERTY FREEDOM ONE \$395

XEROX TELECOPIER® 7010 TERMINAL

- Don't settle for just a facsimile!
- Send and receive a copy in less than 30 seconds
- Distribution-quality Copies
- Automatic Document Feeder: 10 originals
- 40-digit Operator Display
- CCITT Groups 3 and 2
- Compatibility and more!

\$1,695

XEROX®, TELECOPIER® and the identifying numbers herein are trademarks of XEROX CORPORATION.

SOFTWARE

We sell all well known brands. Order correctly — OPENED SOFTWARE IS NOT RETURNABLE!

- NEWSTAR NEWWORD 2 \$169
- NEWSTAR NEWWORD 3 \$269
- BD SOFTWARE C COMPILER 8" \$95
- MICROSOFT * ASHTON-TATE -40%

- MICROPRO * MULTIMATE * PFS -45%
- BORLAND TURBO PASCAL, ETC. -45%
- MOST OTHER PC PROGRAMS -45%
- VENTURA DESKTOP PUBLISHING CALL

NETWORKING

CALL FOR DEALER PRICING, INTEGRATION, AND/OR CONSULTATION.

- COMPUPRO NET 100 \$369



- PC-SLAVE/16 1MB, 8MHZ, V20 \$655
- RTNX MSDOS 2.1 \$109
- ATNX MSDOS 3.1 \$189
- NTNX MSDOS 3.1 \$295



- EARTHNET-P.C. ARCNET HALF CARD \$259

NOVELL

ADVANCED NETWORK STARTER KITS, SERVERS, AND SOFTWARE CALL

CABINETS

- MPS 10005 6 SLOTS 3HH DRVS \$395
- MPS 5401 HARD DISK FH OR 2HH \$195
- JMR 1H5 PC-STYLE FH OR 2 HH
HARD DISK DRIVE \$185
- INTEGRAND 2915 DUAL HORIZ. FH
HARD DISK DRIVE \$199
- INTEGRAND 800 AD/15 15 SLOTS
2 HARD DISK (5") \$440
- INTEGRAND LASER 3310 4 SLOTS,
2 HARD DISK (5") \$387
- PARA DYNAMICS 3020D 20 SLOT
DESK TOP \$595
- PARA DYNAMICS 2300-G35" HD
& 2x8 FLOPPY \$339
- PARA DYNAMICS 5820-S 58 AMPS,
20 SLOTS, MANY DRIVES \$1,795

HARD DISK SYSTEMS

- 40MB Q540 W/DISK 3, CABINET,
CABLES, ETC. FOR COMPUPRO \$1,575
- 80MB MC-1325 FOR COMPUPRO \$1,795
- COMPUPRO DRIVE ENCL., DISK 3,
QUANTUM Q540, 5" & 8" FLPYS \$2,395

DRIVES

- SEAGATE ST-225 20 MB, 65mSEC \$319
- QUANTUM Q540 40MB, 45mSEC \$995
- QUANTUM Q540 REFURB 1 YR \$839
- SEAGATE 4051 50MB, 40mSEC \$789
- MICROPOLIS 1325 85MB, 28mSEC \$1,295
- MAXTOR XT-1140 140MB, 30mSEC \$3,195
- ALLOY 17.7MB TAPE SUBSYS \$1,595
- IRWIN 10MB TAPE (CDOS4, 1/CPRO) \$369



All merchandise new. Advertised prices are cash prepaid only. AM. Express - add 5%. MC, Visa & P.O.'s from qualified firms - add 3%. Wires, COD's (\$5 min. fee) with Cashiers Check/MO & APO's accepted. Shipping: minimum \$4 first 3 lbs. Tax: AZ RES ONLY add 6 1/2% sales tax. All returns subject to 20% restocking fee or credit towards future purchases. All prices subject to change without notice.

When Our Ad Says "Call", We Will Not Be Undersold!

S-100™

S-100 BUS PRODUCTS

CompuPro

We will integrate a system of some or all CompuPro components
"HAVE SPECS WILL QUOTE!"

SYSTEM 816/C-2	\$7,279
CPU286 8MHZ 3 CYCLE	\$509
CPU286 8MHZ, 2 CYCLE	\$655
CPUZ 8MHZ Z80	\$199
CPU 8085/88 6/10MHZ	\$255
CPU 8086 10MHZ	\$288
SPUZZ 256K, 8MHZ, UP TO 4 USERS	\$279
SP186 10MHZ, 512K RAM SLAVE	\$475
Mdrive/H 512K	\$363
Mdrive/H 2MB	\$800
RAM 23 64K..... \$180 RAM 23 128K	\$255
RAM 22 256K STATIC	\$435
RAM 24 1MB STATIC ON "ONE BD"	\$2,195
RAM 16 64K STATIC RUNS W/CPU286	\$99
INTERFACER 3 8 SERIAL	\$435
INTERFACER 4 3 SERIAL/1 PARAL	\$329
DISK 1A 5" & 8" FLPY CTRL	\$435
DISK 3 ST-506 H.D. CTRL	\$508
ENCLOSURE/2-DESK 21 SLOT	\$729
ENCLOSURE/2-RACK 21 SLOT	\$800
DRIVE ENCLOSURE-DESK	\$508
DRIVE ENCLOSURE-RACK	\$655
PC VIDEO BOARD	\$362
NET 100	\$369
HYBRID HUB..... \$255 PASSIVE HUB	\$47
SYSTEM SUPPORT 1	\$292
CONCURRENT DOS 4.1D	\$4.95
CONCURRENT DOS 4.1E	\$635
CPM/816..... \$255 CPM/80	\$182
COMPUPRO DISK DRIVE BRACKETS	\$25

MACROTECH INTERNATIONAL CORP.



MI-286 80286, 8MHZ, 2-CYCL & Z80H, A "DROP-IN" REPLACEMENT FOR CPU 8085/88	\$666
MSR-II 1MB DRAM	\$808
MSR-512 512K DRAM	\$555
ST-II 1MB STATIC RAM ON "1 BD"	\$1,999
ADIT-8 INTELLIGENT 8 PORT SERIAL	\$777
AMADIT-8 FOR ALPHA MICRO SYS.	\$959

LDP

Let us integrate a system per your specs! Ask about our stylish 6 slot 1.5 drive cabinet.

THUNDER + 8MHZ 80186, 512K RAM, 2S, P, FLPY CTRL, CDOS 4.1 - LITERALLY A SYSTEM ON A BOARD	\$895
THUNDER + 10MHZ 80186, 1MB RAM, ETC. AS ABOVE	\$1,095
LIGHTNING 286 8MHZ CPU	\$819
LIGHTNING 286 10MHZ 80286 CPU	\$897
CONTROL-IT-LL H.D. & FLPY CTRL	\$485
HAZITALL 2 S, 2 P, CLOCK	\$245
MEGARAM 512K DRAM	\$445
MEGARAM 1MB DRAM	\$595
MEGA-S-RAM 16K-1MB STATIC RAM	\$298
OCTAPORT 8 PORT SERIAL	\$298
NV-DISK SOLID STATE 2MB DISK	\$745
CDOS 4.1 MULTI-USER	\$350
COLOR MAGIC S-100 BUS PC COLOR GRAPHICS EMULATOR FOR BUS	\$445

INDUSTRIAL COMPUTER DESIGNS

CCT-100 CLOCK TIMER	
CALENDAR BOARD	\$249
CCA-100 CAL/CLK ALARM	\$175
D/A 64-100	\$298
A/D 64-100	\$225

Inter-Continental Micro

CPZ48004 MASTER: 4MHZ Z80, 64K 2S, 2P, FLPY, 8&16 BIT MEMORY OR I/O MAPPED APPLICATIONS	\$495
CPS-B8A SLAVE: 8MHZ Z80, 128K, P, 2S	\$469
CPZ-186 MASTER: 8MHZ, 1MB RAM, 2S, WILL RUN 8 AND/OR 16 BIT APPLICATIONS AND SLAVES	\$1,196
CPS-16 SLAVE: 8MHZ 8086, 256K RAM, 2S, P	\$549
CPS-186 SLAVE 10MHZ 1MB RAM	\$1,049
CPS-216 DUAL SLAVE: 8MHZ 8086 & 512K	\$1,269
OMTI-5300HD & TAPE CTRL CONNECTS WITH SCSI PORTION MASTER, MOUNTS ON DRIVE	\$519
LANS-100 ARCNET CONTROLLER	\$349
LAN-PC ARCNET CONTROLLER (WORKS WITH NOVELL)	\$349
TURBODOS MUTD-1 FOR ANY COMBO, 8 & 16 BIT	\$875
NOV-86NOVELL NETWORK 86 SW	\$1,119

VME BOARDS

DUAL

VMPU-32 PROCESSOR 68020 16 MHZ, 1MB RAM, CLOCK, BATT, PAGED MEMORY MGMT	\$4,200
VSMD CACHING SMD 3 DRV. CTRL	\$2,392
VMEM 2MB, 85mSEC ACCESS	\$3,196
V9TRK 9-TRK TAPE CTRL, 6250 BPI	\$999
VIOP VO CPU, 10MHZ 6800, 512K	\$1,197
IO-S8 8 SERIAL UP TO 84.4K BAUD	\$397
SCM SYS CONTROL MODULE, 16 MHZ	\$397
VMBD-12 12 SLOT BACKPLANE	\$669

APPLE PRODUCTS

brother

M-1109AP 100 CPS W/ALL CABLES	\$239
----------------------------------	-------

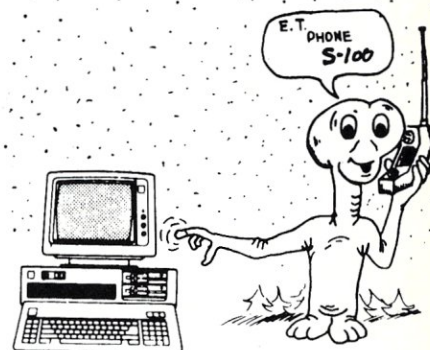


300C MODEM FOR IIC - FITS ON BACK OF COMPUTER AND YOU DON'T LOSE YOUR SERIAL PORT	\$65
---	------

PRACTICAL PERIPHERALS

PRO CLOCK	\$99
PRINTER FACE	\$49
GRAPHIC CARD	\$69
SERIAL	\$99

EVERYONE'S PHONING HOME
ABOUT THIS ADVANCED "AT"
COMPATIBLE SYSTEM



"ET-286" BASIC SYSTEM

12MHZ ONE WAIT STATE OR 10MHZ NO
WAIT STATE SPECIAL ORDER - ALLOW 5
WEEKS, 10 & 6 MHZ 80286 CPU ONE WAIT
STATE, 1 - 4MB RAM ON MOTHER BOARD
ALSO ON MOTHERBOARD:

- 6 AT COMPATIBLE SLOTS
- 2 XT COMPATIBLE SLOTS
- 3 RS232 OR RS422 SWITCHABLE SERIAL PORTS
- 2 CENTRONICS PARALLEL PORTS
- CLOCK/CALENDAR W/BATTERY BACK-UP
- 100% IBM-AT COMPATIBLE BIOS
- CABINET, 200 WATT P/S, & KEYBD..... \$1,395

OPTIONS TO COMPLETE YOUR
"STATE OF THE ART" SYSTEM:

FLOPPY-HARD DISK 16 BIT CTRL	\$200
1.2, MB FLOPPY DRIVE	\$139
26MB 65mSEC HARD DISK DRIVE	\$309
51MB 35mSEC HARD DISK DRIVE	\$809
85MB 28mSEC HARD DISK DRIVE	\$1,295
140MB 30mSEC HARD DISK DRIVE	\$3,319
20-60MB (2MB/MIN.) TAPE BACK-UP	\$707

TATUNG TEGA-22 DUAL FREQ, 256K RAM	\$259
TATUNG CM-1380F 13" ENHANCED DUAL FREQ. RGB MONITOR W/G&A SW	\$450
TATUNG PRIVATE LABELS IBM & MICROAGE MONITORS	CALL
TATUNG 1422A DUAL FREQ. TILT 'N' SWIVEL AMB/GRN 14" 640x400	\$139
MACROTECH MSR-AT 12MHZ, 0-3MB RAM, NO WAIT STATES	\$185-\$625
NEC 1401 MULTISYNC	CALL
PC-DOS 3.2	\$119
XENIX OPERATING SYSTEM	\$395
LINK 125 GRN 14" TERMINAL, 1 YR WARR, WY-50 "LOOK-ALIKE"	\$379
ALLOY PC-SLAVE/16 8MHZ 8088 1MB RAM PER USER	\$655
ALLOY AT/IX NETWORKING SOFTWARE, UP TO 31 USERS	\$189
AUTO CAD ADE-0	\$259

S-100 DIV./696 CORP.
14455 NORTH 79th ST.
SCOTTSDALE, AZ 85260

800-528-3138 Orders Only
602-991-7870 Customer Support
TELEX 9103806778 SONE HUND



minor way with initial handshaking and the setting of the serial-port speed.

LOOKING TO THE FUTURE

The V.32s will be on the shelves shortly, if manufacturers' promises can be taken seriously. Due to the complexity of V.32 recommendations, the designs will be more expensive than their V.29 cousins. The rigid nature of V.32 recommendations regarding EDC, though, should make them compatible with each other. Compatibility between V.29 modems is not possible now, as each manufacturer is striving to get his EDC protocol more widely accepted. Add to that fray the proprietary Packetized Ensemble Protocol of the TrailBlazer, and you have a slugfest of major proportions over whose high-speed method will reign supreme. Part two of this review will contain evaluations of V.32 9600 baud modems. §

PRODUCT INFORMATION

UPTA 96

V.29 9600, 7200, 4800-bps

Price: \$895

Snap-in 103, 212A and V.24-bis modem components.

Recommended Software: MEX-PC, Crosstalk-Fast, BLAST, Respond, Relay Gold, Carbon-Copy, Q-MODEM, Professional-YAM. Call manufacturer for mini-to-mini, PC-to-mainframe, and other software needs.

Electronic Vaults, Inc.
12347-E Sunrise Valley Dr.
Reston, VA 22091
(703) 620-3900

Racal-Vadic 9600VP

V.29 9600, 7200, and 4800 with 103 and 212A asynchronous modes

Price: \$1495

Recommended software: George (from Racal-Vadic), call manufacturer about other PC-to-PC software or PC to mainframe and mini software packages.

Racal-Vadic
1525 McCarthy Blvd.
Milpitas, CA 95035
(800) CALL-VADIC

Telebit TrailBlazer RA12E-T1

9600+, V.22, V.22-bis, 103 and 212A emulation

Price: \$2300

Recommended software: MEX-PC, Crosstalk-Fast, BLAST, Carbon Copy, Relay Gold, Respond.

Telebit Corp.
10440 Bubb Rd.
Cupertino, CA 95014
(408) 996-8000

Modem	bps transmission including turnaround
Racal-Vadic 9600VP	5.609k
EVI UPTA 96	6.167k
Telebit TrailBlazer	7.971k

Table 1. Modem throughput with 196k ASCII file

Modem	Time in milliseconds
Racal-Vadic 9600VP	NA *
EVI UPTA 96	253 ms
Telebit TrailBlazer	136 ms

Table 2. Turnaround times, transmission mode

Modem	Time in seconds
Racal-Vadic 9600VP	2.5
EVI UPTA 96	3.5
Telebit TrailBlazer	1.5

Table 3. Turnaround, interactive mode

Modem	Retransmission
Racal-Vadic 9600VP	4.0
EVI UPTA 96	3.5
Telebit TrailBlazer	2.5

Table 4. Retries, retransmissions per 196k transfer (average)

Note: turnaround times in interactive mode were timed at the keyboard.

** The transmission turnarounds are manufacturer supplied.*

Retransmissions were averaged from Crosstalk-Fast transfers after software and modem were fully prepared and operational.)

ONE COMPUTER & TWO PRINTERS?

Stop Plugging & Unplugging! use your choice of TWO printers from ONE parallel cable with the simple flick of a switch.



\$65⁹⁵
Plus \$3.25 Shipping

30 Day Money Back Guarantee
Model #2100

One Parallel Plug In... Two Parallel Plugs Out!
Connect any Standard Centronics Parallel Cable directly to the switch. Two 3 1/2' cables (included) attach to each printer.

Simply Call 1-800-TO-ASK-US! 8am-5 pm PST
800-862-7587
(CA 415/567-4067)

To Order: Write or call Tipz Direct, Inc. Visa/MC or Money Orders please. Shipping within 48 hrs. via UPS.

Offering a FULL LINE of Switches, Cables, & Accessories

Tipz

**P.O. Box 690,
San Francisco, CA 94101-0690 MSJ-210**

S-100 BARE BOARDS

8086/8087 CPU - plus
2764 or 27128, 8253, 8259

8088 Auxiliary Processor
I/O mapped, 4K EPROM,
4K RAM, prototype area

\$45.00 Each

Call or write for brochure.

Terms: Check or money order only. CA residents add sales tax. Prices include UPS shipping.

Applied Innovations
3000 Scott Blvd. Suite 106
Santa Clara, CA 95054
(408) 748-1875

Turbo Tech Report Speaks Your Language.

Turbo Pascal
Articles and
Reviews

News and
commentary

A disk filled
with Turbo Pascal
code!



The newsletter/disk publication for Turbo Pascal® users

Are you a devoted Turbo Pascal programmer, tired of reading about other languages? Are you looking for powerful utilities written in Turbo Pascal that you can use to develop software or incorporate into your programs? Are you interested in improving and expanding your Turbo Pascal programming skills?

Then you deserve a subscription to *Turbo Tech Report*, the bimonthly newsletter/disk publication from the publishers of *Dr. Dobb's Journal* and *Micro/Systems Journal*. Each issue delivers more than 250K of Turbo Pascal source code programs on disk, and 20+ pages of articles, Turbo Pascal software and book reviews, and analysis and commentary. It's the only publication delivering such focused technical articles with code on disk—and it doesn't waste your time with information about other programming languages. Each valuable issue contains:

- **Articles** on topics like speedy 3D graphics, mathematical expression parsers, creating global gotos, memory resident and AI applications and more—all written by Turbo experts.
- **Reviews** of the latest Turbo Pascal software programs from companies like Borland

International, Blaise Computing, Media Cybernetics, Nostradamus, TurboPower Software, and more!

- **News and commentary** detailing the latest products and developments in the Turbo Pascal programming community.

- **A disk filled with Turbo Pascal code!** You'll get the Turbo-Pascal utilities and routines discussed in the newsletter's articles, as well as applications developed by Turbo users from around the world. You'll receive programs that make labels, generate menus, provide faster screen access, transfer files between CP/M and MS-DOS computers, and more!

If you're an expert Turbo Pascal programmer or a novice interested in expanding your Turbo skills, you need a publication that speaks your language: *Turbo Tech Report*. Subscribe today at the special price of just \$99—that's 33% off the regular price of \$150. To order by credit card, call toll-free 1-800-528-6050 ext. 4001 and ask for item 300. Or mail the attached coupon with your payment to *Turbo Tech Report*, 501 Galveston Drive, Redwood City, CA 94063.

Turbo Pascal is a trademark of Borland International Inc.

Experimenting with Protected Mode on the AT

by Marshall Brain

For most users, the IBM AT and its compatibles, are nothing more than accelerated PCs. The AT has the same 640K memory limit as the PC and runs DOS programs in exactly the same way the PC does, but it does its work three or four times faster than the older system.

The AT's CPU chip, the 80286, can actually do a great deal more than simply emulate the 8088 used in the PC, because the 80286 has two separate modes of operation. It has a Real mode, in which the 80286 behaves just like an 8088 or 8086. This mode allows the AT to act like a fast PC. The 80286 also has a Protected mode, in which the chip behaves like a mainframe CPU. DOS currently runs only in the Real mode, so the extra capabilities of the 80286 are never seen by the normal user.

This article discusses the differences between the Real mode and the Protected mode. It explains the requirements of Protected mode operation, and uses a simple Turbo Pascal program to demonstrate taking the AT into Protected mode and back out to Real mode. This article assumes that you have some knowledge of assembly-language programming techniques on the 8088. You may also wish to consult the books and articles listed in the bibliography for additional information on the AT BIOS calls used and for more information about the Protected mode.

DIFFERENCES BETWEEN THE PC AND THE AT

The AT system board has three features that distinguish it from the PC system board. First, the AT system board has not only the main CPU chip, but also a second complete microprocessor called the 8042. This microprocessor is used primarily as a keyboard interface, but it has other capabilities that are important to use of the Protected mode on the AT. Second, the AT has a CMOS clock/RAM chip built onto the system board. This chip provides a software equivalent for the DIP switches found on the PC system board, but it also has memory locations that are needed when you work in Protected mode. Finally, the AT system board contains an 80286 processor instead of the PC's 8088.

The presence of the 80286 gives the AT its speed boost in three ways: the 80286 runs at a higher clock rate; the 80286 has a 16-bit data bus, compared to the 8088's 8-bit bus; and the 80286 has internal pipelining that speeds up instruction execution. When running in Real mode, the AT uses all of these advantages to provide noticeably better performance than the PC.

The very fact of the 80286's existence also gives the AT Protected mode capabilities. Once the 80286 is switched to Protected mode, the programmer is able to access the following features: (1) a 16Mb address space, as opposed to the Real mode's 1Mb space; (2) virtual memory support; (3) hardware multitasking support for context switching and task separation; and (4) an impermeable protection mechanism that isolates different tasks from one another. The AT has all of the essential features of a mainframe or superminicomputer built in, due to the presence of the 80286.

SEGMENT REGISTERS

The main difference between the Real mode and the Protected mode on the AT lies in their use of the segment registers. A new use for segment registers is the key to the capabilities of the Protected mode. On the 8088, and on a 80286 running in Real mode, the segment registers are used simply to address a memory space greater than the 64K that would be allowed by the 16-bit width of the registers and instruction pointer. Whenever a memory access takes place, it is always done relative to one of four 16-bit segment registers; the value of the appropriate segment register is multiplied by 16 and added to the 16-bit offset value. This allows the CPU to reference any location in a 1Mb address space.

In the Protected mode, you are able to access up to 16Mb of physical memory (and up to several billion bytes of virtual memory), because the segment registers are used in a new way. Instead of being used as a physical memory pointer, the segment registers are used as 13-bit pointers that point into tables of "descriptors." Descriptors are 8-byte (64-bit)-long values that are used to describe a segment. Such a

description includes the segment's location in physical memory, the segment's length, and the segment's access rights.

Many different types of descriptors are possible, because many different types of segments are defined in Protected mode. There are code segments, data segments, and task-state segments. In general, though, there are two main types of segments (and therefore segment descriptors): memory descriptors and control descriptors. This article concentrates on memory descriptors. Control descriptors are used to implement protection and multitasking features of the 80286, and will be largely ignored in this article.

To access descriptors, you use 16-bit segment register values to point into descriptor tables. These tables can be thought of as arrays of descriptors, with each descriptor being an 8-byte-long element of the array. There are three different types of tables defined. The Global Descriptor Table (GDT) is potentially accessible by any task in the system (although access may be denied). A Local Descriptor Table (LDT) can be defined for each task and is private to the task. An Interrupt Descriptor Table (IDT) is also defined and replaces the 256 vector pointers found in the first 1000 bytes on the 8088. These tables are stored in memory and are actually another type of segment themselves.

This article concentrates on memory descriptors in the GDT. Just remember that, in Protected mode, any segment register can be used to point into a descriptor table so that a segment descriptor can be retrieved. The information contained in the segment descriptor is used to access memory.

HOW DESCRIPTOR TABLES WORK

Think about the following statements when they are made in Real mode or on the 8088:

```
MOV AX, 40H
MOV DS, AX
```

They seem almost insignificant. A 16-bit value is moved to AX, and then this 16-bit value is moved to the data segment register DS. These statements allow you to

access data beginning at memory location 400H.

In Protected mode, this instruction pair works differently. AX gets loaded as usual, but the loading of DS sets in motion the following chain of events:

- 1) The upper 13-bits of the segment register are used as an index into the appropriate descriptor table.
- 2) The 8-byte descriptor is retrieved and placed into a descriptor cache for the appropriate segment register. On the 80286, all four segment registers have the same 16-bit segment register found on the 8088, but there is also a hidden 8-byte register that holds the current segment descriptor for that segment register.
- 3) The cached information is then used anytime a reference is made to the given segment.

This chain of events, shown in Figure 1, (turn to page 50 for article Figures and Code) occurs whenever any segment register is loaded with a new 16-bit value. Many instructions load segment registers, including MOV (as shown), POP, LDS, far CALLs and JMPs.

Figure 2 describes the format of information contained in segment registers and descriptors in Protected mode. As can be seen, it is the descriptor that actually contains the physical memory address for the segment. The descriptor also contains the length of the segment and access rights. When you try to access information using a given segment, all of this cached information comes into play. For example, assume that the following instruction is executed:

```
MOV AX, DS [69H]
```

The following events take place:

- 1) The 80286 makes sure that 69H is within the length of the segment. If the segment is only ten bytes long, a processor fault will occur.
- 2) The 80286 adds the 24-bit physical memory address found in the cache to the offset value 69H.
- 3) The memory location addressed by this sum is loaded into AX.

All of this takes place in hardware, using information that has already been loaded, so there is no time penalty for the extra work done in Protected mode. The only speed penalty occurs when a segment register is loaded. At that point, the 8-byte descriptor value must be retrieved and moved from the appropriate table in memory into the cache register.

Descriptors give the Protected mode its flexibility. For example, assume that DS is loaded with 40H at one point. At this time, the descriptor may indicate that this segment is located at memory location 1000H. At a later time, however, 40H

Bus Specific Networks

NEW SOLUTION

Complete network solutions for the price of a single card.

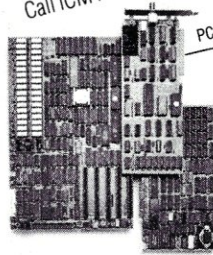
Entry level costs to networking are reduced by ICM's unique integration of the most common components of both file servers and network workstations.

ICM's WS-286 contains all the common local area network components on one board—including ARCnet, SCSI, 1 MByte RAM, and 5 expansion slots. This greatly reduces system complexity, repair costs, and stock of multiple costly components.

The on-board SCSI controller provides easy interface to ICM's single-board disk and tape controller—so the WS-286 may be used as both a file server and inexpensive diskless workstation, supporting Novell and TurboDOS. It fits into standard IBM-XT enclosures without software or hardware modification. The 80286-based workstation operates at 10 MHz with no wait states when accessing on-board peripherals and RAM, while the expansion slots operate at 4.77 MHz to allow full XT expansion card compatibility.

Call ICM for more information and Reseller pricing.

ONLY \$1595.



PC, ARCNET, AND S-100 APPLICATIONS.



ICM

TOMORROW'S BREAKTHROUGHS TODAY

InterContinental Microsystems

4015 Leaverton Ct., Anaheim, CA 92807

Phone: (714) 630-0964

Telex: 821375 SUPPORT UD

Easylink 62562040 BBS (714) 632-8750

might be moved into DS again, and this time the retrieved descriptor might indicate a physical address of 74500H. The program has no idea that the segment has been moved, because it used 40H to reference the data in both cases. It is also important to note that each segment is treated as a specific, fixed-length entity, which can be moved, or altered. This is somewhat different from the way segments are treated on the 8088.

GETTING INTO PROTECTED MODE

To take advantage of the capabilities of Protected mode, the 80286 has to be running in Protected mode. Fortunately, this is fairly easy, because the AT BIOS contains a call that will switch the 80286 into Protected mode. Before this call can be used, however, some preparation is required.

For the 80286 to work in Protected mode, it must have a GDT. The GDT is needed because the first thing that the 80286 does when it gets to Protected mode is to try to get the descriptor for the CS register. Without this it cannot execute any instructions. Also, the first instructions executed once in Protected mode are instructions that load the SS, DS, and ES registers, so that the program can access data and make subroutine calls. These four descriptors will be found in the GDT, so the GDT must be set up in Real mode before the switch to Protected mode is made. Once these four descriptors are cached, the

80286 can work on its own in Protected mode by creating or modifying descriptor tables to give it access to any memory location. The Protected mode needs the Real mode, though, to get things started.

To switch to Protected mode, an AT BIOS routine can be used. This routine is accessed using interrupt 15H, function 89H. As shown in the AT *Technical Reference Manual*, this routine requires that a GDT be set up before it is called. It also requires that an IDT be created. Normally, the IDT is set to handle at least the 14 processor faults possible on the 80286. These processor faults are shown in Figure 3. A problem arises with these faults, however, because the original PC architecture defined hardware and software interrupts at many of these locations. The BIOS routine therefore allows the programmer to remap the hardware-interrupt controllers so that their vectors don't occur within the first 14 interrupts.

This is all that is required to go to Protected mode. It would be nice, however, to be able to get back to Real mode eventually through software, rather than having to turn the machine off and back on to get back to DOS. Unfortunately, the 80286 has no provision for getting back to Real mode from Protected mode. Apparently the designers assumed that the Real mode would be used once to initialize a GDT and IDT, and that there would be no reason to return.

To get around this problem, a sort of Rube Goldberg machine was built into the AT that allows it to return to Real mode. It is used by the VDISK utility, for example, to allow it to access expanded memory. VDISK switches to Protected mode so that it can access the full 16M address space, stores or recalls information above the 1M memory limit of Real mode, and then returns to Real mode. To take advantage of this mechanism, you must set it up before entering Protected mode.

The mechanism works like this: To get back to Real mode from Protected mode, the 80286 sends a message to the 8042 microprocessor on the AT system board. This message says, "Reset the machine." The 8042 does this, sending a hardware reset across the AT system board. This is essentially the same signal as the signal created by powering off and back on. The 80286 resets, and the BIOS starts executing its normal power-on sequence in Real mode. (Whenever the 80286 is reset, it comes up in Real mode.) Just before it starts its power-on self-test routines, however, it looks into the CMOS RAM. If it finds one of several different values in location 0FH of that RAM (the CMOS RAM has its own address space), it knows that the reset signal was not a "power on" signal, but instead a "returning from Protected mode" signal. The value 05H in location 0FH tells the BIOS to do a far jump directly to whatever location it finds in memory at 467H and 469H.

To use this mechanism, the programmer is required to set the CMOS RAM so that an 05H appears in its location 0FH and must also set up 467H and 469H (locations in the BIOS reserved memory area) to contain the address of the code that should begin executing when the computer returns to Real mode. If this is done correctly, the 80286 can go to Protected mode and then back to Real mode without any problems.

AN EXAMPLE PROGRAM

Keeping all of this in mind, examine the example program. The main routine spells out the steps necessary to get into Protected mode.

The first statement sets up the GDT. The BIOS call being used requires eight descriptors, as described in Figure 4. This routine simply sets the physical memory address, segment size, and appropriate access-rights byte into each descriptor. For most, the value 92H in the access-rights byte indicates that the segment is present, has a privilege level of 0 (used for the protection features—0 is the highest level), is not executable, and is both readable and writeable. For the code segment, the access-rights byte indicates that the segment is present, has a privilege level of 0, is executable (the 80286 will not allow data segments to be executed, or code segments to be written into), and is readable (you can make code segments nonreadable to

keep other routines from stealing the code). For more information, refer to the Intel manual in the bibliography.

The next step is to set up the IDT for the 14 processor fault vectors. The descriptors here are used as "call gates" (a type of control descriptor). For example, the statement "SET_IDT_DESC(0, . . ." sets up the IDT descriptor 0 (also vector 0 and fault 0), so that an interrupt 0 goes to the routine FAULT in the code segment. The "+7" makes sure that the code put in place by Turbo Pascal to allow local variable access is ignored. Don't try to access local variables from the fault routine (once FAULT calls something else, the called routine can access local variables). All fault routines point to the same fault handler that beeps the speaker twice.

The next step saves the segment registers needed to run in Real mode (SS, DS, and ES). When you return to Real mode, these values are reloaded into the appropriate segment registers. Once this is done, Turbo never knows that it was out of Real mode. These values are stored in typed constants, which reside in the code segment. The values can be accessed, since DS and SS are unknown.

When the BIOS call for Protected mode executes, it remaps the hardware interrupts to whatever location is requested. It also masks off all hardware interrupts so that they can't get in to bother the 80286 while it is in Protected mode. When returning to Real mode, the interrupts are remapped to their normal positions, but they remain masked off. To allow them to work again, you must set the masks in the interrupt-controller chips back to their original states. The SAVE_8259_REGS routine saves the contents of the mask registers so that they can be correctly restored upon return to Real mode.

The next routine sets 05H into location 0FH of the CMOS RAM. It sends address 0FH to port 70H (the CMOS RAM address port, which gives this RAM its own address space), and then writes the value 05H into port 71H. This causes the CMOS RAM chip to move the 05H into location 0FH.

Next, locations 467H and 469H are set to the address of the exit-point routine that will be used when the return to Real mode occurs. Again, the "+7" is used to circumvent the local-variable-address setup code. The exit-point routine cannot access local variables.

Finally, the 80286 can be taken to Protected mode. A small piece of inline code is used to set the registers to their appropriate values and call the interrupt. You cannot use the INTR procedure here because the BIOS call destroys BP, and INTR tries to use it.

Once you enter Protected mode, all of its capabilities are opened to you—the 16M address space is available, as well as virtual memory support, protection, etc. So why then, after all of this work, does the program simply beep once and drop

back to Real mode? This occurs for several reasons. First, the hardware interrupts are off, so the keyboard does not work. The screen cannot be accessed because the video BIOS call is not entered in the IDT, and a descriptor for the video memory space does not exist. The disk drives don't work for a variety of reasons. Essentially, the AT becomes a naked machine, and the speaker is the only accessible output device. While in Protected mode, you can do most non-I/O and things unrelated to the heap that you normally do in Turbo Pascal, although the results of any calculations must be sent out in beeps.

To get out of Protected mode, the 8042 is told to create a hardware reset by sending 0FEH to port 64H. The HLT command halts the processor until the reset can take effect. The Rube Goldberg machine then takes over and eventually begins executing the program again at the exit-point routine. This routine restores all segment registers, replaces the mask registers, and returns. If you make a drawing of the stack contents through all of this, you will find that what is on the stack at the time of the exit point's return is the address of the caller of GOTO_PROTECTED_MODE. The exit-point routine returns to the main routine, which returns to DOS.

There are several things you should bear in mind while Turbo Pascal is running in Protected mode. First, no screen, keyboard, or disk I/O is possible for the reasons mentioned above. In addition, no BIOS or DOS function calls are available. You cannot access the heap, because the heap manager loads many segment values that have no entries in the GDT. The compiler directives shown must be off, or the program will not work. Typed constants (whose values are stored in the code segment) cannot be written to while in Protected mode, because the 80286 guards against any modification of executable code segments. Finally, local variables or parameters should not be used in the routines mentioned above, because the BP register is either lost or destroyed. Any routines called by these routines, however, can have local variables and parameters.

EXPERIMENTING

Once you have the Protected mode program working—that is, it runs, beeps once, and returns to the Turbo Pascal environment—you can try several easy modifications and experiments. One of the first might be to create processor faults while in Protected mode.

You can create processor faults in several ways, but the easiest involves the creation of a nonpresent segment. For example, the access-rights byte for the Data descriptor in the GDT (descriptor number 3 in Figure 4 and in the SETUP_GDT procedure) has an access-rights byte of 92H, or 10010010. Figure 2 shows that bit 7 is used as a "present" bit; you can mark a

segment "not present" by changing bit 7 to zero. This is normally done in a virtual memory system. When a segment is swapped out to disk, the descriptor is marked "not present." The next time that segment is referenced, a not present fault (fault 11) occurs, which allows the operating system to pull that segment back into physical memory, set the descriptor to indicate "present," and then restart the faulting instruction.

To try this, simply change the access-rights byte of descriptor number 3 in the GDT from 92H to 12H. Now, when the program executes, it will try to load the descriptor for DS, and a not present fault will occur. You will hear two beeps instead of one to indicate that the fault took place.

You can also do this with the extra and stack segments. The stack segment will use fault 12 instead of 11, although you won't hear any difference. Do not try this on the code segment, however. If you change the code-segment descriptor to 1AH, the program will crash. This occurs because the 80286 never gets a code-segment descriptor loaded and can therefore do nothing.

When experimenting with faults, or when debugging, it is nice to know exactly which fault is occurring. You may wish to create 14 different fault routines—one for each possible fault. Fault 0 might beep once, fault 1 twice, etc., so that you know exactly which fault occurred. Create the 14 routines, and then change the descriptors in SETUP_IDT so that each one points to the appropriate fault routine instead of to the generic fault routine. Then recompile the program. (A note on compiling: All tests for this article were performed using an IBM AT and Turbo Pascal Version 3.00B. It is unknown how the program will behave on AT compatibles or on older versions of Turbo Pascal.)

It would be nice to have some output device other than the speaker. You may have noticed that an extra descriptor is included in SETUP_GDT. The TEST_SCREEN routine can use this descriptor to access the screen as an output device from Protected mode. To make use of this routine, first remove the comment braces from around the TEST_SCREEN call. Then modify the physical address for descriptor 8 in the GDT to match your system. If your system uses an EGA or CGA board, the call to ADDR24 should use the value \$B800 for the physical address. If you use a monochrome adapter, the address should be \$B000.

Once you have made these changes, compile the program and run it. It should print "HELLO FROM PROTECTED MODE" onto your screen several times. This routine takes full advantage of the redirectability of segments to access the screen. An absolute array SCREEN is declared at the beginning of the program. It has been declared to point to \$40:0. If you decode 40H in light of the Protected mode

Bus Specific Networks

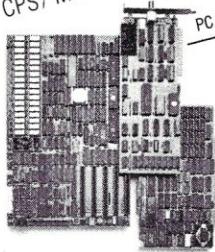
NEW FUTURE

Finally PC-DOS capabilities on S-100 systems—a major breakthrough the noncompatibility barrier.

For the first time, you can run monochrome PC-DOS and MS-DOS—including Lotus 123, Wordstar, and Sidekick—on your existing S-100 system. All with the brand new CPS-16F, a powerful single board network processor, only from ICM. It fits existing installations without a LAN hookup and has low entry requirements for TurboDOS Systems. The CPS/MS-1000 package (with 1000 opportunities) consists of the CPS-16F processor; M/STER—ICM's monochrome graphics PC-emulating terminal; PC-DOS; TurboDOS/PC; and ICM's exclusive MS-1000 software.

ICM brings PC-DOS to the S-100 industry. Call ICM for Reseller pricing on the CPS/MS-1000 package and info about our upcoming products.

PC, ARCNET, AND S-100 APPLICATIONS.



ICM

TOMORROW'S BREAKTHROUGHS TODAY

InterContinental Microsystems
 4015 Leaverton Ct., Anaheim, CA 92807
 Phone: (714) 630-0964
 Telex: 821375 SUPPORT UD
 Easylink 62562040 BBS (714) 632-8750

use of segment values, you will find that this corresponds to descriptor 8. Descriptor 8 has been directed (in the GDT) to point at the physical memory address that contains the screen memory buffer. The routine DISPLAY simply writes information into the appropriate address in the screen buffer by accessing the segment value \$40. The absolute address feature of Turbo is convenient in this case. Whenever the screen array is accessed, the segment value \$40 is loaded into a segment register. This retrieves the descriptor needed to access the screen.

You can use the DISPLAY routine to create a CLRSCR procedure for Protected mode. Simply display 25 lines of spaces on the screen.

CONCLUSION

This article has shown how to get Turbo Pascal to run, with limitations, in Protected mode, but it has barely scratched the surface of the Protected mode's capabilities. You will find that you can use Turbo Pascal to create a comfortable environment for further experimentation with Protected mode features. §

REFERENCES

Claff, William J., "Moving From the 8088 to the 80286; Important Differences You Need to Know to Make Your Programs Transportable," *Byte Special Issue, "Inside the IBM PCs,"* Volume 10, Number 11, 1985.

IBM Corporation, *IBM Technical Reference Personal Computer AT*, 1984.
 Intel Corporation, *iAPX 286 Programmer's Reference Manual*, 1985.
 Quendens, Guy, and Webb, Gary, "Switching Modes; Getting Into and Out of the iAPX 286's Protected Virtual Mode Helps Take Full Advantage of the AT's Power," *PC Tech Journal*, August 1985.

The code that appears in this article is available on disk and can be obtained by sending \$6 (to cover postage and cost of a disk) to the author at Box 37224, Raleigh, NC 27627.

Marshall Brain is an instructor at North Carolina State University, where he is finishing his master's degree in Computer Studies. He can be reached at Box 37224, Raleigh, NC 27627.

COMP Computing Standard MUMPS

by Michael Guttman and Robert Engle

A Microcomputer Version of the MUMPS language, Intended For Writing Database Applications

MUMPS is a little-known but well-established database application development environment, developed for minicomputers and mainframes. Its unusual moniker is an acronym for Massachusetts General Hospital Utility Multi-Programming System, a hint of its rather obscure origins. MUMPS has largely outgrown its original pedigree, however, and is now in use around the world on many different machines and for many different applications.

