Implementing PC-DOS on Non-IBM Compatible Computers

see pages 26-30

Also in this Issue

TurboCharge Your 8086/8088 Computer - Part II ................. 32
Assembling An AT Clone ......................................... 44
Build An S-100 EPROM Emulator .............................. 60
Intersystem Processing ............................................. 66
Scientific & Technical Word Processors ......................... 50

Complete Table of Contents on Page 3

January / February 1986

Vol. 2 / No. 1

$4.50

U.S.A.
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 onboard 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.
Intelligent Computer Designs XL series S-100 products offer the latest in state-of-the-art designs.

Our slave products, the 8-bit XL-DLZ80 and the 16-bit XL-DL88, feature two slaves on one card, enabling system integrators double the number of users previously possible.

The XL-M180, a high-performance system master for single or multi-user applications, complete with hard disk, streaming tape and floppy capabilities introduces something new. An onboard Local Area Network with up to 5Mbps transfer speed. With it, interfacing other XL-M180’s or even IBM-PC’s is a snap!

If you think ICD has solutions for your systems’ needs, write or give us a call.

**XL-DL88**
- 2 completely independent slaves
- 8mhz 70108 processor
- 128k to 512k memory expansion
- 4 serial communication channels
- DMA transfers to/from system master
- CP/M 86, MS-DOS, PC-DOS, TurboDOS compatible

**XL-M180**
- High performance HD64180 uPD
- 512k memory with burst refresh
- 5Mbps local area network
- 4 bi-directional parallel ports
- 2 serial ports
- 1 clocked serial port
- Pipelined architecture
- Z SYSTEMS O/S available

**XL-DLZ80**
- Hard disk controller/Streaming tape option
- All digital floppy disk controller
- Extended instruction set
- Programmable instruction set
- 20 level interrupt structure
- Over 80 I/O ports
- Software programmable baud rates

All ICD products have a one-year part/labor warranty.

9252 Garden Grove Boulevard, Garden Grove, CA 92644 • 714 638-3622/Telex 312228
One Success After Another.

Since 1968 Teletek has been a leader in the design and manufacture of single-board computers, controllers, memory boards and interface boards. Teletek offers five distinct single-board computer (SBC) models with its own unique architecture to meet the varied needs of the system integrator. Based on the 8086/8088 and 68088 8-bit microprocessors, Teletek's SBCs can be further augmented with an integrated 8087 math coprocessor for numeric intensive applications.

Teletek's Systemaster II provides two RS232C serial ports and two Centronics-compatible parallel ports or may be optionally configured to provide a SCSI interface or an IEEE-488 interface to support many laboratory testing and measuring instruments. Teletek also offers a dual controller board, the HD/CTC, which will control any two ST506-compatible hard disk drives and any QIC-02/QIC-24-compatible cartridge tape drive. This unique design saves hardware cost as well as space in the mainframe.

Teletek's SBCs run under the multi-user, multi-processing operating system TurboDOS by Software 2000, supporting up to 16 users and capable of running PC-DOS application programs. For single-user systems, Teletek is supported by the CP/M operating system.

Teletek provides a 30 day evaluation program to qualified customers. For more information, call our Sales Department at 916-920-4600 or write for our information package.
STAFF

Publishers
Sol & Lennie Libes

Editors
Sol & Don Libes

Associate Editor
Susan Libes

Contributing Editors
A.G.W. Cameron
David Carroll
Ian Darwin
Nelson T. Dinerstein
Dave Hardy
Ken Jackson
Henry Kee
Steve Leon
Bruce Ratoff
William Wong

Editorial Assistant
Lenore Melillo

Advertising
Lennie Libes

Circulation/Administration
Lennie Libes
Diane Overholser

FEATURE ARTICLES

Implementing PC-DOS on Non-IBM Compatible Systems – Part I
Christopher Cochran & Kristofer Sweger

TurboCharge Your 8086/8088 Computer – Part II
Stephen R. Davis

Program Interfacing To MS-DOS – Part V
William G. Wong

Assembling An AT Clone
Sol Libes

Build An S-100 EPROM Emulator
Robert A. Rioja

Intersystem Processing
Hank Volpe

PRODUCT REVIEWS

Turbo-DOS PC
Michael Guttman & Vince Mills

Scientific & Technical Word Processors – Part II
Steven Bosak

DEPARTMENTS

Editor's Page
Sol Libes

News, Views & Gossip
Sol Libes

Mail
Sol Libes

The SIG/M Public Domain
Steve Leon

The PC/Blue Report
Hank Kee

The C Forum
Don Libes

Turbo Pascal Corner
David W. Carroll

The UNIX File
Ian F. Darwin

The Data Base Forum
Nelson T. Dinerstein

Classified Ads

AUTHORS: Micro/Systems Journal is always seeking good articles. Please write or call first to see if we are interested in the 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 writings.

Send a stamped self-addressed business size envelope for a copy of our Author's Guide.

TRADEMARK ACKNOWLEDGEMENTS

Macintosh
Apple Computer Company

UNIX
AT&T

GEM, CPM
Digital Research Inc.

Concurrent DOS
Digital Research Inc.

8088, 8086, 80286
Intel Corporation

IBM PC, PC/XT, AT
International Business Machines

MS-DOS, XENIX
Microsoft

GW Basic
Motorola

68000, 68010, 68020
Motorola
SETTING THE RECORD STRAIGHT

IBM has long been considered the “Master of Computer Marketing” and the PC just another in a long series of mighty triumphs. There is no doubt that the PC is a huge success. But let it not blind us to IBM’s failures and not-so-successful products.

I would therefore like to “set the record” straight by reviewing IBM’s track record in the personal-computer field. It is my opinion that another track record in the personal-computer company, without the huge financial resources of IBM, would have long been out of the business had they made the same marketing errors.

A BIT OF PC HISTORY

Virtually everyone I come in contact with seems to think IBM’s first entry into the personal-computer field was the PC, introduced in 1981. Nothing could be further from the truth. In fact, it was IBM’s third attempt, the first two being colossal failures that would have driven a smaller company quickly into the Chapter-11, bankruptcy court.

IBM’s first desktop, personal computer (the 5100) was actually introduced in early 1978 and was called the 5100. It had a 5” CRT display (I bet you thought Osborne was the first), a standard keyboard, and a tape cartridge for storage. Two versions were available, with either Basic or APL in ROM. There was very little program support, and no compatibility with other systems on the market. In addition the price was over twice that of the Altair and IMSAI (the market leaders, at the time) and the machine was only sold by IBM’s internal sales staff. It was apparent that IBM did not think much of the desktop market at the time, since virtually no promotional effort was put into the product.

At the time, Apple was still a garage operation selling a computer on a PC board (the Apple I). And Commodore was still showing prototypes of its PET computer and trying to get it into production. By the fall of the year, Apple was organized into a real company and had introduced the Apple II. Commodore was shipping the PET and Radio Shack introduced their TRS-80 Model I. IMSAI was also selling a desktop unit running Microsoft Basic. The machines were being bought mostly by computer hobbyists.

The 5100 fell flat on its face. In 1979 IBM introduced the 5110. By this time the Radio Shack, Commodore, and Apple computers were being shipped in large numbers to several hundred computer stores around the country. Users were starting to add 5-1/4” floppy disks to their systems. VisiCalc had been introduced for these machines, as well as some decent word-processing and database software. IBM salesmen were shocked to find that personal computers were appearing on the desks of their business customers.

The 5110 was essentially an upgraded version of the 5100. It still used IBM’s proprietary 8-bit microprocessor, but now had floppy-disk drives. The unit looked like an oversized calculator with a large floor-standing cabinet, housing dual 8” floppies. The screen was upgraded to display 16, 64-character lines. Applications software, which included the traditional business applications for dentists, travel agents, etc., was rented rather than sold outright. The system cost was still more than twice that of other personal computers on the market and users were completely dependent on IBM for software and support.

By mid-1980, IBM realized that the 5110 was also a failure, as Apples, TRS-80s, PETs multiplied on office desks, invaded classrooms, and seemed to be popping up everywhere. It began to look like a repeat of the same situation that had occurred in the minicomputer field when IBM was only a minor player and the minis were growing in power and taking over applications that once were the exclusive domain of large mainframes.

THE PC TO THE RESCUE

IBM realized that it had to do something quickly. It therefore decided if it could not sell a desktop system in its traditional way, it would try to emulate its competitors in system design and marketing. Thus was created the 5150, better known as the “PC”.

IBM looked around for a quick solution. The company decided to bring out a machine using the popular Z80, 8-bit microprocessor running the CP/M operating system and MBasic and sell it via the same outlets that carried the Apple computer. They undertook negotiations with Digital Research Inc. (developers of CP/M) and Microsoft (developers of MBasic). They found DRI to be very independent, while Microsoft was very cooperative. Bill Gates of Microsoft pointed out that 16-bit microprocessors (e.g. the 8088 and 8086) were already available and that a nearby company (Seattle Computer Products) had already migrated MBasic to the 8086/8088 and created a 16-bit version of CP/M. And, that if DRI was being independent, maybe IBM should consider a 16-bit machine instead of an 8-bit one. Microsoft offered to handle both the operating system and Basic language processor. IBM, under time pressure, bought the deal.

The machine was thus essentially a copy of machines like the Apple and TRS-80 with few improvements. Like the Apple, it came with 16K of memory, a cassette interface, some slots for add-on boards, Basic in ROM, color graphics with about the same resolution, and an optional disk drive with only a little more storage than the Apple’s. The software IBM introduced was virtually the same that was available for the Apple. The price was essentially the same as the Apple’s and IBM offered it through Computerland and Sears, Apple’s two biggest retail outlets. One could make a strong argument that IBM created an Apple-like clone. One thing is for sure…IBM did not create an advance in technology. Rather, it was attacking a marketing problem with a quick and cheap solution.

Where IBM did pioneer was in the marketing of personal computers. They
invested a tremendous amount of money in advertising, promotion, and training of dealers. They insisted on dealers investing money in transforming their disheveled, little low-overhead computer stores from havens for computer hobbyists to carpeted, fancy-furniture establishments with pin-stripe, high-pressure salespeople.

One other thing happened to help IBM. The government decided to drop their antitrust suit against Big Blue. With that problem off its back IBM decided to really show it could compete in the PC marketplace. IBM took advantage of its incredible manufacturing and marketing capability to set new standards for computer pricing that has allowed it to assume the dominant position in the market, forcing many competitors out. It is beginning to look like a repeat of what IBM did in the mainframe business, when IBM forced companies like RCA and GE out of the business. IBM has already forced companies like Osborne, Otrona, Gavilan, Hyperion, Columbia, Actrix, and Victor Technologies out and left several others gasping. IBM's recent aggressiveness in the PC software area has already done-in several software companies and more are on their way out. We may again see the government haul IBM into court on antitrust action . . . but by then it may be too late, as it was in the mainframe business.

We should also add to IBM's failure list the PCjr and PC Portable. And there is some question as to the success of the AT as most dealers are heavily discounting the unit in order to move accumulating inventory (see "News" column).

**BOBCAT 3**

**DISK CATALOG**

The most versatile and powerful catalog program available

- creates, adds, updates, and deletes a filename catalog
- seven report formats
- hard disks, multiple drives, and user numbers
- individual file titles
- wildcard searches for filenames and file titles
- CP/M or PC MSDOS

8″ CP/M SSD or Popular CP/M or PC MS DOS 5 1/4″

- U.S. residents $49.95 U.S.
- Canadian residents $49.95 Can
- Other countries $64.95 U.S.

plus $3.00 P & H

NAME, NUMBER, EFFECTIVE & EXPIRY DATES

Bank drafts: certified checks, money orders, company checks

R & L Micro/Services, Inc.

Box 19955, Station F

Ottawa, Ont.

K2C 3S8 613-225-7964 THE HOME OF THE BOBCAT

---

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.

P O Box 2368

Cambridge MA 02238

617 • 576 • 3828
RUMORS & GOSSIP

The PC Technical Group of the Boston Computer Society has undertaken an on-going study of Scientific Word Processors. After 10 months of work the group (composed of 12 scientists from MIT, Harvard, Brandeis, and industry) have issued their first report. It is without a doubt the most thorough study I have seen so far and I recommend it to anyone considering the purchase of a SWP. For a copy send $8 (U.S., Canada & Mexico), $10 elsewhere, to: Carl A. Hein, Dunster House Apt Z, Swanson Road, Boxborough MA 01719. Make the check payable to Carl.

Jon Shirley, President of Microsoft, recently boasted that Xenix has an installed base of 120,000 systems. This is probably 70% of the Unix systems in operation today. Of course the most popular Unix-based system is the Tandy Model 16. With Xenix’s new System V compatibility and Microsoft’s promise of MS-DOS/Xenix software compatibility, Xenix systems will surely continue to far outnumber other Unix implementations. Shirley also recently disclosed that Microsoft is working on programs to do speech-to-text files and videopictures-to-graphics files.

Atari has signed a Unix System V licensing agreement for use with its 68000-based ST series and its future 32-bit computer lines. Such an agreement typically sells for $50,000. It is rumored that Atari will introduce a new entry in the ST series shortly running System V, with built-in hard disk and 1Mbyte of memory. The 32-bitter is expected later in the year.

Apple is expected, at its upcoming stockholders meeting, to introduce new versions of the Ile and Mac. Look for the Mac to have 1Mbyte of RAM, more ROM, 800K floppy drive and expandability. The Ile will get a new operating system and limited Mac compatibility. In the fall, expect a new version of the Mac based on the 68010, a separate larger monitor, 4Mbyte memory expansion, built-in hard disk options and a bus with slots for expansion.

Early word is that AT&T may finally have a winner in the new 6300 machine. It has a 286, is AT-compatible, faster, has more expansion slots and runs UNIX with MS/DOS as a task. Both share the same file structure so that files can be easily moved between the two. In fact a user, while in UNIX, can run DOS programs and execute DOS commands. Software developers should love it.

And, look for something new in spelling checker programs. Some newer ones sit in background checking spelling as text is entered.

KayPro continues to lose money. They posted a loss for the fourth quarter ending August 31st due to poor sales. The company had reported losses for the previous five quarters.

CD ROMs may be the hot new item for 1987. Microsoft and IBM are known to be working hard to develop such systems for personal computers. Toward this end, Microsoft will run the “First International Conference On CD ROM” in March.

Racal- Vadic has announced a new modem for personal computer users that can blaze along at 9600 baud. It also supports 300 and 1200 baud transmission. Cost is $1,495.

Digital Research, in an attempt to get Concurrent PC-DOS off dealer shelves has modified it to work with the AST Rampage extended memory spec (a superset of the Lotus/Intel/Microsoft standard to access memory beyond 640K). So far it is estimated that DRI has sold only 15,000 copies of Concurrent plus some OEM deals.

PC-Net WHERE ARE YOU?

Novel, 3Com and AST are beating IBM and Microsoft to the punch. They are expected to momentarily release upgrades of their networking systems to MS-DOS 3.1 compatibility. Are we seeing a situation where IBM is developing standards only to lose out to competition? It is interesting to note that although IBM has 80% of the PC/XT sales in the U.S. they have less than 50% outside the U.S.

IBM has finally released its first PC-Net interface/controller card. It looks like it contains more processing power than a PC. I haven’t heard the price yet. But considering the high cost of the card and the interconnected coaxial cabling system there is some doubt as to whether this is going to make it. In the meantime companies like Novel, etc. are having great success with lower cost twisted-pair networking systems. I have heard of some organizations using some of the unused telephone circuits in their buildings for networking interconnections. There is no doubt there have been significant improvements in digital signal processing circuits that make it possible to ship data at 10Kbits/sec, and higher, over lower quality lines.

TIS THE SEASON

The Christmas season has become the make-or-break time of the year for computer hardware vendors and dealers. And ’85 Yuletide sales have become even more important after a very slack first three quarters. If Christmas sales are not good we will see more shakeouts in the first quarter of ’86.

The key players are IBM, Apple, Commodore and Atari. Recognizing that the Apple II and Mac are selling well in the home market and Commodore’s Amiga and Atari’s ST520 are priced lower, Apple cut prices in early October. The Ile was cut 23% to $995 with a color monitor and printer, while the IIe and Mac were cut 17.5% and 10.5% respectively.

IBM is going with dealer rebates which few dealers are expected to pass on to customers because of discount and volume pressures from IBM. IBM is offering the system to dealers for $500 (minimum order, 40 systems) with deferred payment until March and full return privileges. The street price is expected to be between $650 and $750 with a color monitor included. However, the general feeling is that this is not enough to move the large number of units still in IBM warehouses and dealer inventory.

Commodore claims to have signed up over 700 dealers for its new Amiga based on a very substantial discount structure and very liberal ordering terms. At $1,295 the system is competing head-on with the two-year old Macintosh. Although by far the most advanced personal computer currently available, it is a very expensive home system and a very limited professional system. I doubt it can succeed where the Mac, with better promotion and distribution, so far, has not been able to.

Atari is betting on the low price ($799) of the ST520 to compete against the Apple II. This is a terrific price for a 68000-based system with 512K RAM, monitor, mouse, disk drive, system software and two languages. Although the basic system has been shipped since early July, peripheral support and application software have been slow to appear. Software developers have found it more difficult to develop software to run under the GEM operating system than under earlier, and simpler, operating systems. This same situation slowed Mac sales and will no doubt also affect the Amiga and ST520.

Atari and Commodore have gotten a very cool reception from chain stores, who are still trying to move large inventories of the Coleco Adam, Commodore VIC20 and C64, and Atari 800. Since both companies need to move very large quantities of machines, this is not a good sign for them.

Micro/Systems Journal January/February 1986
DUAL GPIB-488 INTERFACE BOARD
A Stand-Alone, Independently Controlled Dual Channel IEEE-488 I/O Processor. Interface Activity Modes for Controller-in-Charge, Controller Assigned or Terminal Bus Slave, and all Interface Functions are handled transparent to Host System CPU through an on-board CPU and DMA controller. User Friendly operation.
A&T, P/N 52748-800-102

RGB COLOR GRAPHICS BOARD
Programmable resolution up to 512 x 512 pixels with 4 local video planes and on-board graphics processor. Color mapper allows 16 colors from a palette of 4096. Light pen input. Plus more...
A&T, P/N 52748-300-101

12-BIT A-D-A CONVERTER BOARD
8 Channel A-D: 12 microsec Conversion, 10KHz Sample Rate, Programmable Gains, Offset and Diff./Single Modes. 8 Channel D-A: 2 microsec. Setting, Bipolar or Unipolar Output. Programmable Reference levels, Dual-Ported Channel Refresh RAM. 16/8-Bit Data Transfers via I/O or Memory Mapped
A&T, P/N 52748-900-101

PERIPHERAL SUPPORT BOARD
Two Serial SYNC/ASYNC Ports with RS-232, TTL or Current Loop Outputs, three 8-Bit Parallel Ports, three Timers, Real Time Clock/Calendar and Response Programmable Interrupts Controller, Small Proto Area with +5 and ±12v.
A&T, P/N 52748-150-101

BAR CODE PROCESSOR BOARD
The BarTender is a stand-alone I/O Processor that reads and prints most common Bar Codes. Includes bi-directional reading, wand interface, clock/calendar with battery. Extensive documentation and software.
A&T, P/N 52748-800-101 Without Wand
A&T, P/N 52748-300-201 With Wand

MULTI-PURPOSE PROTOPYING KIT
Industrial Quality with Plated-Thru holes for Wire-Wrap or Solder projects. Complete with +5, ±12v Regulators, Bus Bar, Filter Capacitors, and Manual.
P/N 52748-450

Also available: Multi-Function I/O Board, Smart Prototyping Kit, 128Kx8/64Kx16 Static RAM Module

INPUT/OUTPUT TECHNOLOGY, INC.
25327 Avenue Stanford, Unit 113, Valencia, CA 91355 • (805) 257-1000

Uncompromising Additions to your S-100/IEEE-696 BUS
MAGAZINES COME & GO

There are two new magazines worthy of note. "80s Journal" is published quarterly by Octoplus Corp., 2426 Wade Ave, Raleigh NC 27607. Subs are $14 (1 yr.) and $25 (2 yrs.). Add $3/yr outside U.S. Their first issue was 48 pages. "MultiUser" is also published quarterly by Octoplus Corp., 2426 Wade Ave, Raleigh NC 27607. Subs are $14 (1 yr.) and $25 (2 yrs.). Add $3/yr outside U.S. Their first issue was 48 pages.

Several magazines have ceased publication. "Compas", published for several years by the International North Star Users Association, ceased publication last fall. "Business Unix" died after only three issues.

THE HIGH COST OF LASER PRINTERS?

Not long ago we bought a copier for our fledgling publishing operation—...a Canon PC-25 copier, the top-of-the-line model. It has two reduction and one enlargement modes, handles legal as well as letter size, plus some other features. The cost was $835 while the list was close to $1,400.

What intrigues me is that I have been given to understand that the basic guts of the machine are the same as that used on the HP LaserJet printer. This set me to thinking. How much could the extra electronics and laser circuitry cost? Surely not the extra $2,000 (HP) or $4,000 (Apple) charges, at discount prices!

It appears to me that HP and Apple must be making a killing on their laser printers. I will have to wait for the competition to develop. I sure would like to get a laser printer with downloadable font and graphics capability!

It appears that I may not have to wait much longer as several printer vendors are preparing products to sell at significantly lower prices. NEC, Epson, Ricoh, and others are expected to shortly introduce systems for shipment starting early this year. It therefore appears possible that we may see laser printers and printers using competing LCD and LED technology selling for well under $2,000 and possibly even as low as $1,000 by year end.

PC & AT CLONE PRICES DROPPING

Over two dozen AT compatible motherboards are already on the market. The result is that AT prices are dropping fast. More than one outfit is already selling a basic AT clone for under $2,000 (see article in this issue). Less than half the cost of a comparably equipped IBM-AT.

The IBM-AT, which is overflowing dealer inventories, got a $350 rebate back. However, IBM introduced a new AT model with a 30Mbyte drive for only $200 more than its 20Mbyte unit. We expect dealers to clear out their 20M ATs...look for the units on the gray market at substantial discounts.

In the meantime PC/XT clone prices are dropping fast. At a recent flea market XT-clone motherboards, fully loaded with 256K of RAM and BIOS ROM were being offered for $135. And a system, with display, 2 drives, 256K RAM and graphics controller and 90 day warranty, could be had for $695. And mail order prices are dropping too—just check out the Aitaz ad in this issue. We will attempt to review some of the low-cost AT clones as soon as we can get our hands on them.
MULTI-USER MICROPROCESSORS & COMPONENTS

8-BIT MASTER PROCESSOR
CPZ-48004 $746
CPZ-48006 $784

8-BIT SLAVE PROCESSORS
CPS-6A $412
CPS-6BA $562
CPS-6BD $467

ARCNET CONTROLLERS
LAN5100 $371
LANPC $371
LANPC-256 $521

ARCNET HUBS
HUB-2A $146
HUB-4P-ENC $41
HUB-8A $319
HUB-8A-ENC $469

HARD DISK CONTROLLERS—Includes cables, TurboDOS Software Drivers
MD-1013 $521
MD-1016 $521
OMTI-5100 $521
OMTI-5300 $596

CARTRIDGE TAPE DRIVES
T-540 $975
T-525 $520

FLOPPY DRIVES
8-242 $1995

HARD DISKS
XT1085 $2239
XT1140 $3495
3051E $1395

CABINETS
C2810 $849
C3810 $1595

DATABASE SYSTEMS
Informa-20 $720
Informa-20 $720
The Language $995

WORD PROCESSING
Word Processor $230
Word Processor $250
The Filer $145
The Speller $145

16-BIT MASTER PROCESSORS
CPZ-186 $1121
CPZ-186-512 $1271
CPZ-186-1MB $1496

16-BIT SLAVE PROCESSORS
CPS-16 $746
CPS-16-512 $971
CPS-16-1MB $1196

ARCNET WORKSTATIONS
WS80-6A $796
WS80-6A-ENC $896

VIDEO DISPLAY DEVICES
ESP-6310 $556

SHEET FEEDERS
Z-300 $1995
Z-200 $1596
Z-100 $495

FLOPPY DRIVES
8-242 $375

CARTRIDGE TAPE DRIVES
Ciper, 90 IPS, 45/60 MB
Ciper, Floppy Tape, 32MB

HARD DISKS
MAXTOR, 30ms, 85MB
MAXTOR, 30ms, 140MB
ATASI, 30ms, 52MB

CABINETS
Para Dynamics Mini-Pronto, 10 slots.
Custom modified Para Dynamics Pronto—all peripherals mounted horizontally in brackets. Holds 2-8” half height floppy drives, 1-5 1/4” cartridge tape drive, and 2-full height 5 1/2” Winchester HD. 20 slots. Custom front bezel. All mounting brackets removed by loosening (not removing) 4 screws. No more cut or scraped knuckles or standing on your head to mount peripherals!

SPECIAL SERVICES
Custom Hardware Configurations
TURBO DOS System Generation
Tape Backup Sub-Systems
**CALL FOR PRICING**

TurboDOS COMPATIBLE MULTI-USER SOFTWARE

DATABASE SYSTEMS
Informa-20 $720
Informa-20 $720
The Language $995

WORD PROCESSING
Word Processor $230
Word Processor $250
The Filer $145
The Speller $145

APPLICATION SOFTWARE
BACs CALL
MCS-3 CALL

OFFICE AUTOMATION
T-Office $995

All prices, specifications & availability subject to change and stock on hand. System House and Dealer inquiries invited. TurboDOS is a trademark of Software 2000, Inc. T-Office is a trademark of Commercial Systems Corp.
Letters to M/SJ

We welcome your letters with comments, compliments, criticism and suggestions. We read them all and publish the most noteworthy, even if they are critical of us. We do not have the staff to answer all letters personally. And all letters become the property of M/SJ and may be subject to editing. Further, 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.