Recently, various PC versions of MUMPS have begun to make their appearance, including the one reviewed here. This is obviously of interest to the rather small community of programmers already using MUMPS on larger machines, but why should the rest of us get excited? Ordinary PC users may be surprised to discover that for the paltry price of Turbo Pascal (\$59.95), they can now purchase a full implementation of a powerful mainframe database package that, in many ways, runs circles around the far more expensive best-sellers of the database genre.

To find out just what MUMPS has to offer, we looked at CCSM (COMP Computing Standard MUMPS), a new version of MUMPS for the IBM PC from Guidance Software, Inc., of Kingwood, Texas, and exclusively marketed by MGlobal of Houston, Texas. Before we delve into the particulars of CCSM, we offer a general discussion of MUMPS.

MUMPS was developed in the late 1960s by a group of programmers at Massachusetts General who were frustrated by the time and overhead required

to prototype and complete database-oriented applications using then-available programming languages such as COBOL and FORTRAN. Like the great PC innovators of today, they decided to throw out the rule book and develop, largely from scratch, an applications development environment based strictly on their own requirements.

The result of their efforts must have appeared astounding in its day. MUMPS emerged as a full language in its own right, directly supporting all the usual operations for logic control and I/O. Moreover, the new language was completely interpretive (still a novelty then), so its users could say goodbye to the cumbersome array of mainframe compilers, linkers, debuggers, and dump translators, thereby speeding development time by at least an order of magnitude.

The most important and novel advantages of the new language, however, were in the areas of data manipulation and database architecture. The new language offered support for arrays of any dimension and flexible typing of variables. It also provided a fast, convenient, and powerful scheme for maintaining indexed data files, including support of variable-length records, fields, and keys, and a full-featured file- and record-locking scheme for multiuser operation.

Not surprisingly, the language was an almost-instant hit in hospital DP shops and was soon ported to new environments and picked up by a number of big clients. One of those clients was the U.S. government, which, through the VA and other agencies, runs a lot of hospitals. Attempts were made to standardize application development around MUMPS, and a canonical version of the MUMPS language was submitted and accepted as an ANSI standard—a distinction applied to very few languages.

Ironically, however, this Cinderella rise from obscurity to acceptance may actually have worked to hide MUMPS from the general public. With excellent versions already in the public domain, MUMPS failed to attract the attention of any major vendor and was thus never hawked to the commercial sector. As a result, it remains something of a cult language, as a host of proprietary application development products have garnered the bulk of commercial attention and dollars.

GETTING STARTED

Our copy of CCSM came with two disks and two manuals. One disk and manual were devoted to this version of MUMPS itself, while the other disk and manual comprised a "cookbook" of generic MUMPS routines that illustrate many of the features available in MUMPS. A page of installation notes was also provided, with which we tried to install the system on our 20-Mb Leading Edge PC clone. It quickly became apparent that both the documentation and the setup files provided were not geared to installing CCSM on a hard disk—a strange approach, given the package's usual function of developing database applications. After a little hacking around, however, we had no real trouble getting MUMPS to work properly on our hard disk.

Once installation is complete, invoking the MUMPS.EXE file brings CCSM up in the interpretive mode, with an asterisk as a prompt. MUMPS commands have a simple, intuitive, and free-form syntax. For example, the command line

```
* SET A = "PRINT SOMETHING"
SET B = "USING MUMPS" WRITE A,B
```

will actually execute two SET (assignment) statements and one write statement, even though no special delimiters are used. For the cryptic-minded, this statement can be further simplified to:

```
* S A="PRINT SOMETHING",
B=" USING MUMPS" W A,B
```

In either case, the result would be simply:

```
PRINT SOMETHING USING MUMPS
```

MUMPS also supports macro substitution with the @ prefix, as in:

```
* S C="A,B" W @C
```

which would also achieve the same result as W A,B.

PROGRAMMING

Fooling around in the immediate mode was certainly fun, but the acid test of CCSM comes in developing real programs. Programs are created with a supplied editor and then executed using the DO command—a process oddly similar to one

in dBASE III. Unfortunately, the editor that comes with CCSM is rather slow and clunky, and its use was essentially mandatory, since all MUMPS programs (and data) are stored on disk in a special MUMPS format that is basically unreadable by other programs. There is a way to read standard ASCII files, but the documentation was not clear as to how this could be used to support the use of an external ASCII editor. Because the programs we wrote were short, we put up with the CCSM editor, but serious MUMPS programmers will need to construct some sort of alternative.

Each line of a MUMPS program consists of an optional label and a series of commands and expressions. Like most programming languages, MUMPS executes from beginning to end, altering the flow in response to GOTO or DO commands, which may refer to internal or external routines. Even though MUMPS is completely interpretive, a reference to an external routine doesn't appear to require any more overhead than to an internal one, and it executes at the same speed.

In addition to DO, GOTO, SET, and WRITE, a complete set of programming commands and operators are provided. These include the commands READ for keyboard input; OPEN, CLOSE, and USE to control the current I/O device; QUIT and HALT to exit from routines or MUMPS; IF, ELSE, and FOR for loop control, and KILL, which gets rid of unwanted variables and arrays. There are many string, numeric, and screen functions and, for multiuser operation, commands for starting and aborting jobs and locking and releasing variables.

STORING DATA

In the example given above, the variables *A*, *B*, and *C* are simple transient scalars that will disappear after MUMPS exits. However, MUMPS also allows the creation of what are called global variables, which are always stored on disk and can be retrieved at any time. Also, both local and global variables can be multidimensional arrays. As in Snobol, array subscripts can be anything at all. For example, the expression

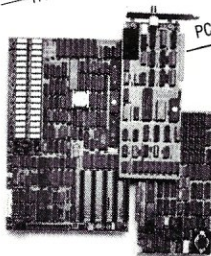
```
* S A("JONES", "RALPH") = "PROGRAMMER"
```

will actually create a variable called A("JONES", "RALPH") and give it the value "PROGRAMMER". In reality, MUMPS handily stores the data "PROGRAMMER" away in a B-Tree index under the keys "JONES" and "RALPH", which makes it possible for you to retrieve the data directly with these parameters. There are few restrictions on how this subscripting can be done. No dimensions need be declared in advance, and all assignments can be made "on the fly." Compared to conventional programming languages and databases, the power of this approach is enormous. You can concoct all

Bus Specific Networks IN HALF A SLOT

Interconnect Novell, Vianet, TurboDOS, and other ARCnet networks with our new half-size LANPC card.

Only from InterContinental Microsystems, now in a new half-slot size card, you can convert hard-working PCs to even harder-working LAN Intelligent Workstations, Diskless Workstations and File Processors. You get IBM PC bus-standard compliance, built-in complete controller for ARCnet modified token passing networks. And you can have a fiber-optic option. All with FCC approval. So you can have ARCnet capabilities in less space than ever before — in only half a PC slot. Just imagine the things you can do with the other half.



PC, ARCNET, AND S-100 APPLICATIONS.



ICM

TOMORROW'S BREAKTHROUGHS TODAY

InterContinental Microsystems
4015 Leaverton Ct., Anaheim, CA 92807
Phone: (714) 630-0964
Telex: 821375 SUPPORT UD
Easylink 62562040 BBS (714) 632-8750

sorts of schemes for organizing data and add new elements and dimensions at any time without having to perform costly and cumbersome conversions. The strings have a length limit of 255 characters, however, and there are some limitations on the number of characters allowed in the various subscripts of a single entry. Even so, it is easy to create simple applications requiring indexed data and not much harder to create and maintain other inverted lists for secondary keys.

A PROGRAMMING EXAMPLE

The program in Listing 1 is a good example of how flexible and compact MUMPS code can be. This routine is a simple, generalized, data-entry program. By using the macro substitution (@) feature described earlier, this program will provide the basic control logic to accept any sequence of data-entry fields. Listing 1A is taken from the CCSM cookbook manual, with minor changes. In Listing 1B, we have taken the liberty of expanding the abbreviated commands to make the listing more readable to those new to MUMPS. A full text explanation follows. (See listings on page 40.)

This program assumes that a global array called LIST has already been created that describes a data-entry system (similar to a format file in dBASE, for instance). Each element in LIST contains four parameters: a prompt name, a logical test in MUMPS code, a reference pointer telling

where the entered data is to be placed (presumably, in some global variable), and an error message should the logical test fail. As programmed here, these parameters are assumed to be separated by commas, in the general form:

```
PROMPT, TEST, REFERENCE,
ERRORMESSAGE
```

The first line of the BEGIN routine sets up a loop (FOR I=1:1) that terminates if the value of the data in LIST(I) is zero or null (QUIT:DATA\$(^LIST(I))). The ' sign here means 'not', and the ^ sign means that LIST is a global, not a local, variable.

Assuming that LIST(I) exists, a local variable XX is then set to it (SET XX=^I), where ^ refers to the last global used. Then the GETDATA routine is called (DO GETDATA), and the loop iterates. When the loop finally ends, all the local variables are killed (KILL I,X,XX,PROMPT,TEST,REF), and the routine is QUIT. The first line of the GETDATA routine parses the PROMPT, TEST, and REF parameters from the local variable XX, using the MUMPS parsing function PS. The second line prints the prompt, then any existing data, and some slashes (WRITE:\$DATA(@REF) @REF). Then the routine requests input from the keyboard (READ X QUIT:X=""), storing it in X and quitting if X is null.

The third line of GETDATA performs the test stored in TEST (IF @TEST), then writes a linefeed (WRITE !), sets the refer-

ence to X {SET @REF=X}, and quits. The final line is only executed if the test fails; a bell is rung using ASCII 7 {WRITE *7," "}, followed by the parsed-out error message. Then the GETDATA routine is called again.

OTHER FEATURES AND EXTENSIONS

Unfortunately, much of the documentation provided for MUMPS commands and operators is cryptic and disorganized. Moreover, some of the information provided in the manuals is incomplete, incorrect, and contradictory, making the experience of learning the finer points of MUMPS frustrating for a new user. For example, several listings from the manual had blatant mistakes, and many forms of the functions used in examples were difficult to find in either manual.

Users who are willing to do some digging will find some interesting and valuable features offered with this version of MUMPS, though. For example, it is possible to read and write ordinary PC-DOS files, to make use of an 8087 chip, to transfer files to and from other MUMPS systems, to customize CCSM for various foreign codes and formats, to list and cross-reference routines and variables, to manipulate CCSM error-trapping and error-message routines, to back-up and copy global variables, to journalize events such as global creation and destruction, to modify screen attributes, and so on.

Our version of CCSM was single user, so we weren't able to test multiuser functions. According to the literature, however, it is theoretically possible to run up to 15 users on a properly configured PC/XT or PC/AT, and it is possible to network up to 255 multiuser nodes. We also did not test an optional PC graphics package that allows users to draw in a variety of shapes, colors and textures. Support is provided for the IBM Color Graphics Adapter and for a number of more advanced graphics boards and monitors.

CONCLUSION

On the whole, we are very happy with MUMPS, and CCSM appears to be a fairly good and efficient implementation of the ANSI standard, along with many useful extensions. We must reiterate, however, that the documentation is confusing and rather inconvenient to use. This should not be any great burden to seasoned MUMPS users, but it is a major annoyance to beginners. If MUMPS is to gain converts from other, more popular, microcomputer languages and databases, more attention to the details of documentation is definitely required. §

Michael Guttman and Robert Engle are independent computer consultants based in Cherry Hill, NJ.

PRODUCT INFORMATION

CCSM—\$59.95
CCSM + Cookbook—\$75.90
Graphics Disk—\$49.95

MGlobal

1601 Westheimer, Ste. 201
Houston, TX 77006
(800) 257-8052
(713) 529-2576 (in Texas)

Typical MUMPS Listing (1A)

```
BEGIN F I=1:1 Q:'D$(^LIST(I)) S XX=^(I) D GETDATA
      K I,X,XX,PROMPT,TEST,REF
      Q
```

```
GETDATA S PROMPT=$P(XX,"/",1), TEST=$P(XX,"/",2),
          REF=$P(XX,"/",3)
          W PROMPT," " W:$D(@REF) @REF," // " R X Q:X=""
          I @TEST W ! S @REF=X Q
          W *7," ", $P(XX,"/",4), ! G GETDATA
```

Expanded MUMPS Listing (1B)

```
BEGIN FOR I=1:1 QUIT:'D$(^LIST(I)) SET XX=^(I) DO GETDATA
      KILL I,X,XX,PROMPT,TEST,REF
      QUIT
```

```
GETDATA SET PROMPT=$P(XX,"/",1), TEST=$P(XX,"/",2),
          REF=$P(XX,"/",3)
          WRITE PROMPT," " WRITE:$D(@REF) @REF," // "
          READ X QUIT:X=""
          IF @TEST WRITE ! SET @REF=X QUIT
          WRITE *7," ", $P(XX,"/",4), ! GOTO GETDATA

      ET @REF=X QUIT
          WRITE *7," ", $P(XX,"/",4), ! GOTO GETDATA
```



Over 90 volumes of public domain "C" software including:

- compilers
- editors
- text formatters
- communications packages
- many UNIX-like tools

Write or call for more details

The C Users' Group

Post Office Box 97
McPherson, KS 67460
(316) 241-1065

FREE

52 PAGE CATALOG AND APPLICATIONS GUIDE

AFFORDABLE ENGINEERING SOFTWARE

CP/M MSDOS

TRSDOS PCDOS

PDP — Plotter Driver Program
For multi and single pen plotters **\$72.95**

XFER — Transfer Function Synthesis
Transfer functions/ Electronic circuits **\$72.95**

TEKCALC — Scientific Calculator Program
Screen graphics/ Statistics/ Programmable **\$72.95**

COMCALC — Communications Design Spreadsheet
Communications budget calculator **\$72.95**

RIGHTWRITER — Report Proofreader Program
Version 2 — Applies 2,200 rules of English **\$97.95**

ACNAP — AC Network Analysis **\$72.95**

DCNAP — DC Network Analysis **\$72.95**

SPP — Signal Processing Program **\$72.95**

PLOTPRO — Scientific Graph Printing **\$72.95**

PCPLOT — High Resolution Graphics **\$72.95**

LOCIPRO — Roof Locus Analysis **\$72.95**

ACTFIL — Active Filter Design **\$72.95**

STAP — Static Thermal Analysis **\$72.95**

MATRIC MAGIC — Matrix Manipulation **\$72.95**

BV Engineering
Professional Software



(714) 781-0252

2200 Business Way, Suite 207, Riverside, CA 92501 U.S.A.

IBM AT Compatible Features:

- 512K RAM expands to 1MB
- 33% faster than IBM AT
- 1.2MB floppy drive
- 8/6 MHZ CPU 80286-8
- USA BIOS fully compatible
- Clock/calendar w/battery
- Hard disk/floppy controller
- AT style Keyboard
- 195 watt power supply
- 48 hour burn in
- Full documentation
- Includes setup software
- One year limited warranty

\$1125*

MADE IN U.S.A.



* Monitor not Included

Hard Disks

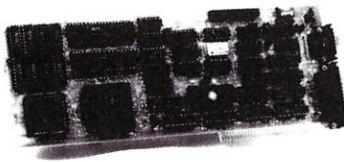
Let PC INNOVATIONS provide, Install, and test the proper Hard Drive for your AT

(drive only)

SEAGATE 225 65MS	\$310
SEAGATE 4026 39MS	\$549
SEAGATE 4038 39MS	\$595
SEAGATE 4051 39MS	\$699
SEAGATE 4096 28MS	\$1095
MINISCRIBE 70MB 28MS	\$1195

10MHz Zero Wait State SI Rating 11.7 **\$1495**

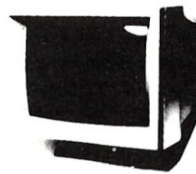
EVEREX EGA GRAPHIC CARD



- 16 colors 640 x 350 res.
- 256K video memory
- Parallel port

\$249

NEC MULTISYNC

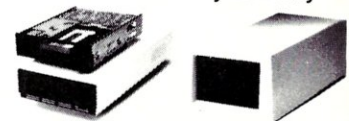


- EGA, PGA, CGA monitor
- 800 x 560

\$569

EVEREX TAPE BACKUP SYSTEMS

The best tape backup money can buy



	INTERNAL	EXTERNAL
20MB	\$589	\$639
60MB	\$759	\$829
100MB	---	\$929
Complete subsystem w/software		

TRY PC INNOVATIONS PERIPHERAL CARDS FCC CLASS 'B' APPROVED

I/O for AT



\$75

- * XT Version available
- Half slot
 - Parallel port
 - Serial port
 - Optional 2nd serial

THE EDGE



\$215

- Color display
- Hercules mono
- 132 column
- Printer port
- Software
- Lotus 123 graphics

HERCULES COMPATIBLE



\$99

- Monochrome graphics
- Requires no software
- Printer port
- Optional serial port
- Runs Lotus 123 graphics

AT MEMORY CARD



\$165

- Up to 3MB
- No piggy back
- Split memory
- Uses 64K or 256K chips
- Software

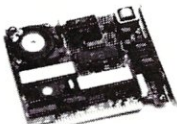
MULTIFUNCTION CARD FOR AT



\$175

- Up to 3MB
- Serial port
- Optional 2nd serial port
- Parallel port
- Game adaptor
- Software

HAYES 1200B MODEM



With SmartCom II **\$349**

2400 BAUD
EVERCOM 1200 BAUD
U.S. Robotics:
2400 Internal **\$389**
2400 External **\$389**

CALL **\$128**

PC INNOVATIONS SPECIAL OFFERS

TEAC floppy disk 360K \$89 1.2MB	\$119	EPSON FX 286	\$539
Toshiba floppy disk 360K \$95 1.2MB	\$125	EPSON FX 85	\$379
Floppy disk controller	\$40	EPSON LQ 800	\$549
Multifunction floppy disk controller	\$95	EPSON LQ 1000	\$739
XT 150 watt power supply	\$75	Toshiba P321 printer	\$525
AT 192 watt power supply	\$109	Toshiba P341 printer	\$757
Memory chips, Eproms, 8087, 80287	CALL	Leading Mfg. 13" RGB display	\$259
		MS400 AT 4 port serial card	\$115
		MS400 XT 4 port serial card	\$110

VOLUME DISCOUNTS AVAILABLE

PC INNOVATIONS

ESTABLISHED SINCE 1980

631 S. Main Street • Plymouth, MI 48170

(313) 451-0664

IBM, IBM AT, Hayes, Hercules are trademarks of their respective companies.

VISA & MasterCard Welcome
Prices subject to change

UNIVERSITY AND CORPORATE BUYERS WELCOME

Our trained sales and technical staff is ready to serve you.

COMPETITIVE EDGE

631 S. Main St. — Plymouth, MI 48170 — (313) 451-0665

Competitive Edge brings you Lomas Data Products Inc.

S-100 CIRCUIT BOARDS

Thunder 186 , 512K, 2 Serial, 1 Parallel, Floppy Disk Controller, Clock, All on One Board with Concurrent DOS			\$746
8MHZ Lightning 286 CPU	\$756	NV Disk 512K, Memory Drive	\$369
10MHZ Lightning 286 CPU	821	Concurrent DOS Single User	280
Control It All Floppy & HD Controller	479	MSDOS™ for 86, 186, or 286	200
Hazitall 2 Serial, 2 Par, Clock	239	CPM22EM 8080 Emulator	55
256K Dram Megaram	358	Mega S Ram 256K Static	371
1024K (1 Megabyte) Megaram	599	8 MHZ Lightning 8086	420
Mega S Ram 16K Static	262	LDP Floppy Contr.	220
Octaport 8 Serial to 38.4K	320	Color Magic Color Graphics Board	476
		512K Dram Megagram	\$448
		Thunder + 186 512K 10MHZ	995
		Thunder + 186 1024K 10MHZ	1146
		Octaport 4 Serial	200
		NV Disk 2048K Wow!!	746
		Concurrent DOS Multi-user	360
		LDP-COM Modem Program	55
		PC DOS Drivers	35

COMPETITIVE EDGE INTEGRATED SYSTEMS

THUNDER 186, 4 SLOT PC STYLE CABINET, 2-5" FLOPPYS, CDOS 512K	\$1595
THUNDER 186, 4 SLOT, 20 MB HARD DISK, 1-5" FLOPPY, CDOS 512K	2595
THUNDER 186, 4 SLOT, 20 MB HD, 1-5" FLOPPY 4 USER 1024K	3195
THUNDER 186, 4 SLOT, 2-5" FLOPPY, COLOR MAGIC, KEYBOARD 512K	2250
8MHZ 286, 1-5" FLOPPY, 1024K, 20MB HD, 10 SERIAL, 15 SLOT, 7-8 USERS	4695
8MHZ 286, 1-5" FLOPPY, 1024K, 40MB HD, 10 SERIAL, 15 SLOT, 7-8 USERS	5295
10MHZ 286, 1-5" FLOPPY, 1024K, STATIC, 40 MB HD, 10 SERIAL, 7-8 USERS	7550
10MHZ 286, 1-5" FLOPPY, 512K STATIC, 40 MB HD, 2 SERIAL & 80287	6495
8MHZ 8086/8087, 1-5" FLOPPY, 512K DRAM, 20 MB HD, 2 SERIAL CDOS or MSDOS	3695
10MHZ 286 1-5" FLOPPY 1024K 120MB HD 10 SERIAL SLOT 7-8 USERS CDOS	8495

LOMAS STATIC RAM NEW 256K TO 1024K PRICES START AT \$412

FEATURING CompuPro® FROM VIASYN™

S-100 CompuPro® CIRCUIT BOARDS

Disk 1A™ Floppy Disk Controller	\$449	Network 100™ Network Board	\$379	Interfacer 4™ 3 Serial 1 Par	\$349
8 MHZ 286 CPU A&T 2 CYL.	679	80287 Option for 286 CPU	295	CPU Z™ 8MHZZ80	219
Ram 22™ 256K Static Ram A&T	449	Concurrent DOS™ 8-16™	739	MDRIVE-H 2048K, 2MB	1339
Ram 23™ 64K Static Ram A&T	199	CP/M® 2.28" or 5"	189	Keytronics 5150 for PC Video	169
Interfacer 3™ 8 Serial Ports	446	Disk 3™ Hard Disk Controller	529	Passive Hub for Network 100	65
SPU Z™ 8MHZ Z80, 256K Multiuser	369	10MHZ 8088, 85/88 CPU A&T	269	HX-12 Hi-Res Color Monitor PC	459
MDRIVE-H™ 512K Memory Drive	559	Ram 23™ 128K Static Ram A&T	275	CP/M® 8-16™	259
PC Video Board for IBM® Compat	379	System Support 1™ A&T	309	SP186 512K Slave	599

HARD DISK SUB-SYSTEMS

40 MEGABYTE HARD DISK, CABINET, DISK 3, SUB-SYSTEM	\$1495
80 MEGABYTE HARD DISK, CABINET, DISK 3, SUB-SYSTEM	2295
ALLOY 17 MB TAPE BACKUP	1795
40 MEGABYTE HARD DISK, CABINET, DISK 3, 5" FLOPPY	1695
8" & 5" FLOPPYS, 40 MB HARD DISK SUB-SYSTEM	2495
SYSTEMS INTEGRATED BY COMPETITIVE EDGE	

COMPUPRO® BUILT SYSTEMS

COMPUPRO MP14 — 14 USER (7 DUAL 186 SLAVES) 10 MHZ 286 80 MB	\$13995.
COMPUPRO 286/80 1024K, 10MHZ 286, 80 MB 4-8 USERS EXPANDABLE	9249.
COMPUPRO 286/40 768K, 8 MHZ 286, 40MB 4-5 USERS EXPANDABLE	6649.
COMPUPRO 10+ 768K (256K ADDITIONAL SLAVE MEMORY) 40 MB 4 USERS	4436.*

Ask About Sperry Service Contracts on CompuPro®

QUANTITY AND DEALER DISCOUNTS AVAILABLE

ALL PRICES SUBJECT TO CHANGE — SHIPPING EXTRA — *LIMITED TIME

IBM is a registered trademark of International Business Machines, CP/M, Concurrent DOS are registered trademarks of Digital Research Inc. MSDOS is a trademark of Microsoft. All above circuit board names are either registered trademarks or Trademarks of Viasys Corporation. CompuPro is a registered trademark of Viasyn Corporation. AT is a trademark of IBM.

NO EXTRA CHARGE FOR VISA OR MASTER CHARGE

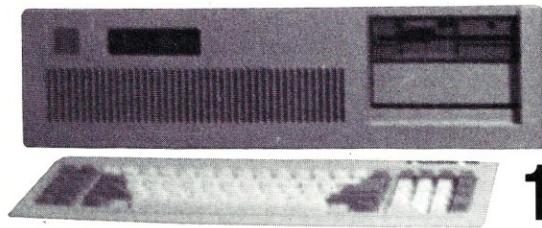
COMPETITIVE EDGE

631 S. Main St. — Plymouth, MI 48170 — (313) 451-0665

INTRODUCING

VELOCITY 286-10™/12

A T™ COMPATIBLE



10/12 MHz OPERATION

1024K on MOTHERBOARD
16 BIT HARD DISK TRANSFERS
SERIAL, PARALLEL, CLOCK

Enhanced Graphic Adapter Now Standard Runs with Color or Mono Monitor
1.2 MB Floppy, A T™ Compatible Keyboard, DOS 3.2

30 MEGABYTE FAST HARD DISK (38MS)

10 MHZ Velocity 286-10 As Above \$2595.

12 MHZ Velocity 286-12 As Above \$2895.

132 Column by 44 Lines No Cost Option

10 MHZ Velocity 286-10 with 70MB 28MS H.D. \$3095

12 MHZ Velocity 286-12 with 70MB 28MS H.D. \$3395.

SHIPPING NOW!!!

— OPTIONS —

80287 Math CoProcessor

Computone Xenix Tape Back Up \$999

30 to 100 MB Hard Drives

20 to 100 MB Streaming Tape Back (2 minutes per 10 MB)

CPM®/Z80H Emulator Card Only **\$395.00**

8MHZ 8088 Slaves (up to 31) for Multi-User

3 MB Memory Expansion Only **\$795.00**

Monochrome & Color Monitors

Basic Xenix® 1024K, 8 Serial, 30MB System **\$4095**

NO EXTRA CHARGE FOR VISA OR MASTER CHARGE

CPM is a registered trademark of Digital Research Inc. Hercules is a trademark of Hercules. A T is a trademark of International Business Machines. Velocity 8 and are a trademark of Competitive Edge. Screen is PC Paint Brush copyright and product of 2Soft, Inc. Xenix is a trademark of Micro Soft.

Program Interfacing To Microsoft Windows

by William Wong

Part I—An Introduction To Windows

Editor's Note:

Bill Wong concluded his ten part series on "Program Interfacing To MS-DOS" in the last issue. Bill now begins a new series on program interfacing to Microsoft Windows.

It is time to take a look at Microsoft Windows, a graphical user interface for PC/MS-DOS. Windows is both a user interface and a programming environment. This article offers a short description of the user interface. Subsequent parts of this series will discuss the programming environment.

Windows requires a graphics adapter such as the CGA, EGA, or Hercules card. It supports but does not require a pointing device, such as a mouse. The interface uses windows, pull-down menus, and icons, similar to those found on the Xerox Star and Apple Macintosh.

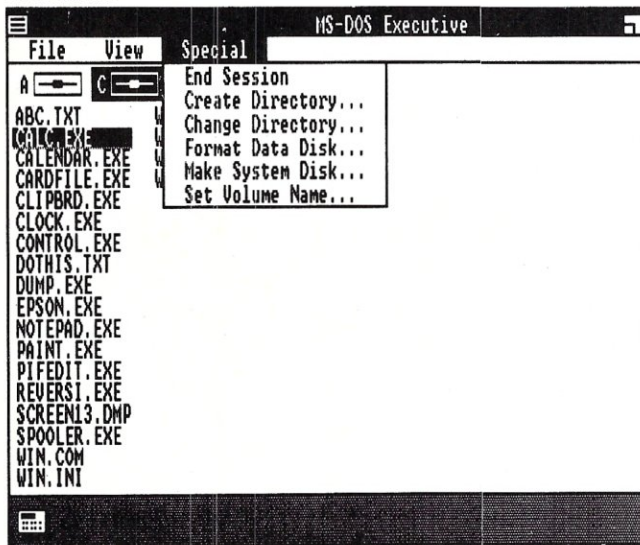


Figure 1 Reprinted with permission from Microsoft Corp.

OPERATION

The top-level Windows screen presents a title and menu line at the top of the screen (Figure 1). You can select menus using the keyboard or by pointing to a menu title, using a mouse. A pull-

down menu is then presented below the menu title. A pull-down menu may have a number of selections. By convention, a line in a pull-down menu with three dots after the text indicates a subsequent interaction, such as the appearance of a dialog box. This feature will be discussed later.

The center of the top-level Windows screen is called the work area. This is the area used by applications. Applications can use the entire work area, or only a portion of that area if multiple application windows are open at one time. In the latter case, the windows are *tiled*. This means that the entire work area will be used and that no application windows will overlap. Removing a window causes others to expand. The converse is true when a new window is added. Pull-down menus and pop-up dialog boxes are an exception since they use the work area.

The bottom of the top-level Windows screen is used to keep icons that represent applications that are loaded but do not use the screen. An icon represents an application that is not currently using a window. Use of icons is a way to keep many applications active without cluttering up the screen. You can open an icon into a window, within the work area, to use the application.

You select items and actions in Windows using the mouse or the keyboard. You use a cursor of varying form to pick an item. The various forms are numerous and are described in the *Microsoft Windows User's Guide*. The important thing to note, at this point, is that it describes an interface that should be followed by all Windows. This consistent interface makes it easier to use new and different applications under Windows. Also, Windows provides support for most of the features used in the top-level Windows interface.

For example, Windows Write, a word-processing application provided with Windows, uses pull-down menus, pop-up dialog boxes, and windows with scroll bars. A scroll bar appears on the right, or bottom edge, of a window; it indicates where in a document the current display appears. It also allows you to move rapidly within a document by moving an indicator within the scroll bar. Windows provides complete support for things such as the scroll bar, including drawing and moving the indicator. The application only needs to keep track of the window and the document. This greatly simplifies the development process of a Windows application.

Windows offers a nonpreemptive multitasking environment. This allows multiple applications to be available at one time. Nonpreemptive means that one application will run until it wishes to give up the use of the processor. It can let another program run by executing an explicit operation, such as delaying for a set period of time or waiting for the mouse to select the application's window. Applications that are moved to the icon area are normally waiting to be opened into a window. Preemptive multitasking systems, such as UNIX, split processor time among programs through various means, such as time slicing and priority mechanisms.

TURBO PASCAL GENERATOR

THE GTP PROFESSIONAL MODEL

Generate error-free Turbo Pascal source code for:

**DATA BASES MENUS
SCREENS REPORTS**

Build complete, working programs in minutes!

FEATURES

- Indexed Data Bases
- Multiple Screens
- Automatic Updating
- Built-in Edits
- Retrieval Facility
- Automatic File Build
- Quick Screen Handling
- Speedy DB Access
- Context Sensitive HELP
- Full Keyboard Support

EASY

1. Paint-the-screen
2. Define fields & calculations
3. Generate, compile & RUN

FLEXIBLE

- 100% Turbo Pascal
- Modify in Pascal under Turbo's Editor
- Extensions are easy with 130 page Programmer Reference manual
- 100 documented source code routines included

NOT Copy Protected NO Royalty Fees NO Run-time Library Required

Requires: IBM PC (100% compatible) 256K RAM 2 Disk Drives
PC-DOS 2.0+ Turbo Pascal 3.0 from Borland International

PRICE \$200.00

CALL FOR QUANTITY DISCOUNTS

VISA/MC — Check — Money Order — No COD or Purchase Orders
Texas Residents: add \$12.25 Sales Tax
Outside US & Canada: add \$25.00 Air Postage and make payment by
credit card or money order in US Funds.

**A
E
F**

ALLEN, EMERSON & FRANKLIN

P.O. Box 928
Katy, TX 77492
(713) 391-8570

B-R-E-A-K THE DIRECTORY BARRIER WITH DPATH + PLUS™

- Run WordStar, dBase, 1-2-3 or any other program from any disk directory, without having to copy overlay files, help files, dictionaries, libraries and the like into that directory.
- Find ANY file for ANY program, even if the program wants its files to reside in the current directory.
- **Output** files can even be created in other directories.
- Assign directories to pseudo-drives, such as F:, and refer to the directory using only the pseudo-drive id.
- **Protect** your hard disk from accidental formatting (true IBM compatibles only).
- DPATH+Plus specs are updatable at any time, even while a program is running!
- Can be deactivated and later reactivated.
- A **trace-mode** facility allows you to see what file-oriented calls any program makes to DOS.
- Operation is completely transparent to your program. **Fast** — written in assembly language.
- Works with hard disks, floppies or RAM disks, in any combination. Requires DOS 2.0 or later. Runs in 4-9 KB of memory, depending on features used.
- **Not** copy-protected. Can be installed on hard disk.

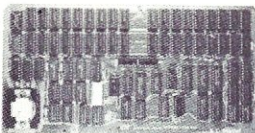
\$45 + \$5 s/h.
COD add \$3.
Visa/MC/check.

Personal Business Solutions, Inc
PO Box 739, Dept P
Frederick MD 21701
☎ 301-865-3376

60-day money back guarantee!

For S100 bus by S.C. Digital, Inc.

MODEL 256KMA



1M DYNAMIC RAM Board Model 256KMA-512 \$375
• 1 Mega bytes • 8/16b data, 24b address • Parity per byte • 175n sec access time • will run Z80/Z8000 to 6mhz, 8086 68000 to 8mhz without wait states • transparent refresh, unlimited DMA • works in most systems including Compupro • with 512KB

MODEL 80286CPU



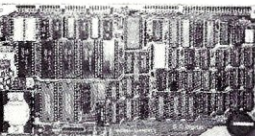
256K/1M DYNAMIC RAM board Model 256KB-256 \$325
• 1 Mega bytes • 8b data, 16/24b address • parity • Memory Mapping in 16K blocks, Ideal for Ram Disk, application notes available • addressable in 128K, 192K, or 256K boundaries • with 256KB

MODEL FDC-1A



128K STATIC RAM/EPROM Model 128KS \$349 no ram \$169
• 128K bytes, for 6264 rams or 2764 eproms, can be mixed • 8/16b data, 24b address • bank select, RAM shadowing, etc. • ROM simulation in RAM • Battery backup with powerfail detector/memory disable, battery holder.

MODEL SUPPORT-1



FLOPPY DISK CONTROLLER board Model FDC-1A \$325
• Single or double density, sides, in any combination of up to four 8", 5 1/4" drives • DMA data transfer with cross 64K boundaries, 24b address, DMA arbitration • built in monitor/boot ROM that accommodates two different processors • serial port with hand shakes, up to 19.2K baud • 765A software compatible, compatible with Compupro's Disk 1A • with CP/M bios.

8086 CPU Board Model 8086 CPU \$325
• 8/4 (or 10/5) mhz SW selectable • 8087 interface • provision to run two processors on a bus such as our Z80 CPU • convertible to 10, 12mhz clock • optimized for DRAM boards • specify clock speed.

80286 CPU Board Model 80286CPU from \$350
• 8/4 mhz switchable • 80827 interface • provision to run two processors on a bus • convertible to 10mhz • separate built in clock for 80287 • optimized for DRAM boards.

Z80 CPU Model Z80CPU \$249
• 6 or 8mhz, specify • memory mapping in 16K blocks, addressable four M bytes.

SUPPORT Board Model Support-1 \$325
• 4 serial, full handshakes, two with software programmable baud rates • Centronics • SASI interface • Real/interval timers • Calendar-clock with battery backup • expandable interrupt controllers for 8086 or 8080/Z80 • CPU switching circuit to run 2 processors on a bus such as our 8086 or 80286 and Z80 CPU boards.

I/O Interface Board Model 3SPC-N \$229
• 3 serial RS-232C with switch selectable baud rates, 110 to 19.2K baud. • 1 parallel.

256K STATIC RAM Board Model 256KS \$375
• 256K bytes • 8/16b data, 24b address • can be loaded in 16Kb increments • fast access time: will run in access of 10mhz when fully loaded with 150nsec chips • Battery backup with power fail detector/memory disable, battery holder • w/ 256KB, less battery.

Please call for latest prices.
Prices subject to change without notice.

S.C. DIGITAL, INC.

1240 N. HIGHLAND AVE., SUITE 4 • P.O. BOX 906, AURORA, ILLINOIS 60507
PHONE: (312) 897-7749

APPLICATIONS

Applications come in two forms: conventional DOS applications and Window applications. Windows allows conventional DOS applications to run, but these applications normally will not use any of Window's extended features and may have to use the entire screen. Also, the application will often have to be terminated to run another application.

Applications can be swapped out to disk to leave more room for additional applications. An application will be swapped in by Windows when it is active, if it is not already resident. Swapping for a non-Windows application is automatic and controlled using .PIF files, which will be discussed later. Unfortunately, the entire area used by this type of application is swapped. Windows applications can make use of the Windows memory manager for more control, because the application has control over each allocated segment. Some can be resident while others are transient. Although this technique is not the same as virtual memory, offers a better solution than implementing such a system within an application.

CONFIGURING WINDOWS

The initial Windows configuration is specified using a text file called WIN.INI, which is read by Windows when it is loaded. The file is divided into a number of sections. The settings within the file are used by Windows and are also available to other applications through Window's supplied functions, to be discussed in a later article. The following syntax is used within the WIN.INI file:

```
[section]
  keyword=value
```

The *section* can be any name and may be specific to an application. The *keyword* and *value* pairs can be any text. Many values can be included with one keyword and are usually separated by commas. An application refers to the values using the section and keyword. An example of part of the standard WIN.INI file is:

```
[Windows]
  DoubleClickSpeed=500
  CursorBlinkRate=450
  Device=Epson FX-80,epson,LPT1:
  Load=clock
  SwapSize=0
  SwapDisk=E
[Ports]
  COM1:=9600,n,8,1,p
```

Windows uses a number of sections, including the one listed above. Applications can be loaded by default using the *Load* keyword. This allows you to have your standard configuration set up when you enter Windows. *SwapDisk* refers to the disk to be used when an application needs to be swapped out. *SwapSize* is the number of kilobytes to be used for the swap file. A zero (0) indicates that the size is to be taken from the first application to be swapped out.

Logical and physical Window devices can be indicated in the *Devices* and *Ports* sections. These sections are often used by applications to determine what devices are available for use. A Program Information section, [PIF], is used to specify what system resources an application needs, or uses. This is important because some non-Windows-specific programs may have different needs, which cannot be set up by the program. The [PIF] section normally contains a list of programs and the required amount of memory in kilobytes.

PIF FILES

In addition, Windows recognizes .PIF files. These files are created and manipulated using PIFEDIT.EXE, a standard Windows support application. A .PIF file contains information about a program, its parameters, the initial directory in which it should run, memory requirements, screen usage, and direct-device usage. The latter allows Windows to recover if, for example, the communications port is set directly by an application.

The screen-usage characteristics are important because they may allow a non-Windows application to run within a window. Those characteristics also indicate memory requirements for applications in which screen capture, or a screen dump, may be available. High-resolution screen-display adapters require more memory to save the screen than low-resolution display adapters.

WINDOWS APPLICATIONS

Three Windows applications included with Windows are the spooler, the control panel, and the clipboard. The spooler allows spooling and redirecting printer output. The control panel allows general Windows attributes to be examined and modified. These applications utilize the Windows environment and available windowing functions. The clipboard application allows exchange of text and graphics between other applications. The actual mechanism and its limitations is quite extensive and will not be covered here. Windows is often supplied with other Windows-based applications.

DYNAMIC DATA EXCHANGE

Another Windows-specific feature is called DDE (Dynamic Data Exchange). DDE is a protocol that uses Windows functions to allow applications to communicate with each other by name. Most current Windows applications do not support DDE, but some newer applications do include DDE support. The DDE protocol will be discussed in more detail in a later article.

A sample scenario would be a communications program that included a DDE hook so that information from an on-line service, like Dow Jones, could be made available. A spreadsheet program could use this hook to get information to present real-time results. The spreadsheet could also be hooked to a graphing program, which would take these results and present a graph in a window, which could then be updated as new results were available. All three applications would need a DDE interface. The communications program would advertise the existence of the on-line information. The spreadsheet would have to access this information and advertise the availability of the results, and the graphing program would have to access the results.