Dear Mr. Libes:

DOS, CP/M-80 and CCP/M-86 and you, so encourage diversity as editorial...although I built my H-100 over a year ago, I believe they are the same.

Dear Mr. Libes:

Strongly discourage IBM specific stuff. The world is flooded with IBM-PC only magazines. Favor generic MS-DOS, CP/M-80 and CCP/M-86 and hardware that is a little more exploratory. Compatibility will surely destroy you, so encourage diversity as editorial policy.

Do not hitch your magazine to a dying horse (UNIX). The great majority of micro-computer users (including advanced users) just skip any article that features UNIX. Your readers ARE NOT USING MINICOMPUTERS! Why pretend that UNIX is for micros?

For general programming illustrations I find C less than ideal. It is often cryptic and not nearly as portable as Turbo Pascal. Turbo code will run under CP/M-80, CP/M-86, MS-DOS and PC-DOS (and everyone can afford it).

Andrew Grygus
La Crescenta CA

Editors,

I bought an IBM PC-compatible computer recently, because it seems to me that IBM-like computers are becoming quite common. I am an advanced computer user interested in all aspects of micro-computers: hardware, software, marketing strategy, public domain, etc. I would hate to see you exclude IBM-like discussions just to satisfy John T. Limnell and Bernard S. Gorman (letters in vol. 1 and 3). Please continue your excellent articles such as S-100 to PC-bus conversion board.

Graeme McRae
Monmouth Junction, NJ

Dear Lennie and Sol:

Carl Herberts to

Dear Mr. Libes:...good news. I bought an IBM PC-compatible computer recently, because it seems to me that IBM-like computers are becoming quite common. I am an advanced computer user interested in all aspects of micro-computers: hardware, software, marketing strategy, public domain, etc. I would hate to see you exclude IBM-like discussions just to satisfy John T. Limnell and Bernard S. Gorman (letters in vol. 1 and 3). Please continue your excellent articles such as S-100 to PC-bus conversion board.

Graeme McRae
Monmouth Junction, NJ

Dear Lennie and Sol:

Thanks for sending the complimentary July/August issue. It is a prized supplement to BYTE and the C-PRO NewsLetter.

As a just-over-a-year owner of a CompuPro 816C, I was greatly impressed with your "new" publication. To support your effort, I am enclosing a check to cover a one-year subscription and copies of the two previous issues.

The articles/reviews were excellent, especially " Concurrent PC-DOS" by Guttmann and Mills. It was a refreshing change from the dista...⊇in 1981 with the IBM-PC, rather than in 1975 with the Altair.)

Guttmann and Mills identified the IBM-PC hardware as a very severe limitation to C/PC-DOS. Having just looked at an early version of the PC and Multimate, I would suggest another problem: non-partitioned programs.

Multimate requires its entire 136k program to be resident in RAM. Contrast this to partitioned programs where only the command file need be resident on RAM. The remaining overlays can remain on disk or in an AboveBoard type RAM-disk. A few "vanilla" examples follow (CMD/OVL): WordStar-Pro (22k/204k), SuperCalc-2 (34k/36k), DBASE-2 (20k/104k).

Having used M/P8-M8-16 v2.1F for over a year, I find the above mentioned "vanilla" programs are excellent in this environment. Based on experience and from studying multiple CPU theory, I perceive no reason these programs should not work simultaneously on an 8588 CPU board.

About windows, I find a normal screen difficult enough to read with reducing the displayed information available. Further, I have found little benefit for a single user to operate on more than one program at a time. (The procedure is easy in MP/M, just move the terminal cable to a new console and sign on.)

The single exception is in debugging programs. When an errant program "hangs" a terminal, just move to a vacant terminal and issue the command: ABORT CMD-PRG #, where # is that of the hung terminal. This saves a reset and preserves the M-Drive configuration exactly as is. (No chance of MFORM crashes.)

I am looking forward, someday, to purchasing the the MI-286/287 board (reviewed in the same issue). Between DR's...on yet another software-oriented periodical.

Regarding 32-bit processors on the S-100 bus: If Peter Thiessen is interested in putting a 68K CPU on it, may I suggest he take a look at Robert Groppo's (of Morrow Designs) article "68000 Bus adapted to 68000-based Unix Computer" in Computer Design, March 1985. Although he doesn't present much meat, Mr. Thiessen may find a starting point. I have not, unfortunately, found any article for the 32000 series. Anyone know of one? Or could you persuade National Semiconductor to recognize the benefits which accrue to its 32016 processor (trailing badly behind Motorola's 68000 in the 16-bit (bus-width!) stakes) by public-domain involvement and present an article detailing an interface?

I don't know what your feelings are on upgrading this bus to accommodate full 32-bit pathways, but I feel a good case could be made for a change in the mechanical specification. In particular, the designers of VME bus have come up with a good scheme, and the S-100 could do worse than follow their example. (For those not familiar with this standard, VME boards are based on 16-buswidth) "hangs" a terminal, just move to a vacant terminal and issue the command: ABORT CMD-PRG #, where # is that of the hung terminal. This saves a reset and preserves the M-Drive configuration exactly as is. (No chance of MFORM crashes.)

I am looking forward, someday, to purchasing the the MI-286/287 board (reviewed in the same issue). Between DR's CB-86 and C compilers, I should have many math-intensive programs to utilize this board.

Adrien J. Dubreuil
Derby, CT

Dear Sirs,

Welcome back! Good to see the return of your magazine - it was sadly missed. I hope you will keep up the level of hardware contributions and stop it from becoming yet another software-oriented periodical.

Regarding 32-bit processors on the S-100 bus: If Peter Thiessen is interested in putting a 68K CPU on it, may I suggest he take a look at Robert Groppo's (of Morrow Designs) article "68000 Bus adapted to 68000-based Unix Computer" in Computer Design, March 1985. Although he doesn't present much meat, Mr. Thiessen may find a starting point. I have not, unfortunately, found any article for the 32000 series. Anyone know of one? Or could you persuade National Semiconductor to recognize the benefits which accrue to its 32016 processor (trailing badly behind Motorola's 68000 in the 16-bit (bus-width!) stakes) by public-domain involvement and present an article detailing an interface?

I don't know what your feelings are on upgrading this bus to accommodate full 32-bit pathways, but I feel a good case could be made for a change in the mechanical specification. In particular, the designers of VME bus have come up with a good scheme, and the S-100 could do worse than follow their example. (For those not familiar with this standard, VME boards are based on the Eurocard form factor, being difficult-to-produce-at-home edge connectors.) I would be very interested in any comments on this — given the seeming lack of anything better other than the high-end buses Multibus-II, which requires licensing by Intel, or VME-bus itself. This is probably the way I will proceed, when I finally get round to bringing a 32-bit system up.

I. Worthington
Romsey Hampshire, England

Andrew Grygus
Norwalk, CT

P.S. Continue the excellent publication.

Dear Mr. Libes:

In response to Robert Savage's letter published in the Sept/October issue, I suggest that he look closer at the "Z-100 Main Board schematics". Although I built my H-100 over a year ago, I believe they are the same.

Sheet 4 of my schematics shows that the 2661's have their own crystal of 4.9152 Mhz for baud rate generation on the serial ports. The 8085-2 on sheet 1 has a 10 Mhz crystal which gives a system clock of 5 Mhz, not 4.9 Mhz that Mr. Savage claims. The only apparent side effect is that the 8085's instruction execution will be slower if he reduced the crystal to 8 Mhz.

I have been planning to increase the 8-bit system clock to 6 Mhz (12 Mhz crystal) to get a 20 percent increase in speed. My 8088 has been upgraded to an 8088-2 that runs at 8 Mhz and haven't had any communication problems.

David S. Podejko
Norwalk, CT

P.S. Continue the excellent publication.

Welcome back! Good to see the return of your magazine - it was sadly missed. I hope you will keep up the level of hardware contributions and stop it from becoming yet another software-oriented periodical.

Regarding 32-bit processors on the S-100 bus: If Peter Thiessen is interested in putting a 68K CPU on it, may I suggest he take a look at Robert Groppo's (of Morrow Designs) article "68000 Bus adapted to 68000-based Unix Computer" in Computer Design, March 1985. Although he doesn't present much meat, Mr. Thiessen may find a starting point. I have not, unfortunately, found any article for the 32000 series. Anyone know of one? Or could you persuade National Semiconductor to recognize the benefits which accrue to its 32016 processor (trailing badly behind Motorola's 68000 in the 16-bit (bus-width!) stakes) by public-domain involvement and present an article detailing an interface?

I don't know what your feelings are on upgrading this bus to accommodate full 32-bit pathways, but I feel a good case could be made for a change in the mechanical specification. In particular, the designers of VME bus have come up with a good scheme, and the S-100 could do worse than follow their example. (For those not familiar with this standard, VME boards are based on the Eurocard form factor, being difficult-to-produce-at-home edge connectors.) I would be very interested in any comments on this — given the seeming lack of anything better other than the high-end buses Multibus-II, which requires licensing by Intel, or VME-bus itself. This is probably the way I will proceed, when I finally get round to bringing a 32-bit system up.

I. Worthington
Romsey Hampshire, England
Dear Sirs:

I hope that you bear with me as I comment on several topics from the September/October issue:

1. DRI acted as if they were trying to kill their own goose with their marketing of CP/M Plus. The documentation for installation of the unconfigured version is so poor that a thorough knowledge of CP/M 2.2 and a good network of knowledgeable friends is needed. I am not expert in hardware or software, but I would have been able to accomplish the job in 20 hours if the documentation had been adequate. Apparently several vendors failed to make the grade without the advantages which I had. Incidentally, I had in my possession a letter from DRI indicating that Westico would supply unconfigured CP/M Plus. I don’t know whether this is true, but I felt little guilt in not forking over money when the realistic additional cost of the time I lost due to their lack of proper support is around $1400. I felt that if the Z80 market were to stay alive, software would need to run under CP/M Plus; however, it appears that only about 2% of software customers are looking for this.

2. Mr. Utley oversimplified the business of running CP/M 2.2 programs under CP/M Plus. I have never seen a program which stuck to normal BDS calls fail under CP/M 3.1; however there are many which don’t stick to these standards. There are so many things which could be taken care of in a general purpose compatibility RSX that it could grow unreasonably large, but here are three which are missing from early public domain programs:
   a. Some programs need to bypass the sector translation table. Since 128 byte sector disks are handled identically by CP/M 2.2 and 3.1, BIOS calls on these disks should be passed through. Many such programs will run find under unbanked CP/M 3.1, but not with the popular compatibility translators.
   b. The method for opening files with wild cards (*) in the name has been covered in the literature but not included in general purpose translators.
   c. Some programs, including those link- ed with Microsoft FORLIB, look for the BIOS copy of the directory record when opening new extents. The method for taking care of this is analogous to the situation in (b). This cancels out the speed advantage which unbanked CP/M + has over 2.2.
   d. Some programs point to the FCB and send the file name out to the screen without terminating it with $. This causes a lot more clutter under 3.1 although it was tolerable under 2.2.

It may be that if we are to look forward to a future for the Z80, we should be giving serious attention to Echelon’s products. I think there is a real need for a memory banking OS and that ZCPR took too much memory to do what even unbanked CP/M

+ would do, but DRI seems to be deserting the market and leaving us in a situation of technical violation of copyright if we do what we need with their products.

3. Hank Kee’s comments about the social rules for software distribution by public domain libraries are interesting. We need these channels, but in my mind their operators would be justified in refusing to handle contributions which carry unreasonable demands or in inviting their subscribers to vote on whether these demands are reasonable.

I have limited such requests in my offerings to requests for additional technical information, for copies of enhanced versions of my software, and for consideration of my business when buying related software. I would have appreciated being notified by those bulletin boards which were carrying my stuff and would have kept them up to date, but I have never had any answers to such requests. About the only compensation I have gained from putting things on bulletin boards which were carrying my stuff and would have appreciated being notified by bulletin boards is the feeling that someone might be helped by my offerings as I have been by theirs, and the ability to ask freeloaders to go find what they want on bulletin boards rather than to expect me to make up a custom program for nothing.

Jim Prince
Birmingham, MI
INDEX TO SIG/M SOFTWARE PUBLISHED

The New York Amateur Computer Club has published an "Index to SIG/M volumes 1 through 241", prepared by Howard Vigotti. This 38 page index, by program category or file name, is available from the NYACC, Box 106, Church Street Station, NY 10008. Categories include different operating systems and languages, specific machines, types of utilities and special interests. No price is marked on the index, but if you want a copy please send them something to cover postage and printing; I would suggest $3 domestic and $4 foreign.

NEW SIG/M SOFTWARE RELEASED

Some months back, Mark Weiss, a graduate student at Syracuse University, contributed a collection of statistically oriented programs to the SIG/M library. We checked them out with an expert in the field and found that Mark's programs were really first rate. The disk, SIG/M Volume 243, called Mark Weiss Shows His Stuff, should be of great interest to anyone with a statistical inclination. The code is written in Turbo Pascal and therefore requires a Z-80 system. However, it should compile for MSDOS or CP/M-86. Mark also sent us his version of LIFE. We had no room for it on Volume 243, but did manage to find a spot for it on Volume 244.

LIFE appears twice in the latest releases. Professor Harold Macintosh of the University of Puebla, Mexico, again demonstrated the versatility of REC (Regular Expression Compiler) with a version of LIFE (called TORU) that runs over a period of some five days. TORU allows you to stop and start at will, so you need not be glued to the machine for that length of time.

Dr. Macintosh, winner of the Computer Hobbyist of the Year award at the 1985 Trenton Computer Festival, has two volumes in the latest releases. TORU is the feature program on Volume 246, MAZE and TOUR on Volume 245 are demonstration programs showing the use of the REC programming scheme for mazes.

The good professor also has some updated HELP material on the disks as well as another demonstration of how simple it is to put your utility routines into a single utility library and then run them directly from within the file. On my Compupro system I move a library of utilities to the RAM disk on startup and run them out of that library throughout the session. If you don't already do that, spend five minutes with ANCI and RUN on Volume 246 to see how simple it is to make life easy.

Harry Van Tassell, CompuServe CP/M sysop, finally broke down and bought a PC Clone at the recent computer flea market run by the Amateur Computer Group of New Jersey. I stopped over at his house the following day to help him get it set up and brought over a collection of public domain utilities that I thought he would need. Included on the disk was VFILER. Rich Conn, developer of CDR, wrote the program for CP/M-80 and we released it on SIG/M Volume 145. Harry rewrote the code for CP/M-86, which SIG/M released on Volume 146. SIG/M included the VFILER source code in its releases.

Well, sure enough - one of the PC people took VFILER and adapted it for MSDOS. Even gave Rich and Harry credit for the original code. However, the rewriter failed to include the source code and sure enough - it does not work on Harry's composite video monitor. No sense rehashing the demon freeware arguments again, but the fact is that by encouraging people to have their hands out for freeware donations, we discourage people who are not seeking money from supplying the source code.

Some other good people have made excellent contributions to SIG/M this month. Gerry Edgar, a frequent contributor, has donated EPRO - Small Prolog (Volume 242). The actual code takes up only 6K and requires a Z-80 to run. M-80 source code is included as is a compiler written in Prolog. The program comes with comprehensive documentation files and references so that with a little effort you can become familiar with the language. Sample code is provided to help you learn.

On the same volume (242) Garry Womack has released, for public domain distribution, a former commercial program - FIP (File Interchange Program). It also requires a Z-80 and, like most programs in the commercial world, does not include source code. A full manual is on the disk. It is a fairly well done file manipulator similar to SWEEP or VFILER, but with enough in the way of features to make it interesting. Such features include file comparison, attribute changing, display of deleted directory entries, wildcard renaming, etc. Also on the disk is Garry's SANDR (Search and Replace) which does just that for binary or ASCII strings.

Adam Fritz has contributed two major works on SIG/M Volume 244. His Whetstone Benchmarks for a variety of languages emphasize integer or floating math, simple or subscripted variables and subprocess or function library routine calls. The languages covered included ALGOL, FORTRAN, BASIC, C, PASCAL and ADA. He has also provided programs to convert LINPAK single precision general system routines. Very technical, very specific - but very useful for those who have a need for it.

TCF-86 IS COMING!

The 11th annual Trenton Computer Festival will be held on Saturday and Sunday, April 19th and 20th. This year we will expand the seminar schedule to cover both Saturday and Sunday. Just about everybody comes to Trenton. We have even invited Irv Hoff!

TCF is run by five computer clubs (including the Amateur Computer Group of New Jersey). In other words it is a show run by computer hobbyists for computer hobbyists.

TCF is the oldest of the computer shows (now in its 11th year) and the biggest and best non-commercial computer event of the year, with a huge out-door computer flea market that covers acres and acres. There is also a large indoor commercial exhibit area, speakers, presentations, user group meetings, and prizes. Sol Libes usually waits for TCF to buy stuff to get the best prices. Incidentally, Sol will be the Master of Ceremonies at the Saturday night banquet and
ORDERING THE NEW SIG/M RELEASES

SIG/M Volumes are available on 8" SS SD Disks for $6.00 each ($9.00 foreign) directly from SIG/M, Box 97, Iselin, NJ 08830. Printed catalogs (40 pages) are $4.00 each ($5.00 foreign). Disks in a variety of formats may be obtained through the worldwide SIG/M distribution network (published in the September/October 1985 issue of M/SJ). The distributor list is also included with the printed catalog.

A disk version of the catalog (Volume 00) is available for $6.00. Many bulletin boards have the software for downloading and most new releases are available on the CP/M Sig on CompuServe. There are also a number of commercial companies which distribute SIG/M software on a sale or a rental basis. For Apple CP/M users, the Philadelphia Area Computer Society, Box 477, Kulpsville, PA 19442, has most of the SIG/M volumes in Apple II format.

Volume 242

Microcomputer Systems Consultants

Volume 243

Microcomputer Systems Consultants

Volume 244

Whetstone Benchmarks and Linpack Conversions

by Adam T. Fritz
released September 20, 1985

LPAKII .DOC 10K system routines to Pascal and C; provides LPAKII-C.LQR 23K and discusses results. In Fortran, C and Turbo Pascal.

LPAKII-T.LQR 24K Turbo Pascal.

LIFE .PAS 13K / 2K drivers to apply and demonstrate routines.

NETWELL. LEB 54K Whetstone benchmarks

LIFE .COM 18K new version in Turbo Pascal

LIFE .FAS 13K /

Volume 245

Convert Programming - LIFE (Toru)
Universidad Autonoma de Puebla
released September 20, 1985

EPRO .LBR 10K datafile listing number from 0 to 100

GETNAME .FRO 1K (support file)

HPIC22 .LQR 8K

LSITOCR .LQR 12K linear least square fits

NAR .COM 3K

NEAR .COM 3K

NORMAL2 .BAS 2K MASTIC best estimate of the true mean

PIAFFPROX.PQS 10K pi approximation - circle in a square -
In a prior column, I made note of how one can amass a very useful collection of software from the public-domain and user-supported software libraries. For example, I had selected the following:

- **PC-WRITE**: a word processor
- **WPK**: a processor for kids
- **PC-CALC**: an electronic spreadsheet
- **FREECALC**: an electronic spreadsheet
- **PC-FILE**: a database management system
- **PC-DBMS**: a database management system
- **5BY5**: a database management system
- **QMODEM**: a communications package
- **XMODEM**: a communications package
- **PCPG**: a PC picture graphics

The functions are similar to those of the **VFILER** library utility in the **PC/Blue** library. **VFILER** was originally written and submitted to the **SIG/M** library by Harry Van Tassel. This is a very flexible file-manipulation utility designed for tagging and untagging files for group operations. For example, a series of unrelated but tagged files can be deleted with a single **ERASE** command. When this software was obtained by **CP/M** systems, it contained source code, but subsequent versions do not include the modified source.

**PCSweep** is similar to **VFILER**, but in addition to the essential functionality of **VFILER**, **PCSweep** has file-attribute handling capabilities. File squeezing and unsqueezing functions are also included. The program is user supported and a donation is unabashedly requested.

Have you ever wanted to modify the date and time stamp of a given file? If you look at the files in a **PC/DOS** diskette, you will find that all have uniform dates and times. For those of us perfectionists, **FDATE** will enable modification of the date and time, one file at a time. The process is done one file at a time. It would have been nice if the source code could have been available so that a wildcard option could be included, if desired. If you are interested in blanking out the date and time stamp on a **DIR** command, this can be done by setting the date and time to 00/00/80 and 00:00:00, respectively.

**Utilities Programs**

**RENDIR** will permit the user to change the name of a subcatalog. This utility saves you from the tedious routine of having to create a new separate subdirectory and then transferring all the files from the old to the new subcatalog.

**LU** is a very popular library utility in the **SIG/M** library. **LU** allows one to put all related files into one compressed file for modem transmission or archiving purposes. **LU**.COM, the original conversion for **PC-DOS**, has been superceded by **LAR.EXE**. The functions are similar to those of the **CP/M** version. Also useful are **SQ** and **USQ**, which can squeeze and unsqueeze a file in **MS-DOS** or **PC-DOS**. I had been using **NSQ** and **NUSQ** for a long time, but these utilities were developed by frequent users of bulletin board systems. Happily, **SQ** and **USQ** eliminate lengthy transmission times required by **NSQ** and **NUSQ**. A new utility combining **LU**, **SQ**, and **USQ**, and then some, is now available. It is called ARC for archive. It was written and made available by System Enhancement Associates, 21 New Street, Wayne, NJ 07470.

**ARC** will create a library file, as well as preserve the original date and time stamp. In addition, it will automatically squeeze down the files more efficiently than **SQ**. Upon extraction, it will, of course, unsqueeze the individual files. Top of all this, it will also sort the files in alphabetic sequence. The great value of this program lies in the large number of useful options that are available.

Usage:

```
ARC [sum|exe|priv|bin] <archive> <filename> . .
```

Where:

- **a,u** = add files to archive
- **d** = delete files from archive
- **x,e** = extract files from archive
- **r** = run files from archive
- **p** = copy files from archive to stdout
- **l** = list files in archive
- **m** = move files to archive
- **t** = test archive integrity
- **c** = convert entry to new packing method
- **b** = retain backup copy of archive
- **w** = suppress warning messages
- **n** = suppress notes and comments

**Editor's Note**: Hank Kee is the librarian for the PC/Blue public domain software library. He is the person who collects, assembles, and checks all the software issued by PC/Blue and then compiles and edits it into the released volumes.

**Copies of the PC/Blue printed software directory can be ordered from Micro/Systems Journal, Box 1192, Mountainside NJ 07092 ($4 U.S., Canada & Mexico; $6 foreign).**

**PC/Blue** disks are available from the New York Amateur Computer Club, Inc., Box 106, Church Street Station, NY NY 10008. Price is $7 per volume which includes media, postage, and handling. On foreign orders, please add $2 per disk.
has been written by Howard Vigorita of the NYACC.

Though there are many different versions of SDIR, their basic formats and outputs are all similar to the version of SDIR in the SIG/M library. However, not all of these directory programs are safe to use. For example, a malicious version of a virtual extended directory program called VDIR has been making the rounds of unsuspecting bulletin-board systems. This program will wipe out your directory permanently. Let this be a sober reminder to all who download from bulletin-board systems. The SYSOP does not always have the opportunity to check out each uploaded file. If you download any file, it should be checked in such a way that the main body of your files are not affected.

There are many more utilities, which I will comment on in a future column. As it is impossible, in this limited space to give a detailed review on all such utilities, I have and will describe only those I use daily.

NEW PC/BLUE RELEASES

During the past two months, I have released several new volumes to the PC/Blue library. They are:

Vol 146/149 Potomac Pacific Engineering (2 diskettes) database/spreadsheet/expert system
Vol 150 Flow System Organizer Visible Pascal (interpretive and interactive)
Vol 151 Miscellaneous utilities ADDRIVE software reset - floppy switch configuration ALUSQIII unsqueeze .Q7F files AXX "Autodore" - file manager BIOSDATE BIOS ROM date display CED command editor CHDRD subdirectory CHANGE COPPYC DISKOPY handling bad tracks DISET displays PC switch settings DIR-X expanded directory DISPMNX displays index II/III index structures DISK8Q squeeze entire diskette DISKNSQ unsqueeze entire diskette EQUIPIO2 equipment display v1.02 RIFIX2 keyboard enhancement v2.2 LAR14 library update program LEDVS displays 1/0 drivers LUM54 ASCII screen editor RDIR rename file RENDIR rename subdirectory RULER2 slide screen column ruler up/down TABS19 TAB utility add/remove tabs from ASCII files WASH file manager WINDOW display background utility W52OS converts Wordstar text
Vol 152 Confidant v2.0 Encryption Program Symphony Macro Conversion Aid & Range Input Command
Vol 153 Fun & Games Archark Aim Bowling Brick Bags Chess Oddball Keno Kidnap Labyrinth Leisure Memo Monopoly Qubert Spacewar Staircase Stocks Stress Tom-Ins Temple
Vol 154 Small Database - SDB v2.0 Ron Cain's Small-C:PC v1.1
Vol 155/156 RBBS-PC v12.5b (2 diskettes)
Vol 157 Public Domain FOCUS by Automata Design Associates
Vols 158/159 AnalyticalCalc-86 v19-04e (2 disks) MS-DOS Integrated Spreadsheet
Volume 160 Genealogy ON DISPLAY version 3.1
by Don Libes

Most people are familiar with C compilers. You hand the compiler your code, it compiles and returns an object module (or a lot of error messages). The linker is then used to combine the object files and any libraries into an executable program (when you get the opportunity to read more error messages). After all this is done, you get your last chance for errors when you run your program.

If the program doesn’t behave as desired (either by acting oddly or bombing), you can change the source code (adding printfs) and recompile it, starting the whole circle again. This could be called the edit-compile-link-run-repeat-the-whole-mess style of programming.

This can be especially painful, because just a simple change like adding a printf still forces the source file to be recompiled and everything to be relinked. This can take a long time for large programs.

Sometimes, the program doesn’t even compile or link, leading to an edit-compile-loop. Assuming that it does compile but doesn’t run correctly, we can run the program under control of the debugger. This means we can set breakpoints and when it is stopped look at variables. Look at the stack. Let the program step through a couple of statements. Look at some pointers. Trace the pointers as they go through many levels of indirection (this is always fun), finally ending up pointing to oblivion.

Symbolic debuggers work on the C-language level rather than the assembler level. You can step through lines corresponding directly to your original program. You can evaluate C expressions or call subroutines. The nicest one I’ve seen (dbx from 4.2BSD UNIX) can print out complete structures formatted appropriately for their types.

All debuggers will let you change variable values, but frequently that isn’t very useful. You need to change the code or unwind the stack a bit and continue on. I don’t know of any debuggers that are this capable.

Indeed, no debugger holds a candle to a good C interpreter. An interpreter bypasses the compile step and often the edit step, also. What’s left? Run, run, run! Yeah!

C Interpreters

INTERPRETERS VS COMPILERS

Let’s review the difference between a compiler and an interpreter. Bear in mind that these are generalizations and differ from implementation to implementation.

A compiler takes the source program and makes several passes over it, typically including scanning, parsing, (optional optimization,) code generation, (more optional code optimization) and linking.

Because of interdependencies in the code, it is much easier to handle one phase at a time. Changing (i.e. fixing) the compiler is easier because impact on other phases is isolated. Machine dependencies can be isolated to only the code generation phase.

The result for the user is that compilers tend to be slow, because of all these phases and also due to the amount of optimization they do. Also, you must have a complete program even to test out portions of an algorithm since the compiler wants to resolve all references.

On the other hand, an interpreter resides in memory while the program is actually running. At each step, it takes the next source statement and performs all the phases at once, thereby executing it. In reality, an interpreter typically performs some of the phases while reading the program in, since many errors can be detected during a simple scan of the program. This is known as incremental compilation but the term interpreter is used to drive home to the buyer of such software that it behaves much more like an interpreter.

C INTERPRETER ADVANTAGES

Not only do interpreters work in ways that seem closer to the ways humans think, but they are more interactive, reporting errors very quickly after they can be detected. Further speedup is gained by having the interpreter and editor be co-resident in memory, and when an error is detected or fixed, control is passed very quickly to the other. Finally, interpreters place little emphasis on optimization under the theory that it is pointless, since you are only debugging anyway.

The end result is that the user does not have to wait for a program to be recompiled and relinked in order to make a small program change. Indeed, program fragments can be tested directly, so that the programmer does not have to waste time building up a small skeleton only to find out that his approach is completely wrong.

Some C interpreters are deeply integrated with an editor so that they can leave the cursor right at the spot where the error was detected as soon as one has been detected. Since this is done so quickly, there is less emphasis on being able to continue compiling in the fact of errors, a continual problem in compiler building. The user is satisfied with one error at a time, since the interpreter picks up just where it has left off.

But what about runtimes? Well, that is where all the interpreters cost. Some interpreters are actually quite speedy, achieving times that are only several times as great as that of compiled code. The slowest interpreters are quite slow, however, and for large programs, verging on the unbearable.

What makes all this livable is the theory that you don’t really care how long programs take to execute while you’re developing them. (Although that’s what people originally theorized about compilers, and yet, no one wants to put up with a slow compiler.)

When you are satisfied that the program performs correctly, you then compile the program with an optimizing compiler. The result is that you have the best of both worlds: a development system that is extremely fast and flexible, and a means for turning out fast code in production form.

THE DISADVANTAGES OF C INTERPRETERS

The drawbacks to working in this manner are few but worth mentioning.

The most important is that the interpreter and compiler can behave differently. There are dark corners of the language, that are not completely defined, even by the new C standard. Implementation dependencies.
(such as language extensions) can be a problem. While all of the interpreters and compilers are full K&R they are not in conformance with many extensions currently expected to be in the C standard and they may differ in disjointed ways. Optimization nuances may cause a compiled program to behave differently than an interpreted one.

There are, of course, ways in which you would want the interpreter to behave differently from the compiler. For example, exit() normally returns to the operating system, but while debugging, it should behave as a return to the interpreter.

Another drawback is that the interpreter will need to be resident while you run your programs along with a lot of extra baggage, such as symbol tables, type and usage information, etc. The result is that you will not be able to interpret programs that are as large as ones that you can compile and run.

The interpreter may do things that get in your way, such as using interrupts or memory that you want to use, or using the screen when you are debugging a graphics program. Of course, a debugger is no different in this respect.

Lastly, you can expect an interpreted program to be (much) slower than a compiled one.

CONCLUSION

When it comes to the point of actually testing running code, a great deal of time is spent making minor changes and then waiting while it is recompiled so that you can test it again. For example, you might discover that you have passed the wrong variable as a parameter or that your index is off by one. Or your might want to try a subtle change in your algorithm. These are all minor modifications that require that you edit your original code, recompile it and then link the whole mess together again.

Being able to stop code at the point of error and modify it or evaluate arbitrary C statements and expressions without constructing a main() and losing your context leads to a whole new style of programming. It brings C programmers to a more modern way of programming. Interactively. Its about time.

Programming in this manner has been available to Lisp and APL programmers for 20 years and to a lesser extent for several other languages. While we slowly get access to C interpreters, new programming environments for these other languages are being refined to make use of even more effective tools, such as graphical display of algorithms and data structures, language-directed editors and program verifiers. These will give C yet another lease on life in the next few years. I can’t wait.
Turbo Pascal Corner

by David W. Carroll

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, Box 699, Pine Grove CA 95665.

Over 400,000 copies of Turbo Pascal have been sold - over 80 percent to MS/PC-DOS users. This has created a sizeable after-market for commercial Turbo tools and utilities - which both small and large software vendors have moved to capitalize on.

These products can be roughly broken down into the following categories:

* Programmer tool kits (ready made source code modules)
* Linkable modules (compiled)
* Program Generators (create Turbo source code)
* Utilities (listers, formatters, cross-referencers)
* Ready-to-go programs
* Training aids

I will discuss a few of the products available in each category in the following sections. A vendor list is provided at the end of the article.

TOOLS

Perhaps the largest category and most useful programs are the programmer toolkits. These typically consist of a number of procedures and functions each of which performs a useful process, like testing printer status, changing the logged drive, getting the date or time, and so on. Of course, many of these processes have been created in the public domain, but these commercial packages have the advantage of a consistent calling protocol and usually provide some level of documentation and some even provide an example program using each procedure or function.

Turbo DataBase Toolbox $54.95
Borland International
Book-Disk package providing routines for building and sorting databases.

Turbo Graphix Toolbox $54.95
Borland International
Book-Disk package which provides using screen graphics and animation routines.

Turbo Pascal Support & Enhancement Products.

AEFTOOLS $29.95
Allen, Emerson & Franklin Inc.
Over 100 utility modules for inclusion in your TP programs. Well written and commented source code. Doc file but no manual. Great value.

Training Wheels for Turbo Pascal $25.00
by Robert E. Brown
Landmark Software
Features 74 utility modules and demo programs for use with Turbo Pascal programs. Large 8.5" x 11" (76 page) manual included. Great value.

Turbo Graphics $39.95
by Mark Edwards
Diversified Educational Enterprises
Complete set of high speed graphics routines and binary files for use with Turbo Pascal - 24 files and large manual (60 pages plus listings).

Turbo Power Tools $99.95
Blaise Computing
This set of utilities includes an excellent manual with over 140-8.5"x11" pages of documentation and three diskettes. This package has virtually everything you could need. The routines include the following:
- String functions
- Screen handling
- Window management
- Keyboard control
- DOS utilities
- File handling
- Directory maintenance
- Memory maintenance
- Program control
- Interrupt service routine support

Turbo Asynch $99.95
Blaise Computing
This set of utilities provides asynchronous communications capabilities in your Turbo Pascal application programs. They will drive any asynchronous device via the RS232 COM ports, like printers, plotters, modems and other computers. A complete manual is provided. Features include:
- Written in assembly code (source included)
- Initialization of COM ports
- Interrupt processing
- Data transfer between circular queues and ports
- Simultaneous buffered input and output to both COM ports
- Transmission speeds up to 9600 baud
- Input and output queues of any size

Super Tools Library $95.00
Paragon Courseware
- Window Management
  Provides Turbo routines for advanced windowing capabilities.
  Features:
  - Open several windows simultaneously
  - Store data in overlapping windows
  - Store environment of each window (color, cursor, etc.)
  - Remove and redisplay windows (popup)
  - Cursor control within each window
  - Window disk libraries supported
  - Supports both color and monochrome
  - Can access video RAM directly (without "snow") for maximum speed, or use BIOS video interrupts for maximum compatibility.

System and File Information Package
These routines provide data on the current system environment.
- Determine run-time hardware (i.e. display card type, number and kind of disk drives, amount of RAM, number of ports, and whether or not a clock exists.
- Determine if a drive is ready, without the DOS error "Abort, Re-try, Ignore".
  Find disk free space. Set default drive. Get and change file attributes.

Math Expression Evaluator
This routine is a single function which accepts a mathematical expression in the form of a Turbo Pascal string. After syntax verification, the expression is evaluated and a REAL type result is returned. All math functions and operators are supported.
The application is a daily reminder program for IBM PC which provides DOS access, screen and keyboard access, demo programs for each routine. Great free space, set or clear attributes, etc. Recently which actually generate Turbo Pascal source code programs, which may contain Turbo routines, all source included.

**FASTSCREEN FOR TURBO PASCAL**
$30.00
Technisoft

**FASTSCREEN** adds fast screen output capability to your Turbo Pascal programs. Display a full screen or window almost instantly. Read a full screen or window containing multiple input fields with a single procedure call. Give your programs a clean, professional look. Inline assembler and Turbo routines, all source included.

**TURBO PASCAL LIBRARY 3.0** $25.00
Success Press

60 useful routines including: pop-up windows, run DOS, DOS commands and other programs from your Turbo programs. Get display mode, use dual displays, get free space, set or clear attributes, etc. Heavily commented source code with a demo program for each routine. Great value.

Pascal's Friend $39.95
J.S. Computing

Four include files and an application program for IBM PC which provide BIOS/DOS access, screen and keyboard access, calendar functions, and 1-2-3 like-wrappers. The application is a daily reminder program.

**EXTERNAL MODULES**

A few products include object code modules which can be linked to the user's program as external procedures to perform operations which are too time critical for anything but assembly code (some screen operations, communications I/O, etc.).

**Turbo-Plus** $29.95
Nostradamus

A library of procedures crafted in assembly language anddesigned to enhance Turbo Pascal versions 2.0 and 3.0. These procedures are extremely fast and combine with user programs as external procedures. Great value. Features include:

- Display Mapping without flicker or snow
- Field input screens
- Screen read and write
- Advanced keyboard control
- RAM Window for debugging
- Enhanced file I/O using file handles
- Pop-up Turbo Reference Guide (memory resident)

**PROGRAM GENERATORS**

A new group of products has appeared recently which actually generate Turbo Pascal source code programs, which may then be run stand-alone or combined with user programs. These programs include a database program generator, several entry screen generators, and even a compiler (parser) generator.

**GTP (Generate Turbo Program)** $99.95
Allen, Emerson & Franklin Inc.


**Screenmaker** US$34.95 plus $4.00 shipping
Stampede Software Designs

Develop a library of up to 9 input screens per program with full editing and help features. Source included.

**Forms Definition System** US$49.95
Compugam

Create data screens and input forms. Allows extensive data editing. Source included.

**Turbo Screen** $49.95
Pascom Computing

Create up to 99 screens and save to disk. Includes screen editor, collator, and code generator. Source not included.

**Turbo Sculptor** $125.00
Software Bottling Co.

Advanced screen generator creates program code from "image" of screen you design.

**QParser** $400.00
QCAD Systems

Compiler program generator - makes TP source code for compiler front-end from a syntax rule table. Sample C and Pascal compiler syntaxes included. Includes TP 3.0. Huge manual.

**UTILITIES**

Utility programs for Turbo programmers are also quite numerous, and include a number of program listers, cross-referencers, and source code formatters. Other utilities include code timers, structure analysers, and optimizers.

**TPF (Turbo Print Formatter)** $29.95
Allen, Emerson & Franklin Inc.

Utility that allows using embedded print commands in TP Source programs for printer control.

**PROGRAMMERS UTILITIES** $95.00
Turbo Power

Especially for power users of Turbo Pascal. Includes a bound manual, three diskettes which include the source plus compiled versions, and quick reference chart. The utilities include the following:

- **PRETTY PRINTER**: Standardize capitalization, indentation, and spacing of code.
- **PROGRAM STRUCTURE ANALYZER**: Find subtle problems the compiler doesn't: uninitialized and unused variables, modified value parameters, "sneaky" variable modification, redefined standard identifiers. Also generates complete variable cross reference and program hierarchy diagram.
- **EXECUTION TIMER**: Summary of time spent in each procedure/function of the program, accurate to within 200 microseconds. Counts number of calls to each subroutine and is fully automatic.
- **EXECUTION PROFILER**: Graphic profile or where program spends its time. Interactive, easy to use. Identify weak code.
- **COMMAND REPEATER**: Go beyond MS DOS batch files to combine a powerful text parser with general purpose command execution capability. Use to copy, print, delete across subdirectories.
- **PATTERN REPLACER**: Find and replace versatile regular expression patterns in any text file. Supports nesting, alternation, tagged words and more. Over dozen programmer's applications included.
- **DIFFERENCE FINDER**: Find differences between two text files, and optionally create an EDLIN script which rebuilds one from the other. Disregard white space, case, arbitrary characters and Pascal comments, if desired.
- **SUPER DIRECTORY**: Replace PCDOS DIR command with extended pattern matching, sorting capability, hidden file display, date filtering, and more.
- **FILE FINDER**: Locate files anywhere in the subdirectory tree and access with a single keystroke. Display the subdirectory tree graphically.

**Turbo Utilities** $20.00
ESD Software Co.

Includes a lister, formatter, and cross-referencer for Turbo Pascal programs.

**TurboRef** $49.95
Gracon Services

Cross-reference and listing utility for Turbo Pascal source programs. Source included.

**ListPascal** $19.95
Karl Gerhard

Versatile List program for Turbo like Borland's "lost" TLIST.

**Tidy** $49.00
Major Software

Turbo or Microsoft Pascal source code formatter utility. No source included.

**PascalPac** $69.00
Major Software

Features four utilities for Pascal programmers:
- **X-REF** creates a cross reference table
- **X-RAY** browses cross-ref table and source code at same time
- **X-PRINT** is a versatile listing program
- **X-PEEK** browses programs or text files

No source included.
4 MHZ ON YOUR Z80 IS LIKE DRIVING 40 ON THE FREEWAY, GO 60 INSTEAD!

SERVO 8 HIGH PERFORMANCE 6 MHZ SINGLE BOARD COMPUTER

- 6 MHZ Z80B CPU — RUNS AT FULL SPEED WITH NO WAIT STATES
- FOUR LAYER BOARD (8.75" x 8") CAN MOUNT DIRECTLY TO MINIFLOPPY
- POWER REQUIRED 5 VOLTS AT 1.4 AMPS, NO OTHER VOLTAGES NEEDED
- UNIQUE FLOPPY CONTROLLER WITH AUTOMATIC SELF-ADJUSTMENT (NO POTS)
- FOR; 3.5" DRIVES, 5.25" DRIVES, 8" DRIVES, 1.6 MB 5.25" DRIVES
- CONFIGURATION MANAGEMENT INCLUDED ALLOWS EASY MENU-DRIVEN SELECTION OF OVER 50 DIFFERENT FLOPPY AND WINCHESTER FORMATS AS WELL AS BAUD RATES, PRINTER PORT SELECTION AND TURN-KEY AUTOLOAD
- S.A.S.I. (SCSI) BUS FOR WINCHESTER CONTROLLER (XEBEC 1410)
- TWO RS232 PORTS WITH SOFTWARE SELECTABLE RATES 300 TO 153.6K BAUD
- STANDARD CENTRONICS TYPE PARALLEL PRINTER PORT
- 50 PIN SYSTEM EXPANSION BUS WITH Z80 TERMS PLUS ADDITIONAL TERMS
- NOT A TOY, SERVO USES MIL-SPEC OR INDUSTRIAL GRADE PREMIUM PARTS

A & T SERVO 8 COMPUTER — $369 FOR CP/M ADD $70 VISA/M/C COD
CP/M 2.2 BIOS SOURCES — $50, INCLUDES WINCHESTER FORMATTER, EPROM, BIOS (Z80 CODE), CONFIGURATION UTILITY (TURBO PASCAL CODE)
SERVO EXPANSION BOARD WITH 128K ADDITIONAL RAM, CLOCK/CALENDAR WITH BATTERY BACKUP, TWO ADDITIONAL SERIAL PORTS, — $266
SERVO CONTROL INTERFACE WITH 6 ANALOG INPUTS AND 8 ANALOG OUTPUTS (12 BIT ADC, DAC) PLUS 16 DIGITAL INPUTS, 8 DIGITAL OUTPUTS — $495

SERVO COMPUTER CORPORATION
360B N. ELLENSBURG ST. BOX 566
GOLD BEACH, OREGON 97444
(503) 247-2021

PC SPEEDUP
V20 REPLACES 8088, V30 REPLACES 8086

90 DAY WARRANTY, HONEST PRICES, NO HIDDEN CHARGES
V20 KIT, V30 KIT: Speed up any 8086 or 8088 based PC. Includes warranty, tool, V20/V30 processor, user guide, BBS access to 3200 free PC programs.

TECHNICAL INFO: 100% compatible, 110 instruction superset, 8080 mode, CMOS, dedicated 16-bit bus, dual bus, pre-fetch pointer, hi-speed block move.

PRODUCTS AND PRICES:
V20/V30 kit in 5 or 8 mhz from $9.50 to $29.95
TURBO SEVEN Kit, switchable to 7 mhz in standard IBM PC. Under $100, CALL VP/N the V20 CP/M emulator — CALL for details. "AT" crystal kits $5 to $6.

Exec-PC, Inc.
P.O. BOX 11191
SHOREWOOD, WI 53211
(414) 242-2165

30 DAY MONEY BACK GUARANTEE
Available for the Lattice, Microsoft, Computer Innovations, DeSett, Mark Williams, and Aztec C compilers under MS-DOS, and most UNIX systems.
DISCOUNTS ON ALL LATTICE PRODUCTS

RAIMA CORPORATION
11717 Rainier Avenue South
Seattle, WA 98178, USA
(206) 772-1515 Telex 9103330300
CALL TOLL-FREE
1-800-843-3313
At the tone, touch 700-992.
**PROGRAM PACKS**

Borland has pioneered the area of ready-to-go programs including source code. Several "program packs" are being offered.

**Turbo GameWorks** $69.95
Borland International
Chess, Bridge, and Gomoku games

**Turbo Editor Toolbox** $69.95
Borland International
Complete WordStar like editor

**TRAINING AIDS**

The final area of Turbo products includes training aids, tutorials, and books. Borland was first here too, with its Turbo Tutor package. Four books on Turbo Pascal have been released to date and at least two more are scheduled to be out by Spring.

**Introductory to Turbo Pascal** $14.95 paperback, 300 pages
Author: Douglas S. Stivison
Publisher: SYBEX, Inc.

Balanced overall coverage of Turbo Pascal for both new and experienced programmers. Many useful examples and routines included in the text.

**Using Turbo Pascal** $19.95 paperback, 350 pages
Author: Steve Wood
Publisher: Osborne McGraw-Hill

Turbo Pascal for the experienced programmer. The book discusses program design and Pascal's syntax requirements, develops a useful application program, and gives an overview of some of the advanced utilities and features available with Turbo Pascal.

**Complete Turbo Pascal** $19.95, 464 pages
Author: Jeff Duntemann, Tech Editor
Publisher: Scott, Foresman

A balanced overall coverage of Turbo Pascal with over 500 text screens, 150 interactive questions, and 34 example programs.

**Product Suppliers**

**Pascal Tutor** $59.95
Gamma Software
Interactive tutorial program on Standard Pascal with over 500 text screens, 150 interactive questions, and 34 example programs.

**Turbo Tutor** $34.95
Borland International
Book-Disk package which highlights hard to understand concepts in Turbo Pascal. Both sample utilities and example programs are provided on diskette.

**Training Aids**

This book was designed for programmers to read for Turbo Tech Journal.

Publisher: Osborne McGraw-Hill

Balanced overall coverage of Turbo Pascal for both new and experienced programmers. Many useful examples and routines included in the text.

**Using Turbo Pascal** $19.95 paperback, 350 pages
Author: Steve Wood
Publisher: Osborne McGraw-Hill

Turbo Pascal for the experienced programmer. The book discusses program design and Pascal's syntax requirements, develops a useful application program, and gives an overview of some of the advanced utilities and features available with Turbo Pascal.

**Complete Turbo Pascal** $19.95, 464 pages
Author: Jeff Duntemann, Tech Editor
Publisher: Scott, Foresman

A balanced overall coverage of Turbo Pascal with over 500 text screens, 150 interactive questions, and 34 example programs.

**Product Suppliers**

**Pascal Tutor** $59.95
Gamma Software
Interactive tutorial program on Standard Pascal with over 500 text screens, 150 interactive questions, and 34 example programs.

**Turbo Tutor** $34.95
Borland International
Book-Disk package which highlights hard to understand concepts in Turbo Pascal. Both sample utilities and example programs are provided on diskette.

Micro/Systems Journal January/February 1986
INTRODUCING
The
COMPETITIVE EDGE
VELOCITY 286-10
A T™ COMPATIBLE

10MHz 80286
1024K on MOTHERBOARD
16 BIT HARD DISK TRANSFERS
SERIAL, PARALLEL, CLOCK
HERCULESTM Compatible Monochrome or Color Card
1.2 MB Floppy, A T™ Compatible Keyboard, DOS 3.1

20 MEGABYTE FAST HARD DISK
VELOCITY 286-10 $3695. Qty 1
DEALER & QUANTITY PRICING AVAILABLE

SHIPPING NOW!!!

— OPTIONS —
80287 Math CoProcessor
Enhanced Graphics Adapter
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 $395.00
Monochrome & Color Monitors

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 I is a trademark of Competitive Edge.
### Multi-Function Boards
- **Teletek 384K Memory Board with OK**: $125
- **Teletek 384K Memory Brd, Clock, OK**: $140
- **Teletek 384K Memory Board with 384K**: $199
- **Teletek 384K Memory Brd, Clock, 384K**: $214
- **PBS 384K Short Card without Memory**: $65
- **Tecmar Captain 64K**: $199
- **PBS 7 Pak 384K**: $197
- **Tecmar Maestro for at w/128K**: $359
- **PBS 7 Pak 640K**: $249
- **Ast 6 Pak + 64K, 1 PAR, 1 SER, Clock**: $239
- **Ast Advantage 128K for AT**: $425
- **PBS AT Extended Memory Card to 3MB OK**: $295
- **Above Board includes Piggy Back Card**: $296
- **STB Super Rio, Ser, Par, Clock 64K**: $395
- **STB Rio Grande 128K for AT**: $199
- **Quadrax Expanded Quadboard OK**: $99
- **PBS I/O - 100 AT Par., Serial for AT**: $345
- **Maynard Accent Board for AT OK**: $99

### Graphics Boards
- **Hercules Monochrome Card**: $299
- **Hi Res PBS Monochrome/Graphics Card**: $125
- **Hercules Color Card w/Parallel**: $159
- **PBS Color Card w/Parallel**: $125
- **Tecmar Graphics Master**: $449
- **PBS Color**: $99
- **Paradise Modular Graphics Card**: $279
- **Persyst Monochrome Card**: $175
- **Persyst Color Graphics Board**: $175
- **Persyst Color Combo 384K w/OK**: $310
- **Persyst Monochrome Combo w/OK**: $310
- **Quadrax Gold Quad Color w/OK**: $389
- **Everex Edge**: $279
- **Everex Graphics Edge**: $299

### Floppy Drives
- **TEAC FD55B**: $119
- **National JA551**: $99
- **Shugart SA455**: $125
- **AT Power Supply 192 watt**: $189
- **Mitsubishi 4851 PC 360K Drives**: $99

### Hard Disk Drive Kits
- **20 Megabyte for IBM PC**: $595
- **20 Megabyte for IBM AT-FAST**: $795
- **32 Megabyte for IBM AT**: $1295
- **70 Megabyte for IBM AT**: $2295
- **80 Megabyte for IBM AT**: $3295
- **112 Megabyte for IBM AT**: $3495
- **Alloy PC-STOR 20MB + Tape**: $2695
- **Alloy PC-STOR 52MB + Tape**: $3795
- **Everex 20MB FAST Tape B/U**: $695
- **Everex 60MB FAST Tape B/U**: $995
- **Expansion Box-for HD.**: $95

### Monitors
- **Amdek 310A**: $159
- **Amdek 600 Color Green Switch Audio**: $409
- **Quadrax Quadchrome**: $449
- **Quadrax Quadchrome II**: $389
- **Samsung Hi Res Green TTL**: $119
- **Samsung Hi Res Amber TTL**: $125
- **Princeton HX-12 Hi Res Color**: $459
- **Princeton MAX-12 Amber**: $459
- **Princeton SR-12 Ultra Hi Res**: $599
- **Princeton SR-12P**: $675
- **Taxan 410 RGB Color**: $349
- **Taxan 420 RGB Super High Res**: $425
- **Taxan 440 RGB Ultra High Res**: $565
- **Tecmar RGB 81140 Hi Res Color**: $595

### XT Clone
- **Low as $795. Qty. 5**
  640K Memory on Mother Board, Choice of Hercules Compatible Monochrome or Standard. Color Card includes one 360K Drive and Parallel Printer Port. 2nd Floppy, Monitor, Hard Drives up to 100 MB, 20, 60 or 100 MB Tape Drives optional.