NAMING CONVENTIONS

Windows is a large and complex piece of software. This article and subsequent articles will use the same naming conventions as are used in the Microsoft documentation. Examples will be written in C for consistency. First, there are the simple data types:

Prefix	Type	Description
c	char	8-bit character
b	BOOL	8-bit Boolean (0 is false, nonzero is true)
n	int	16-bit integer (default if prefix)
l	LONG	32-bit integer
l	long	same as LONG
f	BYTE	8-bit unsigned integer
w	WORD	16-bit unsigned integer
dw	DWORD	32-bit unsigned integer
h	HANDLE	16-bit unsigned integer
p	NEAR	16-bit pointer
lp	FAR	32-bit pointer
pt	POINT	32-bit coordinate (X,Y) two 16-bit numbers
rgb	RGB	32-bit RGB color value R+(G*256)+(B*256*256)

POINTERS AND HANDLES

Pointers and handles actually come in several different forms. They are listed here and will be used in definitions in later articles. Like the previous types, these new types use the same prefix convention, except the type name is capitalized.

Type	Description
PSTR	NEAR pointer to character string
PINT	NEAR pointer to signed 16-bit integer
LPSTR	FAR pointer to character string
LPINT	FAR pointer to signed 16-bit integer
LPRECT	FAR pointer to RECT
LPMSG	FAR pointer to MSG
FARPROC	FAR pointer to a function
HWND	HANDLE to a window
HSTR	HANDLE to string resource
HCURSOR	HANDLE to cursor resource
HICON	HANDLE to icon resource
HMENU	HANDLE to menu resource
HDC	HANDLE to display context
HPEN	HANDLE to physical pen
HFONT	HANDLE to physical font
HBRUSH	HANDLE to physical brush
HBITMAP	HANDLE to physical bitmap
HRGN	HANDLE to physical region
GLOBALHANDLE	HANDLE to global memory
LOCALHANDLE	HANDLE to local memory

Of course, the pointers introduced a couple of new structures. These are listed here:

```
typedef struct
{
    int x ;
    int y ;
}
POINT ;
typedef struct
{
    int left ;
    int top ;
    int right ;
    int bottom ;
}
```

```
RECT ;
typedef struct
{
    HWND hwnd ;
    WORD message ;
    WORD wParam ;
    LONG lParam ;
    DWORD time ;
    POINT pt ;
}
MSG ;
```

POINT and RECT have obvious uses and definitions. MSG, on the other hand, is specific to Windows. It is a structure used to send a message between windows, which is the way applications can communicate with each other and with Windows. The elements of the structure are: the window receiving the message; a message number; a WORD and LONG parameter whose meaning is specified by the message number; the time the message was posted; and the mouse position, listed in screen coordinates, when the message was posted. The actual use of this structure will be covered in the next article, which discusses basic window functions.

LOOKING TO THE FUTURE

As you may have guessed, Windows is driven by handles and pointers just like DOS, except handles are no longer just for files. The way these items are used will become clear in the following articles, so stay tuned. These articles will address the support functions for Windows; the Graphics Device Interface (GDI); and system resources, which include managers and functions for modules, memory, tasks, resources, atoms, communications, sound, and access to the WIN.INI file. §

Bill Wong is president of Logic Fusion, Inc., 1333 Moon Dr., Yardley, PA 19067, a systems software development firm.

What you see is what you get . . . and send!

Transfer Protocol: Modem7/CRC Packet Size: 128 Files: 1

Block #	of	Kbytes	%	Time Remaining	Errors			Total Kbytes
					Consec	File	Total	
31	522	3	5	5:06	0	0	0	3

Errors:
Status: Transfer in progress
[MEX File Transfer]



[CTL-C to abort]

Sending: >ANYFILE.AQC

Announcing Version 1.6 of **MEX**, the communications software with a view from the top. Regardless of your level of sophistication, **MEX** can put you on top of the data transfer game and keep you there. For the executive on the go, our new pull-down transfer screen and easy-to-use menus reduce the complexities of modem communications to a few keystrokes. For the advanced user, **MEX's** greatly enhanced script processor offers a complete programming language for development of highly secure custom applications. If communication is money in your business, **MEX** may be the best investment you make this year.

Two options available:

MEX-PC is the most complete modem software you can buy. Allows you to switch between menu-driven and command-driven communications at will. Makes full use of Hayes AT command set, with overlays available for most other modems. Features include: complete script processor programming language; user-definable keystrokes; auto-dial and auto-baud-set phone libraries; all popular protocols, including MODEM-7 batch transfers **\$59.95***

MEX-PAC — All the features of **MEX-PC**, plus:

- (1) A remote module that allows you to run your office computer from home, and vice versa; and
- (2) Terminal emulation that lets your PC masquerade as a DEC VT52/100 or Televideo 925 terminal for on-line communication with mainframes. **\$99.95***

XMODEM CRC/XMODEM CHECKSUM/KERMIT/COMPUERVE A/128 or 1K BLOCKS

Versions available for IBM-PC and compatibles, Tandy 2000 and most CP/M machines.

MasterCard, VISA welcome

**Give us a toll-free call at
1-800-NITEOWL**

In Wisconsin, 1-414-563-4013

* plus shipping; Wisconsin residents add 5% sales tax.

NightOwl Software, Rt.1 Box 7, Ft. Atkinson, WI 53538

MSO
MICRO
SUPPLY
ORGANIZATION INC.

dare to compare.

**Don't be fooled.
No hidden cost!
Complete MS-DOS/CPM
Super Turbo**

In keeping with industry trends MSO is bringing our customers high performance P.C. compatibles and accessories. MSO takes the P.C. compatible to maximum performance with its SUPER TURBO. The SUPER TURBO features the V20-8 chip which runs at three times the speed of the IBM-PC XT* and also runs CPM 8080 software.

The SUPER TURBO comes complete with the MS-DOS operating system, Read and Run CPM, full Instructional Documentation, Utility software, **plus for our first 100 customers MSO is offering the Micropro Wordstar Professional Software package. This package includes: Wordstar, Mailmerge, Correctstar, Starindex, Datastar and a G.L. Accounting System.** The SUPER TURBO is a complete turnkey system with everything necessary to plug in and operate.

**USER SUPPORT
HOT LINE**

805/393-2247

All systems carry full 90 day warranty.

CASH PRICE ONLY

Check in advance. Add 3% for VISA/MC. Shipping & handling charges will be added to each order.

For our catalog with complete details and prices, send \$2.00 to:

Micro Supply Organization, Inc.
4909 Stockdale Hwy. #180
Bakersfield, CA 93309

MON. - FRI. 7am - 5pm PST • SAT. 9am - 5pm PST

*IBM is a registered trademark of the IBM Corporation.
15% restocking on returned orders

Super Turbo
Super Price:
\$1299



**The Super Turbo P.C. runs IBM
software and CPM 8080 programs**

- CPU - V20-8 8mHz Super Chip runs *IBM compatible software at 3 times the speed of the IBM-XT and CP/M 8080 software.
- 8087 Math Processor optional
- 256K RAM on mother board expandable to 640K
- ROM 8K Bios
- 6 empty slots for expansion
- 2 serial port one optional with expansion kit
- 1 parallel port
- 1 game port
- Clock calendar with software
- Hi-Res monographics video board
- Floppy controller
- Dual Floppy Drives 360K ea.
- 135 watt XT Power Supply
- 5150 style compatible keyboard
- Hi-Res TTL Green or Amber 12" monitor
- MS-DOS operating system and manual.
- Instructional Documentation and Utility Software
- Assembled and tested in U.S.A.
- **Optional internal 20 meg sub system for Super Turbo add \$499**

**Special printer
pricing with
purchase of
above computer.**

MSO
MICRO
SUPPLY
 ORGANIZATION INC.

the giveaway.

MSO, the leader in cost effective computer systems, is offering the Sanyo MBC 1200 at a **giveaway price!**

The MBC 1200 Accounting Software -G/L, A/R, A/P, Inventory and Payroll modules retail for \$398 per module. **Now from MSO you pay only \$99 per module or all five for \$299!** Second disk drive with fan-option available. **COMPAT** disk utility available.

SPECIFICATIONS:

- Two Z-80A (main and subsidiary) CPUs with no-wait mode for fast execution, substantial memory capacity (RAM 64KB, ROM 4KB).
- High-resolution full graphic function with 640 x 400 dot matrix display.
- Choice of 33 or 40 line text mode.
- CP/M operating system complete with interpreter, editor and all utilities.
- Easy-to-use Sanyo graphic BASIC.
- One (MBC 1200) or two (MBC 1250) internal double-sided, double-density, double-track, 5¼" slim-type 640KB formatted mini floppy disk drives.
- Special design featuring detachable ergonomic keyboard with coiled cable.
- Interfaces for Centronics printer and one RS-232C port provided.

20 meg internal hard drive sub-system for Sanyo MBC 1100, 1160 & 1200

\$599 (with purchase)

Printer specials with purchase

805/393-2247

All systems carry full 90 day warranty.

CASH PRICE ONLY

Check in advance. Add 3% for VISA/MC. Shipping & handling charges will be added to each order.

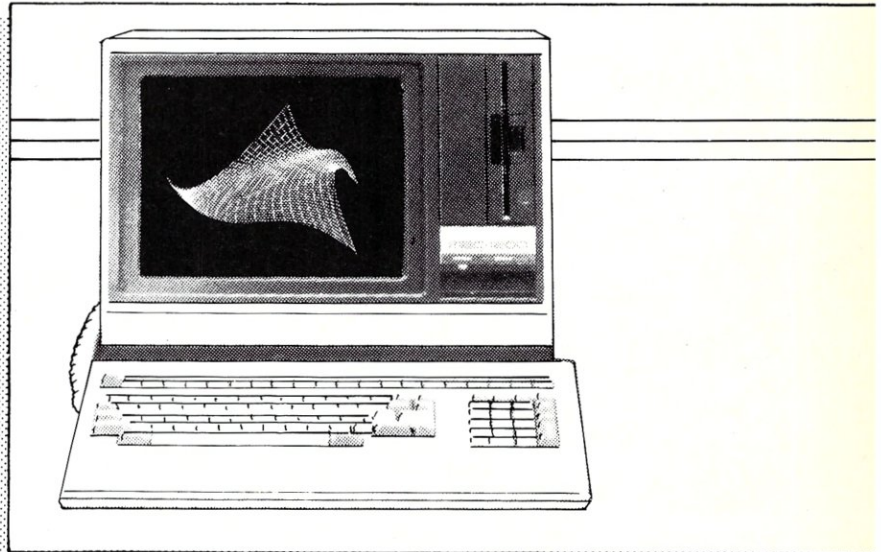
For our catalog with complete details and prices, send \$2 to:

Micro Supply Organization, Inc.
 4909 Stockdale Hwy. #180
 Bakersfield, CA 93309



15% Restocking on Returned Orders

SANYO
MBC 1200



FREE SOFTWARE

- Wordstar • Spellstar
- Calcstar • Mailmerge
- Infostar • Basic

\$399

DeSmet C

— now with —

LARGE MEMORY OPTION

C88 Compiler \$109

Full K&R + V7 extensions
inline **asm**
assembler, linker, librarian
full screen editor (SEEtm),
8087 & S/W floating point

D88 Debugger \$50

Set Breakpoints by line number
or function name
Examine Global/Local variables
by name
Show C source while debugging
Both D88 and User displays

Large Memory Model Option \$50

32-Bit Pointers (Full Megabyte
Addressing)
Fast Access of Static Data
Works with D88

— plus —

DOS Link \$35

convert to .OBJ files

Graphics (+src) \$35

Hacker \$25

source for start-up,
RAM.COM & more

Make \$50

full UNIX-level

Tools (+source) \$35

XARRAY \$39

Large Arrays (+source)

C WARE CORPORATION

505 W. OLIVE, SUITE 767
SUNNYVALE, CA 94086 USA
(408) 720-9696 TELEX: 358185
We accept
VISA, MC & American Express

FIGURES & CODE FOR MARSHALL BRAIN ARTICLE - Experimenting With Protected Mode On The AT

Figure 1 - Sequence of events used whenever a new value is loaded into a segment register.

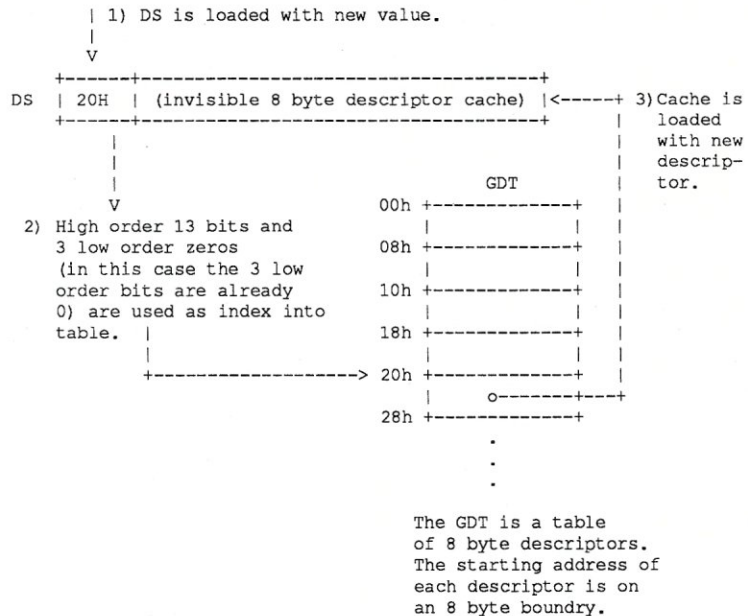
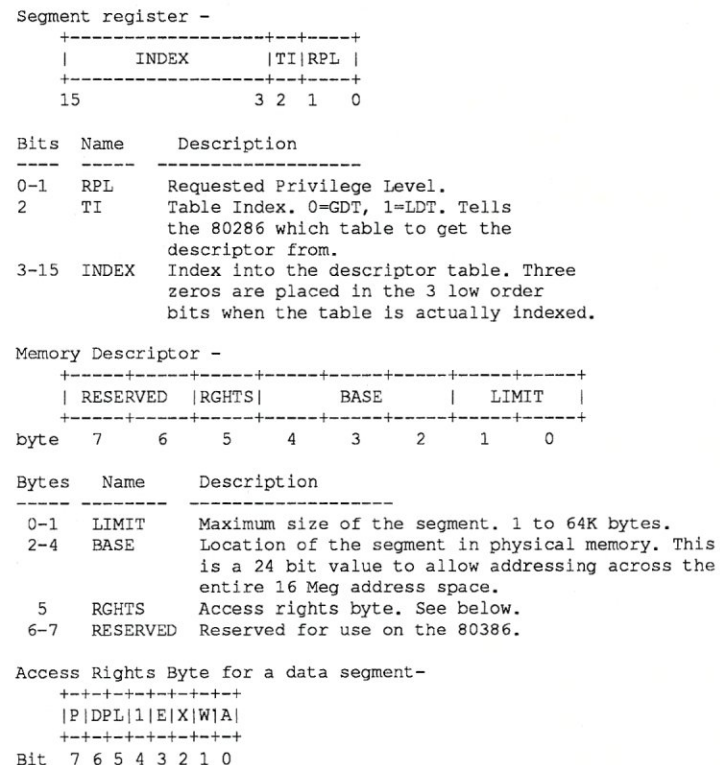


Figure 2 - On the 80286, a segment register consists of the 16 byte segment register found on the 8088, as well as 8 bytes of cached information from a descriptor table. The individual bits of each of these values are defined as shown.



Bits	Name	Description
0	Accessed	1 Indicates that segment has been accessed.
1	Writeable	For a data segment, 1=writeable, 0=read only. For a code segment (bit 3 = 1), 1=readable, 0=execute only.
2	eXpansion	1=expand down, 0=expand up.
3	Executable	1=executable segment, 0=data segment
4		Indicates Memory descriptor (as opposed to Control)
5-6	DPL	Descriptor privilege level.
7	Present	Indicates that segment is present in memory. 1=Present.

Figure 3 - Processor faults on the 80286. Taken from the Intel manual.

Number	Description
* 0	Divide Error
* 1	Single Step interrupt
* 2	Non-Maskable interrupt
* 3	Breakpoint interrupt
* 4	INTO Overflow
5	Bound Range Exceeded exception
6	Invalid Opcode exception
7	Processor Extension Unavailable
8	Double Exception Detected
9	Processor Extension Segment Overrun
10	Invalid Task State Segment
11	Segment Not Present
12	Stack Segment Overflow or Not Present
13	General Protection Error

*=same as 8088

Figure 4 - Segments expected to be found in the GDT by the AT BIOS call Interrupt 15H, function 89H.

Number	Name	Description
0	Dummy	Segment 0 is invalid, so initialize to 0.
1	GDT Alias	Since the GDT is itself a segment, there must be a descriptor available if it is to be changed. A descriptor created to access a segment that would not normally be accessible is called an "alias".
2	IDT Alias	Alias for interrupt table.
3	Data Seg	Descriptor to point to the user's data segment.
4	Extra Seg	Descriptor to point to user's extra segment.
5	Stack Seg	Descriptor to point to user's stack segment.
6	User Code Seg	Descriptor to point to user's code segment.
7	BIOS Code Seg	The BIOS routine fills this descriptor so that its code can continue executing once the jump to protected mode occurs. Once in protected mode, the BIOS routine loads the segment registers with the user defined values, and then jumps to the user's code.

CODE STARTS HERE

```

program protected_mode_and_back;
{$u-,k-} {These directives MUST be set off like this}

{Marshall Brain Version 1.0 September 15, 1986}
{This program demonstrates what is required to go into protected
mode, and then come back to real mode. Interrupt 15, function
89 is used to get into protected mode. See article or AT ROM BIOS
listing for details.
The program will beep once (low tone) from protected mode and
return to DOS if everything works correctly. A processor fault
will cause 2 beeps (high tone) and halt the system.}

```

LOOKING FOR "NET" RESULTS?

EARTH COMPUTERS
has the
solution
to your
Networking
problems.

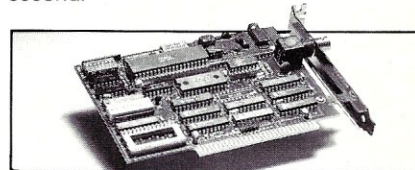


EARTHNET-PC.™

EARTHNET-PC is the most flexible networking card on the market. It has been designed for high performance and maximum functionality.

EARTHNET-PC is fully compatible with SMC networking cards and runs popular networking software such as NOVELL's NETWARE, ViaNet, and TurboDOS, all of which support the new LAN Standard and DOS 3.1 record locking.

EARTHNET-PC's 5-1/2 inch card fits in any short slot of an IBM-PC/XT or compatible system and uses advanced Token-Passing technology. Data transfers are made at 2.5 Megabytes per second.



YOU CAN STOP SEARCHING...**EARTHNET-PC** IS THE SOLUTION TO YOUR NETWORKING PROBLEMS! Order your **EARTHNET-PC** today! Call **EARTH COMPUTERS**, the company that's setting the standard for LAN standards.

ATTENTION DEALERS! If you've been searching for ways to increase your NETWORKING profits, call **EARTH COMPUTERS** and find out about our attractive, profit-generating dealer program.

EARTHNET-PC is a trademark of EARTH COMPUTERS
NETWARE is a trademark of Novell
ViaNet is a trademark of Vianetix, Inc.
TurboDOS is a trademark of Software 2000
IBM-PC/XT is a trademark of International Business Machines, Inc.

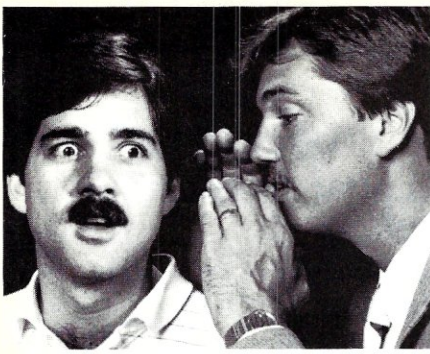


EARTH COMPUTERS

P.O. Box 8067, Fountain Valley, CA 92728
TELEX: 910 997 6120 EARTH FV

(714) 964-5784

Ask about EARTH COMPUTERS' other fine PC and S-100 compatible products.



If You Could Hear The Language They're Using!

What they're saying would astound you.

They've switched from BASIC, and PASCAL, and a host of other languages and even stand-alone database programs.

They've switched to COMP Computing Standard MUMPS...CCSM, the Database Language.

"The best software product of its kind, that I have come in contact with." *Computer Language Magazine*

"I've found the best in CCSM...fast in development and fast in execution...no data-typing problems, no concerns for program size, no concerns for file or device opens..." *R.D. Ashworth, Ph.D.*

"...5 years in Basic, Pascal, C, dBase, and Dataflex... I have never worked with a language/programming environment as responsive, easy to use and as powerful as CCSM" *P.K. Wayne, MD, Ph.D.*

CCSM, the Database Language is a powerful, flexible string-oriented language, with the features of a complete development and operating system. Typical programs are written in 1/3 to 1/5 the amount of code, compared to BASIC or PASCAL.

Multi-Tasking Available, Too!

CCSM, the Database Language offers a true multi-tasking option. Run multiple, concurrent background processes for data searches, report generation, etc.

Order CCSM, the Database Language today.

In a very short time, you'll be surprised at the language you're using.

IBM's and compatibles...128K min.

1-800-257-8052

in Texas 713-529-2576

MGlobal

1601 Westheimer, Suite 201
Houston, TX 77006

AMEX, VISA AND MC accepted by phone

Single User disk and operations manual	\$59.95
"CookBook of MUMPS" (includes disk)	\$24.95
SPECIAL Single user & "Cookbook"	\$75.95
Multi-tasking	\$149.95
Programmer's Toolkit I	\$49.95
Graphics option	\$49.95
Multi-user	\$450.00
MacMUMPS (Macintosh version)	\$149.95
Shipping and handling	\$3.00

Texas residents add 6 1/8% sales tax

IBM is a trademark of International Business Machines;
Macintosh is a trademark licensed to Apple Computer

```

const
  gdt_size=8;      {Max number of descriptors in GDT minus 1}
  idt_size=13;    {Max number of descriptors in IDT minus 1}
  code_desc={6*8}48; {Code segment selector is #6 in GDT}
const {typed constants are variables in the code segment}
data_seg:integer = 0; {storage space for segment registers}
stack_seg:integer = 0;
extra_seg:integer = 0;
type
  descriptor=record {from AT Bios listing}
    seg_limit:integer; {segment limit (1-65536 bytes)}
    base_lo_word:integer; {24 bit physical address (0-(16M-1))}
    base_hi_byte:byte;
    data_acc_rights:byte; {access rights byte}
    data_reserved:integer {reserved for 80386 compatability}
  end;
  string80=string[80];
var
  gdt:array[0..gdt_size] of descriptor; {Global descriptor table}
  idt:array[0..idt_size] of descriptor; {interrupt dsctrptr table}
  result:record ax,bx,cx,dx,bp,si,di,ds,es,flags:integer; end;
  mask1_8259,mask2_8259:byte; {storage space for 8259 masks}
  screen:array[0..$2000] of byte absolute $40:0;

```

```

procedure setup_error;
{Displays a message if you try to setup a descriptor outside the
boundaries of a descriptor table.}
begin
  writeln;writeln('There has been an error during setup of
  'descriptor tables. ');
  halt;
end;

```

```

procedure set_gdt_desc(num,seg_lim,base_lo:integer;
                      base_hi,acc_rights:byte);
{This procedure sets up descriptor number NUM in the GDT with the
values that are passed.}
begin
  if (num>=0) and (num<=gdt_size) then
  begin
    with gdt[num] do
    begin
      seg_limit:=seg_lim;
      base_lo_word:=base_lo;
      base_hi_byte:=base_hi;
      data_acc_rights:=acc_rights;
      data_reserved:=0;
    end;
  end else setup_error;
end;

```

```

procedure set_idt_desc(num,seg_lim,base_lo:integer;
                      base_hi,acc_rights:byte);
{This procedure sets up descriptor number NUM in the IDT with the
values that are passed.}
begin
  if (num>=0) and (num<=idt_size) then
  begin
    with idt[num] do
    begin
      seg_limit:=seg_lim;
      base_lo_word:=base_lo;
      base_hi_byte:=base_hi;
      data_acc_rights:=acc_rights;
      data_reserved:=0;
    end;
  end else setup_error;
end;

```

```

procedure beep;
{beeps the speaker.}
begin
  sound(300);delay(400);nosound;delay(400);
end;

```

```

procedure errbeep;
begin
  sound(2000);delay(400);nosound;delay(400);
end;

```



```

procedure fault;
{This routine is entered if a processor fault occurs while in
protected mode.}
begin
  errbeep;
  errbeep;
  {pop the 4 words put on the stack when fault occurred.}
  inline($5b/$5b/$5b/$5b); {pop BX 4 times.}
  port[$64]:=$fe; {return to real mode}
  inline($f4); {Halt to wait for reset to take effect.}
end;

procedure setup_idt;
{this procedure loads the idt with the appropriate values to handle
all 80286 processor faults.}
var x:integer;
begin
  {The first 14 selectors are all 80286 faults, and are directed to
the fault routine for now.}
  set_idt_desc(0,ofs(fault)+7,code_desc,0,$86);
  set_idt_desc(1,ofs(fault)+7,code_desc,0,$86);
  set_idt_desc(2,ofs(fault)+7,code_desc,0,$86);
  set_idt_desc(3,ofs(fault)+7,code_desc,0,$86);
  set_idt_desc(4,ofs(fault)+7,code_desc,0,$86);
  set_idt_desc(5,ofs(fault)+7,code_desc,0,$86);
  set_idt_desc(6,ofs(fault)+7,code_desc,0,$86);
  set_idt_desc(7,ofs(fault)+7,code_desc,0,$86);
  set_idt_desc(8,ofs(fault)+7,code_desc,0,$86);
  set_idt_desc(9,ofs(fault)+7,code_desc,0,$86);
  set_idt_desc(10,ofs(fault)+7,code_desc,0,$86);
  set_idt_desc(11,ofs(fault)+7,code_desc,0,$86);
  set_idt_desc(12,ofs(fault)+7,code_desc,0,$86);
  set_idt_desc(13,ofs(fault)+7,code_desc,0,$86);
end;

procedure addr24(segment,offset:integer; var a24w:integer; var a24b:byte);
{To work in protected mode, physical addresses must be expressed
in 24 bit values. Addr24 puts the lower 16 bits of the 24
bit address in a24w, and puts the upper 8 bits in a24b.}
var x:integer;y:byte;
begin
  inline(
    $8B/$86/segment/ {mov ax,segment[bp] }
    $BA/$10/$00/ {mov dx,16 ; move segment over 4 bits}
    $F7/$E2/ {mul dx }
    $03/$86/offset/ {add ax,offset[bp]; add offset and send }
    $83/$D2/$00/ {adc dl,0 ; carry bit to dl }
    $88/$96/y/ {mov y[bp],dl ; save result }
    $89/$86/x); {mov x[bp],ax ; save result }
  a24w:=x;
  a24b:=y;
end;

procedure setup_gdt;
{This procedure sets up the GDT with the descriptors required by
Int 15, function 89.}
var a24w:integer;a24b:byte; {24 bit physical addr storage}
begin
  set_gdt_desc(0,0,0,0,0); {# Function }
  addr24(dseg,ofs(gdt),a24w,a24b); {0 dummy descriptor}
  set_gdt_desc(1,sizeof(gdt),a24w,a24b,$92); {1 GDT alias }
  addr24(dseg,ofs(idt),a24w,a24b);
  set_gdt_desc(2,sizeof(idt),a24w,a24b,$92); {2 IDT alias }
  addr24(dseg,0,a24w,a24b);
  set_gdt_desc(3,$ffff,a24w,a24b,$92); {3 Data descriptor }
  set_gdt_desc(4,$ffff,a24w,a24b,$92); {4 Extra descriptor}
  addr24(sseg,0,a24w,a24b);
  set_gdt_desc(5,$ffff,a24w,a24b,$92); {5 Stack descriptor}
  addr24(cseg,0,a24w,a24b);
  set_gdt_desc(6,$ffff,a24w,a24b,$9a); {6 Code descriptor }
  set_gdt_desc(7,0,0,0,0); {7 Temp Bios CS }
  {the following is an additional descr used to access the screen.}
  addr24($b800,0,a24w,a24b);
  set_gdt_desc(8,$2000,a24w,a24b,$92); {8 Screen descriptor }
end;

procedure save_segs;
{Saves the current segment values in typed constants so they can
be retrieved when program returns from protected mode.}
begin

```

Multi-User Networking Specialists

System Manager PC

List Price \$375

Complete PC/Compatible Network Security Package

- All menu driven
- Restrict drive and directories
- Log ON/Log Off record keeping
- Full electronic mail
- Display system status
- Talk-send messages to another processor
- Acct. manager-menu driven acct. editor
- Runs on all PC's/compatible's

We support large TurboDOS systems
with multiple masters

We support the following OEM's

ICD ICM Earth Computers
L/F Technologies Teleteck

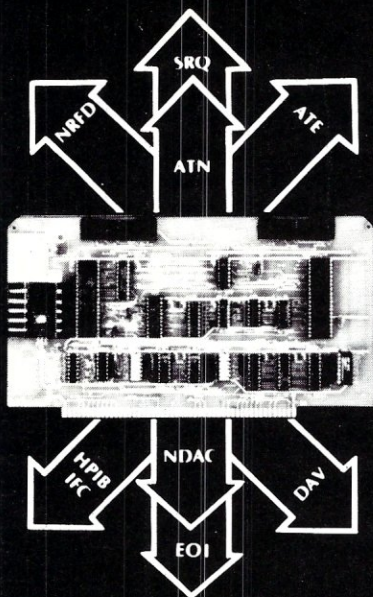
Dealer discounts

Lodden
Technology
Limited

2834 N. Milford Rd.
Milford, MI 48042

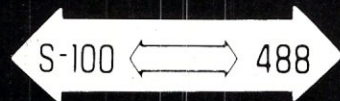
(313) 685-8538 ORDER
(313) 685-3877 TECH
(313) 685-7947 COMPUTER LINE

TurboDOS is a registered trade mark of Software 2000, Inc. System
Manager Lodden Technology Ltd.



THE 488+3

IEEE 488 TO S-100
INTERFACE



- Controls IEEE 488 (HP1B) Instruments with an S-100 computer
- Acts as controller or device
- Basic and assembly language drivers supplied
- Meets IEEE 696 specification
- Industrial quality burned in and tested up to 125K bytes/sec under software control 3 parallel ports (8255-5)
- \$375



D&W DIGITAL, INC.
20655 Hathaway Avenue
Hayward, California 94541
(415) 887-5711

```

data_seg:=dseg;
extra_seg:=dseg;
stack_seg:=sseg;
end;

procedure save_8259_regs;
{Saves current 8259 masks so they can be replaced on return to
 real mode.}
begin
  mask1_8259:=port[$21];
  mask2_8259:=port[$a1];
end;

procedure set_cmos_for_shutdown;
{sets cmos ram so that the reset that brings this program back to
 DOS jumps to the correct address in this program.}
begin
  port[$70]:=$0f;
  port[$71]:=$05;
end;

procedure protected_mode_exitpoint;
{When program returns to real mode, if comes here.}
begin
  {recover segment registers}
  inline(
    $2E/$A1/data_seg/      {mov ax,cs:data_seg }
    $8E/$D8/                {mov ds,ax }
    $2E/$A1/extra_seg/     {mov ax,cs:extra_seg}
    $8E/$C0/                {mov es,ax }
    $2E/$A1/stack_seg/     {mov ax,cs:stack_seg}
    $8E/$D0);              {mov ss,ax }
  {reset 8259 masks}
  port[$21]:=mask1_8259;
  port[$a1]:=mask2_8259;
  {when this procedure returns, it will return to the place where
 goto_protected_mode was called from (see last line of pgm).}
end;

procedure setup_jump_locations;
{set up appropriate memory locations with address of routine
 to use when returning to real mode.}
begin
  memw[$0040:$67]:=ofs(protected_mode_exitpoint)+7;
  memw[$0040:$69]:=cseg;
end;

procedure display(col,row:byte;ln:string80);
{routine to display information on the screen by writing directly
 into screen memory. Col,row express the coords that LN should be
 displayed at.}
var x,addr:integer;
begin
  addr:=row*80+col;
  for x:=0 to length(ln)-1 do
    screen[(x+addr)*2]:=ord(ln[x+1]);
end;

procedure test_screen;
{writes a message on the screen.}
var x:integer;
begin
  for x:= 1 to 5 do
    display(5,x,'HELLO FROM PROTECTED MODE. ');
end;

procedure goto_protected_mode;
{Take program into protected mode and perform whatever routines
 are needed while in protected mode. Returns to real mode when
 done. Do not use any turbo function requiring BP register in this
 routine.}
begin
  {Call interrupt 15, function 89}
  inline( {do interrupt 15h, function 89h to get to protected mode}
    $8c/$d8/      {mov ax,ds } {load address of GDT into es:si}
    $8e/$c0/      {mov es,ax }
    $be/gdt/      {mov si,ofs(gdt)}
    $b7/$08/      {mov bh,8} {bh,bl contain new location for}
    $b3/$08/      {mov bl,8} {hardware interrupt vectors}
    $b4/$89/      {mov ah,89h}
    $cd/$15);     {int 15h}

```


POPPY. A true multi-user computer.

Charter Information's POPPY is a new variety that will make your office blossom.

POPPY handles all facets of your business — purchasing, sales analysis, and general ledger functions. Accounts payable, inventory control, materials requirements planning, and payroll. Plus desktop publishing and a host of other wonderful things that you need to do.

POPPY consists of a processor board, SmartCable, keyboard, and monitor. Plant multiple POPPY boards in a System 4 and watch your productivity bloom.

POPPY runs programs that operate on an IBM PC, including those that use graphics. And when it comes to dazzling displays, POPPY provides text and graphics in both color and black and white. Roses pale in comparison. Each POPPY has its own processor, one that is powerful enough to make a PC wilt. So you get to see what you want to see and do what you need to do as soon as you are ready.

What is really outstanding is POPPY's ability to fit in. The System 4 gives POPPY access to shared hard and soft disk drives, tape drives, printers and other devices. All under the control of a master processor that fulfills each POPPY's every need. The master takes care of spooled printing, controlled access to discs, file and record locks and operation of printers.

POPPY is no garden variety product.

If you are looking for a rainbow of colors to make your business ideas look better, wait until you see what POPPY can do — for all your users. You see, unlike standalone computers which require separate cards for each graphics mode, POPPY provides all the most popular graphics modes as standard equipment. It lets you switch among them through software — no fumbling with tiny dipswitches. That includes monochrome, color, EGA and Hercules graphics.

It's no wall flower, POPPY communicates.

Each POPPY comes with a PC compatible serial port. That means that each user can have a serial printer, a mouse or an external modem of its very own. And there is an optional 1200/2400 baud modem built in, if you specify it. Rooted in the S-100 technology of the System 4, communications between your users and shared information is faster than summer lightning. And that cultivates user productivity like never before.

How did they do it?

The designers of POPPY and System 4 have had years of experience implementing multi-user installations, in both manufacturing and office situations. They realize that multi-user environments demand something more than just another garden variety machine to insure adequate throughput and make information sharing practical among all your users. So, they created a multiple processor system. Every POPPY has its own processor and memory. Then, using the fastest communications available over a common data bus and the power of TurboDOS, they designed in a master processor to take care of the things that usually slow you down. And, up to sixteen POPPYs can be arranged in one System 4, working like one big bouquet of computing power.

How do you use it?

Our consultants can help you design the system that best fits your needs. The System 4 can be a complete computer for a small company or a departmental system. Or it can be a part of a distributed network for larger companies, providing multi-user access to data. If you have ordinary PCs that you want to link into a POPPY network, you can do so over a conventional serial LAN.

If I pick it, will it be easy to use?

Absolutely. If you already know PC-DOS, you're in the driver's seat. And if you need assistance, applications have help screens that take you through the rough spots. TurboDOS, the operating system for the System 4, lets you use all the most popular programs that run on the IBM PC — so if you're already using programs like Lotus 1-2-3, Wordstar, dBASE III+, and Flight Simulator, you won't have to transplant anything. You can merge data on your PC with information from POPPY. And all your users can share all the information.

CP/M already in use? No problem. You can run CP/M 80, CP/M86, and PC-DOS programs at the same time under TurboDOS, with file and record locking. (So the data goes to the people you want sharing it and no one else.)

Let's plant the seed.

Before you settle for just a garden variety computer, you're invited to pick POPPY. A computer with the power and flexibility to grow with you. Please contact one of our representatives today by phone or fill out the coupon and mail it. Who knows? Our relationship may bloom into something big.



Specifications:

Workstation:	
Processor	80188
Clock Speed	8MHz (no wait states)
RAM	512K
Display	Monochrome, CGA, EGA, and Hercules
Keyboard	Selectric style
Port	Serial port (IBM PC Com 1 compatible)
Modem	Integral 1200/2400 baud modem (optional)
SmartCable	11 pair shielded (200' max.)
System 4	
Backplane	IEEE 696
Master processor	ICM memory mapped
Diskette	720K or 320K (software controlled)
Hard Disk	20, 46, or 76 MB
Tape backup	Stream out, media or file by file restore (optional)
Max. Poppys/case	Sixteen
Max. Printers/case	Sixteen

You've planted a seed in my mind.

Please send me more information on how I can use POPPY to cultivate productivity.

Name _____
Title _____
Company _____
Address _____
City _____
State _____ Zip _____
Telephone _____



Charter Information Corp

Where good ideas improve.

2421 Rutland
Austin, Texas 78758
(512) 835-1111

#1 Lint for MS-DOS

KILLS
C BUGS
FAST

PC-lint

The professional diagnostic facility for C

PC-lint lets you zap swarms of C bugs and glitches at a time.

Now you can uncover the quirks, inconsistencies, and subtle errors that infest your C programs . . . waiting to bite you. PC-lint finds them all . . . or as many as you want . . . in one pass. Set PC-lint to match your own style.

Outperforms any lint at any price

- Full K&R support and common ANSI enhancements (even MS keywords)
- Finds inconsistencies (especially in function calls across multiple modules!)
- Modifiable library descriptions for 8 popular compilers
- Super fast, one-pass operation
- Suppress any error message
- Zillions of options

PRICE \$139 • MC • VISA • COD

Includes USA shipping and handling. Outside USA, add \$15. In PA add 6%.

ORDER TODAY, 30-day guarantee

Runs under MS-DOS 2.0 and up, and AmigaDOS. Uses all available memory.

Trademarks: PC-lint (Gimpel Software), MS, MS-DOS (Microsoft), Amiga (Commodore)

GIMPEL SOFTWARE

3207 Hogarth Lane,
Collegeville, PA 19426
(215) 584-4261

```
{procedures to be executed in protected mode should go here}
beep;
{test_screen;}
port[$64]:=$fe; {return to real mode}
inline($f4); {Halt to wait for reset to take effect.}
end;
```

```
begin {main routine}
setup_idt;
setup_gdt;
save_segs;
save_8259_regs;
set_cmos_for_shutdown;
setup_jump_locations;
goto_protected_mode;
{protected_mode_exitpoint will return to here.}
end.
```

BSW-Make

A practical and efficient

software configuration manager

for MS-DOS, VAX/VMS, and VM/CMS

At The Boston Software Works, we routinely work with a number of different operating systems and development environments. One tool we have found to be indispensable is **BSW-Make**. BSW-Make is a complete implementation of the UNIX *make* utility. It automates the tedious task of rebuilding your software after an editing session; BSW-Make does only the minimum work required to update your program after a change, saving time and preventing missed compiles.

We carefully constructed BSW-Make to be portable, and have used it successfully under MS-DOS, PC-DOS, VAX/VMS, and VM/CMS. We wouldn't want to start a major software project without it, and we think you won't either, once you've tried it.

Highlights of BSW-Make:

- Works with any compiler, assembler, linker, or text processor
- Not copy protected
- Indirect command file generation facility overcomes operating system command length limitations
- Macro facility for parameterized builds
- Syntax compatible with UNIX *make*
- 30-day unconditional money-back guarantee

MS-DOS
\$89.95

VAX/VMS
\$495.00

VM/CMS
\$550.00/yr

BSW-Make for MS-DOS runs on any MS-DOS machine. It requires MS-DOS or PC-DOS version 2.00 or later, and is shipped on IBM PC 5¼ inch diskettes.

BSW-Make for VAX/VMS runs on any VAX or MicroVAX running VMS version 4.0 or later. It is shipped on 9-track magtape or RX50 diskette.

(Available soon) BSW-Make for VM/CMS runs on any IBM 370-series, 43xx, 308x, or 309x system running VM/CMS. It is shipped on 9-track magtape.

All prices include shipping within the United States and Canada. Foreign orders (except Canada) add \$10.00 handling; actual shipping cost will be billed. We accept checks, MasterCard or VISA, or company purchase order.

The Boston Software Works, Inc.

120 Fulton Street, Boston, MA 02109
(617) 367-6846

Creating MYLIB in Modula-2

by Charlie Foster

A Modula-2 Tutorial—Creating Library Modules

Any language is only as good as the extent to which it helps the programmer. Being inherently lazy, I needed a structured, self-documenting, powerful language that I could learn fast. As an ex-Pascal advocate, I can say that Modula-2 fits the bill. I don't want to beat the drum about all of its power, but I do want to share with you one aspect that I have found outstanding.