### Accelerator Cards for IBM PC
- **Earth Computer Turbo Accel-286 Software Transparent 286 Power for PC 512K**: $995
- **True AT Power with Turbo Accel-286, Fast 16 Bit Memory & Accepts 80287**: $395
- **Earth Computer 8MHZ Z80 H Slave Run CP/M 2.2 Software on Your PC, XT or AT**: $395
- **Earth Computer 8MHZ Z80H S-100 Slave Runs Under Turbodos**: $395

**ALL PRICES SUBJECT TO CHANGE WITHOUT NOTICE — SHIPPING EXTRA**

IBM is a registered trademark of International Business Machines, CP/M is a registered trademark of Digital Research, Inc. All circuit board names are trademarks of the respective manufacturer.
COMPETITIVE EDGE
P.O. Box 556 — Plymouth, MI 48170 — (313) 415-0665

Competitive Edge brings you Lomas Data Products Inc.

S-100 CIRCUIT BOARDS

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thunder 186</td>
<td>256K, 2 Serial, 1 Parallel, Floppy Disk Controller, Clock, All on One Board with Concurrent DOS</td>
<td>$895</td>
</tr>
<tr>
<td>8MHX Lighting 286 CPU</td>
<td>$821</td>
<td></td>
</tr>
<tr>
<td>10MHZ Lighting 8086 CPU</td>
<td>$520</td>
<td></td>
</tr>
<tr>
<td>Control It All Floppy &amp; HD Controller</td>
<td>467</td>
<td></td>
</tr>
<tr>
<td>256K Dram Megasram</td>
<td>580</td>
<td></td>
</tr>
<tr>
<td>1024K (1 Megabyte) Megasram</td>
<td>599</td>
<td></td>
</tr>
<tr>
<td>Ram 67 128K Static 100ns Chips</td>
<td>506</td>
<td></td>
</tr>
<tr>
<td>Octopart 8 Serial to 38.4K</td>
<td>320</td>
<td></td>
</tr>
</tbody>
</table>

COMPETITIVE EDGE INTEGRATED SYSTEMS

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>THUNDER 186, 4 SLOT PC STYLE CABINET, 2-5&quot; FLOPPYS, CDOS</td>
<td>$1595</td>
<td></td>
</tr>
<tr>
<td>THUNDER 186, 4 SLOT, 20 MB HARD DISK, 1-5&quot; FLOPPY, CDOS</td>
<td>2595</td>
<td></td>
</tr>
<tr>
<td>THUNDER 186, 4 SLOT, 20 MB HD, 512K, 1-5&quot; FLOPPY 4 USER</td>
<td>3195</td>
<td></td>
</tr>
<tr>
<td>THUNDER 186, 4 SLOT, 2-5&quot; FLOPPY, COLOR MAGIC, KEYBOARD</td>
<td>2250</td>
<td></td>
</tr>
<tr>
<td>6MHZ 286, 1-5&quot; FLOPPY, 1024K, 20MB HD, 15 SLOT, 7-8 USERS</td>
<td>4695</td>
<td></td>
</tr>
<tr>
<td>6MHZ 286, 1-5&quot; FLOPPY, 1024K, 40MB HD, 15 SLOT, 7-8 USERS</td>
<td>5295</td>
<td></td>
</tr>
<tr>
<td>8MHZ 286, 1-5&quot; FLOPPY, 1024K, STATIC, 40 MB HD, 10 SERIAL, 7-8 USERS</td>
<td>7550</td>
<td></td>
</tr>
<tr>
<td>8MHZ 286, 1-5&quot; FLOPPY, 512K STATIC, 40 MB HD, 2 SERIAL &amp; 80287</td>
<td>6495</td>
<td></td>
</tr>
<tr>
<td>8MHZ 8086/8087, 1-5&quot; FLOPPY, 512K DRAM, 20 MB HD, 2 SERIAL CDOS or MSDOS</td>
<td>3695</td>
<td></td>
</tr>
</tbody>
</table>

A COMPLETE LINE OF TERMINALS, PRINTERS and SOFTWARE FOR YOUR SYSTEM NEEDS

FEATURING CompuPro® FROM VIASYN™

S-100 CompuPro® CIRCUIT BOARDS

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Disk 1A&quot; Floppy Disk Controller</td>
<td>$371</td>
<td></td>
</tr>
<tr>
<td>8 MHZ 286 CPU A&amp;T</td>
<td>671</td>
<td></td>
</tr>
<tr>
<td>Ram 22&quot; 256K Static Ram A&amp;T</td>
<td>446</td>
<td></td>
</tr>
<tr>
<td>Interfacer 3&quot; 8 Serial Ports</td>
<td>371</td>
<td></td>
</tr>
<tr>
<td>SPU Z+ 8MHZ Z80, 256K Multisuser</td>
<td>256</td>
<td></td>
</tr>
<tr>
<td>MDRIVE-H+ 512K Memory Drive</td>
<td>371</td>
<td></td>
</tr>
<tr>
<td>PC Video Board for IBM® Compert</td>
<td>396</td>
<td></td>
</tr>
</tbody>
</table>

Your Choice $263 — SYSTEM SUPPORT 1, 85/88-10MHZ, INTERFACER 4, RAM 23-128K HARD DISK SUB-SYSTEMS

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 MEGABYTE HARD DISK, CABINET, DISK 3, SUB-SYSTEM</td>
<td>$1295</td>
<td></td>
</tr>
<tr>
<td>40 MEGABYTE HARD DISK, CABINET, DISK 3, SUB-SYSTEM</td>
<td>1895</td>
<td></td>
</tr>
<tr>
<td>80 MEGABYTE HARD DISK, CABINET, DISK 3, SUB-SYSTEM</td>
<td>2595</td>
<td></td>
</tr>
<tr>
<td>LOMAS 24 MB TAPE BACKUP SUB-SYSTEM</td>
<td>1495</td>
<td></td>
</tr>
<tr>
<td>40 MEGABYTE HARD DISK, CABINET, DISK 3, 5&quot; FLOPPY</td>
<td>2095</td>
<td></td>
</tr>
<tr>
<td>8&quot; &amp; 5&quot; FLOPPYS, 40 MB HARD DISK SUB-SYSTEM</td>
<td>2395</td>
<td></td>
</tr>
<tr>
<td>2-40 MB HARD DISK DRIVES IN CABINET, DISK 3, SUB-SYSTEM</td>
<td>3095</td>
<td></td>
</tr>
</tbody>
</table>

SYSTEMS INTEGRATED BY COMPETITIVE EDGE WITH COMPUPRO® COMPONENTS

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>8MHZ 286 SYSTEM</td>
<td>$4795</td>
<td></td>
</tr>
<tr>
<td>8MHZ 286 &amp; 512K STATIC RAM</td>
<td>9 SERIAL PORTS, SS1 &amp; I/O 3</td>
<td></td>
</tr>
<tr>
<td>ONE 5&quot; FLOPPY &amp; 20MB HARD DISK</td>
<td>15 SLOT CABINET, 30 AMPS &amp; 8</td>
<td></td>
</tr>
<tr>
<td>CONCURRENT DOS 8-16 MULTI-USER</td>
<td>EM8080 SOFTWARE &amp; BIT EMULATOR</td>
<td></td>
</tr>
<tr>
<td>OPTIONAL 8MHZ 280H SLAVES</td>
<td>COMPONENTS ASSEMBLED &amp; TESTED TOGETHER, ALL INTERNAL CABLES SUPPLIED</td>
<td></td>
</tr>
<tr>
<td>8MHZ 286 SYSTEM</td>
<td>$5395</td>
<td></td>
</tr>
</tbody>
</table>

10MHZ 8088-85/88 SYSTEM

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>85/88 CPU &amp; 128K STATIC</td>
<td>65/68 CPU &amp; 128K STATIC</td>
<td></td>
</tr>
<tr>
<td>SYSTEM SUPPORT &amp; I/O 4</td>
<td>4 SERIAL PORTS &amp; 1 PAR</td>
<td></td>
</tr>
<tr>
<td>ONE 5&quot; FLOPPY &amp; 20MB HD</td>
<td>SYSTEM UPGRADEABLE TO MULTI-USER &amp; 280 SLAVES ALL</td>
<td></td>
</tr>
<tr>
<td>CP/M-8-16 &amp; 16BIT</td>
<td>15 SLOT CABINET</td>
<td></td>
</tr>
<tr>
<td>5&quot; FLOPPY WILL READ IBM</td>
<td>5&quot; FLOPPY WILL READ IBM</td>
<td></td>
</tr>
</tbody>
</table>

ALL PRICES SUBJECT TO CHANGE — SHIPPING EXTRA

IBM, 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.
# COMPETITIVE EDGE

P.O. Box 556 — Plymouth, MI 48170 — (313) 451-0665

HIGH PERFORMANCE PRODUCTS RUNNING UNDER TURBODOS

TELETEK ENTERPRISES S-100 MASTER/SLAVE BOARDS

<table>
<thead>
<tr>
<th>Product</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Telemaster</td>
<td>$495</td>
</tr>
<tr>
<td>Systemaster I</td>
<td>$795</td>
</tr>
<tr>
<td>Systemaster II</td>
<td>$795</td>
</tr>
<tr>
<td>8MHz SBC 86, 512K</td>
<td>$963</td>
</tr>
<tr>
<td>5MHz SBC 86, 512K</td>
<td>$795</td>
</tr>
<tr>
<td>6MHz SBC 1, 128K</td>
<td>$375</td>
</tr>
<tr>
<td>HDC</td>
<td>$375</td>
</tr>
</tbody>
</table>

| 4MHz Z80A MASTER SYSTEMASTER I, 2 SERIAL, 2 PAR, FOC, CLOCK | $495 |
| 8MHz Z80H 128K MASTER, SYSTEMASTER II®, 2 SER, 2 PAR, FDC CLK | 795 |
| 8MHz 8086 512K SLAVE SBC 86 RUNS WITH EITHER MASTER ABOVE | 963 |
| 5MHz 8086 512K SLAVE SBC 86 RUNS WITH EITHER MASTER ABOVE | 795 |
| 8MHz 8087 OPTION FOR 8MHz 8086 SBC 86 | 175 |
| 5MHz 8087 OPTION FOR 5MHz 8086 SBC 86 | 109 |
| SBC I 6MHz Z80B 128K SLAVE RUNS WITH EITHER MASTER ABOVE | 375 |
| HDC HARD DISK CONTROLLER — ST506 INTERFACE RUNS WITH ABOVE | 375 |
| SBC II DUAL SLAVE BOARD (TWO Z80’S PORTS ETC.) | 689 |

**COMPETITIVE EDGE INTEGRATED SYSTEMS**

<table>
<thead>
<tr>
<th>Product</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>4MHz Z80A 64K, CPM 2.2, 2-5” FLOPPYS, 4 SLOT CABINET</td>
<td>1295</td>
</tr>
<tr>
<td>8MHz Z80H 128K, SINGLE USER TURBODOS WITH SPOOLING, 2-5” FLOPPYS, 4 SLOT CABINET READY TO RUN</td>
<td>1695</td>
</tr>
<tr>
<td>8MHz Z80H, 128K MASTER, 4-6MHz 128K SLAVES, 1-5” 20MB HD</td>
<td>4795</td>
</tr>
<tr>
<td>10 SLOT 1-5” FLOPPY READY TO RUN WITH TURBODOS</td>
<td></td>
</tr>
<tr>
<td>8MHz Z80H, 128K MASTER, 4-8MHz 512K SLAVES, 1-5” 20MB HD</td>
<td>7495</td>
</tr>
<tr>
<td>10 SLOT 1-5” FLOPPY READY TO RUN WITH TURBODOS</td>
<td></td>
</tr>
<tr>
<td>4MHz Z80A, 64K MASTER, 4-6MHz 128K Z80B SLAVES, 1-5”</td>
<td>4395</td>
</tr>
<tr>
<td>20 MB HD, 10 SLOT 1-5” FLOPPY READY TO RUN WITH TURBODOS</td>
<td></td>
</tr>
</tbody>
</table>

SYSTEMS AVAILABLE WITH UP TO 16 SLAVES AND 2-112 MD HD (224 MB TOTAL)

**EARTH COMPUTER PRESENTS**

8MHz Z80H TURBO MASTER 8, 64K, 2 SERIAL, PARALLEL PRINTER PORT, FLOPPY AND ST506 WINCHESTER CONTROLLER ALL ON ONE BOARD

<table>
<thead>
<tr>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>$795</td>
</tr>
</tbody>
</table>

8MHz Z80H TURBO MASTER 8, 256K INSTEAD OF 64K

<table>
<thead>
<tr>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>$895</td>
</tr>
</tbody>
</table>

**EARTH COMPUTER TURBO SLAVE**

8MHz Z80H-128K SLAVE 2 SERIAL TURBO SLAVE I RUNS WITH ALL TURBODOS S-100 SYSTEMS

<table>
<thead>
<tr>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>395</td>
</tr>
</tbody>
</table>

All Prices Subject to Change — Shipping Extra, Visa M/C Accepted

Systemaster I & II, HDC, SBC I, SBC II are either registered trademarks or trademarks of Teletek Enterprises. CPM is a registered trademark of Digital Research Inc., Turbodos is a trademark of Software 2000 Inc.
Implementing PC-DOS on Non-IBM Compatible Computers

By Christopher Cochran & Kristofer Sweger

INTRODUCTION

When IBM introduced the IBM Personal Computer with the MS-DOS operating system, the battle between CP/M-86 and MS-DOS was decided. At the time, MS-DOS had only a slight edge over CP/M-86 in functionality, its main improvements being dated files and faster disk access. Since then, Microsoft has expanded upon the original operating system to provide some truly exceptional capabilities for a microcomputer operating system. These include tree-structured file systems, installable device drivers, redirection of console input and output to other devices or files, piping the output stream of one program to the input stream of another, and greater disk capacity.

A recent study by Input, a Mountain View CA market research firm, projects that by 1990, PC-DOS or MS-DOS will be used on 74 percent of business microcomputers. CP/M on only one percent, and other proprietary operating systems (i.e. Unix or manufacturers' systems) will account for the remaining 25 percent. In 1984, comparable figures were 33 percent PC-DOS, 13 percent CP/M and 54 percent other. If this forecast is accurate, the market share for PC-DOS will more than double, CP/M will almost vanish, and other operating systems will lose half their market share.

Because MS-DOS is now the operating system chosen by most software developers, you will need to have MS-DOS on your computer to run the majority of 8086 (and 8088) software being written today. So you skip down to your local computer store with $65 to buy a copy of MS-DOS for your nice, fast S-100 10 Mhz 8086 computer, only to discover that you can only buy it for an IBM-PC or PC-compatible! You see, MS-DOS is distributed only for hardware whose manufacturers have obtained a license to sell MS-DOS for their specific equipment. For most low unit-quantity computers the cost per copy of the license agreement with Microsoft plus BIOS customization has been prohibitive. Furthermore, until now there has been no publically documented procedure for porting MS-DOS to these machines. Your only choice was to enter into a licensing agreement with Microsoft, which, if they will even talk to you, is rather expensive for a one-time port.

The purpose of this article is to explain how you go about modifying an implementation of MS-DOS designed for another computer so that it correctly executes on yours. If you embark on such a project, be sure that you purchase a copy of MS-DOS so that you have the legal right to use and modify it without violating the law.

FEASIBILITY OF THIS PROJECT FOR AN INDIVIDUAL USER

Porting IBM PC-DOS to an arbitrary computer is not a simple task to be done in a few leisurely afternoons or evenings. It is debatable whether this project is even worthwhile at all for a single installation, given the amount of work it requires. To begin with, the following equipment, software tools and expertise are prerequisite for a successful implementation:

- A development computer with an 8086, 80286 or 8088 CPU. 256k bytes or more of main memory, one or more floppy disk drives, a video console and some sort of hard copy device.
- If your development computer can't read soft-sectored 5.25" PC-DOS diskettes, you will need access to an IBM-PC or other computer running PC-DOS. This is necessary to extract the DOS system files from the IBM release 5.25" floppy disks. Some means must then be provided to transfer these files to your development system via a serial or parallel port which can transfer 8-bit binary data.
- Necessary software tools include an 8086 assembler, disassembler, machine code debugger, and editor. All of these are available under any system that already has CP/M-86 (or CCP/M, MP/M-86), MS-DOS (or PC-DOS), or RMX up and running. You would be well advised to also have a good high-level language on hand so that you can develop, as needed, various source code scanners and filters to assist the development process.
- You should have experience in assembly language programming and debugging. Operating system work is notoriously difficult to debug because when the system does not come up, your machine is dead and debugging facilities cannot be loaded without destroying the evidence left by the crash, i.e. you need a chicken before the egg. The primary way to overcome this problem is to write clear, well-documented assembly code that can be painstakingly followed by 'hand' until the error is found. Built-in tracing output can also be helpful in locating the failure. Modular design and other structured programming techniques are a must. (Yes, structured programming is possible in assembler!)
- You should also have prior experience in software drivers for CRTs, printers, disk drives and boot ROMs. You will likely have to modify the boot ROM on your computer if you ever want to load the DOS with your reset button. If you modify a ROM, you will also need access to the appropriate EPROM programmer and associated software.
- This article assumes you understand the DOS well enough to write programs in assembler that access the operating system functions with software INT calls. We also assume you can implement an MS-DOS device driver and attach it to...
the system at boot time with the appropriate DEVICE = command in a CON-
FIG.SYS file.

Given that you possess the above prerequisites, you should also have some good reasons for spending the effort to successfully port DOS to your computer. See if some of the following apply:

- Having control over your own operating system implementation provides independence from one manufacturer's software and the power to choose and tailor your system to your own needs.

- The ongoing flexibility of your own operating system compared to a commercially available system will have major advantages.

- You have a substantial investment in high-performance hardware and/or more than one or two computers for which IBM-PC DOS is both desirable and available, and purchase of an IBM-PC or compatible instead of porting the DOS would be more costly or would not provide all desired capabilities.

- You already possess numerous I/O drivers that can be worked into a new BIOS for PC-DOS, significantly reducing development cost.

- Your hardware base is continually growing and you need an operating system that can grow with it. For example, our own systems have each grown to 10 I/O devices, 768K RAM, 72 megabytes of hard disk space, dual 8" floppy drives and dual 5.25" floppy drives. Everything is easily accessible under PC-DOS.

- You do not feel comfortable unless all the system software on your computer is under your total control. After all, manufacturers and distributors come and go, and the support for your computer can suddenly disappear leaving you stranded.

- You want to learn as much as possible about how PC-DOS works, and are not too concerned about the amount of time you will need to invest.

WHICH IBM-PC DOS IS BEST?

For the purpose of this discussion, we will assume that you will be porting the MS-DOS version known as IBM DOS 3.1 to your computer. Earlier versions could be used, but they lack many desirable features and your conversion effort will be roughly the same. Avoid completely versions prior to DOS 2.0 (ie. versions 1.1 and 1.25) or DOS 3.0 because of a variety of bugs and unimplemented features. DOS 3.1 supports greater disk capacity, shared files and network capabilities. It has several new commands and has resolved various bugs present in earlier versions. Our version for the CompuPro (Viasyn) S-100 computer has been using DOS 3.1 since July 1985 without detecting a single operating system malfunction.

There is also a generic MS-DOS on the market, which is used by most of the IBM-PC look-a-likes. We have seen some bugs in these versions which do not seem to be present in the IBM versions at the same revision level. Although we have no direct evidence of this, we suspect the IBM versions are more thoroughly tested and debugged than the generic versions. Because of this and because the IBM-DOSes are much more available and reasonably priced, we suggest you avoid the generic MS-DOS versions altogether.

DOS & BIOS STRUCTURE

The DOS, as we will refer to it from now on, is contained in one visible and two hidden files on the IBM release diskette named COMMAND.COM, IBMIBIO.COM, and IBMIBOS.COM, respectively. The IBMIBOS.COM file is the disk operating system itself, free of all hardware dependencies other than requiring an 8088 or 8086 CPU. The IBMIBIO.COM file contains the machine code that provides an interface between the running IBMIBOS.COM and the underlying hardware. COMMAND.COM is the DOS command shell, which accepts user commands and controls program execution. To access the hidden files, you need to use a utility capable of finding and opening hidden files, such as DEBUG.COM or CHMOD.EXE (a public domain program), or you must write your own utility that uses extended FCB techniques.

The IBMIBOS.COM and COMMAND.COM files will run totally unchanged on any 8086-family computer system. However, you will be replacing the entire IBMIBIO.COM file with a new implementation which provides an interface to your particular hardware configuration. Virtually all the work in porting the DOS to your system is in the development and testing of your own custom BIOS (Basic Input/Output System). This BIOS will contain all the routines that are specific to your computer. To work properly, it must provide all the services and structures expected of it by the DOS. The BIOS can be divided into four major components:

- The first component is initialization of the BIOS from cold-boot. This involves moving the entire system to the base address of 0070:0000, determining the size of memory, calling the ROMBIOS initialization routine (if any), initializing any other hardware specifics (such as interrupt controllers), and jumping to the Startup Module to complete the cold-boot.

- The second chain of boot-in devices. The boot-in devices are provided simply as a linked list of device headers and associated code. These device drivers are designed in an identical manner to the custom device drivers attached at boot time by the CONFIG.SYS mechanism. Device drivers are described in some detail in the IBM DOS Technical Reference.

- The third component contains the ROMBIOS emulation routines. The IBM-PC has a ROM in high address space to provide a low-level interface between the IBMIBIO and the hardware. Since standard entry points into the ROMBIOS are called by many of the supplied utilities, we recommend you provide your own emulation of ROMBIOS functions to support these utilities. ROMBIOS emulation is described in more detail later on, is not hard to implement, and provides the compatibility necessary to run many IBM-only programs.

- The final component is the Startup Module. This relatively independent module within the BIOS is responsible for directly interacting with the DOS to finish all remaining initialization tasks. This code should be extracted from the end of the original IBMIBIO.COM file and disassembled into a usable source file. We will give some details on how to do this later in this article.

BIOS INITIALIZATION

The BIOS is entered from the bootstrap loader at its base origin of 0070:0000. However, most 8086 loaders on S-100 systems load a system at 0040:0000, just above the interrupt vector table (0000:0000), and then do a long jump to 0040:0000. Furthermore, if your development computer is running CP/M-86, it is desirable to boot your initial DOS port directly from CP/M as a command file (.CMD), which will of course load the system into some arbitrary base address. Therefore, your BIOS should, from the very start, always relocate itself to the required base address of 0070:0000 as the first initialization action.

Both the BIOS and DOS are internally relocatable code, so moving them to 0070:0000 is a simple task of copying the loaded memory image from its load address (ie. CS:0000) to 0070:0000 with one block move instruction. If there is any possibility that the copied image will overlap the loaded image, you must be sure to implement the move in the proper direction, so each byte is copied before it is overwritten with different contents. Because of this overlap problem, it is wise to PUSH a FAR RET address of 0070:START onto the stack, where START is the starting offset where your code continues from after the move. The block move instruction is then immediately followed by a FAR RET instruction, causing the CPU to jump to the correct location even if the move has overwritten the FAR RET before it is executed. This is possible because the 8086 CPU will already have the FAR RET instruction in its prefetch queue. However, you must of course ensure that the stack contents will be preserved throughout all of this.

If you have more than one boot device,
**EDITOR’S NOTE**

We have been in touch with several readers who have implemented PC-DOS on their S-100 systems. As we went to press with this issue we received a copy of another article on the subject. Because of space limitations we will not be able to publish it (it is really a small book). It includes a complete, well documented, BIOS. A copy can be obtained, on disk ($15) from the author: Fred Green, 4003 W. Iliff Lane, Lakewood CO 80227; (303) 986-6651.

Readers should also be aware that Computer House (see ad in this issue) sells an implementation of PC-DOS for CompuPro systems.

for example a floppy and a hard disk, your boot-strap loader should communicate to the BIOS which device the DOS was loaded from. This information is needed later on to determine which drive to search for COMMAND.COM and CONFIG.SYS, and which default drive to 'come up' on. Therefore your BIOS should at this store your BIOS will be initialized as soon as the 

Next, your BIOS should fill the entire interrupt vector table with a vector to a special diagnostic routine that traps an unexpected INT calls and informs the user at the console whenever such an event occurs. This is very important to implement right from the start, because uninitialized interrupts are a likely cause of system crashes in early versions of your BIOS port with this feature. It can wait until you have reached the final 'bells-and-whistles' stage. Instead, assume that your system always boots from and defaults to the A: drive.

A memory search should now be made to determine how much memory is installed in your computer. You could make this a fixed quantity in your BIOS, but you would then have to modify your BIOS whenever your memory size changed. To determine your memory size, just cycle through memory, paragraph by paragraph, looking for memory that cannot be modified. A sample memory search routine is shown in Listing 1.

The final action of the BIOS initialization before jumping to the Startup Module is to pass parameters to that module by way of PUBLIC variables:

- The total memory paragraphs currently installed in the system;
- The base segment address where the DOS image currently resides;
- The segment address where the BIOS ends, including all appended working variables and constants it will ever require. This is where the DOS image will be moved and where it will permanently reside;
- The drive number to be used as the default drive, and where the command shell and CONFIG.SYS files are to be found;
- The address of the device driver header in the linked list of built-in devices.

All remaining startup and initialization tasks are performed under control of the Startup Module, which is free of hardware dependencies.

**BUILT-IN DEVICE DRIVER CHAIN**

Although I/O drivers can be attached to this operating system at boot time, there are certain drivers that must be built into the BIOS for the system to start up. After all, how can the system process a CONFIG.SYS file without having any initial disk drivers? Character drivers also must be present to support the standard I/O devices: CON, PRN and AUX.