I don't mind having to write a new procedure, but I get bored when I have to rewrite it time after time. In Modula-2, library modules provide programmers with an external hierarchical structure. Program modules are at the highest level of the hierarchy and can import identifiers, from library modules. Library modules naturally export identifiers but they can also import identifiers from lower-level library modules.

It was obvious that once I wrote a good, debugged, working general procedure, all I had to do was save it in a library, and it would always be available for any program I might write.

Thus, you now know what I want to share with you. I will show you how to create your own library called "MYLIB." The process is simple, and the library you create can help you speed up your programming tremendously.

Creating a library module requires the writing of two compilable units: a Definition module and an Implementation module. Normally, these files have the same file name but different extensions: for example, MYLIB.DEF and MYLIB.IMP.

The DEF or Definition module contains constant, type, and variable declarations and procedure headings. This is all the compiler has to know to check type consistency across compilation units. It is also the only information essential to any programmer who wishes to make use of this module. The IMP or Implementation module contains the bodies of procedures and the statement part of the module.

The modules are hierarchical, so changing definitions part means you have to recompile all lower modules. When you change the implementation part of the modules, however, only that part has to be recompiled. That is what you would usually be doing—modifying the body of your procedures rather than the headings. The result is that you rarely have to recompile a module because you have changed a lower one.

Each library module, once successfully compiled, is composed of two components: a symbol file (MYLIB.SYM) and an object file (MYLIB.MCD). My particular compiler uses M-code as an interim device for speed of development. Once ready to go, an option flag is flipped, and the program then compiles into a Z80 COM file. That is the reason for the type MCD: It is the M-code file type.

Together the symbol and object files form a library module. The only thing you need to do when you want to use the procedures within the library module is to import them. For example:

```
MODULE Example;  
FROM MyLib EXPORT Beep;  
VAR
```

```
  identifiers;
```

```
  etc, etc
```

The example program above shows that I only needed to ask the compiler to go to MYLIB and export the procedure BEEP. I could have said:

```
FROM MyLib EXPORT Beep, AppendExt,  
  StripExt, GetExt;
```

and the compiler would have gone to MYLIB and gotten all of those Procedures for me from MYLIB. So I have the option of getting one or more procedures from my personal library. I will go into depth later about those procedures and how to make them into a library module but I want to expand on some additional benefits of using this type of a module approach to programming.

An important capability provided through module usage is data abstraction. Data abstraction is accomplished by first declaring the type as an identifier in the Definition module, without describing the

structure of the type—for example:

```
TYPE AnIdentifier;
```

and then placing the description of the type in the corresponding Implementation module.

```
TYPE AnIdentifier = WHATEVER;
```

This technique effectively hides the structural details of the type from users of the module. A type declared in this way is often referred to as a *hidden type*. Any user only needs to look at the Definition module for his programming information. The details of the procedures are hidden and are of no interest, because you only want to use them. You should try to put in extra comments in the Implementation module because, once it is written, you probably won't see these commands again for long periods of time and will naturally forget why you did what you did.

Back to our library: So that you can see the steps I took to develop the library, I will present them one at a time. First, I wrote the Definition module:

```
DEFINITION MODULE MyLib;
```

```
TYPE
```

```
  Extension = ARRAY [0..2] OF CHAR;
```

```
  FileName = ARRAY [0..12] OF CHAR;
```

```
PROCEDURE AppendExt
```

```
  (VAR DiskFileName : FileName;
```

```
   Ext : Extension);
```

```
PROCEDURE StripExt
```

```
  (VAR DiskFileName : FileName);
```

```
PROCEDURE GetExt (str : FileName;
```

```
  VAR Ext : Extension);
```

```
PROCEDURE Beep;
```

```
END MyLib.
```

Not much to it, is there? If I were to publish MYLIB, I would only have to explain the Definition module, since what is in the Implementation module makes no difference to the user. There can be no conflict of identifiers, because the variables are local and thus invisible to any procedure or module outside of that procedure.

The Implementation module is a little more complex than the Definition module because it is actually a normal procedure or module written with comments or any-
Continued on page 72

B-Tree/ISAM File Handlers

by Andrew L. Bender, M.D.

Have you ever been frustrated by the capabilities or performance of a database-management package? Then why not use an off-the-shelf B-Tree/ISAM file handler to do the job? There are several such packages currently available that may suit your needs and surprise you with their capabilities. Add a CRT screen handler program for input, a report-generator program for output, and some C code to glue them together, and you will have a high performance, sophisticated database administration system tailored exactly to your needs.

I tested three database file handler packages. Two of the packages were written in a neutral dialect of C. By this I mean that any C compiler which supports the C language as described in Kernighan and Ritchie's book *The C Programming Language* would be able to compile these programs by making a few minor changes with some global definitions. The other package was supplied as an executable file. It makes itself resident; you invoke it by interfacing subroutines that permit you to call the program not only from C, but from a variety of languages such as BASIC, COBOL, Pascal, or FORTRAN. Actually, you can use any language you desire for interfacing with this package by writing a trivial subprogram.

C-TREE

The first package I tested, from FairCom, Inc., is called c-tree. This B-Tree/ISAM file handler subroutine package, written entirely in C, is supplied as source code. You can modify the source code to do anything you desire. The source code is straightforward and can be compiled with several C compilers. The documentation is excellent—easy to read and apply.

Several weeks after the program arrived, I received some clearly written instructions from FairCom that corrected some minor bugs and added some enhancements to the package.

To test the c-tree subroutines, I used c-tree to create and generate two files with a relational structure. A second test procedure read both files sequentially and then used an index key to perform a random record retrieval from one file, extract a key from that file, and update the other file. I encountered no bugs in this process. c-tree allows for fixed- and variable-length

records, and shared- and exclusive-use files on virtual or permanent devices. The keys can be segmented, unique, binary, or character, but they cannot be signed integers. In other words, c-tree makes magnitude comparisons. You can, however, use your own collating sequence, which, in a sense, lets you do your own comparisons. If these features are still not enough, you can rewrite anything you want, since you have the source code.

In addition to these features, keys can be of variable lengths, and the key segments can be located in variable-length fields of a variable-length record. To round out this impressive list of features, the keys can also be floating point, with trailing alphanumeric characters and, as noted above, you may define any collating sequence for key comparisons.

c-tree supports networking by means of record locking. For those systems such as MS-DOS that support file locking by exclusive use only, dummy lock files are created to lock the region of the file in use.

Naturally, when you use this package you must be able to read and understand the documentation. The manual was clearly written and well printed. A reference card answers most questions once you are familiar with the product. A generous number of source code examples are also provided; they allow you to take a look at the way a certain function was employed, in case the written word wasn't quite enough. A large portion of the manual is given over to improving the performance of c-tree, hints for use, compiling c-tree programs and the like. I rate the documentation A+.

If you develop a package with FairCom's c-tree, you may incorporate any part of the compiled code as a part of the program, subject to certain restrictions. You may then sell the composite binary program without payment of royalties to FairCom. The license agreement seems fair. At \$395 for a single-user license, this program is a good product and reasonably priced. c-tree runs on several computers under different operating systems. It has also been ported from the iAPX 86/88 to the DEC VAX series, the Motorola 68000 and 6809 series, National Semiconductor's 32032, the AT&T 3B2, and the IBM Series/1, all under different operating systems.

BTRIEVE

Next, I investigated SoftCraft's Btrieve, another B-Tree/ISAM file handler that runs under MS-DOS. Unlike c-tree and the other B-Tree/ISAM file handlers I investigated, Btrieve is supplied as a binary module (an .EXE file) that you must load into memory as if it were a program you were going to run by itself.

It comes with an impressive manual that consists of lucid, well-presented examples. It interfaces with many languages. You communicate from your program to, and from, Btrieve by means of interrupts. Btrieve sets the interrupt vector to point into itself; every call to Btrieve becomes a software interrupt call. This allows a high degree of isolation between your program and the package. While there is admittedly some increased overhead on a software interrupt, the number of actual software interrupts executed is small when compared to the rest of the body of program instructions.

Suppose SoftCraft were to change the features in Btrieve to improve their operation? I think that it would be easier just to change the package as an alternative to recompiling or relinking the application programs out there that use Btrieve. Also, I hate waiting for the linker to do its thing. Keeping Btrieve out of the linking process speeds up development and makes overlay programming easier. The down side to this is that a newer version of Btrieve may cause your program to stop working even though you didn't recompile it. This happened, for instance, because a newer version 4.x uses "handle" I/O and the older version 3.x used basic I/O; some older programs actually ran out of handles.

Another recent enhancement, variable-length records, causes Btrieve to return the length of each record read. This could clobber the calling sequence of a program designed to run under the older version of Btrieve; where the record length was only an input to the interface subroutine. In these cases, it is best to stick with the old version of Btrieve until all the enhancements and their consequences are carefully thought out.

The interrupt-calling feature also allows any compiled program, or interpreter, to call Btrieve with a simple subroutine. SoftCraft gives you many examples of these subroutines. Btrieve

comes in a networking version (Btrieve/N) that I did not test. I was assured by numerous Btrieve users that SoftCraft is helpful and courteous.

In addition to supporting a variety of keys—segmented and nongsegmented, duplicated and unique, binary and character—Btrieve allows the user to supply an "own code" collating table for each key.

A powerful feature of Btrieve is transaction processing. You can mark any transaction on a file, or group of files, by means of two operations, "Begin transaction" and "End transaction." You can use an optional "Abort transaction" code to exit from the transaction at any time. The files involved in a transaction are not updated until the transaction ends normally. As a practical example, suppose you were updating a file, or several files, and you determined that for some reason the update had to be discontinued. You could simply abort the transaction, leaving all of the files in their original state without having to back-track and clean up the mess. This process has some overhead, but I used it extensively in a complicated program and saw no performance degradation.

Btrieve performs well and is well documented. It is the fastest B-Tree/ISAM handler I have used. I chose it for two applications over c-tree because I needed its transaction-processing features. It, unlike c-tree, is written in assembly language. Naturally, it can only be used to interface with MS-DOS and cannot be ported to other operating systems or processors. This is a distinct disadvantage if you are writing applications for different machines or operating systems. Nevertheless, as with any program supplied in source-code form, you trade off some flexibility to the mercy of your C compiler, which may not produce highly optimized code, and which requires recompilation and relinking if changes are necessary.

Like c-tree, Btrieve comes with a no-royalty distribution license for the incorporated composite programs. SoftCraft requires that you register your program with it first.

SOFTFOCUS B—Tree ISAM Handler

The last product I evaluated was a collection of three C subroutines from SoftFocus. These subroutines fall into three categories: ISAM file handlers, B-Tree handlers, and routines to handle variable-length records. Not nearly as complete as the previous products, these subroutines are supplied in source form with terse but complete documentation. Networking is not supported directly, but you could modify the source code to do any kind of networking you desired.

I used this set of subroutines in the two-file relational database that I tested c-tree against. I found that both c-tree and SoftFocus gave similar results. I did not test the variable-length features of either package.

The manual gave an excellent introduc-

tion to B-Tree theory and ISAM files. Spending the few dollars for this sturdy, well-thought-out, although simple, package is a good investment if you do not need the "heavy hitting" features of c-tree or Btrieve.

IN SUMMARY

These packages illustrate some of the products available for building a sophisticated database system. Before embarking on such a project, you should define your goals, the data, and the requirements. Purchase your software from a reliable source and try to get it on evaluation. If you have a serious application, you might need to buy several packages to evaluate their suitability for your needs.

There is no neat program to tie the data-

base file handler to the screen handler. Report generators that work with Btrieve and c-tree files are available, but I did not use or test these programs; rather, I wrote my own. All tests were performed on an Alloy PC+ System with a 640K IBM PC, an Alloy 85Mb Drive and a 60Mb tape backup. NTNIX, a proprietary Alloy Computer Products operating system, was used to operate two V-20 PC slaves and one 8088 slave PC under MS-DOS 3.1. With three disk partitions, it was possible to test the simultaneous operation of three programs running in parallel on each slave card. Lattice C, Version 3.1, and MASM 4.0 were used for interfacing and compiling. §

Andrew Bender, M. D., is a neurologist with 25 years programming experience.

Supports
Microsoft 4.0 Libraries

Complete C Programs in Half the Time, with *Instant-C*TM

Instant-C helps you create a working, well-tested program faster than any other interactive C development system. Much faster than traditional compilers, linkers, and debuggers.

Hard to believe? Here's how we do it. Because *Instant-C* is a high-performance interpreter there are **no compile or link delays**. Change your program, then test it immediately. No matter how large your program, the turnaround time is just seconds.

"Instant-C means instant gratification."—*PC Magazine*, **Editor's Choice** for best C interpreter, 10/29/85
"Time after time, the Instant-C prompt was starting back just barely after pressing Enter."—*PC Tech Journal*, 5/86

Source-level debugging saves your time.

- set any number of conditional breakpoints in your program;
- stop execution from keyboard;
- single-step by source statement;
- examine and change variables or code, and continue execution;
- execute any statement or function directly for instant testing;
- display source code back-traces;

New! source code animation;
New! monitor data changes;
New! full-screen and multi-screen support, even with non-standard graphics devices.

"The resulting debugging and testing capabilities are fantastic and the detailed trace/debug/display commands make it easy."—*The C Journal*, 5/85

Run-time checking stops your program as soon as errors occur, when bugs are easiest to understand and fix.

New! pointer references checked for reasonableness;
New! array indexes checked within declared bounds.

Not only does *Instant-C* help you quickly change, test, check and debug your code, but it runs your program faster than any other C interpreter: **Fifty to 500 times faster!** Fast enough for *real* programs, even fast enough for real-time programs.

"It is much faster than any of the other products mentioned and was the only one able to complete the standard SIEVE in a reasonable time. Clearly, this high speed allows much more complex problems to be attacked with Instant-C than with any of the other products discussed."—*Computer Language*, 2/86

Immediate feedback and more than 400 diagnostics makes *Instant-C* great for learning C. Full K&R and the ability to **New! link compiled object code and libraries** (Lattice and Microsoft) makes *Instant-C* compatible with your existing programs. **New!**

"When you get right down to it, I don't think there's a better way you could learn C."—*Programmer's Journal*, 3/85
"Clearly, Instant-C is the performance champion."—*PC Tech Journal*, 5/86

The bottom line for your business is increased productivity. The result for you is a job well done, and quickly.

"We sincerely feel that Instant-C can have a major positive impact on programmer productivity."—*Computer Language*, 2/85
"Instant-C by Rational Systems is a C programmer's dream."—*Micro/Systems Journal*, 3/86

Version 2 is available for MS-DOS and PC-DOS, and comes with a full 31 day **money back guarantee**. *Instant-C* is only \$495. Order today! Call or write for full information.

Rational Systems, Inc.

P.O. Box 480
Natick, MA 01760
(617) 653-6194

#1 C interpreter



The professional C development environment

Your C compiler creates great final code . . . but as a programming tool, it's too, too slow. With C-terp you can edit, debug, and run without the wait. Nothing, but nothing, is faster for developing professional C programs.

Choose the perfect C-terp companion for your C compiler

C-terp/Microsoft	C-terp/XENIX
C-terp/Lattice	C-terp/Aztec
C-terp/Mark Williams	C-terp/C86

Link in all your compiler's functions, your own functions, add-on libraries, assembly routines, and data objects. Get instant access to everything in the C-terp interactive environment.

Only C-terp offers all this and more

- Full K&R with common ANSI enhancements
- Source level interactive debugging
- Software paging for your big jobs
- Complete multi-module support
- Run-time pointer checking
- Unsurpassed reconfigurable screen editor
- Dual display and full graphics support
- Large model ■ Call-in

ORDER C-terp TODAY (specify compiler)

C-terp runs on IBM PC, AT or compatibles.

Price:

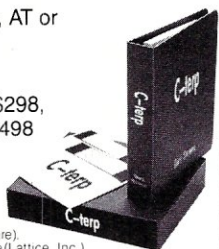
MS-DOS 2.x and up - \$298,

Xenix System V 286 - \$498

MC, VISA, COD

30-day money-back

GUARANTEE



Trademarks: C-terp (Gimpel Software), C86 (Computer Innovations), Lattice (Lattice, Inc.), Xenix, Microsoft, MS-DOS (Microsoft, Inc.), Aztec (Manx Software), Mark Williams (Mark Williams Company), IBM (International Business Machines, Inc.)

GIMPEL SOFTWARE

3207 Hogarth Lane, Collegeville, PA 19426

(215) 584-4261

He uses computers in his private neurology practice and designs hardware and software for neurophysiological data acquisition and research.

WHAT IS A B-Tree ISAM FILE?

You need know nothing about B-Tree indexing, ISAM files, or database design to use the subroutines discussed in this article. A complete treatment of the subject is beyond the available space in this issue. If you are only interested in getting a small application running, do not worry about the theory of database indexing. If you are working on a large database with multiple related files, I suggest you read the Comer article and Wiederhold book (see References). The way you organize your data will have important effects on the performance of different systems.

The term B-Tree has a nebulous beginning, but probably it was called B-Tree after R. Bayer, or after Boeing Aircraft's Scientific Research Lab, Bayer's employer. Later, the nature of the tree led users to say the B was for "balanced" or "bushy." A B-Tree index contains pairs. Each pair contains a key and a location in a file where the data associated with that key may be found. This organization scheme is similar to that of a book index in which you look up the topic you are interested in (the key) to find the page (the location of the associated information or data).

The associated data is usually a record of related items, and the file of these associated records is a relational database. Therefore, you can use the index to locate records in a relational database.

A certain number of pairs are located on the index page of the book, just as there are only a certain number of pairs on the index page of a B-Tree file. The maximum number of possible pairs of index keys and locations on an index page is fixed and is two times the "order" of the B-Tree. A single pointer, on each page, without any associated key, is used to direct the B-Tree search to an index page that contains lower-value keys than are present on the current page.

The locations in the B-Tree index may point to the actual data related to the index key, or they may point to more index pages. An indicator is used to signal the search routine as to whether the target data is file data or another index page. The B-Tree file-organization method for inserting and deleting records from the data file always leaves the B-Tree balanced, minimizing time required to search it.

Different methods of organization of a B-Tree file are given different names. A B+ -Tree file contains a separate index file with pointers into a sequential file containing the actual keys. A B* -Tree file's index nodes are two-thirds full instead of one-half full. This is called a "bushy" tree

by some.

The records of a relational database are generally organized in an indexed sequential scheme. The ISAM, or Indexed Sequential Access Method, was first designed to relate physical storage addresses to data records in a large magnetic drum data-file. The ISAM file contains one primary index made up of keys and pointers. The key's pointer points to the record containing that key. This organization lends itself particularly well to the B+ -Tree searching method where the keys are in the data-file and the index file is separate. When the ISAM file is read sequentially, records are returned to the user in the order of the primary index. The index part of the ISAM file is physically part of the data part of the ISAM file. Secondary indices can also be defined in an ISAM file, so that when references are made to the file and a secondary index is selected, the records are returned in the order of the secondary index. Although the physical storage addresses are not usually used to retrieve data records from a data file, the method is still known as ISAM, and it still is a useful method of organizing data in files that will not require later sorting.

REFERENCES

Comer, D., "The Ubiquitous B-Tree," *Computing Surveys*, Vol. 11, pp. 121-137, June 1979. This article most clearly describes B-Trees.

Knuth, D., *The Art of Computer Programming. Volume 1: Fundamental Algorithms*, Addison-Wesley Publishing Co., Reading, MA, 1968, and *Volume 3: Sorting and Searching*, Addison-Wesley Publishing Co., Reading, MA, 1973. A wealth of B-Tree background information.

Wiederhold, G., *Database Design*, McGraw-Hill, New York, NY, 1977. A high-level yet pragmatic discussion of ISAM files and database design. A newer, second edition is also available.

PRODUCT INFORMATION

c-tree

\$395 (single-user license)

FairCom, Inc.

2606 Johnson Dr.

Columbia, MO 65203

(314) 445-6833.

Btrieve

\$245 (nonnetworked version)

(extended support options are also available.)

SoftCraft, Inc.

P.O. Box 9802

Austin, TX 78766

(512) 346-8380

B-Tree ISAM File Handler

\$115

SoftFocus, Inc.

1343 Stanbury Dr.

Oakville, Ontario

Canada L6L 2J5

(416) 825-0903.

Up To Your Ears In Alligators?

If that sounds familiar, you need Write-Hand-Man™, the multi-function pop-up desktop organizer that works neatly with existing software for CP/M™ 2.2 and 3.0 systems. Write-Hand-Man eliminates that swamped feeling with tools that will get you organized. Write-Hand-Man comes with a 4-function, floating-point, 14 digit Calculator – Notepad – Two-week Appointment Book, File and Directory viewing – Phonebook with dialing – Cut and Paste – Key Redefinition – ASC II table. Even add your own applications.



Clear the swamp from your desktop.
Order Write-Hand-Man today. \$49.95

CA residents add 6.5% tax. Sorry, no credit cards or purchase orders.
Specify: 8" or which 5" format
CP/M 2.2 or 3.0 format

30 day guarantee

™ Write-Hand-Man – Poor Person Software

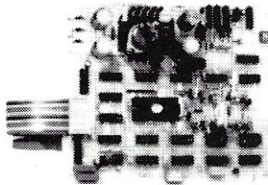
™ CP/M – Digital Research

**Poor Person
Software**

Dept. 204
3721 Starr King Circle
Palo Alto, CA 94306
(415) 493-3735

CP/M, MS-DOS EPROM PROGRAMMING SYSTEM

2708
2758
2716
2516
2532*
2564*
68764*
2816A



2732/A
2764/A
27128/A
27256
27512
27CXXX
2864A
8751*_{MICRO C}

– FAST ALGORITHM – STAND ALONE BOARD USES NO BACKPLANE SLOT
– NO PERSONALITY MODULES – INSTALL PROGRAM FOR SOFTWARE
– USES 24 VOLT XPMR FOR POWER – LARGE COMPREHENSIVE MANUAL
– PROGRAMS 26,25,21 & 12.5V E/EEPROMS – ALL SUPPLIES ON BOARD
– NOT A SERIAL PROGRAMMER, NO FILE DOWNLOADING REQUIRED
*SOCKET ADAPTER REQUIRED FOR DEVICES MARKED WITH ASTERISK(S) (ADAPTER DIAGRAMS INCLUDED)

PARALLEL PRINTER INTERFACE

CONNECTS TO ANY PARALLEL PRINTER INTERFACE – USES 8 OUTPUT DATA BITS AND ONE BIT FOR DATA INPUT (PRINTER BUSY LINE)

CONTROL PROGRAM COMMANDS

– PROGRAM EPROM(S) FROM DISK – SAVE EPROM(S) TO DISK
– READ DISK FILE INTO RAM – PROGRAM EPROM(S) FROM RAM
– READ EPROM(S) INTO RAM – COMPARE EPROM WITH RAM
– VERIFY EPROM IS ERASED – COPY EPROM
– DISPLAY/MODIFY RAM – (MONITOR MODE) WITH 14 SUB COMMANDS
FILL–DUMP–XFER–EXAM–MODIFY–BIAS–PRGM–VERIFY–CKSUM, ETC)

ASSEMBLED & TESTED UNIT WITH COMPLETE DOCUMENTATION AND SOFTWARE ON DISKETTES ----> \$199
PARTS KIT WITH SOFTWARE AND DOC->\$179 PCB SOFTWARE & DOC->\$69
SOFTWARE ON 8 & 5 1/4 DISK FOR KAYPRO, AMPRO, IBM & OTHER PMTS

TO ORDER SEND CHECK, MONEY ORDER, WRITE OR CALL
ANDRATECH
P.O. BOX 222
MILFORD, OHIO 45150
(513) 752-7218

CALL OR WRITE FOR MORE INFORMATION --- ADD \$4.00 FOR SHIPPING
OHIO RES. ADD 5% TAX --- VISA/MC ACCEPTED --- \$3.00 FOR COD

TURBO PROFESSIONAL™

SERVICE INTERRUPTS
No assembly required

RESIDENT PROGRAMS
Easy, pop-up routines

EXECUTIVE PROGRAMS
Run ANY DOS program

DISK SECTOR I/O
Lowest level access

FAST TEXT WINDOWS
Virtual windowing system

KEYBOARD MACROS
Simple, powerful

LOTS OF EXAMPLES
21+ full example programs

MUCH MORE . . .
Over 140 routines in all

"If you never thought Turbo Pascal was a systems programming language, you've never seen Turbo Professional."

Darryl Rubin
Computer Language

For programs that move with technology—Turbo Professional—a truly professional library of subroutines.

150 page reference manual.
Full source—many example programs.
No royalties charged for applications.

Requires IBM compatible,
DOS version 2.0 or greater,
Turbo Pascal 2.0 or greater.

Sunnyhill Software not affiliated with Borland International.
Turbo Professional, trademark of Sunnyhill Software
Turbo Pascal, registered trademark of Borland International

Dealer Inquiries Invited.

\$69⁹⁵ plus \$5.00 S&H

Washington residents add 7.9%
International orders add \$5.00
VISA and Mastercard accepted.

To order Toll-Free
call 1-800-367-0651

**Sunny Hill
Software**

PO. Box 55278
Seattle, WA 98155
(206) 367-0650 M-F, 8-6 PDT



TASKVIEW™

EASY TO USE

OPTIONAL MENUS

TIME SLICING

VIRTUAL MEMORY

EMS SUPPORT

EGA SUPPORT

PRIORITY DRIVEN

CUT AND PASTE

Compatible, efficient DOS multi-tasking.

We designed Taskview with efficiency in mind. During normal operation, TASKVIEW hides behind DOS, providing you with control of up to 10 concurrent or non-concurrent programs. Just the touch of a key instantly switches a program to the foreground. Included desktop utilities let you cut and paste from program to program. Simple to use and reasonably priced, no well equipped PC user should be without it.

Requires: PC/AT/Jr compatible,
DOS 2.0-3.1, 256K RAM, 1
Floppy drive.

Sunnyhill Software not affiliated with Borland International
Taskview trademark of Sunnyhill Software
Sidekick registered trademark of Borland International

Dealer Inquiries Invited.

\$79⁹⁵ plus \$3.00 S&H

Washington residents add 7.9%
International orders add \$5.00
VISA and Mastercard accepted.

To order Toll-Free
call 1-800-367-0651

**Sunny Hill
Software**

PO. Box 55278
Seattle, WA 98155
(206) 367-0650 M-F, 8-6 PDT



Intercepting XIOS Calls

by Alex K.H. Soya

Certain applications, under Concurrent CP/M (CCP/M), may need to intercept XIOS calls in order to reroute XIOS functions or to implement device-drivers for hardware not normally supported by the XIOS. The days of modifying your BIOS or XIOS device driver code, to perform the above, are unfortunately over. Few OEMs include the source code and object modules necessary to generate a new operating-system image with their systems software. If the device-driver source code is available to the end user, it is usually at a prohibitive cost.

In this short article, I wish to propose a method to intercept XIOS calls. By intercepting XIOS calls it is possible to dynamically install device drivers or rerouting procedures to perform what in the past could only be done by altering the XIOS. The method I use is generic to any CCP/M system and is thus not dependent on a particular implementation.

A DIRTY TRICK

Before going any further, let me say that I believe that intercepting XIOS calls, or even making calls to the XIOS directly, is an extremely bad habit. At all times programmers should avoid playing with XIOS calls and only use such dirty tricks as a last resort. By making XIOS calls one cannot guarantee that the program will function correctly with another version of CCP/M, or even another manufacturer's particular implementation. However, in certain situations, there is just no other solution but to make calls to the XIOS, or even worse, intercept calls to the XIOS and perform or emulate XIOS functions in an application program.

Now that I have expressed my feelings about intercepting calls to the XIOS, let me present my method: reroute a XIOS call to a handler in an application program and let that handler perform a particular XIOS function rather than the XIOS.

NOT THIS WAY

At first you may say easy: just obtain the current vector to the XIOS entry point in the SYSDAT page, save it, overlay the vector with a value pointing to a routine in the

Install Device Drivers or Special XIOS Function Handlers Dynamically Within Application Programs Under Concurrent CP/M

application program, and it's done. As soon as the application program has completed execution, restore the old vector and all is back to normal.

Unfortunately several problems can occur using this obvious method. If the program runs more than once, or another program using the same method is executed, it will perform the same action. That is OK, until the original program terminates and restores the XIOS vector. Now the second program no longer intercepts the XIOS calls and thus will no longer function. Even more catastrophic things can happen.

The problem of multiple programs intercepting XIOS calls is not the only one. Some OEM implementations of CCP/M make use of the XIOS ENTRY vector in the SYSDAT page to locate certain data structures within the XIOS. A typical example would be the FORMAT program under

CompuPro's CCP/M-3.1D. The FORMAT program obtains drive parameters from the XIOS depending on the type of disk selected. As the XIOS-ENTRY vector points to the handler in the application program, the FORMAT utility is unable to obtain the required structures. Another example would be the Gifford disk cache buffering program used for their RAM drives. Overlaying the XIOS ENTRY jump vector in this case causes the system to crash with 'Wild Interrupt' messages.

MY METHOD

I propose the following method be used for intercepting XIOS calls by application programs: create a queue for the sole purpose of managing a buffer for redirection vectors. As the buffer is in the same segment as the XIOS code segment, the JMP address at XIOS ENTRY (0C03h in SYSDAT page) can be modified to jump to the buffer. In the buffer, a JMPF instruction would cause the XIOS call to be redirected to the application program. This avoids the problem of overlaying the XIOS vector in SYSDAT, and thus lets programs such as FORMAT perform correctly. In addition, each process that intercepts XIOS calls would be inserted into a doubly-linked list. When a process exits, it would delete its entry in the linked list and thus leave the other processes intercepting XIOS calls undisturbed. The last process to terminate deletes the queue from the system and restores the original JMP offset at the XIOS entry point.

The above scheme will only work reliably if there is no chance of the process being aborted from an external program such as ABORT (or STOP under CCP/M 4.1). Therefore the process must set itself to be non-abortable and every effort must be made to handle all possible errors from

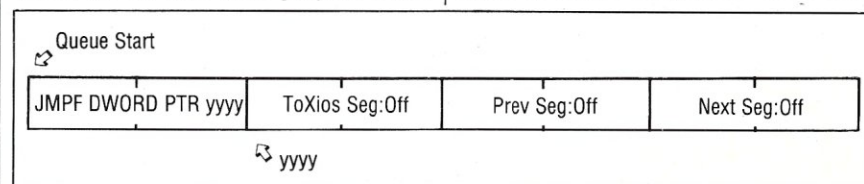


Figure 1. Fields assigned to the queue buffer.

BUSINESS COMPUTERS & UPGRADES

Compu Pro/Viasyn-S100

CPU 286-8 3CY	542	SYS.SPT.1	319	RAM 23 128K	262
CPU 286-8 2CY	674	INTF 4	333	RAM 24 1MEG	2249
CPU 86 -10	295	DISK 1A	445	MDRIVE/H.5M	CALL!!
CPUZ-8	209	RAM 22 256K	446	MDRIVE/H.2M	CALL!!
SPUZ-256	289	RAM 23 64K	187	INTF 3-8	445
				DISK 3	520

LOMAS MEGARAM 2-4 MEGABYTES
MODIFIED FOR USE WITH DISK 3 SYSTEMS

CALL!!

HARD DISKS FOR S-100 & AT SYSTEMS | **MAXTOR** 1085 - 85MB - \$1889 | 1140 - 140MB - \$2669 | **NEC** 4126 - 26MB - \$395 | 5146 - 51MB - \$649

60 MB PC TAPE BACKUP INTERNAL \$795 / EXTERNAL \$995

We can custom configure any system to meet your needs. Installation & training available in some areas.

WATCH FOR OUR 60 MEG S100 TAPE BACKUP...

SOFTWARE AND OTHER MODELS/CAPACITIES AVAIL.

UNIX SYSTEM-V MICRO

- 80286 @ 10 MHZ
- 4 MB RAM
- 72 MB HARD DISK
- 60 MB TAPE
- 8 SERIAL PORTS
- GRAPHICS CONSOLE

STARTING AT

\$8895

LINK 220 terminal

OT-700 700 CPS Printer \$499

MULTITECH

MT-700PC 8 MHZ! \$1195

640K • 2 FLOPPY • RS 232

PARALLEL • MONO GRAPHICS

W/20 MB HARD DISK \$1595



DIGITAL DECISIONS SYSTEMS

229 Lakeview Ave. • Clifton, NJ 07011

1-800-862-6233 Wait for tone, dial Ext. 0014762 **(201) 772-6749**

Call/Write for our **FREE 140 page Color Catalog**

Announcing a New Product...

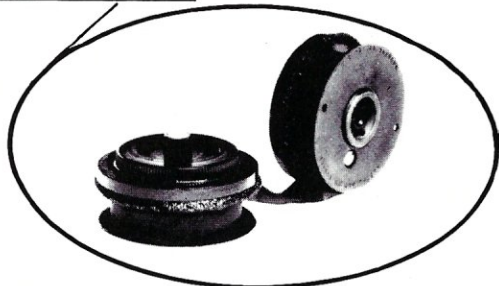
OPEN SPOOL PRINTER RIBBON

THAT RE-INKS ITSELF

FOR USERS OF

Okidata and Other Open Spool Printers Including Teletype

MONEY BACK GUARANTEE



OUR SELF-INKING DEVICE WILL LAST 15 TIMES THE RIBBON YOU ARE NOW USING.

- 15 TIMES LESS RIBBON CHANGES
- NO MORE MESSY HANDS

CONTROLLED PRINTOUT DEVICES, INC.

P.O. BOX 869, BALDWIN ROAD
ARDEN, NC 28704

(704) 684-9044 • TELEX: (FILMON-AREN) 577454

CONTACT US BY MAIL, PHONE OR TELEX AND WE WILL FORWARD YOU A BROCHURE

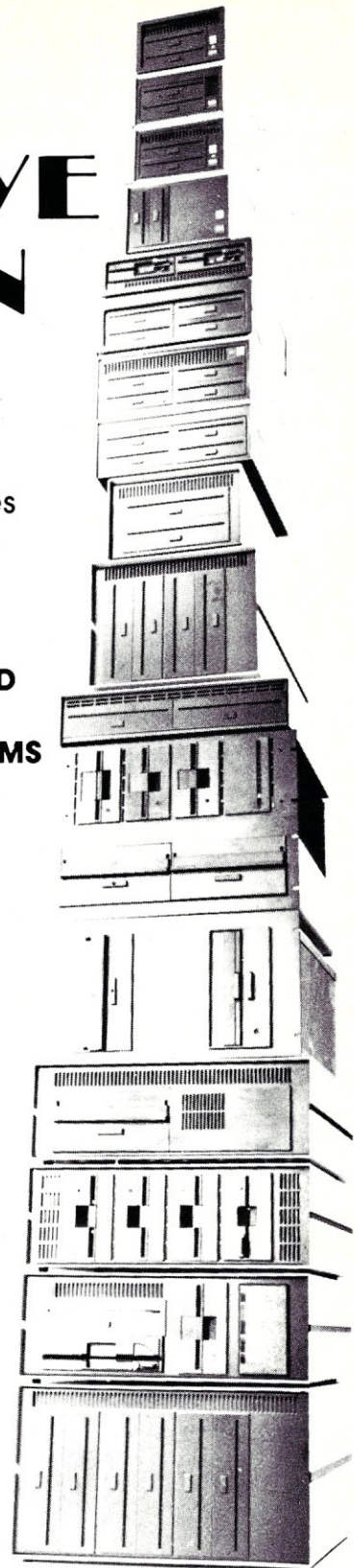
DRIVE INN

Enclosure & power supplies for

FLOPPY, WINCHESTER, TAPE DRIVES, SINGLE BOARD COMPUTERS & S-100 SYSTEMS

8 inch
5 inch
3 inch

Call or write for free catalogs & application assistance



NEGRAND

RESEARCH CORPORATION

8620 Roosevelt Ave. • Visalia, CA 93291

209/651-1203

We accept BankAmericard/Visa and MasterCharge

within the program. Before the process terminates, it will delete itself from the linked list.

THE QUEUE STRUCTURE

Figure 1 shows the fields assigned to the queue buffer. The buffer is located within the SYSDAT segment of CCP/M. Normally queues are used to exchange messages between processes. In this case, the only reason for the queue is to obtain a portion of RAM located in the SYSDAT segment. As the JMP instruction at the XIOS entry point is an intra-segment jump, only locations within the same segment as the XIOS can be reached. Digital Research has not provided for any locations within the SYSDAT segment to be used by application programs. As long as no process writes to the queue, this method is safe.

Notice the JMPF instruction at the start of the queue buffer, it causes the XIOS call to be redirected to the address pointed to by the ToXios field. The fields PREV and NEXT in the queue buffer are the root pointers to similar structures in the application program's code segment (see Figure 2). Each program which needs to intercept XIOS calls contains the same fields as the queue buffer. The TOXIOS field points to the next processes XIOS intercept routine. If the field is contained in the last, or only, process in the list, then TOXIOS points to the location which the JMP instruction at the XIOS entry point used to jump to. Thus, TOXIOS is the link from the XIOS entry point to the resident device drivers of CCP/M. The PREV field points to a process previous to the current process and can thus be used to backtrack the linked list when deleting a process from the XIOS intercept list. The PREV field in the queue buffer will always contain a null value as the queue buffer is the root of the list. As expected, the NEXT field points to the next process in the XIOS intercept list. The last process in the list has a null value in the NEXT field.

PSEUDO CODE

As the routines to perform the actual insertion and deletion from the linked list are rather complicated, I include the pseudo code listings I used to write the actual assembly language routines. Listing 1 shows how to insert a process into the XIOS intercept list. The first process to install itself must create the queue and locate the queue buffer in the SYSDAT segment. Also, the operand of the JMP instruction at the XIOS entry point must be modified to point to the queue buffer. As the 8086 does not use absolute addresses in the operand, the relative offset to the buffer must be computed. Note that all interrupts must be disabled while the pointers are modified to prevent any XIOS calls from being made during this procedure. If interrupts are not disabled, a XIOS call could be made while the pointers are only partially filled and thus

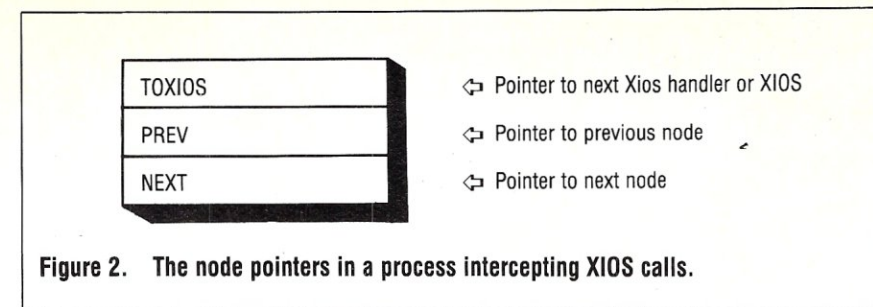


Figure 2. The node pointers in a process intercepting XIOS calls.

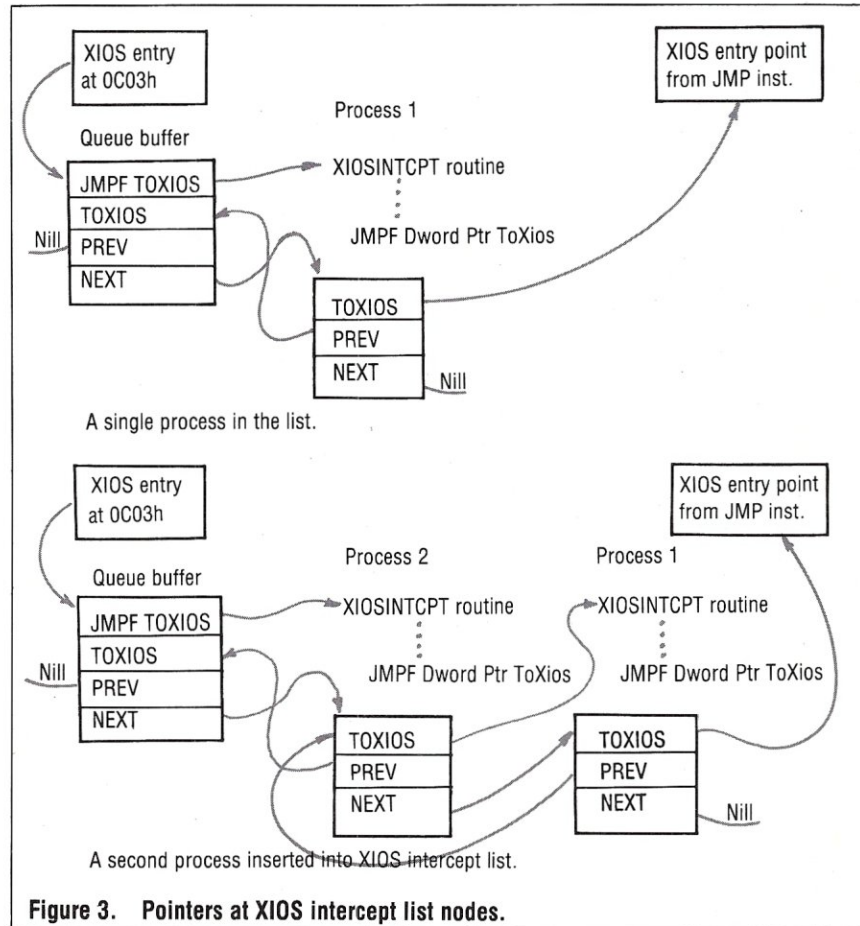


Figure 3. Pointers at XIOS intercept list nodes.