All the device drivers in the BIOS are designed just as if they were to be installed at boot time via a CONFIG.SYS file with DEVICE= commands. See the DOS 3.1 Technical Reference Manual for information about designing such drivers. Several important aspects of built-in device drivers include:

- All built-in drivers are linked together into a single list, using the link field of the driver header (as described in the DOS Technical manual). The last driver in the chain must point the address FFFF.FFFF to indicate that no more drivers follow.
- The minimal set of device drivers must include a consoledriver named CON, a printer driver named PRN, an auxiliary I/O device named AUX, a time/date driver named CLOCKS, and a block device driver for at least one disk drive. It is recommended that in your final system you also support devices named COM1, COM2, LPT1, LPT2, and LPT3, since these are the device names included in IBMBO.COM, and may be referred to by name by programs developed for the IBM-PC.
- For the sake of simplicity in your first attempts, PRN and AUX devices can merely be place holders which perform their actual I/O operations with the console. However, the console (CON) must be implemented as fully as possible, and in particular, the non-destructive input command (command 5) must operate properly.
- The CLOCKS device may also be implemented in a minimal way, by merely returning a 6 byte string of zeros on input requests and ignoring all output or other requests. However, if you do have a clock, the benefits of a full implementation are well worth the effort. Remember that you must set Bit 3 in the attribute field of the CLOCKS device header to tell the DOS that this is the clock device. Also, note that dates are passed to and from the CLOCKS device as the number of days since January 1, 1980. If your hardware maintains the date in month-day-year format, you will have to convert all dates input and output to day counts to support it properly.
- The STRATEGY and INTERRUPT entry point protocols for all built-in drivers are so similar that you can economize your efforts by combining them into common entry points used by all device drivers.
- The initialization entry point (command 0) for each device is very important to implement. This operation is called exactly once for each device driver when the system first comes up, and is never called again. You should place all necessary initialization for each device into its respective INIT entry point. These individual INIT calls are performed in the order that the device drivers are linked together. If you have any order dependencies between your various drivers, be sure that they are linked in the appropriate way.
- The IOCTL entry points (commands 11 and 12) are never accessed by the standard DOS utilities, by the DOS itself, or by most other programs that run under the DOS. They exist primarily for custom applications and you should therefore leave them out of your implementation, at least initially.

Be careful to set the attribute field of the device headers with the proper bit settings to reflect each device driver type. Just one wrong bit can easily prevent the
The console device CON has different attribute bit settings from the other character drivers: Bit 0 and Bit 1 will both be on to signify that CON is the “standard” input and output device. Optional for devices CON is Bit 4, which tells the DOS that the console supports INT 29h for character output and ROMBIOS INT 10h for the CLS command. If you set Bit 4, you must implement both an INT 29h handler for console output and ROMBIOS emulation in your BIOS. Interrupt 29h is used by the DOS instead of normal CON driver STRATEGY and INTERRUPT calls to speed up console output, and can double its speed. The INT 29h handler simply takes the byte in AL and outputs it to the CON device, preserving all registers that are used.

Do not implement a NUL device. This device is actually present in the DOS as the real head of the device chain and there is no way to remove it or replace it for any reason.

You are advised to implement your built-in device chain as a separately assembled source module which is one of several object modules linked together to form the entire BIOS image. Your device drivers will likely be the most volatile area of your system and keeping them separate will help you manage the code as well as protect the other source modules in the system.

ROMBIOS EMULATION

Many useful program files are included on the PC-DOS release disk. Naturally, you will want as many of these as possible to work properly on your system. However, to run a PC-DOS program on a non-IBM computer, you must be certain that any hardware dependencies are removed or properly handled. Hardware dependence in a program can manifest itself in three major ways: use of IBM-PC ROMBIOS software interrupt calls, direct access to IBM-PC input/output ports, and direct access to fixed IBM-PC memory locations. All three are frequently present in programs developed for the IBM-PC because software developers have had to bypass the operating system in order to achieve adequate performance. More subtle but equally catastrophic dependencies can also appear because of programmed-in assumptions regarding device characteristics, control sequences, unit numbers, execution timing, or in the way IBMIBIO.COM is implemented.

Tables 1a, 1b and 1c show the PC-DOS program files grouped according to their hardware dependence. The first group of programs in Table 1a will require no changes or special attention to work properly. The second group (Table 1b) includes programs with at least one software interrupt call to the ROMBIOS routines which are built into the IBM-PC hardware but not included in PC-DOS, and so will not operate correctly unless you can deal with their ROMBIOS calls. The remaining DOS program files (Table 1c) are so hardware dependent that it is not practical to get any of them working without a major rewrite.

To use a program which makes ROMBIOS calls, it must either be patched to bypass each call (but preserving the desired effect), or your BIOS must provide an equivalent software interface to the program.

Since you won’t need any ROMs to emulate the ROMBIOS software interface, you can include ROMBIOS code directly in the BIOS. This is the preferable solution because:

- None of the DOS system files will have to be patched, so you won’t need to identify all the locations in each program which must be changed, and you won’t have to figure out what each patch should be;
- Programs or systems which were written for an IBM-PC environment have a better chance of working properly on your system without alteration (or with fewer changes) if the ROMBIOS environment is provided in your BIOS; and
- Implementation of future DOS releases will be much easier, since you probably won’t need to patch any of the new system files.

IBM has defined ROMBIOS interrupts numbered from 10h to 1Fh for the PC. These are listed in Table 2, classified according to their relative implementation difficulty. All programs access these routines only through software interrupts. Four of the routines are trivial to create: two simply return a word value in register AX, and two are IRET instructions. Five more will not be needed in normal operations: resident Basic, bootstrap, video initialization, diskette parameters, and video graphics characters. The remaining seven range from simple to difficult to develop, with interrupt 10h (video display) both the most difficult and most important to provide. For more complete documentation on ROMBIOS requirements see the IBM-PC, PC/XT or PC/AT Hardware Technical Reference.

Interrupt 10h for the video display is very screen capability dependent. Unfortunately, this is the one ROMBIOS routine which is most frequently used by programs, and complete emulation really requires hardware like the IBM-PC memory-mapped screen, which contains 25 lines of 80 characters and an attribute byte for each character. However, useful results can be obtained by emulating interrupt 10h calls to the extent possible on an ordinary CRT terminal. If you do this, programs which call the ROMBIOS to clear the screen, address the cursor, or perform other simple video functions will operate properly.

Like the other major components in the BIOS, the ROMBIOS should be implemented as a separately assembled module which is later linked together with others into an executable image. The ROMBIOS module should have a PUBLIC entry point which is called to initialize the interrupt vector entries for the ROMBIOS routines, and which initializes the ROMBIOS data area at location 0040h:0000.

STARTUP MODULE STRUCTURE

The Startup Module initializes the DOS, processes the CONFIG.SYS file, sets up the necessary device tables, and finally loads and starts the command shell. This module is rather complex and will vary in its details from one DOS version to another. But because it is free of hardware dependencies, it can be extracted from the IBMIBIO.COM file and employed in your system. The steps you should follow to create a working Startup Module are as follows:

1. Load IBMIBIO.COM into a debugger with disassembly capability. It would be wise to perform this on an IBM-PC. Note where the end of this file is in memory.

2. Locate the Startup Module within the memory image of IBMIBIO.COM and save all the data from this location to the end of the memory image. Use the map given in Figure 1 to help you find its starting location.

3. Disassemble the resulting file into a usable assembly source file. This is not a trivial task and represents the most significant barrier to porting this operating system to your computer. If you can purchase a top notch disassembler for this task it will save you much work, not to speak of any sanity it preserves. The process of converting a disassembler’s output into a usable source file will consist of the following steps:

(a) Many instructions will contain ‘immediate values’ that have been erroneously changed to labels. You must identify what these are and change them back into constants.

(b) Many instructions will have code labels on them which are not referenced anywhere in the source. These should be deleted so that ‘real’ labels can be identified and renamed with meaningful names.

(c) There are many calls in the Startup Module to INT 21h functions. Since they are documented in the IBM 3.1 Technical Reference Manual, you can use this knowledge to deduce the function of the surrounding context from which they are called. Table 3 lists all the INT 21h functions that are called from this module, plus some internal DOS.
calls whose function we have identified.

(d) Add the necessary assembler directives so that the new source file can be assembled without assembler errors. The Startup Module has all data and code in the same segment (8080 model), but frequently changes its segment registers.

(e) Before you make any substantive modifications to the code or data areas, make sure that your new source can be assembled, linked, and that the resulting image produced exactly matches the original image from which your disassembly was constructed. This is a good way to catch typos and other mistakes while they can still be caught.

When creating an acceptable source file from a disassembly file, you should strive for completely commented source with all disassembler-generated labels renamed to meaningful names. Like the other major components in the BIOS, the Startup Module should be implemented as a separately assembled module which is later linked together with the others into an executable image. Listing 2 shows the layout of the parameter data area in the Startup Module for DOS 3.1, which you must fill with appropriate values before jumping into the module.

In the next issue the authors will conclude with discussions of implementation strategy, loading the system into memory, preparing DOS-readable disks, getting the system to work, and extensions to the completed system.

References

1. IBM Disk Operating System Version 3.10 Reference, IBM Corp. #6138519.
2. IBM Disk Operating System Technical Reference (covers Versions 3.1, 2.1, and 2.0), IBM Corp. #6024213.
3. Disk Operating System by Microsoft, Inc. (Version 2.0), IBM Corp. #6936752.
4. Technical Reference, IBM Personal Computer, IBM Corp. #1502234.
5. Technical Reference, IBM Personal Computer/XT, IBM Corp. #6322508.
7. IBM Personal Computer Macro Assembler, IBM Corp. #6172234.

Kristofer Sweger has over fifteen years experience as an independent consultant and programmer. His assignments have included design and construction of custom systems and application software, development of computer data for litigation and research, and modeling and analysis of economic, production and transportation systems. He has extensive experience with a wide selection of programming languages on both large and small computer systems. His education includes graduate studies in Electrical Engineering and Computer Sciences at the University of California, Berkeley and a B.S. in Computer Science from California State Polytechnic College, San Luis Obispo. He is a member of ACM and the IEEE Computer Society.

Christopher Cochran has eleven years of experience that includes work in statistical analysis and mathematical programming, computer architecture, operations research, system design, programming language methodology and systems programming. He majored in computer science at the University of California in Berkeley and is the author of MegaBasic, a high performance, extensible 8086-based semicomputer designed for large scale programming use. Mr. Cochran has interests in astronomy, quantum mechanics, electronics, mathematics and is an accomplished jazz and classical musician on flute, saxophone and clarinet. He is a member of ACM.
THE PROGRAMMER'S SHOP™ helps save time, money and cut frustrations. Compare, evaluate, and find products.

SERVICES
- Programmer's Referral List • Dealer's Inquire • Compare Products • Newsletter
- Help find a Publisher • Rush Order
- Evaluation Literature FREE • Over 700 products • BULLETIN BOARD - 7PM to 7AM 617-826-4086

AI - Expert System Dev't
- ExpertEase - Inductive, PC DOS Call
- EXSYS - All RAM, Probability, Why
- Trees, Solid, files, popular PC DOS $359

INSIGHT 1 - Probabilities, required thresholds, menus, fast PC DOS $95
- INSIGHT 2 - adds backward, forward, partial
- tions, db2, lang. access. PC DOS $449
Others: APES ($359), Advisor ($494), ES Construction ($100), ESP ($845), Expert Choice ($449), more.

AI - LISP
- GC LISP - "Common", rich
- Interpreter - Interactive Tutorial 495 Call
- LARGE Model - 2 to 15 meg $495 649
- Compiler and LISP Interpreter 1190 1045
- TIC LISP - "LISP-Machine" - like, all
- RAM, classes,urtle graphics, 807.
- compiler. CPM-86, MSDOS $235
- WALTZ LISP - "FRANZ LISP" - like, big
- nums, debug. CPM-80, MSDOS $149
Others: C-terp ($439), IQ LISP ($155), Trans-Lisp/PC ($75),
- BYSO ($125), MultiLISP-86 ($199)

AI - PROLOG
- ARITY PROLOG - full, debug, ASM, C, virtual
- Compiler $1950 MSDOS $495
- MPROLOG - Rich syntax, editor, segment
- Prolog-86 - Learn Fast. Standard,
- Sources, portable. PC DOS $275
- Prolog-86 - Learn Fast. Standard,
- tutorial, samples MSDOS Call
- Others: Prolog-1 ($359), Prolog-2 ($1895),
- MicroProlog ($229), Prof. MicroProlog
- ($359).

Editors for Programming
- BRIEF Programmer's Editor - undo, windows, reconfigure PC DOS Call
- C Screen with source 80/86 $ 75
- EMACS by UnixPress - powerful,
- multi-file, windows, DOS, MultiLISP-programs.
- Source: $495 $299
- Entry System for C-Bellesoft PC DOS $325
- FirsTime by Spruce - Improve
- productivity. Syntax directed for Turbo
- ($69), Pascal ($229), or C ($239)
- EMATE - well liked 80/86 $159
- VEDITE - multi-tasking PC DOS $ 95

Feature
- Paragon PASCAL - for performance;
- extensions like "packages"; "iterators";
- 5 memory models; 64 bit 8087; strings.
- Space vs. speed optimization options. MSDOS $895

SERVICE: FREE NEWSLETTER
Software development and AI on micros: trends, forecasts, controversies, innovations, and techniques. Plus announcement of 80 NEW tools. CALL for "Newsletter Packet".

C Language - Compilers
- BOS C - solid value, fast CPM80 $125
- 86 by CI - 8087, reliable MSDOS Call
- Lattice C - from Lifeboat MSDOS $289
- Lattice C - from Lattice MSDOS $339
- Consulair Mac C/toolbar MAC $299
- Magnax - tight, full
- Microsoft C 3.0 - new, tight. MSDOS $259
- Q/C 88 by CodeWorks - Compiler source,
- decent code, cross/native MSDOS $295
- Williams - source debug. MSDOS $399
- Wizard C - Lattice C compatible,
- full Sys. III, lint, fast. MSDOS $399

C Language - Interpreters
- C-terp by Gimpel - full K & R, OBJ
- and CPM, large progs. MSDOS $249
- INSTANT C - Source debug, Editto
- Run 3-seconds MSDOS $399
- INTRODUCING C - Interactive C
- learn fast, tutorial PC DOS $115
- RUN/C - improved MSDOS $109

C Libraries - General
- Blists C Tools 1 ($109), C Tools 2 $89
- C Food by Lattice - ask for source $119
- C/LIB by Vance $125
- C Utilities by Essential - 300 + $149
- Greenleaf Functions - portable & ASM $149
- Polyfort - for Lattice, ASM source $ 99
- Software Horizons - Pack 1 $ 129

C Libraries - Applications
- COMMUNICATIONS: Asynch Mgr $175
- Greenleaf - full, respected $149
- Software Horizons - Pack 3 $139
- FILES: Btrieve - multi-language $199
- C Index by Trio - full B + Tree,
- vary length field, multi compiler
- /File is object only $ 89
- /Pro is partial source $179
- /Plus is full source $349
- C Tree by Faircom - source, port. $349
- dbISAM by Lattice for db2 or 3 $219
- dbVISTA - full indexing, plus optional
- record types, pointers, Networks
- Object only - MS C, LAT, C86 $179
- Source - Single user $459
- Source - Multiuser $929

Ask about Atari ST, Amiga
Note: All prices subject to change without notice.
Mention this ad. Some prices are specials.
Ask about COD and PO'S. All formats available.

Call for a catalog, literature, and solid value
800-421-8006
THE PROGRAMMER'S SHOP™
128-M Rockland Street, Hanover, MA 02339
Mass: 800-422-8070 or 617-826-7531 1085

C Support - Systems
- C Debug - Source debuggers - by
- CompleteSoft ($269), MSD ($149).
- C Sharp - well supported, Source,
- realtime, tasks MS DOS $600
- C Sprite Debugger by Lattice $149
- C ToolSet - DIFF, xref, source $135
- PC Lint - full C program checking
- and small model. All C's $119

Low Cost Languages
- ECO C/88 by Ecosoft $ 50
- Introducing C - Step by step training $109
- Trans-Lisp-PC - "Common Lisp", tutorial,
- graphics, 230 functions, samples $ 75
- Modula 2 by TIC - Windows, $80
- Prolog-86 - enhanced, DOS, Editor
- Call Quick BASIC by Microsoft - Compile
- BASIC, Link $ 79
- Snobol4 + by Catspaw - Strings $ 85
- Turbo Edit/Assembler $ 85

TURBO PASCAL and SUPPORT
- BORLAND: Turbo 3.0 $ 49
- 3.0 with 8087 or BCD $ 79
- 3.0 with 8087 and BCD $ 85
- Turbo Graphix - graphs, windows $ 85
- Turbo Toolbox or Editor $ 55
- Turbo Tutor $ 29
- TURBO . . . Asynch by Blassie, full
- MetaWindow by Metagraphics $ 49
- Power Tools by Blassie - library
- Power Utilities - profiler, pp $ 89
- Professional - interrupts, macros. $ 50
- OTHERS: FirsTime ($99), Screen
- Sculptor ($99), Pascal Pac ($100),
- Type ($45), Multi Halo ($95).

Fortran & Supporting
- Fortilis - by Alpha - graph, comm. $ 59
- MACFortran by Microsoft - full text $73
- PolyFortran - xref, pp, screen $149
- Prospero - 66, reentrant $390
- RM Fortran - enhanced "IBM Fm" $429
- Scientific Subroutines - Matrix $149
- Strings and Things - registers, shell $ 59

MultiLanguage Support
- Advanced Trace 86 Symbolic, rewrite
- Assembler $149
- Btrieve/N ($469), single user $199
- Codesifter - executive profiler $109
- LMK Make by Lattice $159
- MultiHalo - full $199
- Panel - Screens, windows $239
- Periscope II Symbolic debugger $129
- PFinish - Profile by line, routine $299
- PLink-86 - 32 levels, overlays $285
- PolyLibrarian - Manage OBJS $ 89
- TexSys - Source code control $ 89
Part II
Run CP/M-80 on Your 8086/8088 Computer

In Part I (Nov/Dec '85 M/SJ) of this two part series, I discussed the advantages of the first two of NEC’s V-series microprocessors over the conventional 8086 and 8088 microprocessors. I will now take a closer look at the ability of V-series processors to run 8080, as well as 8086 code.

HOW IT’S DONE

Intel’s 8080 microprocessor, being a relatively early arrival, had a fairly simple internal structure which kept the transistor count down to a manageable level (the 8080 was actually a 3-chip set). Thus, except for special purpose registers, the instruction set of the 8080 is reasonably straightforward.

The next significant microprocessor improvement from Intel was the 8086/88 family. Being a direct descendant of the successful 8080, the architecture of this new processor family has an inescapable similarity to that of the 8080.

Thus, when NEC decided to add a second instruction set to their new V-series, the 8080’s met all requirements: 1) a large user base, 2) relative simplicity, and 3) architectural similarity to the 8086. (Although the Motorola 68000 might have been an even more attractive alternative from the standpoint of user base, it was probably impossible to implement because it did not fulfill the latter two requirements.)

But how would you design a chip to execute two different instruction sets? As mentioned in Part I, most modern microprocessors rely on an internal program, known as the microcode, to interpret the actual machine instructions. If your microcode instruction area is large enough, you could include two separate microcode programs each designed to carry out separate, independent, machine-instruction sets on the same underlying hardware. This is apparently what the designers of the V20 and V30 microprocessors did. (Without actually adding an instruction set, designers at IBM pulled a similar trick when they rewrote the microcode of the 68000 processor in the PC XT370 to have it execute IBM 370 instructions.)

We will follow NEC’s convention of calling the “8086/88-like” mode, native mode, and “8080-like” mode, the emulation mode. The name, emulation, is not meant to imply that the chip is running some emulation program. Emulation programs which run the 8080 instruction (and the Z80 superset) on MS-DOS machines have been around for some time, but all of them suffer from a lack of performance. In the V-series’ emulation mode, a 5-MHz 8088 has roughly the throughput of a 6- to 7-MHz 8080. Apparently, there was not enough microcode space to implement the Z80 superset of instructions. When the V20 or V30 encountered these instructions, the results are unpredictable.

The chip keeps track of which mode it is in, by using two previously unused bits of the flag register. One of these flags, the emulation flag, is set to 1 whenever the chip is in native mode and cleared, when in emulation mode. This maximizes compatibility with the 8086, in which this flag would normally be a permanent.

Many of the internal registers of the V20 and V30 serve dual roles, being available both from native mode and emulation mode. This serves to reduce the overall number of hardware registers required by the chip, as well as facilitate communication between the two modes. The register correspondence between native and emulation mode is outlined in Table 1. Although the segment registers cannot be modified from emulation mode, the code, data, and stack segments are used just as they are in native mode; all opcode fetches are through the code segment, all data fetches through the data segment and all pushes and pops through the stack segment. Because of this, 8080 programs running under the V-series processor actually have a 192K, albeit segmented, address space.

EXECUTING 8080 CODE ON A V-SERIES CHIP

Several instructions have been added to the native and emulation mode instruction sets to facilitate switching from one mode to the other (see Table 2).

The instruction INTEM (INTerrupt into EMulation Mode) operates like an INT (INTerrupt) instruction, except that after pushing the status register onto the stack and before vectoring to the interrupt code, it clears the emulation bit, putting the chip in emulation mode. The emulation equivalent of the native mode IRET (Interrupt RETurn), RETEM (RETurn from EMulation) causes the status register, with the emulation bit set, to be popped back off the stack restoring the chip to native mode. This instruction also pops the return address back to the native-mode code as an IRET would.

While in emulation mode, however, calls can be made to native code through use of the CALLN (CALL Native mode) instruction. CALLN is the emulation-mode equivalent of INTEM, except that it sets, rather than clears the emulation flag. The called, native-mode program can return to
the emulation program by executing a normal IRET, which pops the status flag, restoring the emulation flag to its previous value.

When the V-series processor receives an external interrupt, it automatically reverts to native mode, as soon as it has pushed the status word onto the stack. Thus, interrupt routines must always be written in native code. When the interrupt handler returns from the interrupt (with the IRET instruction), the status word is popped off the stack, returning the chip to whatever mode it was in when the interrupt occurred.

**RUNNING CP/M-80 ON THE V-SERIES CHIPS**

The question does remain as to why you would want to run a different instruction set on the V-series processor. If you were a systems developer who had invested many hours in developing software on the 8080 for some particular application, you might welcome the opportunity to upgrade your product to run on a faster chip with a larger address space and an expanded instruction set. However, in all fairness, this group of programmers is probably not very large.

But if CP/M-80 programs could be made to run under MS-DOS, we might interest those CP/M-80 owners who have hesitated to buy an MS-DOS machine, since it would not be able to run their software. Likewise, MS-DOS owners who would like access to as yet untranslated CP/M-80 programs, might also be interested.

CP/M-80 is a particularly good candidate for emulation. Since MS-DOS was patterned after CP/M in the beginning, there is an almost one-to-one correspondence between the first 20 MS-DOS system calls and the CP/M BDOS calls (see Table 3). For a more detailed discussion of this topic see "Program Interfacing to MS-DOS, Part III" (M/SJ Sep/Oct ’85). This similarity of system calls means that a CP/M emulator program would not have to reimplement the CP/M BDOS calls.

While it is true that virtually all CP/M machines today use Z80’s as their processors, rather than 8080’s, most CP/M programs limit themselves to the 8080 instruction subset so as to retain compatibility with the largest number of users. Thus the absence of the Z80’s capabilities is not such a big disadvantage when running CP/M on the V-series processors.

What is needed, is a CP/M-80 emulator that runs CP/M programs in emulation mode and presents a CP/M BDOS/BIOS interface. One such package is the EMULATOR sold by GFI Electronics. To illustrate the principles involved, GFI has made available much of the source code for this product. Heavily commented excerpts from that source code appear in Listing 1 and are explained below.

Although the principles of CP/M emulators can be relatively simple, many tricks are necessary to make programs run which don’t follow the rules of CP/M. Thus GFI’s EMULATOR package is well worth the $49.95 purchase price (GFI Electronics, 1800 Avalon St., Olathe, KS 66062, 913-829-0157). My experience with EMULATOR shows it is capable of running most CP/M programs.