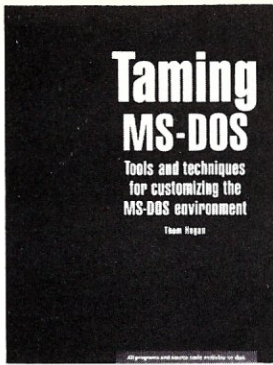
cause catastrophic results. Listing 2 demonstrates how to delete a node from the list. Note how much simpler it is to delete an item from a linked list than it is to insert an item. Refer to Figure 3 for a graphic representation of the various pointers for single and multiple nodes in the list.

THE REAL THING

And now to the real thing. Listing 3 is the assembly language version of the above mentioned pseudo code. These routines should be included by your application program (if you use RASM86 you can just link the routines with your program). Note that the routine, XiosIntcpt, is your routine which performs the actual Xios emulation and is thus not shown in Listing 3. The current implementation requires that the process has a unique code segment if multi-

ple copies of the same task are run. In the near future, I hope to modify the routines to allow shared code segments. Contact me if you require your program to use shared code segments. Any future versions of the routines in Listing 3 that I develop will be compatible with the present routines.

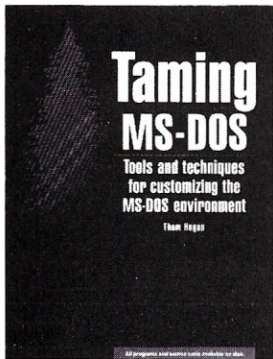
Listing 4 presents an example program, named CONNECT, making use of XIOS intercept routines. The program is rather primitive and requires a lot of refinement before being used in critical applications. It is merely here to demonstrate how to use the intercept routines from an application. The purpose of CONNECT is to make a logical connection between two virtual consoles allowing the operator to monitor and control the activities on another terminal. The destination console must be in dynamic mode or in the foreground for CONNECT to work



Taming MS-DOS

by Thom Hogan

Learn how to make DOS work for YOU! **Taming MS-DOS** will take you beyond the basics, picking up where your DOS manual leaves off. This advanced users guide will show you how to extend the power of DOS so you can work more accurately and efficiently.



- Learn to maximize your batch files with routines using redirection, filters and pipes. You'll find routines that prevent accidental reformatting of your hard disk, redefine function keys and locate files within subdirectories. You'll learn to implement a DOS help system with help text files, a menu system that interprets keyboard input, and a routine for quick redefinition of function keys.

- You'll learn how to customize CONFIG.SYS to maximize the performance of your system and how to use ANSI.SYS to tailor your system prompt and monitor attributes to fit your needs.

- **Taming MS-DOS** includes nearly 50 ready-to-use programs that increase DOS's functionality. Now you can easily rename directories and disk volumes, change file attributes, check available RAM and disk memory, display a memory-resident clock, and assign DOS commands to ALT keys.

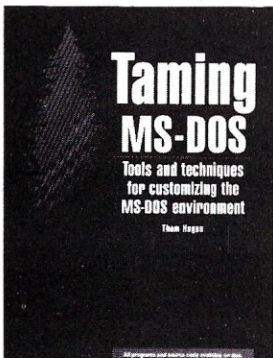
- Quick reference charts provide easy access to batch command syntax, CONFIG.SYS syntax and ANSI.SYS command strings.

Taming MS-DOS shows you how to alter programs and customize DOS to fit YOUR needs, saving you time and frustration every time you use it!

The programs, including batch files and DOS enhancements, are also available on disk along with full source code.

Taming MS-DOS
Taming MS-DOS with disk

Item #060 \$19.95
Item #061 \$34.95



To order: return this coupon.

M&T Publishing, 501 Galveston Dr., Redwood City, CA 94063

YES! I want to make DOS work for me. Please send me:
___ copies of **Taming MS-DOS** at \$19.95 each
___ copies of **Taming MS-DOS** with disk at \$34.95 per set

Subtotal _____
CA residents add appropriate sales tax ___ % _____
Add \$1.75 per item for shipping _____
TOTAL _____

Check enclosed
Charge my VISA M/C Amer. Exp.
Card # _____ Exp. Date _____
Signature _____
Name _____
Address _____
City _____ State _____ Zip _____

correctly. The routine, XIOSINTCPT, in Listing 4 is the XIOS function handler. It will intercept any console output calls for the console of interest and store them in a local circular queue. The call is then forwarded to the actual XIOS for output to the destination console. The CONNECT task continuously reads the circular queue and forwards any characters found to the local console. If a key on the local console is pressed the character will be sent to the Virtual Input queue (VINQ) for the target console. This example will work on any version of CCP/M. The only critical part is the flag number used. Make sure you set this flag to a value that does not conflict with your system's implementation.

CONNECT demonstrates how to re-route output from one console to another console. I have also used the above XIOS intercept routines to write a device driver for a special I/O board which was not supported by the XIOS as implemented by the manufacturer of my system.

I would suggest that you use the above routines anytime you need to intercept XIOS calls from an application program. I feel that this method is a relatively safe procedure and should work fine as long as all programs use this very same method. I am still open to suggestions for any improvements and would welcome such. The next step would be to develop a generic XIOSINTCPT routine allowing shared code segments to be used by application programs. I have made these routines available on my Concurrent RCPM system at (305) 727-0331 for your convenience. And as a final comment let me stress again that intercepting XIOS calls should be avoided if at all possible. (μ)

Alex Soya has been involved with CP/M for over 8 years. He is currently a computer science senior at Florida Institute of Technology. His interests include Operating Systems, Data Communications, System Utilities, Real Time and Scientific Programming.

Listing 1

Inserting a process into XIOS intercept list.

```

Procedure Hook_in_xios_list
  [ Insert Process into Xios intercept list ]
Begin
  If XiosVect Queue does NOT exist then
    Begin
      Make Queue
      Disable Interrupts
      Locate XiosVect Queue buffer
      Que.Buffer Start := JMPF DS:DWORD Ptr (Buffer_Start+4)
      ToXios SEG       := XIOS SEGMENT
      ToXios^OFF      := address([XIOS SEG:ENTRY+1])
      XIOS SEG:ENTRY+1 := displacement(Que.Start)
      Que.Prev        := NIL
      Next            := NIL
    end
  else
    Begin
      Disable Interrupts
      Locate XiosVect Queue buffer
      ToXios       := [Que.ToXios]
      [Que.Next].Prev := ToXios
      Next         := [Que.Next]
    end
  Que.ToXios := XiosIntcpt
  Que.Next  := ToXios
  Prev     := Que.ToXios
  restore interrupt state
end

```

Listing 2

Deleting a process from the XIOS intercept list.

```

Procedure UnHook_From_Xios_List
  [ Delete Process from Xios intercept list ]
Begin
  Disable Interrupts
  Prev.ToXios := [ToXios]
  Prev.Next  := [Next]
  If (Next = NIL) and ([Prev].Prev = NIL) then
    Begin
      Restore XIOS entry JMP address
      Restore Interrupts
      Delete_Queue
    end
  else
    If ([Next] <> NIL) then
      begin
        [Next].Prev := [Prev]
      end
    Restore Interrupts
  end
end

```

Listing 3

;XINICPT.A86 - Xios Intercept Routines

```

;-----
; 8-20-85 Initial Version. Alex K.H. Soya
; 1.0
;-----
;This set of routines allows process to insert & remove itself from
;linked list structure allowing an application program to intercept
;Xios calls & emulate such in a local routine. Also allows special
;device drivers to be installed in a dynamic manner.
;
;Following routines allow multiple such processes to insert & delete
;themselves from list without interfering with each other.
;
;As method of intercepting XIOS calls is very sensitive to changes,
;causing incompatibility between programs if not handled in uniform
;manner author publishes these routines under following conditions:
;
;Routines should also work with MPM-86, but were only tested under
;Concurrent CPM 3.1 and Concurrent PC-DOS 4.1.
;
; ----- C O P Y R I G H T -----
;
;Following routines are COPYRIGHT (C) 1985 by Alex K.H. Soya
;
;All commercial rights to these routines remain with author unless
;agreed differently in WRITING by author.
;

```


;You may use these routines in public domain application programs
;after notifying author of intentions to do so. If your program
;is for commercial purposes (rather than public domain) you will
;require author's consent in writing before selling programs
;using these routines.

;Above is so that author may be able to notify you in case of any
;changes. A lot of planning and work has gone into method used to
;to intercept XIOS calls.

; Alex K.H. Sova
; Computerse [70406,1452]
; Concurrent RCMP COLIASH (305) 727-0331
;
; P.O. Box 510121
; Melbourne Beach, FL 32951
; U.S.A.
;
; Voice Phone (305) 724-9921
;

;You can put these in your program head with your other equates
;if you use this as an INCLUDE file under ASM86

```
WRITESTR equ 9 ;Write string to console
Q_MAKE equ 134 ;Make a System Queue
Q_OPEN equ 135 ;Open a system queue
Q_DELETE equ 136 ;Delete a Queue
SYSYSDAT equ 154 ;Return SYSYSDAT address
False equ 0
true equ not false
RASM86 equ true ;Set to TRUE if using RASM 86
;and declare following public
;labels as external in your code segment
```

If RASM86

```
public Hook_In_Xios_List ;link into XIOS intercept list
public UnHook_From_Xios_List ;delete from xios intercept list
public ToXios_Off ;your MINICPT routine uses this
;vector to forward calls to XIOS
public XIOS ;declare this as extrn XIOS:DWORD in
;your application program & call
;XIOS by a CALLF XIOS instruction if
;you need to call xios directly.
extrn XiosIntcpt:near ;This is application programs Xios
;function handler. It should handle
;function you want to intercept and
;do JMPF DWORD PTR ToXios Off to get
;to next node in list and thus
;actual xios.
```

ENDIF

```
; locate_XiosVect_Queue_Buffer
;
; Entry -> bx = offset SYSYSDAT page
; es = segment SYSYSDAT page
;
; Return <- Bx = offset of XiosVect Queue buffer
; es = Segment of XiosVect Queue buffer
; ax = 0 = found buffer
; ax <> 0 = Buffer not found
;
```

```
QLR equ 74h ; Queue List Root
locate_XiosVect_Queue_Buffer:
add bx,QLR ; point to Root of QUEUE list
Search_For_buf:
mov bx,es:[bx] ; Get next Queue D offset
cmp bx,0
jnz got_one_more ; If queue not found then
or al,0Ffh ; notify caller
ret ; return
got_one_more:
mov di,bx ; get Qname offset
add di,6
mov si,offset XQNAM ; prepare for comparison
mov cx,8 ; 8 characters in name
cld ; incrementing comparison
repe cmpsb
jne search_for_buf ; until queue found
mov bx,es:[bx] ; get Buffer offset
xor ax,ax ; buffer was found
ret
```

```
; Hook_in_Xios_List
; Overlay XIOS jump vector to point to a
; redirection vector in a Queue Buffer.
; Initialize the redirection vector in the
; queue to contain address of a local xios
; function intercept handler.
```

; to make code more readable:

```
Que_ToXios_Off equ es:Word Ptr [DI]
Que_ToXios_Seg equ es:Word Ptr 2[di]
Que_Prev_Off equ es:Word Ptr 4[di]
Que_Prev_Seg equ es:Word Ptr 6[di]
Que_Next_Off equ es:Word Ptr 8[di]
Que_Next_Seg equ es:Word Ptr 10[di]
P_ToXios_Off equ es:Word Ptr [DI]
P_ToXios_Seg equ es:Word Ptr 2[di]
F_Prev_Off equ es:Word Ptr 4[di]
F_Prev_Seg equ es:Word Ptr 6[di]
F_Next_Off equ es:Word Ptr 8[di]
F_Next_Seg equ es:Word Ptr 10[di]
```

```
Hook_In_Xios_List:
mov dx,offset XIOSQPB ; try to open XIOS intercept
mov cl,Q_OPEN ; queue
int 224
or dx,ax
jnz Make_Queue ; If queue NOT exist then
jmp queue_already_exists
Make_Queue:
mov dx,offset XIOSQPB ;make queue
mov cl,Q_MAKE
int 224
or ax,ax
jz xios_q_made ;if que make NOT success then
mov dx,offset BAD_XQ_MAKE
mov cl,C_WRITESTR ;tell operator
int 224
or al,0Ffh ;and return error code and
ret ;return to caller
xios_q_made:
mov dx,offset XIOSQPB ;Open Queue so we can delete it
mov cl,Q_OPEN
int 224
```

```
mov cl,S_SYSYSDAT ;obtain SYSYSDAT segment
int 224
pushf ;disable interrupts
cli
push ax
call locate_XiosVect_Queue_Buffer;Locate Xios_Vect Quee buffer
or ax,ax
pop ax
jz Que_Buffer_located ;If Queue buffer not found then
Que_Buffer_not_located:
popf ;restore interrupt state
mov dx,offset No_Buffer_msg ;send error message
mov cl,C_WRITESTR
int 224
mov dx,offset XIOSQPB ;Delete Queue just in case
mov cl,Q_DELETE
int 224
or ax,0Ffh ;Set Error Flag
ret ;return to caller
```

```
Que_Buffer_located:
mov es:[bx],2effh ;Que.Start := JMPF DWORD PTR
mov dx,bx 1 add dx,4
mov es:2[bx],dx ;(Que.Start+4)
mov di,bx
mov bx,ax
add bx,28h
mov dx,es:2[bx] ;Get Xios Segment
mov ToXios_Seg,dx ;ToXios SEG := XIOS SEG
mov bx,es:[bx] ;Get XIOS Entry offset
mov dx,es:1[bx] ;Get JMP displacement
add dx,bx 1 add dx,3 ;make address
mov ToXios_Off,dx ;ToXios OFF:=add([XIOS SEG:ENTRY+1])
mov ax,di ;get address of Que.Start
sub ax,bx 1 sub ax,3 ;compute displacement
mov es:1[bx],ax ;XIOS SEG:ENTRY+1 := Disp(Que.Start)
add di,4 ;Point to Que.ToXios
mov Que_Prev_Off,0 ;Que.Prev := NULL
mov Que_Prev_Seg,0
mov next_off,0 ;Next := NULL
mov next_seg,0
jmp common_insert
```

```
Queue_already_exists: ;else (XiosVect Quee exists)
mov cl,S_SYSYSDAT ;locate Sysdat page
int 224
pushf 1 cli ;clear interrupts
call locate_XiosVect_Queue_Buffer ;Locate Queue Buffer
or ax,ax
jnz que_buffer_not_located ;if Buffer not found, abort
mov di,ex 1 add di,4 ;point to Que.ToXios
mov dx,Que_ToXios_Off ;ToXios := [Que.ToXios]
mov ToXios_Off,dx
mov dx,Que_ToXios_Seg
mov ToXios_Seg,dx
push es
mov dx,di
les di,Dword Ptr Que_Next_Off ;[Que.Next].Prev := ToXios
mov P_Prev_Off,offset ToXios_Off
mov F_Prev_Seg,CS
mov di,dx
pop es
mov dx,que_Next_Off ;Next := [Que.Next]
mov next_off,dx
mov dx,que_Next_Seg
mov next_seg,dx
```

```
common_insert: ; end
mov Que_ToXios_Off,offset XiosIntcpt ; Que.ToXios := XiosIntcpt
mov Que_ToXios_Seg,CS ; Que.ToXios := XiosIntcpt
mov Que_Next_Off,offset ToXios_Off ; Que.Next := ToXios
mov Que_Next_Seg,CS ; prev := que.ToXios
mov prev_off,di
mov prev_seg,es
popf ;Restore Interrupts
call locate_real_xios ;keep address of real XIOS, for
;those who need to call XIOS
;functions directly.
xor ax,ax ;No error
ret ;end
```

```
; The following must be contiguous in order given
;NOTE* This current version does NOT allow your process to run
; as SHARED CODE if execute more than once.
ToXios_Off dw 0 ; or real XIOS
ToXios_Seg dw 0 ; Previous node
Prev_Off dw 0 ; Next Node
Prev_Seg dw 0
Next_Off dw 0 ; Next Node
Next_Seg dw 0
XIOS rd 1 ; vector to actual xios entry
```

```
; UnHook_From_Xios_List
; Removes process from Xios Intercept list
;
; Entry -> None
; Return <- None
;
```

```
UnHook_From_Xios_List:
pushf 1 cli ; Disable Interrupts
les di,Dword Ptr prev_Off
mov dx,ToXios_Off ; Prev.ToXios := [ToXios]
mov P_ToXios_Off,dx
mov dx,ToXios_Seg
mov P_ToXios_Seg,dx
mov ax,Next_Off ; Prev.Next := [Next]
mov P_Next_Off,ax
mov dx,Next_Seg
mov P_Next_Seg,dx
or ax,dx ; If ([Next]=Null)
or ax,P_Prev_Off ; and ([Prev].Prev = Null) then
or ax,F_Prev_Seg
or ax,dx
jnz Not_Last_Item ; Restore XIOS entry JMP address
mov ax,ToXios_Off ; Restore interrupts
mov bx,28h ; Delete queue
mov bx,es:[bx] ; make displacement
sub ax,bx 1 sub ax,3 ; and restore XIOS ENTRY JMP disp.
mov es:1[bx],ax
mov cl,Q_DELETE
mov dx,offset XIOSQPB
popf ; Restore interrupts
int 224 ; Delete queue
ret
```

```
Not_Last_Item: ; else
mov ax,Next_Off ; If ([Next]<> NULL) then
or ax,Next_Seg
jz Not_Last_Node
les di,Dword Ptr Next_Off
mov dx,Prev_Off ; [Next].Prev:=[Prev]
```



```

mov P Prev Off,dx
mov dx,Prev Seg
mov P Prev_Seg,dx
Not_Last Node:
  popf
  ret
; Restore Interrupts

; locate_real_xios - This routine will locate the actual XIOS
; entry point and store a vector pointing there
; in the double word XIOS. An application
; program may then make calls to the XIOS
; without intercepting it own calls.
;
; Entry -> ToXios established
; Return -> XIOS filled in with vector to actual XIOS
;
locate_real_xios:
  mov di,offset ToXios_Off ; point to our node
  mov ax,cs ! mov es,ax
  pushf ! cli ; make sure nobody moves us
While_Not_Nill:
  mov ax,P Next_off ; While [Next] <> Nill do
  or ax,P Next_Seg
  jz found_terminal Node
  les di,Dword Ptr P Next_Off ; Point to next node
  jmps While_Not_Nill ; endwhile
found_terminal_node:
  les di,dword ptr P ToXios_Off
  mov Word Ptr XIOS,di ; XIOS := [ToXios]
  mov Word Ptr XIOS+2,es
  popf
  ret
DSEG ; end proc locate_real_xios
; If you use ASM86 and have not yet defined DSEG
; offset put your ORG here
XIOSQPB dw 0,0 ; The Queue Parameter Block
dw 0
dw offset dum_buf
XQNAM db 'XiosVect'
XIOSQDD dw 0,0,0 ; Queue Descriptor
db 'XiosVect'
dw 1
dw 18
dw 0,0,0,0
dw offset dum_buf
dum_buf rb 18
BAD_XQ_MAKE db '+++ Unable to open XiosVect Queue +++','$'
No_Buffer_Msg db '+++ Could not locate XiosVect Queue buffer +++','$'
; Now its time for a beer.

```

LISTING 4

;CNCT11.A86 - Connect current Virtual console to another virtual
; console. For Concurrent CPM only.
;CONNECT Vers 1.1 - modified to make use of XINTCPT routines
; and to assemble with RASM86
; 08/20/85
; Renamed to fit in CompuServe Directories

; To Assemble: SET CPCDOS41 equate to whatever you have
; abd then assemble as:

```

RASM86 CNCT11 SPZSZ
LINK86 CONNECT=CNCT11,XINTCPT

```

After executing type a Ctrl-Z to get out
of CONNECT

Original Version :
Author : Alex Soya - Nov. 1984

CompuServe [70406,1452]
Concurrent RCPM Goliath (305) 727-0331 300/1200/2400 bps
P.O. Box 510121, Melbourne Beach, FL. 32951

```

false equ 0
true equ not false

CPCDOS41 equ FALSE ; True if for Concurrent PCDOS 4.1
; else for Concurrent CPM 3.1

; CPM function calls
P_TERMCPM equ 0 ; Terminate & release all resources
C_READ equ 1 ; Read character from Default Console
C_WRITE equ 2 ; Write character to Default Console
C_RAWIO equ 6 ; Direct Console I/O with Default Console
C_WRITESTR equ 9 ; Print ASCII string to default console
C_STAT equ 11 ; Obtain Status of Default Console
DEV_WAITFLAG equ 132 ; Wait for System Flag
DEV_SEITFLAG equ 133 ; Set system Flag
Q_OPEN equ 135 ; Open A system queue
Q_WRITE equ 139 ; Write message to System Queue
S_SYSDAT equ 154 ; Return address of System Data Segment
FPDADR equ 156 ; Return address of process descriptor
CF equ 13
IF equ 10
CNTRL_Z equ 'Z'-40h
OPTION equ 05dh ; Command line Option can be found here.
BASEFLAG equ 80 ; BaseFlag + console # used for qbuffer
; YOU MAY NEED TO CHANGE THIS FLAG
; NUMBER TO ONE NOT USED BY YOUR XIOS
; # of bytes in output Circular Queue buffer

QDEPTH equ 40

; Sysdat Offsets:
;
;
SUPENR equ 0h ; Supervisor entry offset
XIOSSYS equ 28h ; Offset to Xios vector
CSEG
extrn Hook_In_Xios_List:near ; Xios intercept list handler
extrn UnHook_From_Xios_List:near ; routines
extrn ToXios_Off:Word ; Pointer structure
public xiosIntcpt ; The actual intercept routine

connect:
  pushf ! pop bx ; No interrupts during stack switches
  cli
  mov ax,ds

```

```

mov ax,ds
mov ss,ax
mov sp,offset stack
push bx ! popf ; Now we have our own stack.
; Keep the Data Segment here
mov cs:DSKEEP,DS ; make us a system process.
call makesys ; Get target console number
; If Console NOT valid then
; Write error message.
valcon:
  jz valcon
  mov dx,offset invmsg
  call C_WRITESTR
  int 224
  jmps qexit ; and terminate
; Save the Target console #
; Make proper Queue name
; open VIRQ que for target console
mov TOON,al
Word Ptr VQNAM, dx
dx,offset VIRQPB
mov cl,Q_OPEN
int 224
cmp ax,0h ; If Not Open then
jz qopnd ; Write an error message
mov dx,offset nopmsg
mov cl,C_WRITESTR
int 224
jmps qexit ; and terminate
endif
qopnd:
  mov cl,S_SYSDAT ; Get Supervisor Segment [1.1]
  int 224 ; [1.1]
  mov ax,es:2[bx] ; [1.1]
  mov cs:SUPVSR+2,ax ; [1.1]
  mov cs:SYSDAT,es ; [1.1]
  call Hook_In_Xios_List ; Insert in Xios intercept list [1.1]
  or ax,ax ; [1.1]
  jnz qexit ; if failure, get out [1.1]
mainloop:
  call tready ; repeat main loop:
  jz clocal ; if (target = ready) then
  call tget ; get target character
  call lput ; put character to local console
  endif
clocal:
  call lready ; if (local = ready) then
  jz mainloop ; get local character
  call lget ; check for abort
  cmp al,CNTRL_Z ; if abort then exit
  jz exit ; put character to target console
  call tput ; until abort
  jmps mainloop ; Delete from intercept list [1.1]
exit:
  call UnHook_From_Xios_List
qexit:
  call makesys ; make us a user process again
  mov cl,P_TERMCPM ; and terminate us
  int 224
  DSKEEP dw 2

```

```

; makesys: Set current process as a systems process in PD flag
; field and set KEEP flag to keep process from being
; terminated.
;
; Entry -> None
; Return -> none
;
; Regs used: ax, bx, cx, es
;

```

```

makesys:
  mov cl,P PDADR ; Obtain process descriptor address
  int 224
  mov bx,ax ; bx := offset(PD); es := segment(PD)
  or es:word ptr 6[bx],3 ; PD.FLAG := PD.FLAG or SYSTEM+KEEP
  ret ; end makesys

```

```

; makes: Reset the system and keep attribute in PD flag
;
; Entry -> None
; Return -> None
;

```

```

makeus:
  mov cl,P PDADR ; Obtain PD address
  int 224
  mov bx,ax ; bx:=offset(PD); es:=segment(PD)
  and es:Word Ptr 6[bx],not 3 ; mask PD.FLAG.System+keep
  ret

```

```

;gtarg Get console number as option from command tail
;
; entry -> none
;
; Return -> DX = 2 byte ASCII value from buffer ( 0 - 99)
; AL = Integer 0 - 99
; FLAG = Z if valid Option, NZ if Invalid option;
;
; Regs used : AX,DX,CX
;

```

```

;gtarg:
  cmp dx,0 ; If No Tail then No Option
  je nooption ; else
  mov dx,word ptr .OPTION ; DX := Option
  mov ax,dx ; AX := Option (in Ascii)
  sub al,'0' ; Convert AX to binary in AL
  cmp ah,'0'
  jb no_tens ; if second digit then
  cmp ah,'9' ; adjust
  ja no_tens
  sub ah,'0'
  mov cl,10
  mul cl ; make first digit the tens
  add al,ch ; AL = binary(DX)
  end if
no_tens:
  cmp al,0 ; is option in range
  jb nooption ; Nope, so no option found
  cmp al,99
  jg nooption

```

```

if CPCDOS41
  cmp dh,' '
  jnz not_single
  mov dh,'0'
  xchg dl,dh
not_single:
  endif
xor ah,ah ; set flag
ret
nooption:
  or ah,0ffh
  ret

```



```

; treadr: See if a character from the target console is ready, i.e.
; is a character in the circular queue buffer.
;
; Entry -> nothing
;
; Return <- Flag = NZ if char is ready, Z if not
; AL = FFh if char is ready, 0 if not
;
; Regs used: None
;
treadr: mov     si, offset OUTQUE    ; Here is our Que
; call    qempt                    ;
; pushf
; jnz    treadr                    ; If Writer waiting then
; cmp    BFULL, 0ffh
; jnz    treadr                    ;
; mov    BFULL, 0h                  ; Set Flag to indicate que empty
; mov    dl, BASEFLAG              ; Compute Flag for this console
; add    dl, TCON                  ; and set it.
; mov    cl, DEV_SEIFLAG
; int    224
; popf
; ret

; tget: Get target character, i.e. Take it out of Queue buffer
;
; Entry -> None
; Return <- AL = Character
;
; Regs used: AX, SI, BX
;
tget:  mov     si, offset OUTQUE    ; This is the Que Buffer
; call    qdel                    ; Go get the char
; ret

; lput: Send a character to local console
;
; Entry -> AL = Character
; Return <- none
;
; Regs used: AX, CX, DX
;
lput:  mov     dl, al               ; Character to send to DL
; mov     cl, C_WRITE              ; Use System C_Write function
; int    224
; ret

; lready: Obtain status of local console
;
; Entry -> none
; Return <- AL = 01h if Char ready, 00h if not ready
; Flag = Z if not ready, NZ if ready
;
; Regs used: AX, BX, CX
;
lready: mov     cl, C_STAT          ; Use System C_stat function
; int    224
; or     al, al                    ; Set Return Flag
; ret

; lget: Get a character from the local console
;
; Entry -> None
; Return <- AL = character
;
;
;
;
lget:  mov     cl, C_RAWIO         ;
; mov     dl, 0Tdh                ;
; int    224
; ret

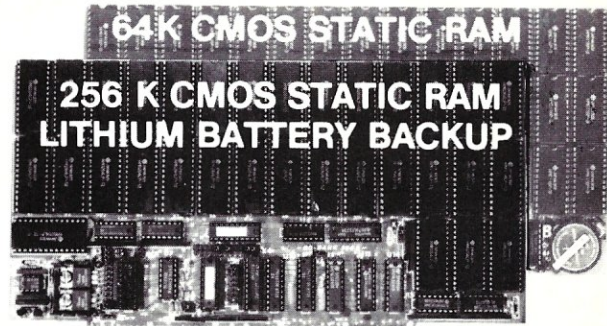
; tput: Send character to target console. Does this by writing
; character to the target consoles VIRQn System Queue.
;
; Entry -> AL = Character to send
; Return <- AX = AX on entry.
;
;
;
;
tput:  push    ax
; mov     ah, 0                    ; Null High Byte
; mov     word ptr qbuf, ax        ; Put character into Queue buffer
; mov     dx, offset VIRQPB       ; DX := Queue Parameter Block offset
; mov     cl, Q_WRITE
; int    224
; pop    ax
; ret

; xiosintcpt: Intercept call to XIOS & handle IO_CONOUT function;
; for target console locally.
;
; Entry -> AL = XIOS function number
; CL = Character to send
; DL = Virtual console to send to
;
; Return <- Jumps to XIOS with all registers preserved
; Except Flag, which is not used by XIOS
; functions.
;
;
;
;
xiosintcpt: push    ds
; mov     ds, DSKEEP              ; Get this Data Segment
; cmp     al, 2                    ; if XIOS function=2 (Console Output)
; jne    jxios                    ; and
; cmp     dl, TCON                ; Console = Target Console
; jne    jxios                    ; then
; call    qinsert                 ; Insert char in queue
; jmpf   cs: Dword Ptr ToXios_Off ; Goto realxios or next process

; qinsert: Insert character into que. If Que is full, wait until
; some space has been made. Uses System Flags to wait for
; empty queue.
;
; Entry -> CL = Character
; Return <- none
;
; Regs used: All Registers preserved except flag Reg
;
;
;
;

```

HIGH PERFORMANCE RAM



✓ COMPARE

	8/16 BIT DATA	Bank Selection MPM, OASIS, CROMIX-D	10 Megahertz Speed	Software Write Protect	Battery Backup Option
Compupro Ram 22	✓	NO	✓	NO	NO
Octagon 256K	✓	NO	✓	NO	NO
Cromemco 256KZ II	✓	NO	NO	NO	NO
Dynamic Boards	✓	✓	NO	NO	NO
BG-Bank 256S	✓	✓	✓	✓	✓

GUARANTEED IN YOUR SYSTEM CROMIX-D • MPM • CCS • OASIS • AMOS

✓ PLUS: 8/16 BIT TRANSFERS • 24-BIT EX. ADDRESSING
8-12 MHZ • 2K DESELECTS • RAM-EPROM MIX
IEEE 696/S-100 • LOW POWER • FULLY STATIC

LITHIUM BATTERY BACKUP avoids power failure crashes intelligently. Unique POWER-FAIL-SENSE circuit allows processor to save register information and disable board before POWER FAILURE CRASHES memory.

BG BANK 256S..... \$695 Battery Backup \$79
BG BANK 64S..... \$299 Battery Backup \$59



BG COMPUTER APPLICATIONS, 206 Brookside,
Bryan, Texas 77801 International orders add 30%.
(409) 775-5009


```

qinsert:  push  ax
         push  bx                ; Save'm
         push  cx
         push  dx
         push  si
         push  di
         push  bp
         mov   si,offset OUTQUE ;Get Circular Que offset
         call  qfull             ;If Que is full then
         jnz  insrt             ; Wait for empty queue flag
         push  si
         push  cx
         mov   Byte Ptr BFULL,0ffh ; Tell Reader to set Flag when empty
         mov   di,BASEFLAG      ; Compute flag for this console
         add   di,Byte Ptr TCON  ; flag := BaseFlag + Console #
         mov   cl,DEV WAITFLAG  ; lets wait for the flag
         call  supif            ; direct supervisor call
         pop   cx
         pop   si                ; end if
insrt:   mov   al,cl             ; Character to send in al
         call  queins           ; insert Character into que
         pop   bp
         pop   di
         pop   si
         pop   cx
         pop   bx
         pop   ax                ; restore them
         ret

```

```

; gempt: Check to see if Que is empty
; entry -> si = que offset
; return -> Que empty: al = 0, Z is Set
; Char in que: al = FF, Z not set

```

```

gempt:  push  bx                ;lets not destroy anything other than al
         mov  bx,word ptr 2[si] ; get rear of que
         cmp  word ptr [si],bx ; if front = rear then
         mov  al,0
         je   qstat             ; que := empty
         mov  al,0ffh          ; else que:= not empty
qstat:  pop   bx
         or   al,al
         ret

```

```

; qfull: test if que is full.
; entry -> si = que offset
; return -> NZ = ok, Z = que FULL

```

```

qfull:  mov  bx,word ptr [si]   ; if front + 1 = rear then
         call incqptr           ; que=full
         cmp  bx,word ptr 2[si]
         ret                    ; Z = full, NZ = ok

```

```

; incqptr: Increment the Que pointer. Takes care of wrap around;
; if end of que area is reached
; entry -> BX = pointer
; return -> BX = pointer + 1 increment

```

```

incqptr:inc  bx                ; point to next que offset
         cmp  bx,qdepth        ; end of que area reached?
         jne  incd             ; if not we are done
         xor  bx,bx            ; yeap... lets wrap
incd:  ret

```

```

; queins:insert a character into que. If que is full character
; is just dropped.
; entry -> al = character
; si = que offset
; return -> none

```

```

queins: push  ax
         call  qfull           ; find out if que is full
         pop   ax              ; Z -> ok, NZ -> full
         je   qind            ; if que full then loos data
         mov  bx,word ptr [si] ; else que[front]=item
         mov  byte ptr 4[si+bx],al
         call  incqptr         ; front = front + 1
         mov  word ptr [si],bx
qind:  ret

```

```

; qdel: delete a character from que
; entry -> si = que offset
; return -> al = character

```

```

qdel:  call  gempt            ;anything in the que ?
         je   qdeld          ;no, so nothing we can get out.
         mov  bx,word ptr 2[si] ;al = que[rear]
         mov  al,byte ptr 4[si+bx]
         call  incqptr         ; rear = rear+1
         mov  word ptr 2[si],bx
qdeld: ret

```

```

; supif: Call CCP/M Supervisor
; Entry -> CX = System call number
; DX = Parameter
; ES = User Data Area
; Return -> AX = BX = Return
; CX = Error Code
; ES = UDA Segment (as on Entry)
; Regs Used: DX, SI, DI, BP are NOT preserved..

```

```

supif:  mov  ch,0              ; Clear High byte of function number
         push es
         push ds
         mov  ds,Sysdat
         mov  si,68h
         mov  es,ds:10h[si] ; Get UDA of current process
         call cs:word ptr SUPVSR
         pop  ds
         pop  es

```

```

ret
SYSDAT dw 0 ; SYSDAT segment
SUPVSR dw 3 ; Entry to supervisor, offset = 3
       dw 0 ; Segment filled in later
DSEGB
TCON  db 0 ; Target Console Number
BFULL db 0 ; Flag indicating buffer is full
OUTQUE dw 0 ; front of que
       rb 0 ; rear of que
       qdepth ; here is the que area.
INVMG db 'Invalid Virtual Console Number.',cr,lf
       db ' usage: CONNECT nn',cr,lf
       db ' nn - Virtual Console number',cr,lf,'$'
NOPMSG db 'Unable to open Input Queue.',cr,lf,'$'
VINQPB dw 0,0,0 ; VINQnn Parameter Block
       dw offset Qbuf
       db 'VINQ' ; Primary Queue name
       If not CPDOS41
VQNAM db ' ' ; Secondary Queue name (Console number)
       endif
       If CPDOS41
VQNAM db '0'
       endif
QBUF  db 40h * 2 ;can have 40 messages times 2 bytes each
       rw 31
       stack dw 0

```

RTC PLUS

TRANSLATE RATFOR TO C PLUS FORTRAN TO C!

- **VERY COMPLETE TRANSLATIONS OF RATFOR.** Over 95% of STUG's RATFOR is supported!
- **NOW TRANSLATE FORTRAN-IV TO C!** *RTC Plus* will now translate your FORTRAN-IV code to C (excluding I/O and F77 character). Tap vast FORTRAN libraries and accelerate your C development efforts!
- **DEMO IS NOW AVAILABLE** containing restricted versions of the *RTC Plus* package.
- **NOW TRANSLATE TWO LANGUAGES FOR THE PRICE OF ONE!**

DEMO: \$10
MSDOS: \$325

COBALT BLUE

ORDER TODAY!

1683 MILROY, SUITE 101
SAN JOSE, CA 95124
(408) 723-0474

The DRC Terminal Kit

by Tony Dean

In the world of supermicros, minis, mainframes, and many single-board computers, ASCII terminals are the most commonly used primary I/O devices. These terminals range in function from Spartan to luxurious and in price from a few hundred to a few thousand dollars. So when it's time to select a terminal, how do you choose?

If your needs are like mine and you need a terminal that is fast, flexible, reliable, and if possible, inexpensive, there are few serious alternatives. I chose to build my own using a terminal-board kit—Digital Research Computers (DRC) ZRT-80—and a separate keyboard and monitor. The terminal took me two weekends to assemble and cost about \$350. Recent price cuts would drop this figure to less than \$200. The DRC terminal board currently sells for \$89.95. The source code for the terminal's ROM is also available, on 8-inch disk, for \$10.

The ZRT-80 controls the video display, keyboard input, and I/O between it and a computer or modem. It uses a Z80-A microprocessor for control, a 6845 CRT controller, and an 8250-baud generator/USART. Its capabilities are impressive, allowing serious hackers to reconfigure the terminal. It offers four terminal-emulation modes—Heath/Zenith H-19, ADM 3A, ADDS Regency 200, and Beehive. I use the H-19 mode with 24 × 80 display format as it is reputed to be the most flexible. You can obtain higher display densities by adding more RAM (positions for an extra 6K are provided). The baud rate is selectable in 16 steps from 75 to 19,200 baud.

To build a complete terminal, in addition to the ZRT-80 kit, you need a power supply, CRT monitor, and keyboard. You can select these items to fit your specific needs. Several surplus-component suppliers, as well as DRC, sell these items. Power supplies start at less than \$20. CRT monitors range from \$29.95 to \$150. Heathkit has a monitor in kit form for \$79.95 that looks nice.

The ZRT-80 accepts an ASCII 7-bit parallel input from the keyboard plus it has a toggle that allows you to change the numeric group into a function-key set like the Heath H-19. Keyboards range from \$19.95 to \$150.

The board comes with a 21-page manual that contains complete information on con-

Assembling A Terminal From Components For Under \$200

struction, theory of operation, and setup. The ZRT-80 has five connectors—video, power, keyboard, option switch, and RS-232 interface. Both split- and composite-video output are provided. The power connector has LED and beeper outputs and a connection for an external reset switch. The option connector allows you to connect the DIP-switch assignments to an external switch set. The board has three eight-switch DIP assemblies that you can set to control the terminal's default baud rate, emulation mode, and keyboard parameters.

The manual covers basic operation to get the terminal through initial power-up and testing for proper use. The terminal-interfacing section is acceptable if you need only the basic configuration of pins 1, 2, 3, and 7 with software protocol, but if you want to use any other RS-232 configuration, it might be a good idea to get a copy of *RS-232 Made Easy* and an RS-232 breakout box. The manual gives brief descriptions of display and emulation modes, and its final page contains a table of escape- and control-code functions for terminal mode.

I found the ZRT-80 easy to assemble. The silk-screening is well done and easy to read, so assembly is a simple pick, place, and solder routine. All ICs are socketed. The only problem I had was having to replace a damaged socket. Rather than wait for DRC to send a replacement, I bought one—and enough sockets to fill the extra RAM positions—from a local electronics store.

Once you have completed the board assembly, you wire the ZRT-80 for power. When you apply power, the red LED should light to indicate proper function. Next you install the keyboard and CRT. When you reach this stage, you can power up the terminal system and run through

the internal tests to see whether the terminal can display characters and read the keyboard properly. When you have set all the switches, made any connections, and run the tests, it is time to use the terminal with your computer. In my case, the ZRT-80 powered right up and worked fine on the first try—an edifying experience indeed. I found that most software offered support for the H-19 display format.

When everything is working well, you can explore such possibilities as expanding the ZRT-80 to support other keyboard capabilities. Analysis of the source code indicates that the addition of support for the eighth bit, which lets you use arrow keys and a variety of special-function, word processing keys, is easy. The big question is what to do with the 12 function keys (they shift to give you 24 functions in all). Sockets are available for another 2732 EPROM and 6K RAM, which gives several possible options.

With this much potential, I feel my ZRT-80 can easily support the rest of the functions of the 8-bit ASCII keyboard to which it is attached. This conversion will require modification of the source code to allow the interpretation of the alternate keypad to be handled differently. With the modifications I plan, my ZRT-80 will be able to convert custom key inputs to escape sequences on some keys. Other keys will pass new codes or full 8-bit codes as would be needed by the function keys.

The ZRT-80 has proven to be flexible, reliable, and expandable, and its standard features have seen more than a year of trouble-free use on my computer. The standard features alone exceed those of terminals costing hundreds more. For hackers who feel that standards are often substandard, the ZRT-80 allows them to customize a terminal to fit their individual needs precisely, and a fully customized terminal cannot be purchased off the shelf at any price.

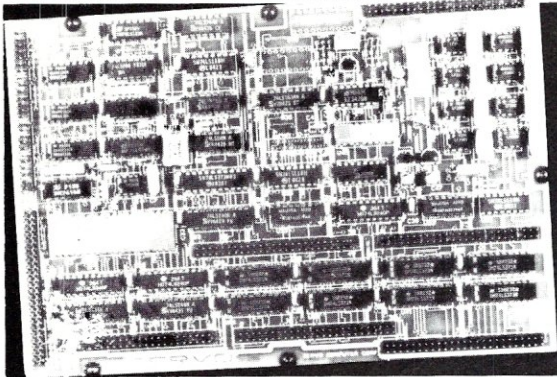
PRODUCT INFORMATION

ZRT-80 CRT Terminal
 \$89.95—Complete kit with 2K video RAM
 \$49.95—Blank printed-circuit board with 2732 monitor and 2716 character ROMs
 \$10—CP/M source disk (8-inch)
 \$7.50—Set of two crystals
Continued on page 72

SERVO INDUSTRIAL PROCESS CONTROL BOARD

14 Bit Lab Accuracy Industrial Durability

The Features of 5 Boards in One.



The Servo Process Control Board is a general purpose multi-channel analog and digital input/output controller which can communicate with and control most medium-scale industrial processes. Designed to work with the Servo Industrial Microcomputer, (or other computers which support the SASI interface), it offers great accuracy at prices starting as low as \$495.

■ American made ■ MILSPEC components ■ 1 year warranty

Maximum configuration includes: ■ 192 analog inputs ■ 64 analog outputs
■ 128 digital inputs ■ 512 digital outputs

■ Based on the INTERSIL 14 Bit A to D converter ■ Communicates with any micro or mainframe computer ■ Easily programmed in high level languages (Turbo Pascal, Microsoft Basic, Forth) ■ PID, FFT, Thermocouple linearization and laboratory DAS software supplied. ■ Custom software and hardware engineering assistance available.



360-B N. Ellensburg St.
Box 566
Gold Beach, OR. 97444
(503) 247-2021

C CODE FOR THE PC

source code, of course

GraphiC 3.0 hi-res color plots	\$300
Panache C Program Generator	\$125
QC88 C Compiler	\$90
Concurrent C	\$45
Coder's Prolog in C	\$45
Biggerstaff's System Tools	\$40
Translate Rules to C	\$30
LEX	\$25
YACC & PREP	\$25
tiny-c interpreter & shell	\$20
C Tools	\$15

The Austin Code Works
11100 Leafwood Lane
Austin, Texas 78750-3409
(512) 258-0785

Free shipping on prepaid orders

No credit cards

Creating MYLIB

Continued from page 57

thing else that will clarify the procedures function. The Implementation module is shown in Listing 1.

The first thing you want to notice is that this Implementation module imported some procedures from another library. In this case, the library was furnished by the compiler's library and was located on my disk drive. So if I wrote a sample program for you, I would import MYLIB, and MYLIB would import the systems library called STRINGS. That would be a four-layer hierarchy. Even if I had used duplicate variables, it wouldn't have made any difference, because they are all invisible to each other.

There are excellent reasons to learn various languages, and, in this case, some of those reasons have nothing to do with anything except programmer comfort. Creating such a library as I have described provides three major benefits.

1. It reduces program compilation time by dividing large programs into smaller, separately compiled pieces.
2. It creates well-defined boundaries for dividing work between programmers.
3. It can be shared by different programs.

I like to think of it as not reinventing the wheel. If you will share your libraries with me, I will share mine with you. §

DRC Article

Continued from page 71

Digital Research Computers (of Texas)
Box 381450
Duncanville, TX 75138
(214) 225-2309

MYLIB Code

LISTING 1

```
IMPLEMENTATION MODULE MyLib;
(* NOTE: I have removed the error trapping code to keep it
simple, but it is needed since 'Strings' is
sensitive to erroneous lengths trying to fit into
other things *)

FROM Strings IMPORT Length, Pos, Append, Copy, Delete;

(***** library stuff *****)

PROCEDURE AppendExt (VAR DiskFileName : FileName; Ext : Extension);
(* this will append a "." and a extension if included *)

VAR
LengthCount : CARDINAL;
BEGIN
LengthCount := Length(DiskFileName);
(* DiskFileName is a array so find the end and
insert a period *)
DiskFileName[LengthCount] := ".";
(* now add the extension *)
Append(Ext, DiskFileName)
END
AppendExt;

PROCEDURE StripExt (VAR DiskFileName : FileName);
(* this will strip the period and the following characters *)

VAR
WherePeriod : CARDINAL;
BEGIN
WherePeriod := Pos(".", DiskFileName);
Delete(DiskFileName, WherePeriod, Length(DiskFileName)-WherePeriod)
END StripExt;

PROCEDURE GetExt(DiskFileName : FileName; VAR Ext : Extension);
(* this will return the extension in Ext *)

VAR
WherePeriod, ExtLength : CARDINAL;

BEGIN
WherePeriod := Pos(".", DiskFileName);
(* HIGH will find the array length *)
ExtLength := HIGH(DiskFileName) - WherePeriod;
Copy(DiskFileName, WherePeriod + 1, ExtLength, Ext)
END GetExt;

PROCEDURE Beep;
(* this will cause the terminal to BEEP *)

BEGIN
WRITE(CHR(07H));
END Beep;

END MyLib.
```

LISTING 2

The following can be used to prove the programs.

```
MODULE ExtTest;

FROM Strings IMPORT Length, Pos, Append, Copy, Delete;

TYPE
Extension = ARRAY [0..2] OF CHAR;
FileName = ARRAY [0..12] OF CHAR;
VAR
Example : FileName;
Ext : Extension;

PROCEDURE AppendExt (VAR DiskFileName : FileName; Ext : Extension);

VAR
LengthCount : CARDINAL;
BEGIN
LengthCount := Length(DiskFileName);
DiskFileName[LengthCount] := ".";
Append(Ext, DiskFileName)
END
AppendExt;

PROCEDURE StripExt (VAR DiskFileName : FileName);

VAR
WherePeriod : CARDINAL;
BEGIN
WherePeriod := Pos(".", DiskFileName);
Delete(DiskFileName, WherePeriod, Length(DiskFileName)-WherePeriod)
END StripExt;

PROCEDURE GetExt (DiskFileName : FileName; VAR Ext : Extension);
```

```
VAR
WherePeriod, ExtLength : CARDINAL;

BEGIN
WherePeriod := Pos(".", DiskFileName);
ExtLength := HIGH(DiskFileName) - WherePeriod;
Copy(DiskFileName, WherePeriod + 1, ExtLength, Ext)

END GetExt;

(***** MAIN *****)

BEGIN
Example := '1234567';
WRITELN('You should get "1234567", you got = ', Example);
WRITELN;
AppendExt(Example, 'abc');
WRITELN('Here you should get "1234567.abc, you got = ', Example);
WRITELN;
Example := '1234567.123';
GetExt(Example, Ext);
WRITELN('The extension should be "123". Its = ', Ext);
StripExt(Example);
WRITELN; WRITELN('The extension should be deleted, its = ', Example);

END ExtTest.
```

Does this look familiar?
What if each change
you made to your
program was ready to
test in seconds instead
of minutes?



"The SLR tools will change the way you write code. I don't use anything else.", Joe Wright

RELOCATING MACRO ASSEMBLERS • Z80 • 8085 • HD64180

- Generates COM, Intel HEX, Microsoft REL, or SLR REL
- Intel macro facility
- All M80 pseudo ops
- Multiple assemblies via command line or indirect command file
- Alternate user number search
- ZCPR3 and CP/M Plus error flag support, CP/M 2.2 submit abort
- Over 30 user configurable options
- Descriptive error messages
- XREF and Symbol tables
- 16 significant characters on labels (even externals)
- Time and Date in listing
- Nested conditionals and INCLUDE files
- Supports math on externals

\$49.95

requires Z80 CP/M compatible systems with at least 32K TPA

SLR Systems

1622 N. Main St., Butler, PA 16001
(412) 282-0864 (800) 833-3061

Running CP/M Under MS-DOS: Part 2

by Robert J. Stek, Ph.D.

In a previous article (Converting from CP/M to MS-DOS on the PC) I reviewed two Z80 coprocessor boards that enable users to run CP/M programs under MS-DOS. At the time of that printing, the only low-cost alternative to these boards was a public-domain CP/M software emulator. Unfortunately, that emulator gave a performance equivalent to that of an 8080 CPU running with an under-1-MHz clock speed.

Since then, however, the NEC V-20 and V-30 CPUs have become widely available as pin-for-pin replacements for the Intel 8088 and 8086 processors. Besides providing a small performance increase for IBM-compatible micros, they also have the unique property of being dual-mode processors in that they can directly execute the 8080 instruction set as well as the 8088 instruction set at whatever clock speed the host microcomputer normally uses. While this does not provide performance quite as good as that offered by the 8-MHz Z80H used in the Earth Computers coprocessor board, nonetheless, it is more than acceptable for running the occasional CP/M program. Two distinct advantages that the V-20 enjoys over the coprocessor approach are: (1) you do not give up an expansion slot, and (2) the V-20 costs less than \$20. Just as a coprocessor board requires some additional software to run CP/M programs, so does the V-20. The price of this software ranges from free (public domain) to nearly \$200.

This article reviews two commercial products that use the V-20 to run CP/M programs under MS-DOS: CP/Mulator by Source Information, and PC RP/M2 by Micro Methods, Inc. Some cautionary notes about some potential problems with the V-20 are given first.

99% & 95/100% COMPATIBLE

I own a two speed "Turbo" clone machine. It can run at the standard 4.77-MHz clock rate of the IBM PC or at a moderately higher 6.66-MHz clock rate. I had hoped that between its Turbo mode and the improved performance of the V-20 over the 8088, I would have the equivalent of a 7-plus-MHz PC. After all, everyone knows that the V-20 is an exact replacement for

Two Programs That Allow Running CP/M Programs On a PC or PC-Compatible Equipped With A V20 Chip

the 8088, and that there would be no problem substituting the new CPU for the old ... HA!

Because my machine does use a dual-clock speed, I was required to purchase the 8-MHz version of the V-20 rather than the standard 5-MHz part. The faster part was a few dollars more, but I figured better safe than sorry. NEC rates the duty cycle of the V-20 somewhat differently than you might expect, and you should not take the chance that the 5-MHz version will "probably" work at a higher clock speed—it won't.

My first task was to make sure that the V-20 worked as well as the 8088 it was replacing. I swapped CPU chips in less than five minutes; after all, the V-20 can only be installed in the socket two ways. And only one way is the right way! (Micro Methods assumes that you have already replaced the chip; Source Information does provide two pages of handholding instructions in case you don't know one end of a chip from the other.)

I booted up PC-DOS 3.1 with no problem. The screen cleared, and I received the normal date and time prompts. So far, everything looked fine. I next wanted to run Norton's SYSINFO program to compare the speed of the V-20 to that of the 8088. I put a disk in drive B and typed DIR, expecting to see a normal directory display.

The disk drive whirred and file names appeared on my screen in a normal fashion—until the display reached the bottom line, at which point you would expect the top line to scroll off the screen. Instead, the display (indeed, the entire machine) just stopped and hung there. Even pressing CTRL-ALT-DEL did nothing. It was necessary to turn off the computer and try again—with the exact same results.

Additional investigating started to provide some clues as to what was going on. Programs that never actually scrolled seemed to work fine. When booting up PC RP/M2 (more about this later, but in one mode, RP/M2 can boot up as a stand-alone CP/M replacement, bypassing MS-DOS completely), the same phenomenon occurred. It booted fine, but it would not scroll past the bottom line. I sometimes use FANSI-Console, a shareware-type device drive that replaces the BIOS screen-writing routines for significantly faster screen updating. When FANSI-Console was loaded, everything was back to normal; screen scrolling worked just fine. A final clue (or perhaps a bit of misdirection) was that scrolling also worked fine if the machine was first put into its Turbo mode. That is, it worked fine with a 6.66-MHz clock but not with a 4.77-MHz clock—quite the opposite of what you might expect, since higher clock speeds have been known to cause problems.

My first thought was to blame the NEC chip itself. I did have access to two other V-20s, so I tried them. Same results! I called the two vendors mentioned above, thinking that they might have had reports of similar problems. Neither had ever heard of such a thing. I spoke with a representative of NEC concerning rumors that there were two versions of the V-20: a current version that was OK and an earlier version that was not. He denied the rumors but did come up with a possible explanation of my problem. Much of the V-20's increased performance was due to the fact that it performed many of its instructions in fewer clock cycles than the 8088. He suggested that perhaps my clone's ROM BIOS depended upon some critical timing for the scrolling routine.

That made some sense. When FANSI-

Console was in place, it completely bypassed the ROM BIOS routines and did the screen handling itself. And when the clock was changed to a higher speed, that changed the timing as well, and the two changes managed to compensate for each other in some unfathomable fashion. A call to the Arizona vendor of my clone revealed that he had heard from one of his dealers who had problems with V-20's as well. To make a long story a bit shorter, he sent me a different ROM BIOS that solved the problem.

The moral of the story is, nothing is ever as simple as it promises to be. If you have an IBM PC, you won't have this trouble at all. But if you do not have an alternate version of a well-written ROM BIOS available for your compatible, you may run into problems.

TRANSFERRING CP/M PROGRAMS TO YOUR PC

In my previous article, I mentioned the problem of transferring your CP/M programs to MS-DOS format. Thank goodness MS-DOS has not had the multiplicity of formats that plague the CP/M world. Perhaps the Uniform utility program is best known as being provided in versions for both CP/M and MS-DOS which allows you to read and write almost all of these different formats (from MicroSolutions, 125 South Fourth St., Dekalb, IL 60115). You will need such a utility to use with CP/Mulator, but PC RP/M2 has a utility that allows it to read and write more than ninety CP/M formats. If you do not have such a utility, or if your unique CP/M format is not supported, then you can still transfer programs via a serial link. Both CP/Mulator and PC RP/M2 provide a copy of a public-domain MODEM-type program to help the transfer along (assuming that you have a compatible program on the other machine).

Whether you use Uniform or the PC RP/M2 utility, you will be able to place a CP/M formatted soft-sectored disk (e.g., a Kaypro disk) into one of the floppy drives on your PC and then transfer files over to an MS-DOS-formatted disk. This could be another floppy or a hard disk; preferably, it would be into a separate subdirectory so that you wouldn't confuse any CP/M .COM files with MS-DOS .COM files. Overlay files, configuration files, data files, etc., are all transferred if needed.

RUNNING CP/M PROGRAMS

CP/Mulator

CP/Mulator suggests that you change the file extension of all CP/M .COM files to .CPM in order to prevent inadvertent execution of a CP/M command file under MS-DOS. It does allow for the odd CP/M application program that might require files with an extension of .CPM. In this case, the extension of the CP/M command files is not changed, and CP/Mulator is run with a command-line option.

CP/Mulator is supplied as a standard MS-DOS .EXE file (CPM.EXE). The emulator is present only when a CP/M application program is executing. Thus, you do not see a CP/M front end with the built-in CP/M commands (ERA, DIR, REN, SAVE, TYPE, USER); you would normally use the MS-DOS equivalent commands. MS-DOS cannot perform a CP/M SAVE, but CP/Mulator does have a command-line option that duplicates the CP/M SAVE command. This allows a SAVE of the CP/M TPA and any modifications made therein. In normal usage, CP/Mulator expects the name of the CP/M program (with its arguments, if any) as its argument—e.g., CPM SC2 to run SuperCalc2, or CPM MBASIC MYPROG to run the MBASIC program MYPROG.BAS.

CP/Mulator also has one other command-line option itself—a DEBUG option that allows for the display or capture of selected information during execution of the CP/M program. Debug information may be directed to the screen, a file or both. Its options are set through a bit mask as follows:

- 80—debug active at program start
- 40—debug output to file (DEBUG.LST)
- 20—debug screen output disabled
- 10—snap current DMA on emulator exit
- 08—reserved for future option
- 04—snap CP/M FCB in and out
- 02—snap regs in and out of emulator
- 01—snap BDOS function codes (nn) and BIOS entry numbers (:nn)

Unfortunately, Source Information provides absolutely no additional information or examples of the DEBUG option beyond listing the above options (the entire manual is just 30 pages long).

During operation of the CP/M application program under CP/Mulator, two "hot keys" are always active—F9 and F10. Actually, they could be called "warm keys," since an outstanding read to the console may delay recognition of these keys until the next keystroke or a RETURN is keyed. F9 is the Menu key that causes a short menu to pop up on-screen as in Sidekick. Here you may enable or disable the debug options selected on the command line, save the TPA to a file, execute an 8080 RST 7 instruction upon resuming the CP/M application (intended for use with an 8080 debugger such as DDT), exit immediately to MS-DOS (without closing any open CP/M files), or continue execution of the CP/M application. F10 is the Abort key that terminates the emulator and exits to DOS immediately.

Source Information notes that some CP/M commands are not supported or are only partially supported under CP/Mulator. Partially supported are programs such as PIP and STAT that work except when referring to USER areas or the I/O byte. Not supported are some hardware-dependent programs such as MOVCPM

and SYSGEN, and SUBMIT and XSUB. Also not supported are programs that give system hardware direct access to read/write hardware port addresses (e.g., communication programs); programs that directly access the CP/M diskette format, such as programs that format CP/M diskettes; and programs that use Z80 instructions. These are reasonable limitations, since most of these programs are not used appropriately under MS-DOS in any case.

Most BDOS and BIOS functions of CP/M 2.2 are fully supported by CP/Mulator. Exceptions include:

- BDOS(3) [reader input]
Waits for and reads a character from AUX (COM1) and returns.
- BDOS(4) [punch output]
Outputs character from register E to AUX (COM1) and returns.
- BDOS(7) [get I/O byte]
Value of I/O byte set by BDOS(8) is returned in accumulator. Not used by CP/Mulator
- BDOS(8) [set I/O byte]
Value of I/O byte is set to value in register E. Not used, but retained for compatibility.
- BDOS(10) [read console buffer]
Similar to CP/M equivalent, but editing of line buffer follows MS-DOS editing rules which only affects operator entry of data.
- BDOS(30) [set file attributes]
Not supported. Accumulator always contains a zero upon return.
- BDOS(32) [get/set user code]
User areas are not supported.
- BDOS(40) [write random with zero fill]
Performs a write random (BDOS(34)) instead.
- BOOT Returns control to MS-DOS.
- WBOOT Returns control to MS-DOS.
- PUNCH Returns with no action.
- READER Always returns a hex 1A, EOF.
- HOME Returns with no action.
- SETTRK Returns with no action.
- SETSEC Returns with no action.
- READ Not supported, returns with an error.
- WRITE Not supported, returns with an error.
- SECTRAN Not supported, returns with an error.

When running commercial programs, these differences actually make no difference. I have tried several dozen application programs and utilities and have had no problems.

Speed of execution was about what you would expect with a 4.77-MHz clock—a

tad faster than when the same program was running on a stock 4-MHz Z80 system. The expected 40 percent speed increase was noted when I ran my PC clone in its 6.66-MHz "Turbo" mode. (My 8-MHz Z80H Earth Computer coprocessor board ran them faster still, but it also cost several hundred dollars more!)

PC RP/M2

PC RP/M2, from Micro Methods, is unique among the dozen or so CP/M emulators for MS-DOS V-20 systems. It is a truly stand-alone CP/M 2.2 replacement. That is, you can boot up RP/M2 from the supplied floppy, bypassing MS-DOS completely. All BDOS and BIOS calls are available, since you are running a complete CP/M-replacement operating system. Since PC RP/M2 occupies a little over one 64K memory segment, any remaining memory in the PC is automatically configured as a memory disk called "M:" when RP/M2 is booted up. In a full 640K PC, this yields a memory disk of over 500K!

It is also possible, however, to boot MS-DOS first and then use the supplied MS-DOS program RPMSYS.COM to bring up RP/M2 under MS-DOS. This allows PC RP/M2 to have read/write access to all DOS files on all DOS drives through the supplied utilities RMPPIP.COM or PCPIP.COM, which allow file transfers from MS-DOS to RP/M2 and back. This is quite different from CP/Mulator, which essentially runs a CP/M application program as a task under the native MS-DOS operating system. With RP/M2 and MS-DOS coresident, you have two independent operating systems available, each with its unique disk format. If you wish to run an MS-DOS program, you must "leave" the RP/M2 operating system with a GODOS command (though you can call a memory-resident program like SideKick from within RP/M2). Similarly, to return to RP/M2 from MS-DOS, you must use the RPMSYS command. Note that if you do leave RP/M2 to return to MS-DOS, you will lose whatever is stored on the M: drive memory disk if you return to RP/M2.

RP/M2 is a complete, CP/M-compatible replacement operating system, so it is supplied with more complete documentation (129 pages of it) than CP/Mulator as well as a number of utility programs. These utility programs duplicate and extend the capabilities of their CP/M counterparts. Other utilities include programs to set foreground, background, and border colors for color monitors; a program to emulate some common display terminals for clearing the screen, positioning the cursor, etc.; an enhanced version of a SUBMIT facility; a DUMP program; an unerase utility; a disk-test program to find bad blocks on floppies; a MODEM7 protocol file-transfer utility; and a program to redefine one of the floppy drives to use one of more than ninety other CP/M disk formats. All these programs have the .COM extension.

RP/M2 was created in 1982 as a "better" CP/M. It has access to user area 0 for all user areas, a paged TYPE command, a hard disk vector that keeps nondismountable disks logged in, batch procedures on any drive, a disk-addressing range of 64Mb, hash-coded vectored directory access, and file date and time stamping like CP/M 3.0. All of these features have been retained in the PC RP/M2 version.

There are no restrictions on the type of programs you can run under PC RP/M2, so I tried DUCOM; it worked just fine. Of course, under RP/M2 you can't directly access an MS-DOS hard drive, so it is perfectly safe. As with CP/Mulator, all standard CP/M programs worked exactly as they should. The few differences in execution speed were minor.

SUMMARY

Both CP/Mulator and PC RP/M2 do what they are supposed to do: allow IBM PCs to run CP/M programs using the NEC V-20 CPU. They are different in price: \$129 for RP/M2 versus \$79.95 for CP/Mulator. They both offer similar performance in terms of execution speed for various benchmarks. So how do you choose?

PC RP/M2 is similar in function to CP/Mulator in that it allows you to use your old CP/M programs on your MS-DOS machine. It clearly has quite a different philosophy, however. It offers several features that are not supplied by CP/Mulator, including terminal emulation so that your CP/M programs will display properly. CP/Mulator suggests that you load ANSI.SYS, but some CP/M programs do not allow the long escape sequences necessary for an ANSI terminal. RP/M2 provides a utility to read many CP/M disk formats, while CP/Mulator has no such utility. RP/M2 supports all BDOS and CBIOS calls, while CP/Mulator does not. On the other hand, CP/Mulator allows your CP/M programs to access data directly on your MS-DOS disks, while RP/M2 requires that you transfer MS-DOS data files from an MS-DOS disk over to an RP/M2 disk first (and then back if required). CP/Mulator assumes you will use familiar MS-DOS commands for housekeeping, while RP/M2 requires that you use one set of housekeeping commands under RP/M2 and a different set under MS-DOS.

For the casual user of well-behaved CP/M programs, I would recommend CP/Mulator because it allows you to run transparently under MS-DOS, using MS-DOS files, and allows a consistent use of operating-system commands. But for the serious hacker or individual who does a fair amount of programming under CP/M, I would rate PC RP/M2 as the better program. Its many utility programs are a nice extra that make it worth the additional cost. §

PRODUCT INFORMATION

RP/M2

\$129 + \$5 shipping (\$10 non-U.S.)
Micro Methods, Inc.
118 SW First St., Box G
Warrenton, OR 97146
(503) 861-1765

CP/Mulator

\$79.95 (includes V-20 chip)
\$49.95 (software only)

Source Information

Box 2974
Warminster, PA 18974
(215) 441-8178

PC-PRO is PC-DOS for CompuPro/Viasyn

- PC DOS Ver 3.1
- PC Graphics Support

for information write

Computer House, Inc.

P.O. Box 709 Woodacre, CA 94973
(415) 453-0865

Trademarks: Viasyn Corp.,
PC-PRO - Computer House.

PORT-A-SOFT

Aardvark to Zorba

We Can Handle It



TRANSFER
PROGRAMS, DATA AND
OTHER FILES BETWEEN
OVER **400** COMPUTERS!

13 OPERATING SYSTEMS

WORD PROCESSORS TOO!!

Prices From \$7.00 Per Disk
Software and Hardware So You Can Do Your Own

**Call or write today for your
FREE CATALOG**

P.O. BOX 1685 555 S. STATE, STE. 12
OREM, UT 84058 (801) 226-6704

ISIS

CP/M  MSDOS

COMPLETE SOURCE CODE INCLUDED!

ICXPDS: eXchanger now supports the 5 1/4" iPDS format. Manipulation of ISIS-II files using your computer system was never easier.

ICXMDS: Same as ICXPDS, but for MDS 8" systems.

IMXPDS: Reads/Writes 5" iPDS disks on PC's and AT's.

TELEDPLUS: Enhanced serial file transfer program for CP/M, ISIS, or MS-DOS.

ISE: Emulator gives the CP/M and MS-DOS user access to all the ISIS-II languages and utilities.

ACCELER 8/16: CP/M-80 emulator for MS-DOS. Enables PC's to run ISE. (no source code, V-20 incl.)

\$89 each

\$250 any 3 above

UDI: The 8086 ISIS Emulator runs all UDI applications. \$300

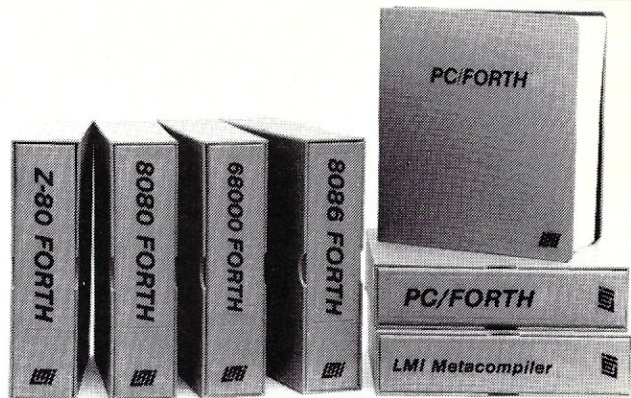
ZAS Development Package: Z-8 and Z-8000
Assembler for CP/M, ISIS, and MS-DOS.

Request a catalog of our products!

Copyrights: CP/M Digital Research, Inc.
ISIS-II and iPDS Intel Corp. MSDOS Microsoft

 **Western Wares** 303-327-4898
Box C • Norwood, CO 81423

TOTAL CONTROL with LMI FORTH™



**For Programming Professionals:
an expanding family of
compatible, high-performance,
Forth-83 Standard compilers
for microcomputers**

For Development:

Interactive Forth-83 Interpreter/Compilers

- 16-bit and 32-bit implementations
- Full screen editor and assembler
- Uses standard operating system files
- 400 page manual written in plain English
- Options include software floating point, arithmetic coprocessor support, symbolic debugger, native code compilers, and graphics support

For Applications: Forth-83 Metacompiler

- Unique table-driven multi-pass Forth compiler
- Compiles compact ROMable or disk-based applications
- Excellent error handling
- Produces headerless code, compiles from intermediate states, and performs conditional compilation
- Cross-compiles to 8080, Z-80, 8086, 68000, 6502, 8051, 8096, 1802, and 6303
- No license fee or royalty for compiled applications

For Speed: CForth Application Compiler

- Translates "high-level" Forth into in-line, optimized machine code
- Can generate ROMable code

Support Services for registered users:

- Technical Assistance Hotline
- Periodic newsletters and low-cost updates
- Bulletin Board System

**Call or write for detailed product information
and prices. Consulting and Educational Services
available by special arrangement.**

LMI Laboratory Microsystems Incorporated
Post Office Box 10430, Marina del Rey, CA 90295
Phone credit card orders to: (213) 306-7412

Overseas Distributors.

Germany: Forth-Systeme Angelika Flesch, Titisee-Neustadt, 7651-1665
UK: System Science Ltd., London, 01-248 0962
France: Micro-Sigma S.A.R.L., Paris, (1) 42.65.95.16
Japan: Southern Pacific Ltd., Yokohama, 045-314-9514
Australia: Wave-onic Associates, Wilson, W.A., (09) 451-2946

The UNIX File

by Ian Darwin

This column discusses the UNIX operating system. If you have comments or questions about UNIX or this column, please write to Ian Darwin at Box 603, Station F, Toronto, Ontario, Canada M4Y 2L8. If you have UNIX mail access to the uucp network, mail "ihnp4!darwin!ian." I can't always answer immediately, but I will get back to you; electronic mail gets answered first!

This installment of your UNIX column finds out what some high-speed UNIX boxes have in common, writes down some unwritten rules, and looks at sharing a phone line between dialin and dialout.

THE VME BUS

Microcomputer makers are always looking for ways to build faster systems. They want good CPU chips, and they want them running at high clock rates. For a given CPU type and speed, the fastest system will usually be a single-board computer with the memory, video or serial I/O, and disk controllers all on one board. But single-board systems typically limit the user's flexibility. A bus in the computer world provides a means of transporting not people but signals from one place to another.

A bus connects several circuit boards together to make a complete system. Many of us grew up with the S-100 bus. The majority of computers sold today use newer, faster buses.

Today a new bus architecture is attracting most of the press attention and manufacturer interest. The VMEbus is derived from Motorola's VERSAbus. The bus is supported by the VMEbus International Trade Association (VITA) and is undergoing the final stages of approval as an IEEE standard (P1014/D1.2) and an International Electrotechnical Commission standard (IEC 821). The architecture is designed and spelled out in considerable detail for total interchangeability (the VITA version of the bus specification is 261 pages long). The spec is intended to allow boards to communicate quickly and reliably. It also goes into detail on the mechanical specification of boards and mountings, to ensure compatibility among manufacturers. The basic format is Eurocard, so that Eurocard cabinets and mountings can be used. One overall design goal is to ensure that system throughput is

limited by the devices making it up rather than by the time needed to transfer data among them.

A detailed article on the VMEbus is beyond the scope of this column. What matters for UNIX users is that many of the top system vendors have adopted the VMEbus. Sun Microsystems' line of Sun-3 workstations runs Motorola 68020 CPUs on the VMEbus with a UNIX derived from 4.2- and 4.3BSD. Dual Systems' Chapparel product line features MC68020 on the VMEbus with a standard System V. Integrated Systems, Inc. (ISI) Optimum/V computers (V is for VME) run Motorola CPUs. The list goes on. There is even an AT&T 32xxx offering.

Hence, the purchaser of a VME-based computer has a wide range of suppliers to turn to for add-on peripheral boards. Indeed, some parts of the Sun-3 computer are not made by Sun; for example, the SMD controller is made by Xylogics. And Dual Systems sells its VME boards to other system builders. Memory boards often have specialized interfaces to the CPU, but most peripheral boards are portable. If you can add a driver to your UNIX kernel, you can add the peripheral to your system.

There are many other reputable makers of VME boards and systems. I list these companies because I know their products, and because they are UNIX vendors.

And their machines are fast. Sun-3 systems offer excellent price/performance. Dual's systems are good, although I haven't yet had a chance to review the speed of their 68020 systems. While the bus design is not the only factor in building a fast computer, a slow bus can limit performance as effectively as any other factor. These vendors' systems, and others that are coming to market prove that it's possible to build fast computers using the VMEbus.

For more reading, see *The VMEbus Specification*, Rev C.1 (or later), publication number HB212 (no ISBN), available from Motorola and elsewhere.

MAKE—SOME UNWRITTEN RULES

In day-long talks I give on Effective UNIX Programming, I spend considerable time talking about unwritten rules—that is, rules that are followed almost instinctively by old-timers but that don't appear in the manuals. Because they're not in the manuals, such rules are often missed by new-

comers. Following them will make your software behave as part of a consistent whole with the rest of the software distributed in the UNIX community.

I will list some of these unwritten rules at intervals in my column. For starters, here is a list of rules for the make utility. These won't make much sense unless you know about make; check out the manual or see my discussion in previous columns for an introduction to this utility.

1. If there is a rule to install the generated program in a bin directory, this rule should be called `install`—not `cp`, not `copy`, not `ins`, not `putit`, but `install`. Software that is consistent is a lot easier to use than software that isn't.

2. If there is an `install` rule, it should depend on the targets that you want to install.

Typing "make install" should do the right thing; it should ensure that the program is built and up to date before you try to install it.

3. There should be a make rule to delete transients such as `.o` files. This rule should be called "clean." You should be able to say "make clean" to put the directory back in a moderately pristine stage.

4. The most important target should be first, since the first target is the default. Typing "make" with no arguments should do something intelligent. Beware of using `.DEFAULT`; it has some unusual side effects on some versions of UNIX.

5. A rule that runs some common software tool such as lint should have the name of that program. Calling the lint rule "lint," not "check" or "verify," is just a matter of saying what you mean in a way that will be understood most readily by another programmer.

6. Every makefile that generates one or more C programs should have a lint rule to run lint on the set of functions in the program.

7. As with any programming language, symbolic constants should be used at the start of the makefile to identify variables, such as `BINDIR` (or `BIN_DIR`) for the directory in which the binary programs get installed, `MANDIR` for the manual page directory, etc. These are much easier to find than path-names scattered throughout the makefile.

8. No make rule should unexpectedly remove or overwrite source language files.

Many of the unwritten rules that I am

trying to preserve are concerned with making software do the right thing—that is, with making UNIX software behave consistently.

SYSTEM V LINE TURNAROUND

Owners of many small UNIX systems want to have a single telephone line that is used for dialin and dialout consecutively. The System V developers have realized this, and now offer a getty variant called uugetty that, combined with a modified cu program, allows a line to turn around automatically. But it's not yet in all the UNIX boxes in the field. Here is a technique you can use on early System V systems and even, I suspect, on System III.

The overall plan is to use two different init states, one in which the line is answered and another in which it's not. I use states 2 and 3, respectively. You need a way to flip from one state to another quickly and easily without disrupting existing logins or dial-outs. My solution to that is a couple of shell files, "dialin" and "dialout." You also need a way to prevent uucp from trying to dialout on the line when it is in dialin mode. My solution to that is the lockit program.

First, is the "inittab" entry I use for my terminal line (shown in Listing 1). The lines beginning with # are comments to init. The first column is just unique names. The second column—2, in both cases—is the init state. That says that in init state 2, both /etc/lockit and /etc/getty will be run; when init is changed to any state other than 2, both programs will be terminated automatically by init. Lockit just creates a dummy lock file. Getty is the program that monitors the phone line, waiting for somebody to try to log in. On my system, /dev/cul0 and /dev/tty03 are links to each other; cul0 is what you had to name your dialout line when I learned UNIX.

You could get by with just the above and manually flip by typing

```
telinit 2
```

or

```
telinit 3
```

to turn the line around. But you'd have to be careful not to do so when anybody was logged in (the user would be kicked off) or when anybody was doing a cu or kermit on the line. Here is a shell file called dialout that makes sure nobody is logged in; if not, it goes ahead and calls init to change state.

```
: dialout -- switch shared line to
dialout
: . . . if nobody is logged in on
it.
```

```
who|grep tty03 && exit
```

```
telinit 3
```

The && exit is a simple way of terminating a shell script when something succeeds that shouldn't. Note that this script relies on grep correctly reporting exit status.

The reverse operation is dialin; if nobody is using cu or kermit, and uucico isn't running, it is safe to enable dialins on the line.

```
: dialin -- switch shared line
to dialin ps -ae | egrep '
(uucico|kermit|cu)' && exit
/etc/init 2
```

By now you've probably realized that you have to be the superuser to run these programs. But set-userid shell scripts are inherently unsafe. What can you do? You can use a simple C program that is setuid, takes no arguments, and passes control to the shell file. If you're the only user on your system, you don't need this—just do

```
su root -c 'dialin'
```

But for a small multiuser system, you need some way of keeping control while allowing other people to dial out. This is shown in listing 2.

Standard() is a local function that standardizes PATH and other important environment variables, to head off certain security attacks. Just compile this program twice, once for dialin and once for dialout.

The lockit program is a page of code that does the obvious: check for the lock file, create it, and pause; catch SIGTERM and remove the lock. This program should also be in C and must be setuid to uucp to create a lock in the uucp directory. If you need a copy, just send me a uucp address or a return-addressed envelope.

There are certainly better ways of solving this. Later releases of UNIX will certainly include them. The best way, from the phone company's point of view, might be to install a second phone line.

That's all for this month. I welcome letters and electronic mail on these and other topics, especially suggestions for future columns. §

VENDORS

Dual Systems
2530 San Pablo Ave.
Berkeley, CA 94702
(415) 549-3854

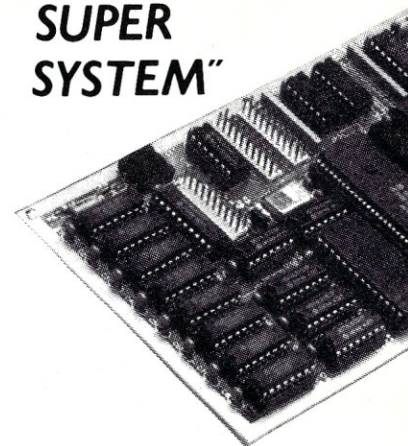
Integrated Solutions, Inc.
1140 Ringwood Ct.
San Jose, CA 95131
(408) 943-1902

Sun Microsystems
2550 Garcia Ave.
Mountain View, CA 94043
(415) 960-1300

Turn to page 81 for Listings

Byte Magazine called it.

"CIARCIA'S SUPER SYSTEM"



The SB180 Computer/Controller

Featured on the cover of Byte, Sept. 1985, the SB180 lets CP/M users upgrade to a fast, 4" x 7 1/2" single board system.

- **6MHz 64180 CPU**
(Z80 instruction superset), 256K RAM, 8K Monitor ROM with device test, disk format, read/write.
- **Mini/Micro Floppy Controller**
(1-4 drives, Single/Double Density, 1-2 sided, 40/77/80 track 3 1/2", 5 1/4" and 8" drives).
- **Measures 4" x 7 1/2"**, with mounting holes
- **One Centronics Printer Port**
- **Two RS232C Serial Ports**
(75-19,200 baud with console port auto-baud rate select).
- **Power Supply Requirements**
+5V +/-5% @500 mA
+12V +/- 20% @40mA
- **ZCPR3 (CP/M 2.2/3 compatible)**
- **Multiple disk formats supported**
- **Menu-based system customization**

SB180-1
SB180 computer board w/256K bytes RAM and ROM monitor
..... \$299.00

SB180-1-20
same as above w/ZCPR3, ZRDOS and BIOS source..... \$399.00

-Quantity discounts available-

NEW

COMM180-M-S
optional peripheral board adds 1200 bps modem and SCSI hard disk interface.

TO ORDER
CALL TOLL FREE **TELEX**
1-800-635-3355 **643331**

For technical assistance or to request a data sheet, call:
1-203-871-6170



Micromint, Inc.
4 Park Street
Vernon, CT 06066

The Scientific Computer User

by A. G. W. Cameron

FORTRAN to DOS Epsilon

FORTRAN TO DOS

Have you ever wished that you could compile and link a FORTRAN program and have it accept parameters from the command line when you run it? Or that you could play some fancy tricks with your video display? Or that you could manipulate bits? Or do any number of other things from within FORTRAN that you have until now only been able to do using DOS commands or C or assembly-language programs?

Actually, this has always been possible: you could interface FORTRAN programs to assembly-language routines, but this has always been something of a hassle. Now it is easy with the FORTRAN to DOS Interface (\$59.95 plus \$4.00 handling from Kris Jamsa Software). This is a set of FORTRAN-callable subroutines and functions that perform these various functions. The vendor claims there are more than 85 of these routines; I counted 87.

The file-manipulation routines enable you to open, delete, rename, and manipulate file dates and attributes, and do other things with files and directories. Under the heading of hardware control, you can initialize the printer and get its status, find your current disk drive and change it, find the memory size and get free disk space, and read and write bytes and words to anywhere in memory. With video control, you can change screen colors, read and write pixels, scroll a region, set character attributes, and freely manipulate the cursor, among other things. System-interface routines enable you to get and set date and time and interrupt vectors, print the screen, define keys and do other keyboard processes, handle reboot and similar processes, play with segments and flags within DOS, and retrieve command-line parameters.