To use this package, the CP/M program must be converted to MS-DOS format. Then the user runs a utility which appends a prefix program onto the beginning of the CP/M-80 .COM file, generating a new .COM file only slightly larger than the original. When the resulting program is executed under MS-DOS, this prefix program is loaded along with the original CP/M pro-

---

**TABLE 1**

<table>
<thead>
<tr>
<th>Equivalent Emulation &amp; Native Mode Registers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Emulation Mode (8080)</td>
</tr>
<tr>
<td>-----------------------</td>
</tr>
<tr>
<td>A</td>
</tr>
<tr>
<td>B</td>
</tr>
<tr>
<td>C</td>
</tr>
<tr>
<td>D</td>
</tr>
<tr>
<td>E</td>
</tr>
<tr>
<td>F</td>
</tr>
<tr>
<td>G</td>
</tr>
<tr>
<td>H</td>
</tr>
<tr>
<td>L</td>
</tr>
<tr>
<td>N</td>
</tr>
<tr>
<td>S</td>
</tr>
<tr>
<td>P</td>
</tr>
<tr>
<td>C</td>
</tr>
<tr>
<td>SP</td>
</tr>
<tr>
<td>PC</td>
</tr>
<tr>
<td>Flags:</td>
</tr>
<tr>
<td>C</td>
</tr>
<tr>
<td>Z</td>
</tr>
<tr>
<td>S</td>
</tr>
<tr>
<td>P</td>
</tr>
<tr>
<td>SP</td>
</tr>
<tr>
<td>PC</td>
</tr>
</tbody>
</table>

Native mode registers not accessible from emulation mode: SI,DI,SP,ES,DS,SS,CS

**TABLE 2**

<table>
<thead>
<tr>
<th>V-series Emulation Instructions</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTM n</td>
</tr>
<tr>
<td>SETEM</td>
</tr>
<tr>
<td>CALLN n</td>
</tr>
</tbody>
</table>

**TABLE 3**

<table>
<thead>
<tr>
<th>Equivalent CP/M BDOS &amp; MS/DOS or PC/DOS System Calls</th>
</tr>
</thead>
<tbody>
<tr>
<td>Call</td>
</tr>
<tr>
<td>-----------------</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>7</td>
</tr>
<tr>
<td>8</td>
</tr>
<tr>
<td>9</td>
</tr>
<tr>
<td>A</td>
</tr>
<tr>
<td>B</td>
</tr>
<tr>
<td>C</td>
</tr>
<tr>
<td>D</td>
</tr>
<tr>
<td>E</td>
</tr>
<tr>
<td>F</td>
</tr>
<tr>
<td>10</td>
</tr>
<tr>
<td>11</td>
</tr>
<tr>
<td>12</td>
</tr>
<tr>
<td>13</td>
</tr>
<tr>
<td>14</td>
</tr>
<tr>
<td>15</td>
</tr>
<tr>
<td>16</td>
</tr>
<tr>
<td>17</td>
</tr>
<tr>
<td>18</td>
</tr>
<tr>
<td>19</td>
</tr>
<tr>
<td>1A</td>
</tr>
<tr>
<td>1B</td>
</tr>
<tr>
<td>1C</td>
</tr>
<tr>
<td>1D</td>
</tr>
<tr>
<td>1E</td>
</tr>
<tr>
<td>1F</td>
</tr>
<tr>
<td>20</td>
</tr>
<tr>
<td>21</td>
</tr>
<tr>
<td>22</td>
</tr>
<tr>
<td>23</td>
</tr>
<tr>
<td>24</td>
</tr>
<tr>
<td>25</td>
</tr>
<tr>
<td>26</td>
</tr>
<tr>
<td>27</td>
</tr>
</tbody>
</table>

**TABLE 3**

<table>
<thead>
<tr>
<th>Equivalent CP/M BDOS &amp; MS/DOS or PC/DOS System Calls</th>
</tr>
</thead>
<tbody>
<tr>
<td>Call</td>
</tr>
<tr>
<td>-----------------</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>7</td>
</tr>
<tr>
<td>8</td>
</tr>
<tr>
<td>9</td>
</tr>
<tr>
<td>A</td>
</tr>
<tr>
<td>B</td>
</tr>
<tr>
<td>C</td>
</tr>
<tr>
<td>D</td>
</tr>
<tr>
<td>E</td>
</tr>
<tr>
<td>F</td>
</tr>
<tr>
<td>10</td>
</tr>
<tr>
<td>11</td>
</tr>
<tr>
<td>12</td>
</tr>
<tr>
<td>13</td>
</tr>
<tr>
<td>14</td>
</tr>
<tr>
<td>15</td>
</tr>
<tr>
<td>16</td>
</tr>
<tr>
<td>17</td>
</tr>
<tr>
<td>18</td>
</tr>
<tr>
<td>19</td>
</tr>
<tr>
<td>1A</td>
</tr>
<tr>
<td>1B</td>
</tr>
<tr>
<td>1C</td>
</tr>
<tr>
<td>1D</td>
</tr>
<tr>
<td>1E</td>
</tr>
<tr>
<td>1F</td>
</tr>
<tr>
<td>20</td>
</tr>
<tr>
<td>21</td>
</tr>
<tr>
<td>22</td>
</tr>
<tr>
<td>23</td>
</tr>
<tr>
<td>24</td>
</tr>
<tr>
<td>25</td>
</tr>
<tr>
<td>26</td>
</tr>
<tr>
<td>27</td>
</tr>
</tbody>
</table>
gram. It then simulates the CP/M-80 environment to the 8080 code underneath.

Once loaded, execution of the program begins at the label 'Start:' in the listing. The first action is a check to make sure that the processor is capable of 8080 emulation. If not, an error message is generated and the program ends. We must now calculate the address of the 8080 code. The 8080 program was loaded along with the emulation prefix by the MS-DOS program loader. Since we know how big the prefix is, we need merely add the current code segment, held in the CS register, to the prefix size in paragraphs to find the segment address of the 8080 code.

Next, we generate the 8080 program's Program Segment Prefix (PSP). Here again our job is fairly simple. Since MS-DOS' PSP is almost identical to CP/M's, we copy our PSP into the first 100 bytes of the CP/M program's segment. Next we copy a jump table into the very high end of the CP/M program's segment (performed at label 'move_bios'). We initialize interrupt 60 to point to the beginning of the CP/M program, 61 to point to the BDOS call handler, and 62 through 69 to point to various BIOS call handlers.

Lastly, we pass control to the CP/M program with the INTEM EM_ VECT instruction at 'start_cpm'. This performs an emulation interrupt to the CP/M code address which was just loaded into interrupt 60. Remember that we calculated the segment address of the CP/M program above by knowing both the prefix's code and size. Also we know that the offset of the CP/M program must be 100 hex. To insure maximum compatibility, we set the data segment register to the CP/M segment.

Now examine the BIOS jump table appearing at label 'cpm'. These jumps appear just as they would in a CP/M environment, except that they each to code which does nothing more than call one of the prefix code's entry points. This is achieved by executing a CALLN through one of the emulation interrupts. Making the code execute these BIOS calls in the native execution area has two advantages: 1) it removes the code from the 8080 address space, thus freeing up a Transient Program Area (TPA) of greater than 63K and, 2) the calls can be coded with the more powerful instruction set of the V20 or V30.

The code necessary to perform the BIOS functions has been left out. Generally, every CP/M BIOS call can be executed by setting up the registers properly and making the corresponding ROM BIOS call. The routine which handles BIOS calls is educational (it starts at label 'bdos'). This routine looks up the BDOS call number contained in the CL register in the jump table 'bdosfun'. All of those CP/M BDOS functions which have a corresponding MS-DOS function call with the same number (most of them do) vector to the routine 'bdos'. This routine does nothing more than perform the MS-DOS call and pass the results back to the CP/M caller.

One of the BDOS calls which has no corresponding MS-DOS call has been included as an example. Since MS-DOS does not have a CP/M version number, the routine 'version' handles the 'return version number' system call by maintaining that it is version 2.2. Similarly, most other BDOS calls which have no MS-DOS equivalent can be responded to with some default response and otherwise ignored. The routine 'nogood', handles BDOS calls which are impossible or have no MS-DOS equivalent and can't be overlooked by putting a message on the screen or exiting to MS-DOS with an error level set.

Notice that since the V-series processor always processes hardware interrupts in native mode, all previously installed interrupt borrowers (such as Sidekick, screen blankers, keyboard enhancers, etc.) continue to work properly, even when invoked while running 8080 programs. Similarly, since all system calls are handled in native mode, CP/M programs should run under such environment managers as Windows, GEM, etc.

**CONCLUSION**

Through this two-part detailed examination, we see that the V-series processors can introduce significant improvements to an 8088- or 8086-based system. Although, they will not replace a turbo card, they do improve performance somewhat. They introduce several extensions to the 8086 family's instruction set, including the entire instruction set of the 8080. With the proper driver software, the V-series microprocessor will execute most CP/M programs at full speed (or better) on MS-DOS machines. All of these improvements are available to an MS-DOS system for around 1% of the system's original cost.

Stephen Randy Davis is a senior systems programmer at a defense contractor in Greenville TX, where he programs various microprocessors. He is working on his masters in physics after having graduated cum laude from Rice University in Houston.

---

**PROGRAM LISTING**

```plaintext
; This segment of a CP/M emulator, written by Jay Spronkle, is 
; reproduced with permission of GFI Electronics. The GFI 
; package is designed to be installed onto beginning of a CP/M 
; .COM file, resulting in slightly larger .COM file which 
; executes on a MS-DOS machine equipped with a V20 or V30. This 
; example is large enough to demonstrate principles of a CP/M 
; emulator for V-Series microprocessors, but for obvious 
; reasons entire package could not be reproduced here. 
; .186 include 'extrn.inc' ; we want extensions... ; ...and unique instructions 
;
; Make sure there is a V20 installed...
;
; org 100H ; this program MUST be built into ; .COM file; if it were .EXE file 
; ; structure would be all wrong for ; loading by MS-DOS 
;
; start: ; n0t nec stop now 
jmp - v20 ;
;
; 8086 is installed, print error and terminate. 
;
; stop_now: ; doncall 9,m8088 ; tell them I can't run on 8088 
; doncall 4CH,2 ; exit with error 
m8088: db 'Cannot run program, V20 not installed!' 
;
; Calculate segment address of CP/M 80-code -- this is 
; done by adding offset of CP/M code from beginning 
; of emulation program to code segment of program. 
; (See article for further explanation) 
;
v20: mov ax,beg of cmap ; get CP/M-80 offset code 
shr ax,4 - - ; convert to segment offset 
mov bx,cs 
add bx,ax 
mov beg of cmap,ax ; store this back 
mov es,ax ; set up es from here on out 
```

Micro/Systems Journal January/February 1986
Create CP/M PSP —

PSP in MS-DOS is almost identical to CP/M PSP, so make copy

of our PSP into CP/M PSP area (main differences, contents of
address 0 & address 5, will have to be set up specifically)

mov di,0
mov si,0
mov cx,100H
rep movs byte

Construct BIOS/BDOS jump table at very top end of 8080
address space. By placing these jump tables this high, we
get TPA of 64K!

mov si,offset header:bseg get jump table address
mov di,biosadr get stuff to move address
move_bios:

mov al,cs:[si] get a byte
inc si increment source pointer
mov es:[di],al write it to CP/M space
inc di increment destination pointer
jnz move_bios go until all copied

place jumps at location 0 and 5

These 2 jumps are used by CP/M-80 program to access CP/M-80
operating-system support routines.

place_jumps:

mov al,0CH ;this is an 8080 jump
mov es:[0],al set 0
mov es:[5],al set 5
mov ax,biosadr ;jump to BIOS emulator address
mov es:[6],ax ;get address of jump
add ax,9 skip to jump for warm start
mov es:[1],ax ;set address of jump

Set up BIOS vectors and BDOS vector —

interrupt 60 points to the CP/M 80 code

es:62-69 BIOS handlers

EM_VECT eqs 60H ;Emulsion int
emulate:

mov ax,beg of cpn get segment address of CP/M
mov ax,100H ;CP/M starting address always 100H
doscalls 25H,em vect set vector to enter CP/M program

([MS-DOS call 25 sets address
of specified interrupt; in this
case, set address of interrupt
60H to beg_of_cpn100H])

mov ax,cs get code segment
mov ds,ax set ds back to us
doscalls 25H,biad,em vect+1 ;BDOS entry-point vector
doscalls 25H,biad,em vect+2 ;invalid BDOS-call vector
doscalls 25H,biad,em vect+3 ;critical BDOS vector
doscalls 25H,biad,em vect+4 ;console-status BIOS vector
doscalls 25H,biad,em vect+5 ;console-input BIOS vector
doscalls 25H,biad,em vect+6 ;console-output BIOS vector
doscalls 25H,biad,em vect+7 ;aux-input BDOS vector
doscalls 25H,biad,em vect+8 ;aux-input BIOS vector
doscalls 25H,biad,em vect+9 ;status-BIOS vector

Now start CP/M-80 program—

note we will start his stack pointer (our BP) at 80, his
code & data segment to be CP/M program we calculated all
along

mov bp,80H ;CP/M stack in BP
mov ax,ds get CP/M program segment
mov ds,ax ;set ds for data ops in 8080 mode
int EM_VECT goto 8080 mode
doscalls 4CH,9 ;we will return here on cold or
warm start, so return to MS-DOS

Jump table to BDOS function handlers

bdosfun dw 0 ;system reset, return to MS-DOS
dw bdosag ;ioind console input
dw bdosag ;console output
dw bdosag ;aux-input
 dw bdosag ;aux-output
 dw bdosag ;direct console I/O
 dw auxost ;***aux input status
 dw auxost ;***aux output status
 dw bdosag ;print string
 dw bdosag ;read console buffer
 dw bdosag ;get console status
 dw version ;***return version number
 dw retdisk ;reset disk system
 dw select ;select disk
 dw BDOS ;open file
 dw BDOS ;close file
 dw BDOS ;search first
 dw BDOS ;search next
 dw BDOS ;delete file
 dw BDOS ;read sequential
 dw BDOS ;write sequential
 dw BDOS ;make file
 dw BDOS ;rename file
 dw BDOS ;**return login vector
 dw bdosag ;return current disk
 dw bdosag ;set dma address
 dw dalloc ;***get addr(allo)
dw bfrptr ;write protect disk
dw bdosag ;get vec
 dw nogood ;***set r/o vector
 dw nogood ;***set file attributes
 dw nogood ;***get addr of dso
 dw exit ;***set/set user code
 dw bdosag ;read random
 dw bdosag ;write random
 dw bdosag ;compute file size
 dw bdosag ;set random record

msg db 'Attempted unsupported BDOS call'
cr1f db 13,10,'","'

The CP/M BDOS emulator

BDOS emulator

bdos:
cmp cl,24H ;see if valid BDOS call
ja bbdosit ;if valid call, jump through table
lret ;not valid, exit

Bdosit:
mov al,cl ;set function number
mov dh,0 ;make 16-bit word
add ax,ax ;double value for word pointer
mov si,ax ;set pointer to table entry
jmp word ptr cs:bdosfun[si] ;jump through table entry

far away most BDOS calls come here, where equivalent
MS-DOS 1.1 call is performed

bdosag:
mov dh,cl ;move function number to dh for DOS
jnt 2IH ;pass through to DOS
lret

some BDOS calls require special handling; here is a sample:

Tell anyone that asks we are version 2.2 of CP/M
version:

mov bx,0022H ;tell them we are CP/M 2.2
lret
return to caller

we just can't handle some BDOS & BIOS calls; better trap out
with an error than to crash

bad BDOS call
	nogood:
doscalls 9,msgb ;print message
doscalls 4CH,2 ;error running
NEW PRODUCTS

SERIAL ADAPTER FOR THE VDB-A2-
- Plugs into keyboard socket
- Direct connect to Keytronic (IBM) serial ASCII keyboard
- Internal speaker for BEEP
- Tx port for non VDB-A2 use
- Hardware programmed Rx & Tx
- Dual baud-rate generator
- Bare-board OR assembled

DOUBLE-HEIGHT S-100 PROTOTYPING-
- Oversized heatsinks for HI-CURRENT reg's: +5, ±12 V.
- Use with wire-wrap sockets OR direct solder connections
- GOLD plated edge connector
- NO PLATING CUTS required
- NO wasted address decoding

WATCH OUR ADS FOR OTHER PRODUCTS COMING SOON

- MODEM ADAPTER FOR THE BIO-1
- 256K RAM SOFTWARE FOR BIO-1
- HIGH FEATURE DISC CONTROLLER
- Z80 C.P.U. / MEMORY MANAGER

1 MEGABYTE DRAM BOARD
Specifications subject to change without notice

EMS Educational Microcomputer Systems
P.O. Box 16115
Irvine, CA 92713
(714) 854-8545

Micro/Systems Journal January/February 1986
The Best C Book
A Powerful C Compiler
One Great C Value $39.95

A good C book just isn’t complete without a good C compiler to go with it. That’s why we give you both. You get a comprehensive 450 page book and a full feature standard K&R C compiler with the Unix V7 Extensions. The book is loaded with examples that teach you how to program in C. And our fast one pass C compiler comes with an equally fast linker so you don’t waste a lot of time watching your disk drives spin. You also get a Unix compatible function library that contains more than 150 functions (C source code included). And if all that isn’t enough, we offer you a 30 day money back guarantee. So what are you waiting for? The exciting world of C is just one free phone call away.

Language Features
- Data Types: char, short, int, unsigned, long, float, double
- Data Classes: auto, extern, static, register
- Type def, Struct, Union, Bit Fields, Enumerations
- Structure Assignment, Passing/Returning
- Structures

Functions

When you’re programming in a high level language you need a high powered editor. That’s why we created a programmable full/split screen text processor. It lets you split the screen horizontally or vertically and edit two files at once. You can move text back and forth between two windows. You can also create your own macro commands from an assortment of over 100 predefined commands. The editor comes configured so that it works just like Wordstar but you can change it if you prefer a different keyboard layout. The editor is a great companion to our C compiler. Because they work so well together we want you to have both. To make sure you do, we’re offering the editor for just $15 when purchased with the C compiler.

MIX Editor
$29.95

When you’re programming in a high level language you need a high powered editor. That’s why we created a programmable full/split screen text processor. It lets you split the screen horizontally or vertically and edit two files at once. You can move text back and forth between two windows. You can also create your own macro commands from an assortment of over 100 predefined commands. The editor comes configured so that it works just like Wordstar but you can change it if you prefer a different keyboard layout. The editor is a great companion to our C compiler. Because they work so well together we want you to have both. To make sure you do, we’re offering the editor for just $15 when purchased with the C compiler.

ASM Utility
$10

The ASM utility disk allows you to link object files created by Microsoft’s MASM or M80 assemblers. Lots of useful assembly language functions are included as examples.

NOT COPY PROTECTED

| Editor | $29.95 |
| C | $39.95 |
| C & Editor | $54.95 |
| ASM Utility | $10.00 |
| TX Residents | $29.95 (61.25% sales tax) |
| Shipping | $10.00 (see below) |

Check Money Order MC/VISA* Exp.

Shipping Charges: (No charge for ASM Utility)

USA $5.00/Order
Canada $10/Order
Overseas $10/Editor $30/C $50/C & Editor

PCDOS/MSDOS (2.0 or later)
IBM PC Single Side
IBM PC Double Side
Tandy 2000
8 Inch
Other
CPM 80 (2.2 or later)
8 Inch
Kaypro II
Kaypro 4
Apple (780)
Osborne I SD
Osborne I DD
Morrow MD II
Other

Orders Only
1-800-523-9520
In Texas 1-800-622-4070

Canadian Distributor
Saraguay Software: 416-923-1500

2116 E. Arapaho Suite 363
Richardson, TX 75081
(214) 783-6001

Ask about our volume discounts.

Unix is a trademark of Bell Labs. CPM is a trademark of Digital Research. MSDOS is a trademark of Microsoft. PCDOS is a trademark of IBM. WORDSTAR is a trademark of Micro Pro.
This installment discusses the remaining functions not covered previously. These include useful and unique operations such as acquiring the MS-DOS version number and two small groups of functions related to memory allocation and program execution.

**DISK ACCESS CONTROL**

Read access to disk files includes error detection to verify the validity of the data being read from the disk. This is normally done using the CRC (error detection information) associated with each sector on a disk.

Unfortunately, normal write access to a disk file only places the CRC information on the disk. An error may occur while writing to the disk but this would not be detected until the sector was read. However, this is usually too late to correct the error. Reading the information after writing verifies that the sector was properly written is the only way to assure that the correct data is written to the disk.

This mode of operation can be built into an application but usually complicates development of the application and always slows it down. MS-DOS provides an alternative, implement the read after write verification in the operating system. Two functions are provided to manipulate this operating mode. The first returns the current setting which defaults to off. The following code will get the current write verify mode.

```assembly
MOV AL, 0
INT 21H
;AL := get verify mode
```

The other function allows the verify mode to be set or reset. The following two code excerpts perform these operations.

```assembly
MOV AL, 1
INT 21H
;AL := set verify mode

MOV AL, 0
INT 21H
;AL := turn off verify mode
```

Now, why not turn verify mode on all the time? Well, it takes at least twice as long to operate in verify mode than it does in non-verify mode because each write requires an accompanying read operation. To top it off, the delay time between a write and the verifying read operation is one disk revolution so the average sector access time changes from half a disk revolution to one and a half disk revolutions. This could bring many applications to a grinding halt or at least slow them down appreciably.

Therefore, there is a tradeoff to be made. The verify mode provides a good deal of valid data assurance at a cost in time. This tradeoff must be balanced with operating time for the application. The slowdown factor will be a function of the number of write operations performed.

In general, disks tend to be very reliable and the verify mode is normally used in only very harsh environments where more errors tend to occur or in environments which have very valuable data that must be accurate.

The advantage of placing the verify operation in the operating system is that the mode can be set independent of the application programs. Thus, you can run your accounting software with the verify mode on and a word processor with the verify mode off.

**GENERAL STATUS**

Now for a number of miscellaneous functions. Knowing the version of the current operating system is valuable since different options are available on each implementation. The following shows how to get the current MS-DOS version number.

```assembly
MOV AH, 3BH
;AH := get version number
INT 21H
;AH, AL := version number
```

Version 2.1 of MS-DOS returns AL set to 2 and AH set to 1.

The control-break option is available on some systems. It is normally a special key, or combination of keys, on the keyboard and allows the user to terminate a program at any time. This option can be disabled by an application or the control-break interrupt vector can be changed by the application as discussed in an earlier article.

The current status of the control-break option is obtained using the following code:

```assembly
MOV AL, 0
;AL := get mode
MOV AH, 33H
;AH := control-break option
INT 21H
;DL := current option
```

Setting the mode on or off is done using the following code:

```assembly
MOV DL, 1
;DL := get option (0 = reset)
MOV AL, 1
;AL := set mode
MOV AH, 33H
;AH := control-break option
INT 21H
;set control-break mode
```

The next function is used to get the country dependent information. This information would be used by applications which display or read information based upon time, currency or large numbers. The function returns this information in a parameter block located in the application referenced by the DS:DX registers. The contents of the data block are described in the following table:

<table>
<thead>
<tr>
<th>Offset</th>
<th>Bytes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2</td>
<td>date/time format</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>currency symbol</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>thousands separator</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>currency symbol</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>decimal separator</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>thousand separator</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>currency symbol</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>always 0</td>
</tr>
<tr>
<td>8</td>
<td>27</td>
<td>reserved</td>
</tr>
</tbody>
</table>

Note, the zero fields allow the various character parameters to be referred to as zero terminated strings. The following code will load a parameter block in the current data segment.

```assembly
MOV DX, OFFSET PARAM
;DS:DX := PARAM addr
MOV AL, 0
;AL := get data
MOV AH, 3BH
;AH := control-break option
INT 21H
;get country depend- ;ent info
```

**DEVICE CONTROL**

Disk and device files can be accessed using the functions described in the previous articles. This access is to the data within a file. MS-DOS provides an additional function to access control information associated with a file or a device. The I/O control function allows an application to determine the type and status of a file. The MS-DOS function code (44 hex) is placed in the AH register while the AL register contains an additional opcode described in the following table. The file handle is passed in the BX register. The handle must have been opened or created using the MS-DOS functions 3C hex (create) and 3D hex (open).
Opcodes 0, 6 & 7 valid for disk files

Opcode Description
0* DX := device info (see below)
1 Set device info (DX = new value)
2 Read CX bytes into DS:DX from control channel
3 Write CX bytes from DS:DX to control channel
4 Read CX bytes into DS:DX from drive (sector 0, default, 1 = A; etc.)
5 Write CX bytes from DS:DX to drive (GL)
6* AL := input status
7* AL := output status

*Opcodes 0, 6 & 7 valid for disk files
Opcode 6 function returns 0 if end of file
Opcode 7 function always returns OFF hex

Device Information

<table>
<thead>
<tr>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>if device is console input</td>
</tr>
<tr>
<td>1</td>
<td>if device is console output</td>
</tr>
<tr>
<td>2</td>
<td>if device is NULL device</td>
</tr>
<tr>
<td>3</td>
<td>if device is clock device</td>
</tr>
<tr>
<td>4</td>
<td>Reserved (must be zero)</td>
</tr>
<tr>
<td>5</td>
<td>if binary mode</td>
</tr>
<tr>
<td>6</td>
<td>0 if end of file character is</td>
</tr>
<tr>
<td>7</td>
<td>End of file indicator (0 = end of file)</td>
</tr>
<tr>
<td>8-13</td>
<td>Reserved (must be zero)</td>
</tr>
<tr>
<td>14</td>
<td>if device can process control strings</td>
</tr>
<tr>
<td>15</td>
<td>Reserved (must be zero)</td>
</tr>
</tbody>
</table>

*Only bits 6 & 7 are used if disk file & bits 0-5 are block device containing file

This MS-DOS function operates differently based upon the type of file which can be obtained using opcode 0. Disk files support only a limited number of opcodes. The primary use of this function with disk files is to determine when the end of the file is reached or to check which drive contains the file.

This function allows the type of a device to be determined. Applications can therefore determine how data is to be read or written to a file. For example, console output may be formatted for 80 character lines while printer output may be formatted for 132 character lines.

Control string operations are device driver dependent and are usually not supported for disk devices. One use could be with communication device drivers where parameters such as baud rate to be modified or examined. This would be much cleaner than going directly to serial I/O ports.

MEMORY ALLOCATION

Programs running with a simple operating system normally use all available memory. More complex programs on such systems may access different modules via overlays or by chaining. MS-DOS allows these types of operations and also allows multiple programs to be resident at one time although only one may execute at a time. In this case each program cannot utilize all available memory, otherwise there would not be enough room for additional programs.

Therefore, MS-DOS provides some basic memory management functions. MS-DOS allocates memory from a common pool in variable sized contiguous blocks whose size is an integral number of 8086 paragraphs where a paragraph is sixteen bytes. Blocks always start and end on a paragraph boundary.

An application can be allocated any number of blocks as long as there is sufficient free memory available. MS-DOS keeps track of the available memory space and where it is located. Programs always reside in an allocated block of memory. In fact, the base page of a program is located at the start of the block containing the program.