Many of these processes are ones that you would take for granted to be available as FORTRAN routines on mainframes and minicomputers. It is sad commentary on current FORTRAN vendors for PCs that

very few of these routines have been provided by them.

The vendor provides two floppy disks providing these routines separately for the Ryan-McFarland, Microsoft, and Lahey FORTRAN compilers. You normally would just add the FTODOS library to your hard disk, but the vendor has also supplied complete FORTRAN and assembly-language source code for all the routines. Thus, you can customize the routines and also learn a lot of interfacing techniques as well with these sources.

It is obvious that many DOS services would not be included among the routines because it would make no sense to do so. Also, not every routine uses a DOS service; for example, changing colors on the screen is accomplished by sending ANSI escape sequences. Furthermore, some of the routines require DOS 3.*; unfortunately, the very useful routines that retrieve command-line arguments are among these.

EPSILON

As I write this, I am awaiting delivery of a Sun 3/260 workstation. This is at the top end of the microcomputer spectrum, and I shall have some things to say in future columns about its characteristics and why I chose it. For now, however, I want only to discuss an editor that I have been using in anticipation of the arrival of the Sun.

The Sun utilizes the UNIX operating system. Unfortunately, that rules out any possibility of using PMATE, my favorite editor (see my discussion of this editor in the July-August, 1986, issue of *Micro/Systems Journal*). For years people have been complaining about the lack of good UNIX software. There are two "official" UNIX editors. One is ED, which is the UNIX equivalent of EDLIN of MS-DOS infamy. Need I say more about that one? The other is VI, which has the advantage of being a screen editor, but I am unable to think of anything else good to say about it. There is also EMACS, a public-domain editor, which has the reputation of being the best that one can get under UNIX. The DOS equivalent of this is Epsilon (from Lugaru Software; I paid \$169 for it by mail order). I purchased this editor in order to become familiar with it in anticipation of getting EMACS for the Sun, and in order to be able to use essentially the same editor on PCs that I would be using on the Sun.

EMACS is written in LISP, but Epsilon is

written in C. It also has extensive capabilities for customization, giving the sources for the various types of operations in what it calls the Epsilon Extension Language, which is almost indistinguishable from C itself. I have not yet tried to customize Epsilon, since I wanted to have some experience with EMACS on the Sun first, and then to customize the two forms of the editor together.

Epsilon has a vast array of commands. Some of these are "bound" to keys on the keyboard, so that they form instant commands. You call others by entering a form of command mode and typing enough of the (usually lengthy) command name to make it uniquely recognizable. The editor makes good use of the keypad keys, by themselves and in combination with the Control and Alt keys. It makes very poor use of the function keys, placing on them commands that are rarely used. In general I dislike commands that require me to depress Shift or Control or Alt in combination with other keys; I would prefer single keystroke commands. However, Epsilon operates by entering ordinary keystrokes as text under most circumstances, so that in general something must be used in combination with ordinary keys to execute commands. That is why I am unhappy that more common commands are not implemented on the function keys.

PMATE suffers from the limitations of its CP/M background in having only a very limited text buffer—so that a lot of slow buffering to disk is necessary—and in not allowing simultaneous display of two or more windows on the screen. To me the most immediate benefit of Epsilon is to be rid of these two limitations. Thus, already I find myself bringing up Epsilon to deal with long files and in order to see two files or two parts of the same file at the same time. Actually, Epsilon can display 12 windows at once, each containing one line of text and one status line. I do not foresee needing that much flexibility.

Epsilon can use macros in the form of sequences of keystrokes which can be stored and executed. Unfortunately, this facility falls far short of that available in PMATE, since Epsilon lacks the conditional statements available in the PMATE command language. Hence, even if Epsilon becomes my editor of choice, I will still need PMATE to do more complicated editing with macros.

The Epsilon manual is terse and lacks

very many examples and clearly is written for users with a programming background. The help facility is useful only if you can remember the often-complicated name of the command you want to see defined. Fortunately, Epsilon comes with a variant on the standard EMACS tutorial, and in fact that is very useful for getting started.

PMATE writes directly to the screen, and so screen changes appear blazingly fast. Epsilon appears to "obey the rules" in using DOS services, so that screen updates are noticeably slower, but they are fast enough not to be really annoying. However, one feature that *is* annoying is the lack of automatic reformatting of the text as changes are made, as is done in PMATE. You must reformat paragraphs or regions manually, and the reformatting procedure takes rather a long time. The procedure must be rather inefficient. Fortunately, since most writing I do is formatted for use with T_EX, having very uneven lines in a paragraph does not affect the final appearance of the output, so reformatting is unnecessary except for the sake of neatness.

There is a considerable amount of awkwardness in erasure and cutting and pasting operations in Epsilon. Erasure of single characters is usually not recoverable, presumably on the mistaken idea that a missing letter will always be obvious. Cutting and pasting involves use of a "kill" buffer: outline a region with a mark and the cursor, copy or move the region to a buffer, move the cursor to the new location, and fetch the text from the buffer. That sequence is fairly standard. But the default configuration is only one kill buffer. You must go to the effort to create additional kill buffers if you want to be able to save several sequences of kills. The PMATE procedure is much better: have one large kill buffer and save there everything that is deleted, including individual characters, as long as there is room. Also, have many additional buffers (by default) into which text can be moved and in which editing can be done. These are the PMATE equivalent of windows, suffering only from the lack of ability to be displayed more than one at a time on the screen. Epsilon can create lots of buffers, and you can attach these at will to a screen window for editing, so in that respect it is somewhat similar in function to PMATE.

The bottom line for me is that I find Epsilon to be a generally acceptable editor, one with some features that are superior to those in PMATE. §

A.G.W. Cameron is Professor of Astronomy at the Harvard-Smithsonian Center for Astrophysics, 60 Garden Street, Cambridge, MA 02138.

ADDRESSES

Kris Jamsa Software, Inc., P.O. Box 26031,
Las Vegas, NV 89126.
Lugaru Software Ltd., 5740 Darlington Rd.,
Pittsburgh, PA 15217; (412) 421-5911.

Unix Code

Continued from page 79

LISTING 1

```
# stuff below here == kludges due to single phone line
# note - tty03 == cul0
lk:2:once:/etc/lockit -w cul0 # keep tip and uucico down
03:2:respawn:/etc/getty tty03 du_1200 # Shared line
# for init state 3, both lockit and getty disappear, allowing dialout
```

LISTING 2

```
/*
 * setuidsh - setuid front end to sh script
 * set up as
 * cc -DSHPROG=
 * mv foo $HOME/bin/foo
 * chmod u+s $HOME/bin/foo
 */

#include <errno.h>

main(argc, argv)
int argc;
char *argv[];
{
extern int errno;

standard(); /* for a clean environment */

setgid(getegid());
setuid(geteuid());

execv(SHPROG, argv);
#ifdef USG
if (errno == ENOEXEC) { /* Sys V can't exec shell files, darn */
execl("/bin/sh", "sh", "-c", SHPROG, 0);
}
#endif
perror(SHPROG);
exit(1);
}

-----
END OF CODE
/bin/sh", "sh", "-c", SHPROG, 0);
}
#endif
perror
```


PC/Blue Report



by Hank Kee

M/SJ ownership has changed but the editorial management remains the same. In future issues of *M/SJ*, there will be a new public-domain software column that will combine Sig/M and PC/Blue Report. Steve Leon, my amiable colleague, will be writing the column. I will continue to edit the PC/Blue library. This will give me time to do some comparative product analysis, which will appear in future issues.

THE SHAREWARE CONCEPT

This, my final column on PC/Blue public-domain software, will be a report to the readers from a different perspective. There has been much discussion about the merits of shareware software. This is a class of software in which authors encourage users of their software to copy and distribute the software without charge. These programs are copyrighted and contain requests for donations. These programs are normally found on remote bulletin-board systems and distribution libraries such as PC/Blue. Their quality is generally quite high. Most have complete documentation. In some cases, such as modem-communications programs, they are the equal of commercially available programs.

When I was librarian (and founding editor) of the Sig/M library of CP/M public-domain software, I only accepted programs for inclusion in the library that included source code and had no distribution restrictions. This was the hacker credo at the time. Although some of the programs may have contained a copyright, permission was always sought and received to distribute them freely for non-commercial use. Steve Leon, the current editor of the Sig/M library, has continued this tradition. The concept of collecting, editing, and distributing the PC/Blue library of PC programs was started in the same tradition.

Then, the late Andrew Flugelman came up with the idea of shareware. His popular PC-Talk communications program was the first such program. His idea for marketing his program was simple. If you tried it, liked it, and decided to use it, he requested a voluntary donation. His concept is now being used by countless aspiring entrepreneurs, so that now most of the contributions to the PC/Blue library are shareware programs.

Each year it becomes more difficult to get started in the software business. Where just a few years ago you could start on a shoestring, today that effort requires large amounts of capital. Marketing (promotion and distribution) has become more important than performance. The result is that many good programmers are discouraged, and their programs never get seen or used by others.

Distribution of software through the PC/Blue and other public-domain software libraries, as well as remote bulletin boards, has given many authors opportunities they would not have had otherwise. For many, it has brought programming talents to the attention of established software publishers and led to jobs they would not have gotten otherwise. The donations have also encouraged many to develop more good software and put it into the public-domain.

Many of the programs that started in the public-domain are now also sold commercially. For example, the 3×5 Information Management program is now sold by MultiMate, a subsidiary of Ashton-Tate, under the name On-File. PC-File, the first shareware database-management system, was an immediate hit, and allowed the author, Jim Button, to quit his job and found his own company. Another program that made the successful transition is PC-Write. This, along with a few other noteworthy programs, makes up the Brown Bag company. Although they have gone commercial, many of these programs continue to be available in the public-domain software sector.

Corporations are no longer looking at public-domain and user-supported programs with disdain. Many of the larger corporations are responsible for the revenues received by user-supported authors.

The software marketing cycle has now come to full circle. Many small, independent software distributors advertise commercial software for as low as \$.75 (yes, that's correct) a program. These programs are not being offered for a voluntary donation, but are sold at an amount so nominal that it is tantamount to free software.

NEW PC/BLUE RELEASES

The following are recent releases in the PC/Blue library of public-domain and user-supported software:

Volume 238

Statistical Process Control v1.1 manufacturing and assembly control

Volume 239

Images from the Mandelbrot Set

Volume 240

CalTech Utilities

CTRLALT v1.0—a set of resident utilities taking roughly 6K RAM

DFIX—tests any IBM-compatible disk or diskette for bad sectors, offers to mark bad sectors if not in use by a file

FM—a visual hex-file editor

KEGELUNIX—a set of small utilities modeled after features of UNIX

NANSI—a replacement and enhancement of ANSI.SYS, with faster screen writing, etc.

SEND—a replacement for ECHO, supporting an extension of the prompt metastrings

Volume 241

Managing Money with Your IBM PC

Stock Trader v2.0—maintains selected stock prices

Disk File Manager v1.0—maintains log of disk files

The Program Wizard v1.0—program start-up menu shell

Volume 242

AT&T 6300 Technical Notes

AT&T 6300 Clock/Calendar Utilities—Set 80 × 50 Video on AT&T 6300

Volume 243

QModem v2.20

Volume 244

PC-Class—a teacher's gradebook by Aiga Publications

COVER26—produces diskette cover listing

MULTGLED—multiple global editing of files

MEGASORT—text file sort utility

Volume 245

Structured Programming Language v1A

Volume 246

Kermit v2.29—modem communications system

Volume 247

Turbo Pascal I/O routines
Spelling Checker v1.2
Programmable RPN Calculator v1.4
Othello v3.0

Volume 248

Micro Development Software BASIC utilities
Advanced Lotus 1-2-3 tutorial
Symphony command language
ARTificial Art

Volume 249-250

RBBS-PC v14.1d—remote bulletin-board system

Hank Kee is the librarian for the PC/Blue public-domain software library. He collects, assembles, and checks all the software issued by PC/Blue and then compiles and edits the programs into the released volumes. §

PC/Blue disks are available from the New York Amateur Computer Club, Inc., Box 106, Church Street Station, New York, NY 10008. Price is \$7 per volume, which includes media, postage, and handling. On foreign orders, please add \$2 per disk. A printed directory is also available (\$5 U.S., Canada & Mexico; \$7 foreign).

THE MISSING LINK

PC-925 TURNS YOUR PC INTO A
FULL FUNCTION TELEVIDEO-925

FEATURING:

- 4 VIDEO PAGES
- SUPPORTS 25th LINE
- 5 VIDEO ATTRIBUTES
- PROTECTED FIELDS
- 40 PROGRAMMABLE FUNCTION KEYS
- FILE TRANSFER
- AUTO DIAL/LOG ON
- KEYBOARD MACROS
- "MOVE" FILE EXCH
- TVI/WS SWITCHABLE CURSOR PAD CODES

Postage Paid **\$79⁹⁵**

MS-DOS & PC-DOS COMPATIBLE
PUT YOUR PC'S TO WORK ON YOUR MULTIUSER...

TELEVIDEO® TELEVIDEO SYSTEMS  **DIGITAL DECISIONS SYSTEMS** WORDSTAR® MICROPRO INC.

229 Lakeview Ave. • Clifton, NJ 07011
1-800-862-6233 Wait for tone, dial Ext. 0014762 (201) 772-6749

Call/Write for our **FREE 140 page Color Catalog**

MORE... VALUE and PERFORMANCE

with Mitek's Relocatable Z80 Macro Assembler and Z80 Symbolic Debugger

NEW Z80 Symbolic Debugger

- Only \$49.95 plus shipping.
- Screen oriented with a simultaneous display of instruction mnemonics, register, stack, and memory values.
- Breakpoints may be set on any combination of fixed memory address, register values and/or memory values.
- Uses Digital Research compatible SYM files.
- Supports Hitachi HD64180.

Relocatable Z80 Macro Assembler

- Only \$49.95 plus shipping.
- 8080 to Z80 Source Code Converter.
- Generates Microsoft compatible REL files or INTEL compatible hex files.
- Compatible with Digital Research macro assemblers MAC & RMAC.
- Generates Digital Research compatible SYM files.
- Conditional assembly.
- Phase/dephase.
- Cross-reference generation.
- Full Zilog mnemonics.
- INCLUDE and MACLIB FILES.
- Separate data, program, common, and absolute program spaces.
- Supports Hitachi HD64180.
- Z80 Linker and Library Manager for Microsoft compatible REL files available as an add-on to Assembler.

ATTENTION Turbo Pascal Users:

Assembler will generate Turbo Pascal in-line machine code include files.

TO ORDER. CALL TOLL FREE: 1-800-367-5134, ext. 804
For information or technical assistance: (808) 623-6361

Specify desired 5 1/4" or 8" format. Personal check, cashier's check, money order, VISA, MC, or COD welcomed.

P. O. Box 2151
Honolulu, HI 96805

MITEK

PRICE LIST

Z80 Macro Assembler: \$49.95
 Assembler, Linker, and Library Manager: \$95.00
 Manual Only: \$15.00
 Z80 Symbolic Debugger: \$49.95
 Manual Only: \$15.00
 Assembler, Linker, Library Manager, and Debugger: \$134.95

Include \$5 for shipping and handling.

Z80 is a trademark of Zilog, Inc. MAC and RMAC are trademarks of Digital Research, Inc. Turbo Pascal is a trademark of Borland International, Inc.

In the SIG/M Public Domain

by Stephen M. Leon

Steve Leon is the SIG/M Disk Editor. He assembles, compiles, and edits all of the SIG/M public-domain software disks. Thus, he speaks with the greatest authority as to what is going on in the SIG/M public-domain software area.

Further, Steve operates a remote bulletin-board system for the Amateur Computer Group of New Jersey. Readers can contact Steve via his RBBS ((201) 886-8041). The system also contains a great deal of public-domain software available for downloading.

One of the three new SIG/M volumes (289) contains some of the software uploaded to the BBS. Some of it is not particularly new, but it slipped between the cracks. FBAD is such a program. It is an improved version of FINDBAD, the bad disk sector lockout program. SETDRU, to set drive, and user path in CP/M 80, and SFILE13, which searches all drives, user areas and libraries for a file, seem quite handy.

Also on the disk is a machine-specific program, by Joe Gugel, to allow Rainbow 100 users to create an MDRIVE, a smart copy program, and Star Trek in Nevada BASIC. (Why another Star Trek? As Brian Beard commented, there are a lot of versions around. But he was running Nevada BASIC and had to write one for it!) I anticipate receiving something like 6-megabytes of Turbo Pascal programs for the BBS; hopefully, some of that material will also make its way onto SIG/M disks.

David A. Danello donated a collection of FORTRAN subroutines and tools. They are to be found on SIG/M Volume 288. Also on the disk is a library containing an unrelated series of miscellaneous programs also provided by Dave. They include a Morse code decoder, a pretty good file-compare program, some interesting astronomy programs, and a mortgage-amortization calculator. I had a lot of fun with Stardate. When it first came in, I was not quite sure what an almanac from 2500 B.C. to 2500 A.D. would do. Ted Campbell's program gives you a whole slew of calendar, almanac, and astronomical data for the period. It is designed for the historian, the writer, or the curious. The file on SIG/M Volume 290 includes both CP/M and MS-DOS versions.

Also on 290 is a Kaypro-to-IBM main-

frame-transfer program. Not having either a Kaypro or access to an IBM mainframe, I could not test it. It looks good. The documentation is professional. The source code is there, so it can be used for machines other than Kaypros. Dave May did what appears to be a good job on it and, as he says in the documentation, it has been used without problems for more than a year.

CHANGES COMING

In the next issue, this column and the PC/Blue columns will be integrated into one column covering public-domain software. The new column will be broadened to cover other areas of public-domain software. It will even touch on public-domain software for the Atari ST and Commodore 128 and Amiga. Readers who access my BBS system ((201) 886-8041) will find a considerable amount of Atari, Commodore, and MS-DOS public-domain software, as well as SIG/M software, available there for downloading.

At last count, there were some 150 public-domain programs for the Amiga, including a variety of languages, graphics and sound programs, utilities, and communications programs. SIG/M does not yet have a way of releasing this material in an Amiga disk format, but all *Micro/Systems Journal* readers with an interest in the Amiga are welcome to download this material and to upload their programs to the system.

On the system too are the recent SIG/M and PC/BLUE releases (all other SIG/M and PC/BLUE volumes are available on request). Also on the system is much of the source code from *Micro/Systems Journal* articles. Much of the same material is on the BBS system Hank Kee runs for the New York Amateur Computer Club ((718) 539-3560 and -3338). The big difference between the systems is that the NYACC board requires registration confirmation to upload and download. The ACGNJ board does not require registration.

SIG/M volumes are available on 8-inch SSSD disks for \$6.00 each (\$9.00 foreign) directly from SIG/M, Box 97, Iselin, NJ 08030. Volumes may also be ordered in most 5-inch formats (other than Apple, Commodore 64, or high-density format) for \$7.00 each (\$10.00 foreign). There is

an additional charge of \$2.00 for formats that require more than one disk to hold a 240K volume (such as SSSD formats). Printed catalogs are \$3.00 each (\$4.00 foreign). Disks in a variety of formats may also be obtained through the worldwide SIG/M distribution network. The distributor list is included with the printed catalog. A disk version of the catalog (Volume 00) is available for \$6.00. It also contains the distribution list. Many bulletin boards have the software for downloading, and most new releases are available on the CP/M SIGs on CompuServe and GENie. §

NEW SIG/M PUBLIC DOMAIN SOFTWARE RELEASES

Volume 288

Utilities for Microsoft FORTRAN-80
Morse code, astronomy, etc.
FORTRAN subroutines
FORTRAN tools

Volume 289

Utilities on the ACGNJ BBS
Compare disks and files in CP/M
Z80 HEX and ASCII file editor
FBAD Bad sector locator
SFILE13 Searches all drives for a file
Nevada BASIC version of Star Trek
Smart copy program
Create MDRIVE on Rainbow 100
Set drive and user path in CP/M 80

Volume 290

Stardate Almanac
Kaypro to Mainframe Transfer program

ELECTRICAL ENGINEERING PUBLIC DOMAIN LIBRARY STARTED

A public-domain software library for electrical engineers has been started by Gerry Harrison, 36 Irene Lane East, Plainview, NY 11803, (516) 822-1697. Four disks are already available. They are in PC format, contain programs in BASIC and Pascal, and cover topics such as optimization of networks, filter design, matching network design, and transient and network analysis.

OUT WITH THE OLD, IN WITH THE NEW

If you are a dealer with old Molecular, OSM, TeleVideo, Altos, Northstar, Dynabyte, CompuPro, IBS, JC Systems, or any other 8-bit or 16-bit S-100 multiuser installation, you can now replace these systems and save your customer thousands of dollars on new high-speed 16-bit S-100+ technology.

For a limited time, when you send us any of these 8-bit or 16-bit computers, L/F Technologies will credit you trade-in dollars toward the purchase of new, high performance LFT 16-bit computer systems.

We will trade you board for board, drive for drive on all complete systems — regardless of present operating condition!

HERE'S WHAT YOU GET

- 1) Huge dollar credits for your old hardware when you buy new, high performance LFT technology.**
- 2) A ONE YEAR WARRANTY on all new upgrades purchased.**
- 3) Super performance featuring 10MHz processors and high speed parallel bus. (We also offer 8MHz, 8-bit systems)**
- 4) Options to add L/F Power, L/F Net, more storage, and more users.**
- 5) Factory technical support and service to replace discontinued company support.**

There are thousands of people with CP/M, MP/M, Concurrent DOS, N-Star, and TurboDOS computer systems, who have been abandoned. Now's the chance to adopt orphans full of growth potential.

Call L/F Technologies today for trade-in prices and upgrade details.



MULTIUSER COMPUTERS FOR BUSINESS AND INDUSTRY

L/F TECHNOLOGIES / CORPORATE OFFICES: 2800 LOCKHEED WAY, CARSON CITY, NEVADA 89701 • TELEPHONE: (702) 883-7611 • TWX: (910) 395-6051

New Products

When contacting vendors, please mention that you read about their products in Micro/Systems Journal.

S-100 HARDWARE

WORKSTATION RUNS IBM SOFTWARE

The Poppy workstation runs IBM PC software in an S-100 environment. It includes a processor board, smart cable, keyboard, and monitor. Based on an 80188 CPU running at 8 MHz, 512K system RAM, 256K display RAM, and 4K ROM, it runs MS-DOS supported by the TurboDOS network operating system. Users can run virtually all popular IBM PC programs such as Lotus 1-2-3, Wordstar, and dBASE III, in a multiuser, multiprocessing environment. Monochrome, CGA, EGA and Hercules display options with automatic switching between modes are available. The Poppy workstation is also Novell-compatible.

The Poppy workstation is \$1,995 and is available from Charter Information Corp., 2421 Rutland, Austin, TX 788758, (512) 835-1111.

14-USER MULTIUSER SYSTEM

The CompuPro MP14 is a multiuser S-100 system supporting up to 14 users. It uses 8 CPUs (80286 and 80186 running at 10 MHz with no wait states), 6.5Mb of RAM, disk caching, high-speed hard disks, and Concurrent-DOS. It is intended for high-performance environments in which all users are accessing shared databases at the same time. It provides PC-DOS compatibility as well as compatibility with all existing CompuPro systems (Model 10, 286/40, 286/80 and C3).

The CompuPro14 is \$18,995 and is available from CompuPro/Viasyn Corp., 26538 Danti Court, Hayward, CA 94545, (415) 786-0909.

NETWORK PROCESSOR

The ICM CPS-16F is an S-100 network processor designed for use in multiuser/multiprocessing systems. It is capable of running CP/M-80/86 and PC/MS-DOS. It contains an NEC V30 microprocessor and is compatible with 8- and 16-bit S-100 systems with extended address capability. When used with ICM's MS-100 software and an M/STER PC-compatible terminal, it allows you to run PC/MS-DOS programs. You can run TurboDOS and PC/MS-DOS on the same system.

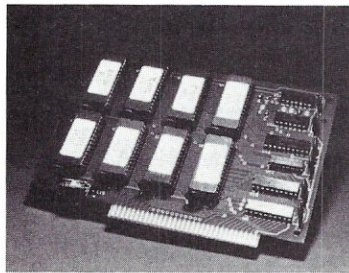
The CPS-16F has 1Mb of RAM (no wait states), real-time clock, and two serial and two parallel ports. For additional information, contact InterContinental Microsystems, 4015 Leaverton Court, Anaheim, CA 92807, (714) 630-3714.

PC-COMPATIBLE HARDWARE

MULTIUSER/MULTITASKING ADD-IN BOARD

The PCplex is a plug-in board for PC/XT/AT systems that allows three users to work on the same system by attaching two ASCII terminals to existing ports. The terminals can be remotely connected via modem. It allows up to 99 concurrent tasks and switching of computer screen and keyboard to different tasks. It supports time slicing, Novell record and file locking, and up to 16Mb of memory via bank switching. It provides user ID and password security, protected file transfer, and a line analyzer of serial ports.

A menu-driven installation program is provided, and the operating system resides in ROM. The basic model is \$199. A version with two COM ports and an advanced system programmers' utility package is \$399. For information contact Xmark Corp., 3176 Pullman St., #119, Costa Mesa, CA 92626, (714) 556-9210.



HIGH-RESOLUTION/HIGH-SPEED GRAPHICS SYSTEM

The Retrograph One VT640 is a monochrome graphics system for IBM PC/XT/AT and compatible systems. It includes a 19-inch monitor displaying 1280 x 960 pixels at one million pixels per second. It also provides intelligent printer output. Vectors, arcs, circles, and paint functions are done in hardware. Features include true zoom and true text over graphics. Plotter output is also provided. Compatible hosts include VT100, DOS, Tek 4010/4014, Plot 10, and Hercules. A mouse option is also available.

The price is \$2,995, and it is available from Retrographics Inc., 65 Commerce Rd., Stamford, CT 06902, (203) 637-1708.



67 TO 268MB BACKUP FOR NETWORKS

The ADIC Model 532 is a .25-inch cartridge-tape subsystem for large networks using the Novell NetWare or Microsoft MS-NET software. The Model 532 stores 67Mb. Up to three Model 530 tape-expansion subsystems can be daisy-chained to a 532 host to provide up to 268Mb of tape storage. The tape unit emulates a disk drive and provides random file access. Programs can be executed directly from tape, allowing the user to default to the tape system in the event of a total disk failure. Software is provided for installation and operation.

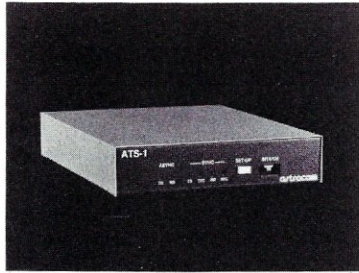
The Model 532 lists for \$3,490, and the Model 530 lists for \$2,640, from Advanced Digital Information Corp., Box 2996, Redmond, WA 98073, (206) 881-8004.

OTHER PRODUCTS

ASYNCHRONOUS TO SYNCHRONOUS CONVERTER

The Astrocom ATS-1 is a new ASCII asynch-to-sync converter. It selectively removes stop bits and synchronizes the remaining bits with the receive clock from the synchronous transmission device. It provides auto-speed adjustment to support data rates from 12 to 19.2K bps and 5- to 10-bit data character sizes. It includes a 64-bit buffer, break detection, and front-panel LEDs.

The ATS-1 is \$395 and is available from Astrocom Corp., 120 W. Plato Blvd., St Paul, MN 55107, (612) 227-8651. §



MODEM/SCSI PERIPHERAL BOARD

The Micromint COMM 180 is an expansion board for the SB180 computer. The 4 × 5-inch board adds the functions of modem and SCSI hard disk controller interface.

The modem is Bell 103-212A compatible 300/1200 baud with Dual Tone Multi-frequency (DTMF) encode/decode or pulse dialing, call progress monitoring, diagnostic capabilities, and voice synthesis capability allowing it to respond verbally to commands entered via touch tone. The modem addresses the bus directly and does not use the SB180 serial port.

The SCSI interface can be used with hard disk drives of up to 50Mb that have SCSI interfaces. The SCSI bus can also be connected to other SCSI interface devices (e.g., laboratory instruments).

Software is provided to operate the modem under the Z-System DOS and for integrating hard disk drivers.

The board may be purchased with only the modem or SCSI components with upgrade later. Prices start at \$150. Call (800) 635-3355 to order. Call (203) 871-6170 for technical information, or write to Micromint Inc., 25 Terrace Dr., Vernon, CT 06066.

19.2K-BPS COMPACT LINE DRIVER

The Prentice DLD is a line driver which allows asynchronous devices to communicate at up to 19.2-bps for up to 3 miles. Twisted-pair and male or female DTE or DCE interfaces permit connections to standard terminals, multiplexers or data switches. The Telco line interface can be either standard RJ11 jack or four-position terminal board. DLD conforms to Bell 43401 up to 9600-bps, and meets CCITT V.24/V.28 recommendations.

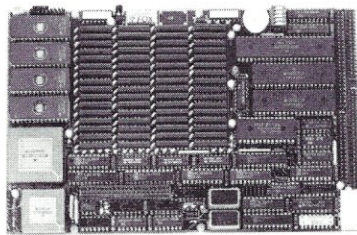
Cost is \$84 each in packs of ten. Contact Prentice Corp., 266 Caspian Dr., Box 3544, Sunnyvale, CA 94088, (408) 734-9810.

68020 SINGLE BOARD COMPUTER

The GMX Micro-20 is a single 8.8 × 5.75-inch board computer that combines 12.5 or 16.67 Mhz 68020 microprocessor and optional 68881 floating-point processor with

2Mb of 32-bit wide RAM, up to 256K of 32-bit wide EPROM, four serial ports, parallel port, 5¼-inch floppy disk controller, SASI peripheral interface, time-of-day clock with battery backup, and a 16-bit expansion connector. Included is a version of Motorola's 020Bug monitor/debugger and hardware diagnostics in PROM. Options include Unix-style operating systems, I/O expansion boards, color graphics interface, cabinet, power supply, and disk drives.

Cost for the 12.5 Mhz version is \$2750. Contact GMX Inc., 1337 W. 37th Place, Chicago, IL 60609, (312) 927-5510.



HITACHI 64180 IN-CIRCUIT EMULATOR

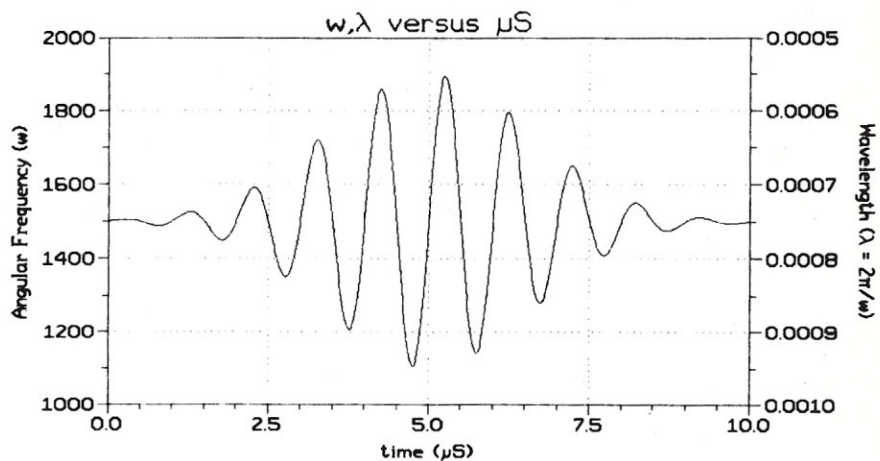
The Softaid ICEBOX is an in-circuit emulator for Hitachi 64180 systems. It provides full-speed, real-time emulation. Built-in automatic tests allow it to isolate problems. A software package is included to automate test and repair operations. It is transparent to the target system and does not alter the system.

ICEBOX performs the following functions: downloads hex and binary files, tests all target system RAM and computes ROM checksums, exercises all I/O ports and memory addresses, finds software faults in ROM and RAM, traces program execution, and provides MMU remap. It supports true hardware transient and permanent breakpoints.

Cost is \$1200. Contact Softaid Inc., Box 2412, Columbia, MD 21045, (301) 792-8096. §

GRAPHICS

starting at \$49⁹⁵



SCI-GRAF: Produces line, scatter, and high/low plots on Epson or IBM compatible dot-matrix graphics printers. Supports images up to 1680x1712 dots (over 3 million pixels!), log & linear scales, graphs in 7 colors (on a IX-80), batch processing. Requires MS-DOS 2 or 3, 256k ... \$99⁹⁵

FONTEDIT: Easy-to-use full screen font editor allows you to create and modify fonts for our SCI-GRAF program. Create Greek, math, and custom symbols in 3 sizes. Supports SCI-GRAF's proportional spacing. Requires IBM compatible 320x200 pixel on-screen graphics & IBM compatible keyboard \$39⁹⁵

GRAF 3.0: Produces bar, pie, line, scatter, and high/low plots on Epson, IBM, C:toh, or NEC compatible dot-matrix graphics printers. Supports 14 fill-in patterns, 8 point-plotting symbols, automatic scaling, labeling, and legend creation. MS-DOS 2 or 3, 192k \$69⁹⁵ CP/M-80: Requires 64k (54k TPA) \$49⁹⁵

MSC Microcomputer Systems Consultants

32 W. Anadamu St. Ste. 190 Santa Barbara, CA 93101 805-963-3412

THE SOFTWARE DIRECTORY

When contacting software publishers please mention that you read about their product in Micro/Systems Journal.

Program Name: PolyBoost

Requirements: PC/XT/AT or compatible

Description: A set of three memory-resident programs that speeds up disk access (via a disk cache), screen display (via direct screen writes), and keyboard input (via a typeahead buffer). The program is compatible with a variety of hardware and software.

Price: \$79.95 (60-day, money-back guarantee)

Publisher: Polytron Corp., 815 NW 169th Pl., Ste. 2110, Beaverton, OR 97006, (503) 645-1150.

Program Name: Power Tools

Requirements: PC compatible with 256K RAM

Description: A memory-resident hard-disk utility. Can locate a file in any directory, access programs located in other directories, "defragment" files, format a diskette while running another program, and execute DOS commands while running programs. Can copy or move files or subdirectories; view, edit or print any file or sector; locate any file or string on a disk; map disk usage; and restore erased data.

Price: \$50

Publisher: MLI Microsystems, Box 825, Framingham, MA 01701, (617) 926-2055.

Program Name: Attach

Requirements: TurboDOS System

Description: Enables TurboDOS users using any processor to attach their processors to any other processor on the same network. Can work between any two TurboDOS processors independent of the master/slave assignment. Extends TurboDOS capabilities by excluding multiple slaves from attaching themselves to the master processor. Allows collection of console outputs into a file for later review.

Price: \$150

Publisher: InterContinental Microsystems, 4015 Leaverton Ct., Anaheim, CA 92807, (714) 630-3714.

Program Name: GTP Development System, Version 2

Requirements: PC or true compatible, 256K RAM, Turbo Pascal V3.0, and two disk drives

Description: An application generator pro-

ducing error-free Turbo Pascal code for screen and database systems. Can create multiple screens, and allows database update and retrieval, automatic field edits, memory-mapped video, color, calculations, error handling, and context-sensitive help. Database manager uses a hashed attribute index scheme that automatically builds and maintains an index for each field. Global search supports wild cards and phonetics. Has a report generator and a menu generator for bundling applications together.

Price: \$150

Publisher: Allen, Emerson & Franklin, Inc., Box 928, Katy, TX 77492, (713) 391-8570.

Program Name: C-Index

Requirements: Lattice C Compiler, Computer Innovations C86, Microsoft 3.0, Consulair, UNIX/XENIX System V, Wizard, Manx Aztec, and Mark Williams.

Description: A data-management tool for C programmers. Provides full B+Tree indexing. Available in three versions: (1) C-Index/Files—includes object code and supports multikey routines for automatic memory management; (2) C-Index/Pro—includes both object and source code; applications can be distributed without royalty; (3) C-Index/Plus—includes transportable source code, no royalty, and telephone support.

Price: Evaluation, \$25; C-Index/File, \$99; C-Index/Pro, \$195; C-Index/Plus, \$395.

Publisher: Trio Systems, 2210 Wilshire Blvd, Ste. 289, Santa Monica, CA 90403, (213) 394-0796.

MEGABASIC
IS FOR SERIOUS SOFTWARE
DEVELOPMENT.
30 DAY FULL REFUND
SATISFACTION GUARANTEE
PRICE: \$395.00

- Full memory utilization (e.g. 500k arrays).
- 32 Bit. integers, up to 18 digit floating point.
- Extensive string processing (including large strings).
- Add your own language feature with ADA-like packages.
- Many features found in no other BASIC (e.g. vector processing).
- Fast execution and program development.
- Network support (3com, Novell, IBM-NET, PC-NET, VIA-NET).
- Runs under PC DOS, XENIX, CONCURRENT / MPM, PCPRO, Turbodos.

For more information call or write
Computer House, Inc.
P.O. Box 709 Woodacre, CA
94973
(415)453-0865

Program Name: CSharp

Requirements: PC/XT/AT or compatible and a C compiler (Microsoft, Lattice, Computer Innovations C-86, or Rational Systems Instant-C)

Description: A C library of support routines for data acquisition and control of hardware. Supports the Metrabyte Dash 8/16 and Data Translation DT2801/2808 analog I/O boards, National Instruments' GPIB-PC IEEE488 interface card, and Keithly Series 500 system.

Price: \$195 (limited to 25 units): \$975 (unlimited units).

Publisher: Systems Guild, P.O. Box 1085, Kendall Square Station, Cambridge, MA 02142, (617) 451-8479. §

d/MULTI
MULTIUSER dBASE
for
TurboDOS

TRUE File and Record Locking as easy as d-BASE-II. Unlimited users can perform the magic of dBASE in the program or interactive mode

- * TurboDOS 1.3 or 1.4
- * No Peeks or Pokes
- * System Date and Time Functions
- * Printspooler Controls up to 16 printers

Martian Technologies . . .
.CREATeIng Multi-users from
Single-users around the world

CALL FOR DETAILS

Martian Technologies
8348 Center Dr., Ste. F, La Mesa, CA 92041
(619) 464-2924

Micro/Systems Journal

Article Index 1985-1986

See page 93 for information on ordering back issues.

Vol-No	Months	Year	Title	Author	Vol-No -Pg
1-1	March/April	1985			
1-2	May/June	1985			
1-3	July/August	1985			
1-4	September/October	1985			
1-5	November/December	1985			
2-1	January/February	1986			
2-2	March/April	1986			
2-3	May/June	1986			
2-4	July/August	1986			
2-5	September/October	1986			
2-6	November/December	1986			
Assembly Language					
			Structured Programming With M80	Dennis Quinn	1-3-26
BASIC Language					
			Assembly Language Extensions for Microsoft BASIC	Ron Kreymborg	1-1-36
			Control Systems Made Simple	Lynwood Wilson	2-5-20
C Language					
			C Forum (Variable Size Arrays)	Don Libes	1-1-18
			C Forum (Writing a Translation Program)	Don Libes	1-2-16
			C Forum (Sets & Bit Operations)	Don Libes	1-3-22
			C Forum (1986 Obfuscated C Code Contest)	Don Libes	1-4-14
			C Forum (Context-independent Macros)	Don Libes	1-5-16
			C Forum (C Interpreters)	Don Libes	2-1-16
			C Forum (getopt-a subroutine for parsing command-line arguments)	Don Libes	2-2-16
			C Forum (International Obfuscated C Code Contest)	Don Libes	2-3-10
			C Forum (Casts)	Don Libes	2-4-8
			C Forum (Reading Input)	Don Libes	2-5-16
			C Forum (Keeping Track of malloc ())	Don Libes	2-6-20
			C and Godbout Disk Controller	Ed Heyman	1-2-46
			C Interpreters—Review	D Libes & G Morris	2-2-22
			C Source-Level Debuggers—Review	Jonathan Sachs	2-2-28
			Declare & Define C Variables In One File	Bill Rogers	2-4-66
			Declare & Define C Variables In One File	Ed Fields	2-6-10
			Faster Floating Point Math	N.T. Carnevale	1-5-46
Concurrent DOS (CCP/M)					
			Concurrent-CP/M Print Utility	Alex Soya	1-4-30
			Concurrent-DOS (Part I—Features, Architecture & Common Problems)	Alex Soya	2-3-44
			Concurrent-DOS (Part II—Processes & Their Data Structures)	Alex Soya	2-6-56

CP/M		
Add A RAM disk To A CP/M System	Howard Peters	2-3-66
Bringing Up CP/M-68K	J Calaway & B Hill	1-5-58
Bringing Up CP/M Plus	Sheldon Kolansky	1-1-20
CP/M Bus (Public Domain Hit Parade)	Bruce Ratoff	1-2-75
CP/M Bus (SB-180, DateStamper, Ztime-1 Z80ASM & DSD80 reviewed)	Bob Blum	2-2-70
CP/M Bus (Patches for Submit & Zsid)	Bob Blum	2-4-44
Enhancing CP/M-80	William Dudley	2-5-28
Extended Single-Density Disk Storage	Willis Howard	1-1-30
Loadable BIOS Drivers For CP/M	Cal Sondgeroth	1-2-66
More Loadable BIOS Drivers For CP/M	Ted Carnevale	2-4-72
New Tricks for CP/M2.2: Logical Name Translation	David Brewer	1-1-46
Syslib, Z3lib & Vlib	Richard Conn	2-5-72
Transfer Files Between CP/M & MS-DOS	Hank Volpe	2-1-66
ZCPR3 & How to Install It	Randy Reitz	1-2-42

Communications		
Setting Up An RCP/M System	Bob Blum	2-3-74

Data Base Forum		
dBASE-II Speed Techniques—Part I	Nelson Dinnerstein	1-5-28
dBASE-II Speed Techniques—Part II	Nelson Dinnerstein	2-1-82
dBASE-III Developer's Release	Nelson Dinnerstein	2-2-74
More dBASE-II Speed Techniques	Nelson Dinnerstein	2-3-82

Forth Language		
Local Variables	Thomas Reno	1-3-38

Hardware (General)		
First Look At 80386	Michael Weinreich	2-5-34
Interfacing Using SCSI Bus	Hul Tytus	2-5-46
Scientific Work Stations	A.G. Cameron	1-5-76

Hardware—PC/XT & Compatibles		
Building an IBM-PC/XT clone	Hank Kee	1-1-74
Build An S-100 to PC Bus Converter	John Monahan	1-2-24
Roll Your Own PC Clone—Part I	Sol Libes	1-2-36
Roll Your Own PC/XT/AT Clone	Sol Libes	2-2-66
Roll Your Own PC Clone—Part II	Sol Libes	2-6-24
The PC Bus	Dave Hardy	1-3-79

Turbocharge Your 8086/8088 Computer—Part I	Stephen Davis	1-5-32
Turbocharge Your 8086/8088 Computer—Part II	Stephen Davis	2-1-32
Assembling An AT Clone	Sol Libes	2-1-44

Converting From CP/M To MS/DOS On The PC	Robert Stek	2-2-80
Building An AT Clone	Leon Suchard	2-3-62
Power To The PC!	Stuart Jones	2-4-22
Experiences Of PC Clone Dealer	Al Levy	2-5-4
Speeding Up The PC/XT	Doug Severson	2-6-34

Cloning In Fast Lane (Review: PC's Limited 12-Mhz AT Clone)	Sol Libes	2-6-42
---	-----------	--------

Hardware—S-100		
Build An S-100 to PC Bus Converter	John Monahan	1-2-24
Build An S-100 EPROM Emulator	Robert Rioja	2-1-60
Build An S-100 HD64180 CPU Card	Roger Stevens	2-3-28
Build A Smart Keyboard Interface	John Monahan	2-4-34
C and Godbout Disk Controller	Ed Heyman	1-2-46
Macrotech MI-286 Review	Charles Strom	1-3-48
Peak 68K8-CP Review	D Hardy & K Jackson	1-5-66

Upgrading CompuPro I/O Boards	R.K. Gerson	1-4-42	Parade)	Bruce Ratoff	1-2-75
Hardware (Product Reviews)			SIG/M Volumes 210-217	Stephen Leon	1-1-14
Baby Blue	Robert Stek	2-2-80	SIG/M Volumes 218-223	Stephen Leon	1-2-12
CompuPro 10 Plus	Michael Guttman	1-4-48	SIG/M Volumes 224-227	Stephen Leon	1-3-12
CompuPro S-100 PC Video Board	Alex Soya	2-2-60	SIG/M Volumes 228-231	Stephen Leon	1-4-24
Earth Computer Turboslave-PC	Robert Stek	2-2-80	SIG/M Volumes 232-241	Stephen Leon	1-5-20
Holliston Challenger XT-186	Charles Strom	2-4-52	SIG/M Volumes 242-246	Stephen Leon	2-1-12
Lomas Data Products' S100-PC	D Hardy & K Jackson	2-2-58	SIG/M Volumes 247-251	Stephen Leon	2-2-12
Macrotech MI-286	Charles Strom	1-3-48	SIG/M Volumes 252-264 & 800-801	Stephen Leon	2-3-14
Magnum Digital PRO-180 & FD-100	Steven Kapplin	2-4-62	SIG/M Volumes 265-271	Stephen Leon	2-4-14
PC's Limited 12Mhz AT Clone	Sol Libes	2-6-42	SIG/M Volumes 272-281	Stephen Leon	2-5-82
Peak 68K8-CP	D Hardy & K Jackson	1-5-66	SIG/M Volumes 282-287	Stephen Leon	2-6-76
Slicer 80186	William Earnest	1-3-56	SIG/M & PC/Blue Distribution Points	Bob Todd	1-4-60
Lisp Language			PC-Blue Volumes 100-100	Hank Kee	1-1-12
16-Bit Lisp & Prolog Implementations—Part I	William Wong	1-1-62	PC-Blue Volumes 111-114	Hank Kee	1-2-14
16-Bit Lisp & Prolog Implementations—Part II	William Wong	1-2-56	PC-Blue Volumes 117-124	Hank Kee	1-3-14
Pascal Language			PC-Blue Volumes 125-137	Hank Kee	1-4-18
Interrupt Borrowing With Turbo Pascal	Stephen Davis	1-4-34	PC-Blue Volumes 138-147	Hank Kee	1-5-18
Intersystem Processing	Hank Volpe	2-1-66	PC-Blue Volumes 148-160	Hank Kee	2-1-14
Turbo Pascal Corner (writing a translation program)	David Carroll	1-2-20	PC-Blue Volumes 161-175	Hank Kee	2-2-14
Turbo Pascal Corner (handling data translations)	David Carroll	1-3-18	PC-Blue Volumes 176-187	Hank Kee	2-3-16
Turbo Pascal Corner (the TYPED constant)	David Carroll	1-4-20	PC-Blue Volumes 188-200	Hank Kee	2-4-16
Turbo Pascal Corner (advanced machine-level interfacing techniques)	David Carroll	1-5-12	PC-Blue Volumes 201-220	Hank Kee	2-5-78
Turbo Pascal Corner (support & Enhancement products)	David Carroll	2-1-18	PC-Blue Volumes 221-237	Hank Kee	2-6-80
Turbo Pascal Corner (Sorting)	David Carroll	2-3-20	Software (Product Reviews)		
Turbo Pascal Corner (Public Domain Utilities)	Stephen Davis	2-5-14	C Interpreters	D Libes & G Morris	2-2-22
Turbo Pascal Corner (Hi Speed Screen Output)	Stephen Davis	2-6-14	C Source-Level Debuggers	Jonathan Sachs	2-2-28
Turbo Pascal V3.0	David Carroll	1-3-72	C/nix Update	R.A. Langevin	1-3-59
MS/PC-DOS			CompuMagic Utility Package	Joseph Sabin	2-5-80
Configuring MS-DOS	Sol Libes	2-3-38	Concurrent PC-DOS	M Guttman & V Mills	1-3-66
Creating A Copy Protected Program	Edwin Thall	2-4-28	ConIX & ConIX Tools	Steven Kapplin	2-3-56
Implementing PC-DOS On Non-IBM Compatible Computers—Part I	C Cochran & K Sweger	2-1-26	CP/Emulator II	Robert Stek	2-2-81
Implementing PC-DOS On Non-IBM Compatible Computers—Part II	C Cochran & K Sweger	2-2-38	dataCURE	Bruce Ratoff	1-1-68
Interfacing to MS-DOS			dBASE III	Scott Patashnick	1-1-56
Part-I The Program Segment Prefix, Access to DOS, and How to Terminate A Program	William Wong	1-2-32	DSD80 (CP/M full-screen debugger)	Robert Blum	2-6-83
Part-II Dealing With Character Input & Output Functions	William Wong	1-3-62	Echelon's Z-System	Morris Simon	2-6-72
Part-III Basic File Access Functions	William Wong	1-4-54	Eureka	Robert Hazelwood	1-5-86
Part-IV Basic File Access	William Wong	1-5-70	FirstTime for C & Turbo Pascal	Stephen Davis	2-3-60
Part-V Programming MS-DOS	William Wong	2-1-38	Fontedit	A.G. Cameron	2-6-50
Part-VI Device Drives—Why and How	William Wong	2-2-50	Hochstrasser's Modula-2	Edward Joyce	2-4-70
Part-VII Device Drives—Structures	William Wong	2-3-40	16-Bit Lisp & Prolog Implementations—Part I	William Wong	1-1-62
Part-VIII Memory Disk Device Driver	William Wong	2-4-54	16-Bit Lisp & Prolog Implementations—Part II	William Wong	1-2-56
Part-IX Printer Device Driver	William Wong	2-5-62	MathCAD	Avram Tetewsky	2-6-53
Part-X Why Move to DOS 3.X?	William Wong	2-6-44	MathCAD	A.G. Cameron	2-6-50
Recovering PC-DOS Files	Edwin Thall	2-6-66	Mex-PC	Dennis Quinn	2-3-52
Using SYMDEB With NMI Breakpoints	James Owen	2-5-66	PC-Pro (PC-DOS on CompuPro)	Alex Soya	2-5-38
Software—Public Domain			Phoenix' PFI-Plus Debugger	Stephen Davis	2-4-48
CP/M Bus (Public Domain Hit			Plotz Graphics	A.G. Cameron	2-6-50
			Plu*Perfect DateStamper	Gord Wiggins	2-5-84
			PMate	A.G. Cameron	2-4-80
			Ratfor	A.G. Cameron	2-6-50
			Reflex	Nelson Dinnerstein	2-4-88
			Ryan-McFarland Fortran Scientific/Technical Word Processors—Part I	A.G. Cameron	2-4-80
			Scientific/Technical Word Processors—Part II	Steven Bosak	1-3-40
			SciPlot	Steven Bosak	2-1-50
			SCI-Graph	A.G. Cameron	2-4-80
			TEX	A.G. Cameron	2-6-50
			Turbo Pascal V3.0	A.G. Cameron	1-4-64
			TurboDOS/PC	David Carroll	1-3-72
			Watcom	M Guttman & V Mills	2-1-42
			Watfor-77	A.G. Cameron	2-4-80
			Xpip	A.G. Cameron	2-4-80
				Edward Joyce	2-4-84

Software Directory			
CP/M Software			
Able One Mailing List	2-2-87	SciPlot Graphics Library	2-2-86
DSD80	1-3-54	SMK-Seidl Make Utility	2-5-91
Electra-Find	1-3-54	Snobol4+	1-5-90
Jontel Business System	2-2-87	Techtype	2-6-94
Locipro	1-5-90	Tutori/o	2-5-91
Masterforth	1-4-82	TurboDos	
MB+ Tools	1-5-90	d/Multi	1-5-90
Megaback	2-6-94	Xenix	
ModemMail	2-5-91	XPD	1-4-82
MTBASIC*ROM	2-2-86	TEX	
Pluto Basic	1-4-82	TEX review	A.G. Cameron 1-4-64
R/Doc-X	2-2-86	TEX developments, Printer Drivers,	
Revas4	1-5-90	Screen Preview and Postscript	A.G. Cameron 2-2-78
Smarts	1-5-90	Metafont	A.G. Cameron 2-4-83
Snobol4+	1-5-90	UNIX	
Write-Hand Main	1-4-82	Coherent System review	Les Hancock 1-3-74
ZAS	1-5-90	Unix File	Ian Darwin 1-2-73
ZDM	1-5-90	Unix File (mailing list system &	
MS/PC-DOS Software		Formatting C programs)	Ian Darwin 1-3-81
db Vista V2	1-5-90	Unix File (macro processing &	
DSD86 & DSD87	2-5-91	Getting On the Unix Net)	Ian Darwin 1-4-76
F77L	1-4-82	Unix File (Hacker's OS & Tuning	
Instant C	1-3-54	Unix Programs)	Ian Darwin 1-5-84
Invisible Optimizer	2-6-94	Unix File (Public Domain Software	
Locipro	1-5-90	& Join)	Ian Darwin 2-1-78
Lattic Topview	2-5-91	Unix File (Usenet & Join)	Ian Darwin 2-2-72
Masterforth	1-4-82	Unix File (Command Portability)	Ian Darwin 2-3-88
Matrix 100	1-5-90	Unix File (Unix Books)	Ian Darwin 2-4-68
MB+ Tools	1-5-90	Unix File (MH Mail Handler)	Ian Darwin 2-5-88
MPS	2-5-91	Unix File (Unix Internals, Binary	
MT8087 Basic Compiler	2-2-86	Files & Termcap Entries)	Ian Darwin 2-6-86
PC-Lint	2-5-91	ZCPR	
Periscope II-X	2-6-94	CP/M Bus (Patches for Submit &	
Pluto Basic	1-4-82	Zsid)	Bob Blum 2-4-44
QROFF	2-2-87	Echelon's Z-System	Morris Simon 2-6-72
R/Doc-X	2-2-80	Syslib, Z3lib & Vlib	Richard Conn 2-5-72
RPG-II	2-5-91	ZCPR3 & How to Install It	Randy Reitz 1-2-42 §
RTC (Ratfor-to-C translator)	2-6-94		

DYNA-MITE v4.0

Terminal Emulation Software with File Transfer for DOS Machines

The latest generation of the classic MITE datcomm package, just \$99.95!

- * XMODEM, XMODEM/Batch, YMODEM, Kermit, MITE and other Protocols
- * Duplicate "menu" and "command" style control mechanisms
- * Emulation of VT100, VT52, TV925, ADDS VP, ADM3a, Z19, IBM3101, others
- * MORSE programming language with variables, cond. branching, etc
- * Lower priced versions (with fewer features) available (incl. CP/M-80)

Mycroft Labs, Inc. / P.O. Box 4106 / Tallahassee FL 32315
Phone: 904 385-1141 Orders: 1 800 MYCROFT BBS: 904 385-MITE

Micro/Systems Journal accepts Classified Ads. The charge is \$6/line (3 lines minimum); 40 characters max./line. Three times frequency

\$15/line; six times \$25; non-profit clubs \$2/line. Logos, special type, etc. are extra charge. Check must accompany ad copy. Send to M&T

Publishing, Inc., 501 Galveston Dr., Redwood City, CA 94063

COMPUTER PROFESSIONALS

with good writing skills to evaluate & review latest computer software products for national syndication. Part/time, contract stringer. Must have access to DEC, IBM PC, MAC or compatible clones. Send resume: PSPA, 830 Bay Ave., #G Dept 7, Capitola, CA 95010

MS-DOS Shareware

\$3.00 each disk postpaid. 000 catalog on disk, 001 expert system, 007 X-lisp, 014 B&W games, 015 color games, SMUG, 39 Hanover St., Asheville, NC, 28806-4158. 2/6/3x

FOR SALE: PMMI S100 Modems, FCC registered, Professionally Maintained, Tested and Guaranteed 90 days private use. 300/600 Baud. SALE \$49.95 + P&H. Prepaid orders from stock, Checks allow 2 weeks. Sorry No CC. Order PMMI/103MS, Send \$54.45 to: Sparks Micro, Box 880, Sparks, MD 21152 (301)472-4880 S100 Specialists Since 1978. 2/6/2x

FOR SALE: COMPUPRO dual 1.2 MB floppy disk subsys, DISK 1, DT-8 Qume's, cables, encl, CPM80. Like new, orig \$1750, ask \$900/offer. IDS, IP-225 dot matrix printer w/graphics, parallel port, w/3P+S board, 160 cps, \$250/offer. Bob Miller 408-371-2677 2/6/3x

JONTEL BUSINESS SYSTEM


The Jontel Business System is a five journal accounting package based on a dual disk CP/M computer. Included with the seven disk package are mailing and inventory programs with 100+ pages of documentation on disk. 8" & 5" disk formats available. \$69.95. Moonlite Computer, 707 Edge Hill Rd., New Bern, N.C. 28560. Source code available for custom installations, compiled CB80. (919)638-6976.

FOR SALE: New Fulcrum Omnidisk controller with CP/M, \$195. Gary Van Cott, Box 1879, Grafton VA. 804-898-3680. 2/6/3x

FOR SALE: CompuPro 20 slot S-100 enclosure/motherboard, Paradyamics model 2200D dual 8" enclosure, 2 Mitsubishi model M2894-63 8" drives, Advanced Digital 4MHz Z-80 SBC, US Robotics S-100 1200B modem, ADDS model VWPNT terminal, Comrex CR-1 Daiseywheel printer, CP/M 2.2 and SCP/80 front end, Pascal MT+ and Aztec C compilers, 100 8" disks, 2 8" flip'n'files. Running system, used < 200 hrs. Package = \$2000. W, 804-253-4006; H, 804-693-5897.

CompuPro System FOR SALE: Near-new home system, CPU-Z, RAM16, DISK1, INTER-FACER 4, CCS 12 slot mainframe, 2 Mit M2896-63 half-ht 8" drives, Tel 510+ term, software + CP/M. Present new value: \$2780. Sell for \$1795 + UPS. 703-463-6793.

CP/M Software. Choose from 400 Public Domain volumes. 100 page catalog \$7.50. New CP/M 2.2 Digital Research Manual \$19.95 + \$3.00 S&H. Send SASE for Flyer. ELLIAM ASSOCIATES, 6101 Kentland Avenue, Suit M, Woodland Hills, Ca 19367 2/4/2x



\$99	BUSINESS BOARD
24 hour business information center using modems at 300/1200/2400 baud. Has a multi-level menu system that is easily customized with no programming req'd. Includes remote PC operations and integrated data base management. Source code additional charge.	
\$99 PC	COMMX
Emulates VT100, Wyse, HP, ADM, TV, IBM, ADDS, file transfers: KERMIT, XMODEM, COMMX mainframe, Telex/TWX. Instant DOS/foreground switch! Unattended macro controls and 700 entry dial directory. Electronic mail sub-system!	
\$59 C DATA ENCRYPTION	
Data Encryption Standard (U.S. government standard FIPS PUB46) in Microsoft "C". Includes compression & telecomm formatting, allowing for faster transmission & storage on any computer or service. Complete "C" source code provided for additional \$249.	
 HAWKEYE GRAFIX Inc	Box 1400, Oldsmar Florida 33557 Call 813-786-8161


RP/M2™ creates

Z80®

CP/M© 2.2 compatible

IBM PC

Now available for Decmation's Blue Thunder softcard, PC RP/M2 is an **operating system**. Either standalone or with DOS present, PC RP/M2 provides the solid base of a genuine operating system reliably distinct from the facade created by an MSDOS interface. **All 2.2 system and CBIOS calls are supported**, with 56.5k TPA, file date and time stamping, fast virtual disk, iobyte redirection, terminal emulation, color console display, auto relog, COM path, addressable SAVE, single key phrase recall. SETDISK redefines a drive to any of over 80 CP/M formats. DOSDISK invokes built-in access to DOS drives. **System disk with manual \$129. Blue Thunder™ softcard \$149. Shipping \$5 (\$10 nonUS)**  



**Micro
Methods, Inc.**

118 SW First St. - Box G
Warrenton, OR 97146
(503)861-1765

The following back issues are available at \$5.00 for 1 issue, \$4.50 each for 2-5 issues and \$4.00 for 6 or more issues. Send orders with payment to M&T Publishing, 501 Galveston Drive, Redwood City, CA 94063. Credit card orders may be placed by calling 415-366-3600, ext. 216.

MARCH/APRIL 1985 (Vol. 1, No. 1): Bringing up CP/M Plus, Assembly Language Extensions for MS-Basic, New Tricks for CP/M2.2, Building an IBM/PC or XT Clone, Extended Single Density Storage, Variable Size Arrays in C; **REVIEWS:** dBase-III and 16-Bit Lisp & ProLog-Part I.

MAY/JUNE 1985 (Vol. 1, No. 2): Build an S-100 to PC-Bus Converter, interfacing to MS-DOS Part-I, Loadable Drivers fo CP/M2.2, Roll Your Own PC-Clone, Bringing up ZCPR-3, C & Godbout Disk-1 Controller, Writing Translation Programs in C and Turbo Pascal; **REVIEWS:** 16-Bit Lisp & Prolog-Part II.

JULY/AUGUST 1985 (Vol. 1, No. 3): Structured Programming With Microsoft M80 Assembler, Local Variables in Forth, Interfacing to MS-DOS Part-II, Data Translation with Turbo Pascal, Implementing Sets with Bit Operations in C, A Unix Mail List System; **REVIEWS:** Scientific & Technical Word Processors-Part I, Macrotech MI-286 S-100 CPU Card, Slicer System, Concurrent PC-DOS, Coherent Operating System.

SEPTEMBER/OCTOBER 1985 (Vol. 1, No. 4): Who Prints on Printer with CCP/M Interrupt Borrowing with Turbo Pascal, Upgrading CompuPro I/O Boards, Program Interfacing to MS-DOS Part-III, SIG/M & PC/Blue Public Domain Software Distribution Points, TurboDos INSTALL program; **REVIEWS:** CompuPro 10+, TEX.

NOVEMBER/DECEMBER 1985 (Vol. 1, No. 5): TurboCharge Your 8086/8088 Computer, Faster Floating Point Math with C, Bringing up CP/M-86K, Program Interfacing To MS-DOS Part-IV, advanced Machine Level Interface Techniques For Turbo Pascal, Context-Independent Macros for C, Build Your Own PC Program Library Using Public Domain Software, dBase-II Speed Techniques-Part I, Tuning Unix Program; **REVIEWS:** Peak 68K8-CP, Eureka For CP/M-80, Scientific Workstations.

JANUARY/FEBRUARY 1986 (Vol. 2, No. 1): Implement PC-DOS on Non-IBM Compatible Computers, Part-I; TurboCharge Your 8086/8088 Computer, Part-II; Assembling An AT Clone; Build An S-100 EPROM Emulator, Transferring Files Between CP/M and MS-DOS Systems; Program Interfacing to MS-DOS Part V; dBase-II Speed Techniques; Unix Public Domain Software; **REVIEWS:** Scientific Word Processors-Part II; Turbo-DOS PC; Turbo Pascal Support & Enhancement.

MARCH/APRIL 1986 (Vol. 2, No. 2): Implementing PC-DOS On Non-IBM Compatible System Part-II (Conclusion), Program Interfacing TO MS-DOS Part-VI, Rolling Your Own PC/XT/AT Clone, Converting From CP/M To MS-DOS On The PC, A C Subroutine For Parsing Command-Line Arguments, Arithmetic Aberrations With dBase-III and the dBase Developer's Release, The Marriage of TEX and Postscript; **REVIEWS:** C Interpreters, C Source-Level Debuggers, Lomas Data Products' S-100 PC, CompuPro S-100 PC Video Board.

MAY/JUNE 1986 (Vol. 2, No. 3): Build An S-100 HD64180 CPU Card, Configuring MS-DOS, Writing MS-DOS Device Drivers (Part 2), Building An AT Clone, Adding A RAM Disk To A CP/M System, Obfuscated C Code Contest, Sorting With Turbo Pascal, More dBase Speed Techniques, Using an RCPM System; **REVIEWS:** Concurrent DOS, MEX-PC, ConIX, FirstTime.

JULY/AUGUST 1986 (Vol 2, No. 4): Power to the PC, Creating A Copy Protected Program, Build Smart Keyboard Interface, Writing An MS-Dos Memory Disk Device Driver, Declare & Define C Variables in One File, More Loadable BIOS Drivers For CP/M; **REVIEWS:** Phoenix' PFI-Plus Debugger, Holliston Challenger XT 186, Magnum Digital PRO-180 & FD-100, XPIP, PMATE, Ryan-McFarland Fortran, WAT-COM, Watfor-77, MicroGlyph SciPlot, Softech Matrix Calculator, Personal TEX.

SEPTEMBER/OCTOBER 1986 (Vol. 2, No. 5): Control Systems Made Simple; Enhancing CP/M-80; A First Look At the 80386; Interfacing Using the SCSI Bus; Program Interfacing To MS-DOS—Part IX; Using SYMDEB With NMI Breakpoints, SYSLIB, Z3LIB & VLIB; **REVIEWS:** PC-Pro; CompuMagic Utility Package; Plu*Perfect Systems' DataStumper.

Here's why you should choose Periscope as your debugger...

You'll get your programs running fast. "It works great! A problem we had for three weeks was solved in three hours," writes Wade Clark of MPPi, Ltd.

You'll make your programs solid. David Nanian says, "I can't live without it!! BRIEF, a text editor my company wrote, would not be as stable as it is today without Periscope."

You'll protect your investment. We won't forget you after the sale. You'll get regular software updates, including a FREE first update and notice of later updates. You'll get technical help from Periscope's author. And you'll be able to upgrade to more powerful models of Periscope if you need to. One Periscope user writes, "...

your support has won over even the heart of this hardened programmer!"

You deserve the best. Thousands of programmers rely on the only debugger that PC Tech Journal has ever selected as **Product of the Month** (1/86). You owe it to yourself to find out why, first hand.

You can try it at no risk. You get an unconditional 30-Day, Money-Back Guarantee, so you can't lose.

Start saving time and money now — order toll-free, 800/722-7006. Use MasterCard, Visa, COD, or a qualified company purchase order. As one user puts it, Periscope is "one of the rare products, worth every penny!"

Periscope I, software, manual, protected memory board and breakout switch \$295
Periscope II, software, manual, and breakout switch \$145
Periscope II-X, software and manual \$115

Add shipping - \$3 US; \$8 Canada; \$24 elsewhere. Ask about air shipment if you can't wait to get your programs up and running!

PERISCOPE

The Periscope Company, Inc.
(formerly Data Base Decisions)
14 Bonnie Lane, Atlanta, GA 30328 404/256-3860

CP/M

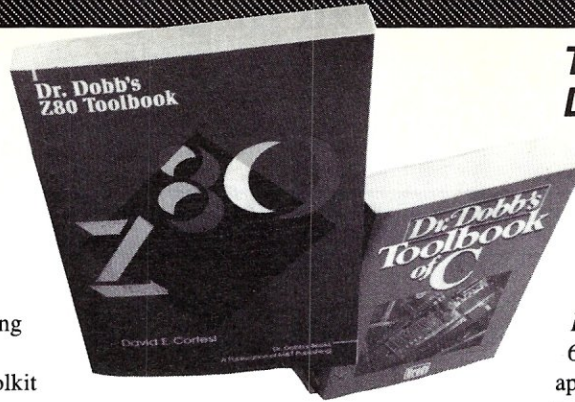
•Dr. Dobb's Z80 Toolkit

Item #022 \$25.00

This book and its companion disk contain everything you need to write your own Z80 assembly language programs. You'll find a method of designing programs and coding them in assembly language, and a complete integrated toolkit of subroutines.

Z80 Toolkit with software on disk Item #022A \$40.00

Formats: 8" SS/SD, Apple, Osborne, Kaypro



Micro/Systems Journal Library

•Programmer's Guide to CP/M Item #219 \$14.95

Edited by Sol Libes, this is a collection of all CP/M articles in *Micro/Systems* magazine from January 1980 through February 1982.

BASIC Programmers

•Basic Booster Library Item #214 \$29.95

This collection of time-saving screen and menu utilities can increase the power and speed of interactive BASIC programs, and simplify the coding required for application screens.

•Turbo Pascal for BASIC Programmers

Item #206A \$14.95

Learn why Pascal is easier to write, read, test and debug than BASIC. You'll find a library of useful programs, appendices including the ASCII character codes, a Turbo Pascal dictionary, and other helpful aides. A disk is also available.

•Companion disk Item #206B \$29.95

•Dr. Dobb's Toolkit of C Item #005 \$29.95

This authoritative reference contains over 700 pages of material by C experts, including the best C articles from *Dr. Dobb's Journal*. You'll find hundreds of pages of valuable C source code including a complete compiler, an assembler, and text processing utilities, all also available on disk.

•Dr. Dobb's Toolkit of Forth Item #030 \$22.95

This collection of useful Forth programs and tutorials contains expanded and revised versions of *Dr. Dobb's* best Forth article along with new material. The screens in the book are also available on disk with full source code.

Toolkit of Forth with software on disk

Item #031 \$39.95

Formats: MS/PC-DOS, Apple II, Macintosh, or CP/M. For CP/M disks specify Osborne or 8" SS/SD.

MS-DOS Utilities & Books

•PC Tools Item #216 \$39.95

PC Tools provides a complete collection of RAM resident system utility programs for IBM PC's and compatibles. It includes many of the same features found in DOS, along with additional features for recovering accidentally deleted files, viewing and editing file contents, and more.

•DS Backup Item #215 \$69.95

This high-performance program has all the features you need to make your backup routine fast and easy. The package contains a disk and manual and is not copy protected.

•Taming MS-DOS Item #060 \$19.95

This advanced user's guide will show you how to enhance and extend the power of MS-DOS so that you can work more efficiently, you'll find nearly 50 batch files and ready-to-use programs with source code so you can customize DOS to fit your needs.

•TallScreen Item #217 \$49.95

This resident screen and keyboard enhancement utility extend the capabilities of DOS to bring you more flexibility and power. You can scroll the screen to see previous DOS output, store multiple DOS commands, use a full screen editor at the DOS command line, and more.

Toolbooks from Dr. Dobb's

•Dr. Dobb's Toolkit of

68000 Programming

Item #040 \$29.95

This Toolkit contains the best 68000 articles reprinted from *Dr. Dobb's Journal*, along with new 68000 programming tools, applications and examples. Software in the book is also available on disk with full source code.

68000 Toolkit with software on disk Item #041 \$49.95

Formats: CP/M 8", Osborne, Macintosh, Amiga Atari 520st, MS-DOS

Turbo Pascal Books & Software Libraries

Turbo-Advantage Item #070 \$49.95

Save time developing your Turbo Pascal programs with this library of 220 routines for Turbo Pascal. Arithmetic operations, bit manipulations, check routines, data compression, menu functions, MS-DOS support, and statistical procedures are just a few of the routines included. The package contains source code and documentation. For MS-DOS systems.

Turbo Display Item #072 \$69.95

Turbo Display is a screen generator for Turbo Pascal. You'll find a screen oriented text editor along with 30 Turbo Pascal procedures and functions. The package includes full source code and documentation. *Turbo Display* is most effectively used with the *Turbo Advantage* routines, above (item #070). For MS-DOS systems.

Turbo Complex Item #071 \$89.95

This library of Turbo Pascal procedures and functions allows you to work easily with complex numbers, vectors and matrices. Some *Turbo Complex* routines are most effectively used with *Turbo Advantage* routines (above). The package includes full source code and documentation. For MS-DOS systems.

Stat Toolbox for Turbo Pascal Item #050 \$69.95

The *Stat Toolbox* contains two complete packages: a Turbo Pascal library of statistical routines including statistical distribution functions, random number generation, basic descriptive statistics, and more; and fully-functioning statistical programs with two data management systems. Source code is included.

Dbase III Programming Guides

Dbase III Advanced Programming

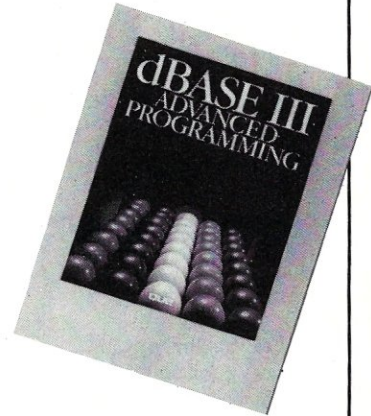
Item #204 \$22.95

Learn how to speed up Dbase programs, and how programs written in assembly, BASIC, Pascal or C can be adapted to run with Dbase applications. This book will show you how to write efficient code, how a particular coding technique works, and when to use it.

Advanced Programming Guide to

Dbase III Item #213 \$28.95

This is the most in-depth book on Dbase available, containing over 600 pages of information on debugging techniques, subroutines, structured programming, terminal codes, reserved words, and more.



To Order:

Order any of Dr. Dobb's *MicroSystems Journal* products, return the order form, or

Call Toll-Free
1-800-528-6050
EXT 4001

and refer to product item number, title, and disk format.

For customer service questions:

Call M&T
Publishing, Inc.
(415) 366-3600
EXT 216

Order Form

Yes! Please send me the following book(s), software with payment by:

Visa Mastercard American Express Check

Quantity	Item#	Description	Disk Format	Unit Price	Total Price

Name _____
 Address _____
 (Please use street address, not P.O. Box)
 City _____ State _____ Zip _____
 Card# _____ Exp. Date _____
 Signature _____

Sub-Total _____
 * Sales Tax _____
 ** Shipping _____
 Total Order _____

*Calif. residents must add applicable sales tax to total

**Add \$2.25 per item shipping

3032

Advertiser Index

Allen, Emerson & Franklin	45
Andratech	61
Applied Innovations	32
Austin Code Works	72
BD Software	5
BG Computer Applications	69
Bob Miller	92
Boston Software Works, Inc.	56
BV Engineering	40
C Users Group	40
C Ware	50
Charter Info Inc.	55
Clary Corp.	7
Cobalt Blue	70
Competitive Edge	41-43
CompuMagic, Inc.	9
Computer House	76
Computer House	88
Controlled Printout Devices	63
D & W Digital, Inc.	54
Digital Decisions	63
Digital Decisions	83
Digital Research Computers	11
Earth Computers	51
Echelon, Inc.	27
Electronic Control Tech.	13
Elliam Associates	92

Exec-PC, Inc.	26
Farbware	19
Fulcrum Computer Prod.	15
Gary Van Cott	92
Gimpel	56
Gimpel	60
Hawkeye Grafix Inc.	92
Integrand Research Corp.	63
Intercontinental Microsystems	35
Intercontinental Microsystems	37
Intercontinental Microsystems	39
L/F Technologies	85
Laboratory Microsystems, Inc.	77
Lodden Technology Ltd	53
M&T Books	94-95
Macrotech International	C-2
Martian Technology	88
MGlobal	52
Micro Methods, Inc.	92
Micro Supply Organization	48-49
Microcomputer System Consultants	87
Micromint	79
Mitek	83
Mix Software	2
Mycroft Labs	91

Night Owl	47
Performics, Inc.	23
Periscope Co., Inc.	93
Personal Business Solutions	45
Poor Person Software	61
Port-A-Soft	77
Professional Software Programmer's Assn.	92
R&L Microservices	20
Rational Systems	59
S-100	30-31
SC Digital	45
Semidisk	C-4
Servo Computer	72
SLR Systems	73
Slicer Computer	19
SMUG	92
Sparks Micro	92
Sunny Hill Software	61
Tee Kay	20
Teletek	C-3
Thinker's Apprentice	21
TIPZ	32
Turbo-Tech Report	33
Viasyn/CompuPro	1
Western Ware	77

Time and Task Management, by Timothy Berry works with your dBASE III package to let you orchestrate the tasks, time and budgets of you and your staff. The system contains dozens of programs, and includes source code so you can customize programs to fit YOUR needs. You'll find:

- **Special Input Program** that allows data entry at any time, by different users on different computers, without requiring dBASE III for each one
- **Data files**, some with sample values, some empty for your own input
- **Program files**, all easy to read and modify
- **Report files**, some reports are included, along with instructions to write your own
- **Format files**, to set the format for data entry and editing
- **Index files**, for fast access to other files

Time and Task Management with dBASE III™ by Tim Berry

For only \$49.95, **Time and Task Management** includes source code and a manual with complete documentation. You'll **spend less time keeping better track** of your work.

Time and Task Management works with dBASE III and your IBM or compatible.

TO ORDER:
CALL TOLL FREE 800-533-4372
 (Mon-Fri, 8-5 Pacific Time)
 In Calif: 800-356-2002
 Or, return your order and payment to:
 M&T Publishing,
 501 Galveston Dr.,
 Redwood City, CA 94063



TURBONET®

One Strong Link Can Strengthen Your Whole System

TurboNET® is Teletek's new local area network that permits reliable high speed data transfers among computers of differing bus architectures. TurboNET is designed to be used in networks consisting of multiple S-100 based and PC based systems. It will allow up to 4000 users, including up to 255 IBM PCs or compatibles, to share a single network and all attached peripherals. The network can be organized in any number of different ways mixing Teletek's 8 and 16-bit multiuser systems and PCs in any combination.

Teletek's Networking Family consists of:

TurboNET PC:

IBM-PC Network Interface Board

Teletek's TurboNET PC board offers IBM-PC Compatibility, CSMA industry standard protocols, 2 megabits/sec network speed, on-board CPU and communications management firmware, and media independence.

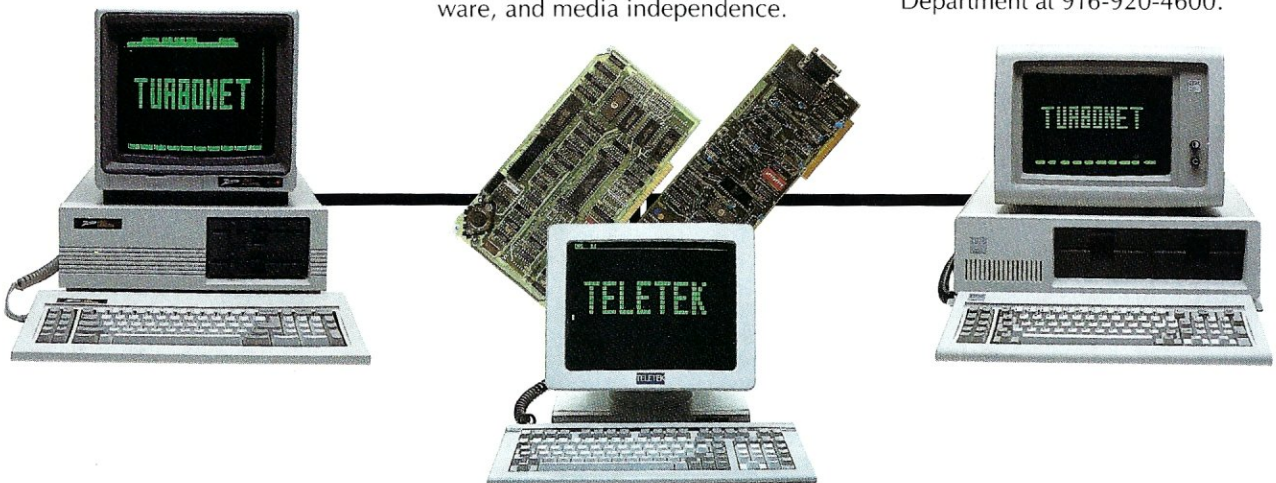
TurboNET S-100:

S-100 Network Interface Board

Teletek's TurboNET S-100 board offers IEEE 696 Compatibility, CSMA industry standard protocols, 2 megabits/sec network speed, on-board CPU and communications management firmware, and media independence.

The benefits are obvious: The cost savings of shared peripherals, almost unlimited system expansion capability, and the use of existing PC workstations with the ability to run the myriad of application software written for MS-DOS and PC-DOS. This coupled with Teletek's 8 and 16-bit multiuser systems running application software written for CP/M and MP/M allows the system the ability to access almost any software library.

For more information on Teletek's TurboNET S-100 and TurboNET PC boards or on any of our full line of S-100 products, please call our Sales Department at 916-920-4600.



TELETEK

4600 Pell Drive, Sacramento, CA 95838
(916) 920-4600 Telex #4991834 Answer back — Teletek

© 1986 Teletek

The Peak of Performance

SCALE THE HEIGHTS OF PRODUCTIVITY

Sure, you've proven that in your hands a computer is a productive tool. But if you haven't teamed up with a SemiDisk you have heights yet to climb!

IT'S NO MERE RAMDISK

SemiDisk has been leading the way for Disk Emulators since their inception. If you've seen RAMdisks you know what it's like to load programs in an

SEMIDISK

SemiDisk Systems, Inc.
P.O. Box GG, Beaverton, Oregon 97075
503-626-3104

instant, and read or write files without delay. Unlike alternatives, the SemiDisk offers up to 8 megabytes of instant-access storage while leaving your computer's main memory free for what it does best - computing!

KEEP A GRIP ON DATA

Go ahead, turn off your computer. Take a vacation. With the battery backup option, your valuable data will be there in the morning even if you aren't. You'll sleep better knowing not even a 5 hour blackout will sabotage your files.

NEW LOWER SEMIDISK PRICES THAT WON'T SNOW YOU UNDER

	512K	2Mbyte
IBM PC, XT, AT	\$495	\$995
Epson QX-10	\$595	\$995
S-100, SemiDisk II	\$799	\$1295
S-100, SemiDisk I	\$595	—
TRS-80 II, 12, 16	\$695	\$1295
Battery		
Backup Unit	\$130	\$130

Software drivers available for CP/M 80, MS-DOS, ZDOS, TurboDOS, and VALDOCS 2.