A program will initially be located in a block which covers the largest free block available at the time the program is loaded. A program should first reduce the size of this block to the minimum size required to run the program before asking MS-DOS to allocate additional space since MS-DOS will typically use any space returned to it by the program. The following code would perform this operation:

```
MOV ES,AX
MOV BX,SIZE
 INT 21H
```

where the result indications are listed in the following table:

<table>
<thead>
<tr>
<th>Result</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>operation successful</td>
</tr>
<tr>
<td>7</td>
<td>memory block destroyed</td>
</tr>
<tr>
<td>8</td>
<td>insufficient memory</td>
</tr>
<tr>
<td>9</td>
<td>invalid memory block</td>
</tr>
</tbody>
</table>

The size of the memory block for a program is normally reduced but it could be increased later. The possibility of increasing its size is based upon the memory allocation scheme and the amount of free memory above the program since data blocks cannot be moved in an 8086 DOS environment. However, applications running on a 80286 or 80386 may be movable depending upon the type of support provided in future versions of MS-DOS.

Even so, the normal mode of operation is to reduce the size of the program segment and then allocate a new block for various data structures. The amount of memory which can be allocated may exceed that freed by a program if there are non-continuous free memory blocks available.

This dynamic memory allocation and deallocation support is very useful for programs which use a large amount of memory for various functions but not at the same time. For example, sorting large arrays in memory uses large amounts of memory. If print spooling is to be performed after the arrays are no longer used then the memory used for the arrays could be used for print spooling. Letting MS-DOS keep track of the various memory blocks is much easier than doing it within the program and allows future extensions to MS-DOS on 80286 and 80386 machines to support expanded and virtual memory in a much cleaner fashion.

The function to allocate memory requires the number of paragraphs to be provided. This value is passed in the BX register and the segment is returned in the AX register. A value of 0 indicates an error and does not conflict with free memory allocation since this area is reserved for interrupt vectors and MS-DOS. An error 8 indicates insufficient memory and BX contains the size of largest free block available.

The following example shows how a small segment is allocated:

```
MOV BX,10
MOV AH,9H
INT 21H
```

Freeing an allocated memory block is as simple as:

```
MOV AX,block to free
MOV AH,8H
INT 21H
```

Errors 7 and 9 may be returned if the block has not been allocated by MS-DOS or the memory allocation system is corrupted. All allocated blocks, with the exception of the program block should be freed before a program terminates.

PROGRAM LOADING/EXECUTION

Loading overlays and other programs has always been difficult on simple operating systems. MS-DOS V1.x provided limited support but made the initiating program do most of the work. MS-DOS V2.x and later provides more sophisticated support along with the memory allocation functions mentioned earlier.

The first function is used to create a new program segment. This has been superceded by the Load/Execute function described later in this section. However, it is sometimes useful for debugging style programs which need to build up a program environment. The create program segment function requires the segment number of a previously allocated block. The segment number can be within an existing block for MS-DOS V2.x and V3.x but it is not recommended. The segment number is passed in the DX register as shown in the following example:

```
MOV DX,PROCSEG
INT 21H
```

The information in the current program segment is copied to the new program segment. The memory block size and error interrupt vectors are also placed into the program segment. The program code must be loaded by the application and the new program must be initiated by the program.

This function is actually superceded by the Load/Execute function (MS-DOS function 4b hex). The function supports two modes specified by an opcode in the AL register. One allows programs to be loaded and executed and the other provides overlay support. Each uses a parameter block referenced by the ES:BX register pair and they both require the name of the file which contains the code to be loaded in an ASCII zero
terminated string reference by the DS:DX registers. This is the same type of file name used with the UNIX-style file open and create functions. The following tables summarize the parameter block information.

Load and Execute (opcode = 0)

<table>
<thead>
<tr>
<th>Offset</th>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2</td>
<td>Environment string segment</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>Command line address</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>Addr of default PCB at 5Ch</td>
</tr>
<tr>
<td>A</td>
<td>4</td>
<td>Addr of default PCB at 6Ch</td>
</tr>
</tbody>
</table>

1. Initiating program must restore all registers including SS and SP.
2. All open files are inherited by new process.
3. Environment string segment contains a set of ASCII strings terminated by a 0 (null string). The strings are of form: parameter=value.

Load Overlay (opcode = 1)

<table>
<thead>
<tr>
<th>Offset</th>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2</td>
<td>Segment to load overlay</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>Relocation factor to be used</td>
</tr>
</tbody>
</table>

The overlay mechanism can also be used to load programs into segments initialized by the MS-DOS Create program segment function described at the start of this section. This call uses the loader portion of COMMAND.COM which is loaded at the top of memory. This function will return a non-zero error code if the loader is not available or the file cannot be accessed.

MS-DOS will allocate a memory block for the load/execute opcode but requires a previously allocated block for the overlay opcode. The memory block will automatically be released by the load/execute initiated program when it terminates unless it uses the MS-DOS function 31, terminate and stay resident.

The terminate and return result code function was mentioned before but its use becomes apparent with the load/execute function. The result code is passed in the AL register by the terminating program using MS-DOS function 4C hex. This result code can be obtained by the program which executed the load/execute MS-DOS function by executing the following code:

```assembly
MOV AH,4EH
INT 21H
AL \ = code, AH \ = mode
```

where the result code in AL is the one specified by the terminating program. The result modes are listed in the following table:

<table>
<thead>
<tr>
<th>Mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>normal termination</td>
</tr>
<tr>
<td>1</td>
<td>control-break exit</td>
</tr>
<tr>
<td>2</td>
<td>critical device error</td>
</tr>
<tr>
<td>3</td>
<td>terminate and stay resident (MS-DOS function 31)</td>
</tr>
</tbody>
</table>

**SUMMARY**

This wraps up the operations available for MS-DOS V2.x. The system is very flexible and provides an upgrade path from CP/M style programs to UNIX-style programs. The memory allocation and program support are useful but allude to more powerful system implementations.

---

**MEX 'n' Match**

**MEX-PC** — Full-featured modem software for your IBM-PC (Versions also available for Tandy 2000 and most CP/M machines). Now can be menu driven • Fully programmable function keys • Simple built-in programming language for custom applications development • Choice of 8 background/foreground colors • All popular protocols (KERMIT available Dec. 85) • Automatic & repeat dialing • Runs DOS commands and programs while online • "Clones" customized versions.

$59.95

**REO** — REmote Operation module with password security. This inexpensive add-on module adds remote access capability to your MEX program. It allows you to call up your office computer from home — or your home computer from your office — and swap data or text files, even programs. Download a spreadsheet, rework it at home, then send it back, updated, to your office. Password security keeps unauthorized persons from gaining access to sensitive files.

$29.95

**TEM** — Intelligent terminal emulation add-on module for MEX-PC. As shipped, TEM emulates an ADM-3A equivalent terminal. TEM expands MEX to emulate either a Televideo 925 or DEC VT52/VT100. TEM is full-function emulation, allowing you to use remote text processors and spreadsheet programs running on mainframes that require a specific terminal interface. Simple installation. Fully documented.

$29.95

**NightOwl Connection** — A multi-user on-line service that provides you with round-the-clock access to MEX support, plus all the latest and best in public domain and user-supported software. The Connection features 36 megabytes of free software — some of it exclusive to Connection subscribers — including 18 megabytes worth of programs for the IBM-PC and compatibles. MEX-PC owners can download all updates to their communications program free of charge!

$100/yr. ($50/yr. renewal)

**MEX PACK**.........................$99.95

As a special introduction to our remote operating module and terminal emulation system, we're offering an all-in-one package for a limited time only. MEX-PACK is a complete communications package that includes: MEX-PC (version 1.50); REO remote operating module; TEM intelligent terminal emulation module; and a six-month subscription to the NightOwl Connection. Purchased individually, the items in this package would cost you $144.85 — a savings of almost $45!

Give us a toll-free call at
1-800-NITEOWL
In Wisconsin, 1-414-563-4013

NightOwl Software Inc., Rt. 1, Box 7, Ft. Atkinson, WI., 53538
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 UPS ground shipping on prepaid orders. Shipping is added to VISA, M/C, and C.O.D. orders. CA residents, please add sales tax.

FULCRUM COMPUTER PRODUCTS
707/433-0202
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. 1-8080, 8015 and 8035 main frames with 21 slot mother boards. CPA front panel and A/D board.
TurboDOS/PC

by Michael Guttman & Vince Mills

The makers and users of multi-user micros have long argued that their systems offer a more reliable and cost-effective solution to most business applications than the ubiquitous IBM-PC. However, even the most sophisticated user now finds it hard to pass up the many useful and innovative software products spawned by the PC.

Fortunately, S-100 users don’t necessarily have to swim against the PC tide. New products and approaches are coming to market that effectively marry S-100 systems to the PC. One of the more intriguing products in this genre is TurboDOS/PC, a software product from Software 2000 that allows a PC to act as a node on a TurboDOS network.

TurboDOS/PC is definitely not a consumer product. It is aimed strictly at system integrators, who must make the appropriate adjustments to run it properly with their hardware. A typical configuration would call for a multi-user micro running TurboDOS to act simultaneously as a network file server for one or more PCs.

One such implementation of TurboDOS/PC comes from Intercontinental Micro Systems, Anaheim CA, a well-known maker of S-100 products. Among their current offerings are ARCNet interface boards that can be used to hook PCs to an S-100 micro running TurboDOS, or to another PC. ICM generously provided us with both the S-100 and PC interfaces, appropriate software, a compatible TurboDOS computer and, for a day, a technician to get them up and running.

The possible uses for such a configuration are virtually endless. The end user gets all the advantages of both the TurboDOS and PC environments and can preserve his investments in both. On the one hand, he can add inexpensive terminals to his high-performance S-100 machine to act as workstations for generic production functions such as word processing, accounting, and the like. At the same time, he can hook up PCs of varying configurations to run popular PC-based software while sharing files and peripherals with S-100 users and other PCs.

WHY ARCNet?

The choice of ARCNet rather than the better known Ethernet is logical given the likely audience of TurboDOS/PC. ARCNet uses a token-passing system, which tends to be one-third the cost of a comparable Carrier-Sensing Multiple Access/Collision Detection (CSMA/CD) system, such as Ethernet (tm).

At the same time, ARCNet’s 2.5 Mbit/s data transfer speed can easily outperform Ethernet’s 10Mbit/s in a lightly loaded network. ARCNet tends to degrade proportionately with network loading, whereas Ethernet has been reported to bog down at even 40% utilization. This performance difference is due to ARCNet’s token-passing arrangement; no time is wasted on data collisions. Another key to the economy of ARCNet is hidden in cabling costs. Whereas Ethernet requires special cabling and expensive taps, ARCNet offers a choice of twisted-pair or coaxial connection. In our tests, every cabling scheme tried worked the first time (even with unshielded telephone wire!).

Although it is relatively inexpensive, ARCNet has plenty of power. Advanced users of TurboDOS/PC can make use of the more sophisticated features of ARCNet to create very elaborate networking structures that provide for a rich mix of multi-user systems and PCs. ARCNet protocol allows linking up to 4000 users per network with 255 nodes per network segment over either coaxial or twisted pair cable. Cable lengths can be up to 40 miles between processors with active repeaters, or hubs. With a passive hub, cables can be stretched 2500 feet without difficulty.

HOW THE NETWORK OPERATES

Using ARCNet, TurboDOS/PC interfaces to MS-DOS or PC-DOS as an installed device driver and interrupt-driven background task that handles ARCNet reception and transmitting. To the user, the network consists of one or more remote disk drives and printer ports. These are mapped into the file server (either an S-100, PC, XT, AT or clone system) with full record-locking and printer buffering.

The master processor is the critical component of the network. It must mediate the network, as well as handle disk and printer requests from the slave nodes. Doing this right requires both exceptional computing performance and fast drives. It also requires a high-performance, multi-user operating system. TurboDOS, a variant of Digital Research’s multiuser MP/M, appears to be a good choice. It has been optimized for split-second DMA, large-capacity drives and file sharing, and offers compatibility with the large pool of CP/M-based software.

TurboDOS will run on S-100 systems with Z80A/B or 8086 origins. ICM sells single-board microcomputers based on Z80A/B and 80186 that are configured for network master use. In the system we tested, an 8 Mhz 80186 chip ran the operating system, with 256K onboard memory. The rest of the card cage contained the ARCNet interface board, an extra 256K memory board and the hard and floppy disk interface.

Optionally, the system can also be configured with one or more Z80 or 8086 slave processors, which are then mapped into the same memory as the network master processor. These can run business and accounting applications compatible with CP/M-80 or M/PM-86 record and file locking, using standard ASCII terminals.

THE SOFTWARE

TurboDOS runs virtually all programs designed for CP/M and MP/M and features full record and file locking for applications that may need it. All the familiar CP/M commands are there, such as COPY, FORMAT, DIR, etc., although some commands use slightly different conventions for assigning parameters. User security can be enforced through LOGON and LOGOFF.

TurboDOS/PC is a PC’s gateway to the TurboDOS network. As a network client, the PC is offered access to the file and print
TurboDOS/PC accommodates both local run to establish the printer server link. Printed output can be sent to the printer server on a character-by-character basis or queued for spooled output. Printing modes are controlled with a PRINT command.

An additional memory-resident program can be run to establish the printer server link. TurboDOS/PC accommodates both local and remote printers. Each of the 16 remote printers is assigned a letter (A, B, C, etc.)

Printer output from your PC can be directly printed on a character-by-character basis or queued for spooled output. Printing modes are controlled with a PRINT command.

For example, to send printed output to the printer attached to your PC, a user merely types 'PRINT LOCAL'. Similar commands specify which remote printer to print to, and whether a queue is to be used. Full print job control is provided with the PRINTER command, which can take a printer offline, restart or abort a queued print job.

Through the TurboDOS software, all file requests are handled transparently through extra disk drives. Limited subdirectory capabilities are allowed, although all subdirectories on the file server are accessible on the 31 user numbers allowed under TurboDOS.

TurboDOS/PC maintains a mapping between each MS-DOS subdirectory path name and the corresponding TurboDOS user number. In addition, the remote file servers return standard DOS error messages. The directory name and the corresponding TurboDOS user number can be used to identify files on the server.

These problems seemed pretty minor compared to generally excellent overall performance and reliability of the network. Cost-wise, the system compares favorably with other networks. From ICM, the TurboDOS/PC software lists for $100, and each network card about $495. (ICM is a hardware manufacturer and software distributor, so prices to distributors and dealers are lower.)

WHITHER TURBODOS-PC?

Software 2000 and ICM have done a rather admirable job in providing a viable connection to CP/M based multi-user micros. For users or vendors with a substantial existing commitment to CP/M software and S-100 hardware, this may be the vehicle that allows them to keep pace with the PC revolution.

The product could also conceivably appeal to existing CP/M users who want to move into multi-user applications and see S-100 systems as a viable alternative. Certainly, current market surveys suggest that many more PC users are beginning to think about multi-user applications.

However, these same surveys also seem to indicate that most users are more likely to wait around for network hardware and software standards to emerge in the MS-DOS environment than to migrate to the older but less familiar S-100, CP/M-based multi-user micros. Fortunately, those who have reason to buck this trend have a good option with TurboDOS/PC.
Assembling An AT Clone

By Sol Libes

In the March/April and May/June '85 issues of M/SJ Henry Kee and I discussed our experiences assembling PC and XT clones. The PC/XT-clone marketplace has changed a great deal in the past year, with prices dropping to well under the $1,000 level for fully equipped systems. Clone makers have also begun shipping AT clones. The AT marketplace has already reached a saturation level and prices on AT clones are dropping rapidly. In this article I will discuss my experiences putting together an AT clone.

WHY AN AT?
If someone had told me a year ago that I would soon have, at home, a full 16-bit system with 1MByte of RAM (plus video RAM and a cache memory here and there) and 33MBytes of hard disk storage, I would have replied that "they were out of their cotton-pick'n minds." But here I sit writing this article on just such a system. The cost was about $3,000... which incidentally is about the same as what I paid for my old S-100 system about 4 years ago. That system has a 6-MHz Z80, 128Kbytes of RAM, two 1MByte 8" floppy drives, 4 ports and a Zenith terminal.

We needed another system here at the M/SJ publishing empire (located in our basement). We have two part-time people working for us now and there are evenings when all of us are pounding the keys. One system may be doing data-base updating, another preparing an article for typesetting, and another editing. We have our old S-100 system, which is today used only for inter­facing to authors who send us 8" CP/M disks. Our other two systems are a very old PC and the XT-clone described in the May/June '85 article (which is still clicking away). The PC was purchased when IBM first introduced the machine in late '81. It still has the old 64K RAM motherboard with the old BIOS ROM and cassette interface. The machine was expanded over the years and now has a 10Mbyte hard disk, 640K of RAM, two floppies, clock/calendar, two serial and one parallel ports and a color graphics display. It has come a long way from its humble beginnings.

I bought the PC at a flea market from a fellow who had bought it only a few weeks earlier and needed the cash for a down payment on a plane. At half the list price I couldn't pass up the deal. It came with single-sided drives and 64K of memory and PC-DOS version 1.0. Version 1 was an almost exact copy of CP/M. At the time, my S-100 system ran rings around the PC. When, in late '82 IBM introduced version 2 of PC-DOS, I expanded the machine to 256K, changed to double-sided drives and bought some of the really good software coming out for the PC. By 1983 I began to use the PC more than the S-100 system.

At the school where I teach my department purchased an AT in the spring of '85 and I began using it. I immediately became spoiled by its lightening fast speed. Disk accesses appeared to be almost as fast as RAM disk and gone were the long waits for sorts and compiles. I began longing for an AT of my own, but the high cost put me off. When the AT clones appeared, at more realistic prices, I decided to jump in.

WHAT DOES IT COST?
There is no doubt that the cost of an AT clone is significantly more than an XT clone.... but, it is about the same price as a comparably equipped IBM-XT.

When I bought my XT clone dealers were only selling parts and it was up to me to assemble the system. Things changed rapidly and today most dealers sell an assembled system for the same price, or less, than the cost of the separate components. The same is pretty much true for the AT-clones.

I bought my AT-clone from PC's Limited, 7801 N. Lamar, Suite E-200, Austin TX 78752 (800-426-5150). I bought a basic system for $1,995.00. The system included an AT-clone motherboard with 1MByte of RAM, 1.2MByte floppy drive, two serial ports and a parallel port, clock/calender, and a disk controller (with cables) to handle up to two floppy drives and a hard disk drive. In addition I decided to splurge and ordered a 30MByte drive for another $849. A 20MByte drive would have saved me only $150. I had shopped around and could have bought a drive and components for probably a hundred dollars less but I had heard good things about this mail order outfit.

I added a standard graphics display controller card ($119) and monitor ($85). Since I already had PC-DOS for my other systems I did not have to buy it again. Thus the total cost was slightly over $3,000....about $1,000 more than a comparably equipped XT-clone!

WHAT DID I GET?
About 2 weeks after I placed my order over the phone a big white box arrived containing the system. I was pleased to find the hard disk already installed in the unit. Documentation consisted of three sheets of paper telling me (in very very few words) how to set the switches on the motherboard and keyboard and how to install PC-DOS on the hard disk......more about this later.

The motherboard was made by Faraday, the disk controller card by Western Digital, the hard disk by Seagate and the floppy drive by Teac, all well known and respected manufacturers. The power supply is rated at 193 Watts and contained no manufacturers identification. The cabinet is a very close copy of IBM's AT cabinet.

The motherboard contained 1MByte of RAM in 4 banks of 256K chips (all socket­ted). This was a pleasure to see after looking at the IBM kluge of soldering 4 banks of 64K memory chips atop of 4 banks to get 512K on their motherboard. Another nice thing was that the 2 serial ports and one parallel port were on the motherboard leaving 6 free slots for plug-ins. At school, I had to install two plug-in cards to get I/O capability, leaving only 4 available slots. Another pleasing thing here was that the ports all used standard 25-pin connectors. IBM's interface cards now come with 9-pin connectors for the serial port and at school I had to make up adapters to go from 9 to 25-pins. Making up the cables was not easy since the documentation included with the AT did not include any pin-out information.

However, there was one small annoyance in that the port connectors did not...
include screw posts to secure the mating connectors in place.

The one sheet of instructions on setting up the motherboard left quite a bit to be desired. For example there are some switch-es and positioners on the motherboard which are not identified. There are also some empty sockets on the motherboard which are not identified. I suspect that, if one wanted, one could buy a technical manual from Faraday.

The same is also true for the floppy/hard disk drive controller card. There is no information provided and there are quite a few jumpers and unused connectors on the card.

Because of this lack of documentation I recommend this system only for the person with experience bringing up a microcomputer system. This system is certainly not a plug-it-in-and-flip-on-the-switch system.

INSTALLATION

I removed the cabinet cover, installed a standard graphics controller card (also a clone), set a jumper on the motherboard for the type of display controller used, connected the monitor, and slid a floppy system disk into the drive. I turned on power and the system booted up.

I was able to access the hard disk as drive C. The dealer had taken the trouble to initialize the hard disk drive for me. Using the standard IBM FORMAT program I installed the operating system on the hard disk and the system was all set.

It should be pointed out that if you buy a hard disk separately you may have to initialize it yourself. To do this you will need a copy of the IBM AT Diagnostics disk, which is not furnished with PC-DOS 3.0. Only the "Standard" Diagnostic disk is furnished. All IBM dealers have a copy of this disk in their service departments.

If, when powering up the system, with no floppy disk in drive A, you get an error message code of "1790" you know that the hard disk has not been initialized and you will have to do this with the SETUP program found on the AT Diagnostics disk. After running SETUP, run the FDISK program on the Standard Diagnostics disk to set the partitioning of the hard disk drive and last format the disk and move the operating system on to the hard disk.

One other important point to know if you buy your drives separately. AT cabinets require that the drives have rails attached to their sides. The drives then can be slid into their compartments and secured from the front of the cabinet. Therefore, make sure that any drive you purchase has these rails.

PERFORMANCE

The Seagate drive was rated for 40 msc and proved as fast as the IBM’s drive. I ran several tests to check boot time, copying files from the hard disk to floppy and vice versa and some sorting and compiling tasks. The clone appears to have essentially the same speed as the IBM. It should be pointed out that performance was almost four times faster than a standard PCXT. (In a subsequent article we will discuss souping up the performance of a PCXT system.)

Like the old IBM AT the processor’s clock crystals plug in and looks like it can be changed easily to provide even faster performance. Several companies sell replacement crystals for this purpose. It should be pointed out that the current production of the IBM-AT now has its crystal soldered in place and a new ROM to prevent altering the clock speed.

The keyboard’s key layout is an exact copy of the IBM-AT. However, there are a pair of switches on the underside of the keyboard which allow its use with a PC/XT and also repositioning of the ESC key to left side of the keyboard (its more common location). The keyboard has a very different feel from the IBM keyboard. The key travel is less and there is no audible key-click. I suspect users with a lighter touch will prefer it. Others, like me, who have a heavy touch and were raised on IBM keyboards will too much. We stock more than 80 boards with experience bringing up a microcomputer system. This system is certainly not a plug-it-in-and-flip-on-the-switch system.

IN CONCLUSION

I have no qualms recommending this system to experienced users who need the added speed. It is certainly not for the uninitiated. The company provides a 30 day, no questions asked return policy, with a 90% refund and they do claim to provide telephone technical assistance…. I did not find it necessary to call them.

It is possible to purchase components separately and put together your own system. For example, one advertiser in this magazine (Atlaw) sells many AT-compatible components (case $119, Keyboard $109, power supply $169 and motherboard $950). If you have a cheap source for drives and controller this approach will cut the cost of putting together an AT clone.

Within the next few months we should begin to see more AT components from the East reach our shores. Prices should really tumble and the price difference between XT and AT clones should lessen. Further, we can expect to see "turbo" AT-clones running at twice the speed of the IBM-AT.

Micro/Systems Journal January/February 1986
EARTH COMPUTERS launches two of the "Hottest" new stars in the S-100 Universe. Both the TURBOMASTER 8™ and TURBOSLAVE I™ are Star performers, featuring high speed Z-80H CPUs.

TURBOMASTER 8

This outstanding new 8-Bit Single Board Computer offers features that are out of this world:
• On-board ST-506 Winchester Controller
• TurboDOS, CP/M, MP/M compatible
• 5-1/4” and 8” Floppy Controller
• Up to 256KB of memory
• 8 MHz, Z-80H CPU
• 2 Serial ports
• 1 Parallel port

TURBOSLAVE I

The perfect companion to the TURBOMASTER 8 or other 8/16-Bit Master processor. This high speed slave utilizes an 8 MHz Z-80H CPU and offers extensive on-board diagnostics... an industry exclusive.
• No paddle boards
• S-100, IEEE 696 compatible
• 128KB of RAM
• 2 RS-232 ports, 50–38.4K Baud
• FIFO communications
• Data transfers to 1 MB

EARTH's new stars are fully compatible with the Multi-user TurboDOS operating system, and will operate in most S-100 systems, including pre-IEEE 696 systems.

EARTH COMPUTERS also manufacturers a growing line of PC compatible stellar performers such as:

• TURBOSLAVE PC™—an 8 MHz Z-80 single board slave processor that runs CP/M applications on a PC and is compatible with the TurboDOS multi-user operating system.

• TURBOACCEL 286™—a high performance 80286 accelerator that boosts PC performance up to five times.

• EARTHNET PC™ and EARTHNET S-100, the low cost, ARCNET-compatible way to tie PC and S-100 systems together.

To put these stars to work for you, call or write EARTH COMPUTERS. BE SURE TO ASK ABOUT HOW YOU CAN WIN A FREE Z-80 CO-PROCESSOR BOARD.
EUREKA!®

In Greek it means "I found it!". In CP/M® it means you'll never lose it again! Eureka! is the premier CP/M® disk cataloger. It puts the critical information about your library of disks and files in one place and allows you to quickly and easily retrieve any of the information. Never again will you have to wonder what disk that letter SMI... is on. or what it is for...Eureka! can tell you in a snap. Lawyers, writers, programmers and many others save time with Eureka!. Haven't you suffered long enough without it???

"We finally chose EUREKA!...largely because it has the ability to read comments directly from a file. EUREKA! is easy to learn and use, has more access and repair choices, lists files by many different ways, and has an attractive price."

T. Bove & C. Rhodes, USER'S GUIDE No 11

EUREKA!® is a trademark of Mendocino Software Company, Inc.

Mendocino Software Company, Inc.

Dept. M - 5
P.O. Box 1564
Willits, CA 95490

(707) 458-9130

VISA & MasterCard accepted

$29.95 plus $4.00 s&h

Write For A Free Info Today.

USA distributor:

SOUTHERN PACIFIC (USA) INC.

P.O. BOX 4427, Berkeley, CA 94704-0427 U.S.A.

Phone: 415-253-1270

Dealers and distributor inquiries welcome

ZENET NETWORK through twist pair

• 6MHz HD648180/780 upward compatible 1.512K byte on board (256K installed). 384K RAM/DSK
• LAN: ZENET Port 800K baud CSMA/CD twist pair bus type upto 500 meters
• Floppy: 3.5" and 8 inch d/s density, d/s sided and d/s track automatic density/format checking
• Hard disc, SCSI interface on board
• Video, 80 X 24 characters (color) and 640 X 200 pixels color graphic. 128K byte video RAM character set is downloaded from disk
• Timer, battery back up calendar

WORLD SMALLEST COMPUTER

• Full personal CP/M system in palm 4mmz 280 256K RAM (128K RAM DISK)
• Serial: RS232C X 2, automatic baud rate checking
• Parallel: Centronics type printer port
• Floppy: 3.5" inch micro floppy disk drive 800K byte (option 8.5" inch drive d/s sided d/s track automatic density checking)

Full featured CP/M® plus system

• Z80 4MHz 128K Byte RAM
• Floppy: 3.5" and 8 inch d/s density, d/s sided and d/s track upto 4 disk drives Automatic density/format check
• Serial: RS232C X 2
• Parallel: Centronics type, 16 bits IQ/7/8 bit keyboard port
• Timer: battery back up calendar
• Video: 80 X 24 high speed CRT controller
• O.S.: CP/M® plus bank version included
• Size: 10 X 6 X 4 1/4 layered

Protected version $50
Unprotected version $75
Additional Licenses $15

Mendocino Software Company, Inc.

add $2.50 shipping
Calif. residents add 6.25% sales tax

A EUREKA!® package is designed to run on only a single system.

MSC-LAT1

$649

MSC-I CO

$499

MSC-MTC

$299

MSC-PCX

$199

MSC-MTP

$189

CP/TREE

Tree-Structured Named Directories for CP/M® 2.2

- Transforms user areas into Unix-like directories
- Provides Unix-like directory commands:
  CD, MKDIR, PWS, RMDIR, TREE
- Includes a CCP replacement featuring:
  • Command and file search path. Use programs like WordStar® from any directory!
  • Erase with query
  • Wildcard rename with query
- Provides output redirection to disk file
- Uses as little as ½ k RAM, never more than 2½ k
- A must for hard disks
- Installs easily: requires no modifications to BIOS or BDOS
- Requires standard CP/M® 2.2 (not 3.0 or Apple), Z80, 48k RAM

$29.95 plus $4.00 s&h

To order: Specify disk format (8" SSD, NorthStar DD). Call for info on others: M.C., Visa, COD (add $1.90), check (delays shipping 2 weeks). MA residents add 6% sales tax. POs not accepted.

Precise Electronics
P.O. Box 339
New Town Branch
Boston, MA 02258

(617) 332-3977

Super assemblers plus the world’s largest selection of cross assemblers!

Z-80 Macroassembler
Power for larger programs! This 2500AD macroassembler includes:
- Zilog Z-80 Macroassembler (with the same powerful features as all our assemblers)
- powerful linker that will link up to 128 files. Com files may start at any address
- Intel 8080 to Zilog Z-80 Source Code Converter (to convert all your Intel source to Zilog Syntax in one simple step)
- COM to Hex Converter (to convert your object files to Hex for PROM creation, etc.)
- 52 page User Manual

8086/88 Assembler with Translator
Available for MSDOS, PCDOS, or CPM/86! This fully relocatable macro-assembler will assemble and link code for MSDOS (PCDOS) and CPM/86 on either a CPM/86 or MSDOS machine. This package also includes:
- An 8080 to 8086 source code translator (no limit on program size to translate)
- A Z-80 to 8086 translator
- 64 page user manual
- 4 linkers included:
  - MSDOS produces .EXE file
  - CPM/86 produces .CMD file
  - Pure object code generation
  - Object code and address information only
- Linker features:
  - Links up to 128 files
  - Submit mode invocation
  - Code, Data Stack and extra segments
  - Handles complex overlays
  - Written in assembly language for fast assemblies.

Z-8000 Cross Development Package
Instant Z-8000 Software! This package allows development and conversion of software for the Z8001, 8002, 8003 and 8004 based machines on a Z-80, Z-8000 or 8086 machine. This powerful package includes:
- a Z-80/8080 to Z-8000 Assembly Language Code Translator
- Z-8000 Macro Cross Assembler and Linker
The Translators provide Z-8000 source code from Intel 8080 or Zilog Z-80 source code. The Z-8000 source code used by these packages are the unique 2500AD syntax using Zilog mnemonics, designed to make the transition from Z-80 code writing to Z-8000 easy.

All 2500 AD Assemblers and Cross Assemblers support the following features:
- Relocatable Code — the packages include a versatile Linker that will link up to 128 files together, or just be used for external reference resolution. Supports separate Code and Data space. The Linker allows Submit Mode or Command Invocation.
- Large File Handling Capacity — the Assembler will process files as large as the disk storage device. All buffers including the symbol table and data overflow to disk.
- Powerful Macro Section — handles string comparisons during parameter substitutions. Recursion and nesting limited only by the amount of disk storage available.
- Conditional Assembly — allows up to 248 levels of nesting.

Assembly Time Calculator — will perform calculations with up to 16 pending operands, using 16 or 32 Bit arithmetic (32 Bit only for 16 Bit products). The algebraic hierarchy may be changed through the use of parentheses.
- Include files supported — Listing Control — allows listing of sections on the program with convenient assembly error detection overrides, along with assembly run time commands that may be used to dynamically change the listing mode during assembly.
- Hex File Converter, included — for those who have special requirements, and need to generate object code in this format.

Cross reference table generated — Plain English Error Messages —
- System requirements for all programs: Z-80 CP/M 2.2 System with 54k TPA and at least a 96 column printer is recommended. Or 8086/88 256k CP/M-86 or MSDOS (PCDOS).
- Cross Assembler Special Features
  Z-8 — User defined registers names, standard Zilog and Z-80 style support. Tec Hex output option.
  8748 — standard Intel and Z-80 style syntax supported.
  8051 — 512 User defined register or addressable bit names.
  6800 Family — absolute or relocatable modes, all addressing modes supported. Motorola syntax compatible. Intel Hex or S-Record format output.
  6502 — Standard syntax or Z-80 type syntax supported, all addressing modes supported.
<table>
<thead>
<tr>
<th></th>
<th>Z80</th>
<th>Z8000™</th>
<th>IBM PC</th>
<th>IBM PC</th>
<th>OLIVETTI</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CP/M®</td>
<td>SYSTEM 8000</td>
<td>UNIX</td>
<td>CP/M 86</td>
<td>M-20</td>
</tr>
<tr>
<td>Z8000™</td>
<td>$299.50</td>
<td>$750.00</td>
<td>$299.50</td>
<td>$299.50</td>
<td>$299.50</td>
</tr>
<tr>
<td>Z80</td>
<td>99.50</td>
<td>500.00</td>
<td>199.50</td>
<td>199.50</td>
<td>199.50</td>
</tr>
<tr>
<td>Z8</td>
<td>199.50</td>
<td>500.00</td>
<td>199.50</td>
<td>199.50</td>
<td>199.50</td>
</tr>
<tr>
<td>8086/88</td>
<td>199.50</td>
<td>750.00</td>
<td>99.50</td>
<td>99.50</td>
<td>199.50</td>
</tr>
<tr>
<td>80186</td>
<td>199.50</td>
<td>750.00</td>
<td>199.50</td>
<td>199.50</td>
<td>199.50</td>
</tr>
<tr>
<td>8748</td>
<td>199.50</td>
<td>500.00</td>
<td>199.50</td>
<td>199.50</td>
<td>199.50</td>
</tr>
<tr>
<td>8044/51</td>
<td>199.50</td>
<td>500.00</td>
<td>199.50</td>
<td>199.50</td>
<td>199.50</td>
</tr>
<tr>
<td>8080</td>
<td>199.50</td>
<td>500.00</td>
<td>199.50</td>
<td>199.50</td>
<td>199.50</td>
</tr>
<tr>
<td>8085</td>
<td>199.50</td>
<td>500.00</td>
<td>199.50</td>
<td>199.50</td>
<td>199.50</td>
</tr>
<tr>
<td>8096</td>
<td>199.50</td>
<td>500.00</td>
<td>199.50</td>
<td>199.50</td>
<td>199.50</td>
</tr>
<tr>
<td>68000,08,10</td>
<td>299.50</td>
<td>750.00</td>
<td>299.50</td>
<td>299.50</td>
<td>299.50</td>
</tr>
<tr>
<td>6800,02,08</td>
<td>199.50</td>
<td>500.00</td>
<td>199.50</td>
<td>199.50</td>
<td>199.50</td>
</tr>
<tr>
<td>6801,03</td>
<td>199.50</td>
<td>500.00</td>
<td>199.50</td>
<td>199.50</td>
<td>199.50</td>
</tr>
<tr>
<td>6804</td>
<td>199.50</td>
<td>500.00</td>
<td>199.50</td>
<td>199.50</td>
<td>199.50</td>
</tr>
<tr>
<td>6805</td>
<td>199.50</td>
<td>500.00</td>
<td>199.50</td>
<td>199.50</td>
<td>199.50</td>
</tr>
<tr>
<td>6809</td>
<td>199.50</td>
<td>500.00</td>
<td>199.50</td>
<td>199.50</td>
<td>199.50</td>
</tr>
<tr>
<td>32000</td>
<td>399.50</td>
<td>750.00</td>
<td>399.50</td>
<td>399.50</td>
<td>399.50</td>
</tr>
<tr>
<td>COP5400</td>
<td>199.50</td>
<td>500.00</td>
<td>199.50</td>
<td>199.50</td>
<td>199.50</td>
</tr>
<tr>
<td>NSC800</td>
<td>199.50</td>
<td>500.00</td>
<td>199.50</td>
<td>199.50</td>
<td>199.50</td>
</tr>
<tr>
<td>6301</td>
<td>199.50</td>
<td>500.00</td>
<td>199.50</td>
<td>199.50</td>
<td>199.50</td>
</tr>
<tr>
<td>6501/11</td>
<td>199.50</td>
<td>500.00</td>
<td>199.50</td>
<td>199.50</td>
<td>199.50</td>
</tr>
<tr>
<td>6502</td>
<td>199.50</td>
<td>500.00</td>
<td>199.50</td>
<td>199.50</td>
<td>199.50</td>
</tr>
<tr>
<td>65C02</td>
<td>199.50</td>
<td>500.00</td>
<td>199.50</td>
<td>199.50</td>
<td>199.50</td>
</tr>
<tr>
<td>1802</td>
<td>199.50</td>
<td>500.00</td>
<td>199.50</td>
<td>199.50</td>
<td>199.50</td>
</tr>
<tr>
<td>F8/3870</td>
<td>199.50</td>
<td>500.00</td>
<td>199.50</td>
<td>199.50</td>
<td>199.50</td>
</tr>
<tr>
<td>NEC7500</td>
<td>199.50</td>
<td>500.00</td>
<td>199.50</td>
<td>199.50</td>
<td>199.50</td>
</tr>
<tr>
<td>NCR/32</td>
<td>399.50</td>
<td>750.00</td>
<td>399.50</td>
<td>399.50</td>
<td>399.50</td>
</tr>
</tbody>
</table>

| Subtotal       | $       | $       | $       | $       | $       |

Name ____________________________
Company ____________________________
Address ____________________________
City ______ State ______ Zip ______
Phone ____________________________
Make and model of computer system ____________________________

□ COD (2500AD pays COD charges)
□ VISA or MasterCard
   Number ____________________________
   Expiration Date ____________________________

Check one: Shipping UPS Blue Label no charge, $20.00 International per unit $ ______
□ 8" Single
□ Osborne
□ IBM PC
□ Cartridge tape
□ Apple (Softcard)
□ Kaypro DSDD Total Order $ ______
□ Other formats available, please call!

Signature ____________________________

TO ORDER. Simply circle the product or products you want in the price columns, and add up your order.
Scientific & Technical
Word Processors

By Steven Bosak

In the concluding part of this series, we review four more technical and scientific word processors. I must note that I was unable to put the screws to some of these products; some manufacturers believed it best not to have new, unproven products reviewed, while others did not get their programs to me in time for Micro/Systems' deadline. Still others submitted only demo disks for "evaluation". The latter case -- a company sending demo disks -- is absolutely insulting when testing programs that are so dependent on intensive use and reliable performance. Demos can do what the average user may not, while glossing over or completely eliminating printer and other performance bugs and glitches. When you test drive a car, do you let the dealer sit behind the wheel and describe how the steering feels?

Another problem, which also pertains to the last systems on my review list, has to do with the Superlok (TM) copy protection program. Two of the packages used this program to defeat copying. Both caused no end of headaches in the installation process. A particularly nasty implementation of Superlok ate three FORMAT.COM utilities, two DISKCOPY.COM utilities and crashed the technical writer's program's own printer drivers and screen installation. Real friendly, huh? Gee, I sure am glad that Superlok, and the companies using its marvelous protection, have their paranoid anti-copy interests so high on their priority list that the "protection" actually DESTROYS the program when it is being legitimately installed. That must be the ultimate protection of all! No installation, no illegal copies! But enough soap-boxing, let's look at the processors themselves.

Once again I used the MS-DOS Zenith Z-150 with 640K RAM and dual floppies to test the programs. I test-printed with the Toshiba P351 high-speed, letter quality dot matrix printer whenever possible. My back-up printer was the "ol reliable Epson MX-80; some programs did not support the P351 or the NEC 35 series which I had at my disposal.

TECHWRITER

TechWriter has a number of features to recommend it, including the ability to produce ASCII files and to automatically save to disk. These are features I'd like to see in most regular word processors. Other extras in the realm of straight word processing include recovery from disk full, print queuing and spooling from within the program, and automatic creation of table of contents and indices.

INSTALLATION: It's a shame that the installation of TechWriter is not as thoughtful as many of its features. Granted, the copy protection is what fouled my experience the most, but the batch operation which guides this installation is riddled with annoying "Are you sure?" interrogations after the most mundane selection of options. For instance, the "Are you sure?" prompt must be answered after choosing between a Hercules Graphics card OR the IBM graphics adapter. These additional rib-pokers made an already long installation that much longer. I won't even detail the exploits of the cannibalistic Superlok.

FEATURES: The opening menu sends you to new documents, editing old documents, over to the print sub-menu or to file management or DOS. The print menu allows you to set all the usual page and line defaults as well as the printer driver you wish to use. A number of drivers can be on disk for the program to load in.

TechWriter, as I said, handles all the usual word processor musts: word wrap, formatting, search and replace, and deletion by character, word line, sentence or block. Headings and footnotes as well as horizontal scrolling are a big plus here.

The character set supplied with TechWriter is better than most, but those needing chemical symbols or graphics displayed within their manuscripts will not find any support here. Still, if you deal with the usual Greek and math characters, you'll find a number of TechWriter's methods for entering alternate characters a welcome relief from many of the programs that require flipping back and forth between modes.

The first mode of TechWriter's alternate character entry which I was impressed with was its use of an "after key" definition. In other words, when entering an "alpha" character, I typed a regular "a" from the keyboard; then defined that "a" as an "alpha" by pressing the F6(Greek and math) key directly afterwards. I really appreciate this approach for two reasons: There were very few keystrokes and no switching in and out of modes, and I found that this "after key" method was an effective mnemonic for familiarization with the
alternate character set. I saw on the screen which regular character would get me a Greek or math character. By seeing the regular character before it transformed into its Greek or math counterpart, I found myself memorizing the keyboard alternatives fairly quickly and referring to the keyboard overlay (which was printed far too small) or the on-screen pop-ups very infrequently.

TechWriter’s built-in macro facility, GNOME, makes large brackets and braces, large sigma and other characters easy to construct, save, and recall. And the GNOME was available for all “regular” word processing tasks also.

Entry of equations was painless and straightforward. Two sizes of sub and superscripts were available, although the smallest is difficult to read on the screen. What you see on the screen is almost what you get. Equations are full line spacings on screen, reduced to half lines during a print.

In my print sample you’ll note curious gaps -- the printer and TechWriter’s printer driver couldn’t quite agree on matters, although the blanks were characters on the screen.

DOCUMENTATION: The documentation was great. The program comes with a training manual and a reference manual in separate vinyl, full-sized binders. The training guide was thorough and well organized. The reference manual was easy to use, to look up features and commands, or to read through beginning to end for a more complete understanding of the system. I only wish there had been more information on the printer drivers, as it seemed from some cursory doing that some drivers did not totally support the package. The documentation was thoughtful and clear all around.

PRINTING: After you provide TechWriter with the defaults, printing is a breeze. You have the option of revising your defaults at any time, as well as changing print drivers before a print session. You need to know the filename of the print driver, however, and there is no way of getting that name without exiting the print section and doing a directory check. If you work with a number of drivers, it may be frustrating trying to remember the correct spelling for a driver. Better to rename your drivers to simple and easily recalled names or, better yet, mark them down somewhere.

I must add, as a person using a number of printers, that switching from printer to printer as well as queuing and spooling from within the program...
Spellbinder allows you to address the character grid pixel by pixel to modify existing fonts or to create wholly new characters which can be saved in Spellbinder's set. An important and impressive aspect of Spellbinder Scientific's generator is its ability to preview on screen what the newly modified or created character would look like on screen before it was saved out to disk. And the touch of one function key sent the "experimental" character to the printer to check results on that end. (figure 1) The generator alone makes this product superior. If font generation is your main concern, you want total control over the fonts displayed and printed, or if you need to create a number of special characters, stop searching -- this is the word processor for you. In addition to its generator, Spellbinder Scientific has extensive equation editing capabilities -- sections of an equation can be marked duplicated, even stored on disk to be retrieved, all with simple cut and paste maneuvers.

The technical text is entered within modes, however. To get a math character you must hit the mode selector to get "math" and then the key which corresponds to the desired symbol. It takes a while to get the hang of finding the right mode. And why these modes were not more logically arranged is beyond me. For instance, when in normal text mode (F9 when editing) you must hit F10 to get to "math". You'd think hitting F9 would get you back to normal type -- wrong, two strokes of the F9 got you back to normal type. These problems would be intolerably confusing if the programmers at Lexisoft hadn't thought to tell you, boldly and clearly, in the upper right corner, just what mode you are in: italics, math, chemical, normal, or raised.

The overprint and raised modes are available for all the character sets -- normal, italics, and math -- making diacriticals and tildes exceptionally easy to enter, create, and subsequently save.

The character sets can be reviewed through menu-driven pop-up keyboards or on the handy, hard plastic reference cards provided with the documentation.

DOCUMENTATION: Spellbinder Scientific came with a "Quick Start" manual to get users entering and editing with the Scientific package within minutes. Not bad for starters. The documentation for Spellbinder's standard word processor is also included, as well as a supplement explaining the scientific and technical processor. The documentation itself was, overall, well written. I only wish Lexisoft had spent more time making it visually appealing. The type was small and difficult to read, the inner margins so thin at times that the three ring binder threatened to devour the text. The layout was logical throughout. The technical and scientific character section concerning character editing and creation was especially well done. The equation editing and chemical graphics segments were illustrated with useful examples. I just wish it was easier to flip through to needed sections or less wearisome on the eyes -- VDT strain is enough for one set of eyes to contend with!

PRINTING: Printing was easy and fairly fast. You could not however, change drivers without re-installing an alternate driver. In addition, I find Spellbinder's format commands even more bothersome than WordStar's. Character after character must be added up on the command line to get the page formatted the way you want. To be fair, this allows extreme flexibility for the final print, and once you have the knack of the codes, they pretty much become natural, many never need to be changed (default margins and page length, for instance). One aspect of the print routine seemed unnecessarily harsh, though. On one print run, I had neglected to turn the printer on. I received, from within Spellbinder Scientific, the DOS message "Write fault to device PRN, Abort Retry or Ignore". I punched Abort and was dumped immediately to DOS -- pretty harsh treatment for forgetting that a printer had been turned off. But overall, printing was a routine procedure with few surprises.

FINAL CONSIDERATIONS: With only a few exceptions, this was the kind of technical and scientific
word processor that didn’t FEEL like a technical processor. Much of the smoothness of operation can be credited to Lexisoft’s adherence to Spellbinder’s ‘‘Scientific’’ into any already solid word processor. If you want one processor to do it all — letters, regular manuscripts, and text, as well as technical documents — this system fits the bill. If you use chemical symbols, the building set provided here might be the deciding factor for you.

Be prepared, however. If you don’t like Spellbinder as a word processor, you are not going to like it for technical and scientific text. I find some of Spellbinder’s commands and routines a bit cumbersome for everyday use — to save a file to disk, for instance, you must go to the top of the file, enter three keystroke commands and, finally, enter the filename. In fact, the entry and editing of the scientific equations might seem easier than mastery of the word processor itself.

SUMMARY: Spellbinder is easy to set up; equations are easy to enter and edit. On the whole this system is the most sophisticated technical and scientific character support available. With the exception of those aggravating modes and the dreadfully typeset manuals, I’d have to say it’s the best all-around system available.

MASS-11

Before we end our MS-DOS search for a technical and scientific word processor, we must make at least one more stop: MASS-11. This side of a VAX, (and the other, for that matter) MASS-11 is a superior word processor. You can spend all day in MASS-11 and never leave the program, so complete are the available features. Split screen editing, vertical typing, column editing and entry, a four-function calculator that performs on either vertical OR horizontal entries, footnotes and headers, auto-embedded date and time in documents, extensive file and list sorts, a spelling checker that is accessible from WITHIN a text file, unerase and flexible cursor jumps (Advance by lines/pages = [FI] [NJx]), I could keep going, the features available seem limitless, including cursor and point addressable graphics!

INSTALLATION: One simple batch file did the trick. I had to answer, as I recall, three questions. The program is big, but don’t let that scare you — it doesn’t behave like a behemoth. The system comes on seven disks, including the start-up installation disk, printer drivers, spelling checker and dictionary and an example/tutorial disk.

FEATURES: I’ve already mentioned, yet hardly covered, all of MASS-11’s great "standard" word processor features (if only MOST of what MASS-11 has were on "standard" word processors!), but the technical and scientific text processing that is included is better than most. It doesn’t have a font generator, but the included character set should be adequate for engineering, general math, and physics.

MASS-11, though an abundantly endowed processor, is never confusing to get around in. Most commands are performed in conjunction with the F1 key. The Greek and math character set is also accessed in this manner, i.e. F1 followed by G and the key representing the alternate character generates the character on-screen. MASS-11 opens up sections in the text for equation blocks, so that moving, reformating text, etc. are still easy to do. Sections of equations can be moved as well. A VERY comforting feature is MASS-11’s ability to subscript and superscript in half and QUARTER spaces.

Despite its size, MASS-11 is fast -- as it uses disk space as virtual memory pages. And there is safety in its size too; the program continually backs-up the file being edited into a disk journal: if there is a glitch, the file is backed-up. When the file is successfully edited and saved, the journal is then deleted, conserving space.

What’s on the screen IS EXACTLY what you get. I had only one print sample -- a laser sample -- and thought it unfair to include a laser representation from only one program. The character set example is from an AMT high-speed dot matrix printer. All the characters are from DEC, the DECtech set in fact, with some addition by the MASS-11 folks.

DOCUMENTATION: Hands down, this is the best documentation I’ve seen for any word processing system. There is a User’s Guide which details — and I mean DETAILS -- every feature of the editor; a Printer Reference Manual which explains the characteristics, interfacing and capabilities of each printer that is supported. A Demo Guide takes you through the training disk; and, also included, is an easel-mounted desk QUICKREF that should be standard operating procedure for all software documentation. The QUICKREF takes up about six inches of desk space and has all the major (and most minor) commands with tips and explanations. The documentation is a boon to the bloodshot eye, elegantly typeset,
printed on heavy rag bond and set into full-sized binders. Content and quality are superb. It seems strange to me that more word processing software vendors don't spend the time to produce quality documentation, especially WORD processing companies.... And if the documentation isn't enough, MASS-II has extensive on-line, menu-driven help.

PRINTING: After setting such Document and Print defaults as footer lines, margins and printer driver. Printing is just a matter of answering the prompt: file to print? MASS-II also supports a very professional print/editing option: redlining. Let's say you have edited a document and made numerous deletions. With the redlining feature, MASS-II will print out those deleted characters, words, lines or paragraphs with an overprint through each; in essence, you can review what you've cut BEFORE you make the final and irrevocable editing decisions. You can restore what you want before a final print run. This is superbly suited to preparing final, edited text -- and is an invaluable feature for working with book-length and publication-ready manuscripts.

FINAL CONSIDERATIONS: Another plus in MASS-11 is its ability to send and receive files from VAX. This gives the program enormous power. If you have access to a VAX on the premises or through a dial-up, you will want MASS-11. Use of the computer as a terminal from within MASS-11 is a big bonus -- as I said, there seems to be little reason to ever return to DOS.

At this time, however, there is just the DECtech character set available. This may pose a problem for some -- check the set beforehand. MEC has found some limitations when using any of the Digital printers, also. Check with MEC.

SUMMARY: If you have a small to large office environment that does heavy word processing with standard technical and scientific characters, get MASS-11...you'll never outgrow it. In fact, if the system will be used for extensive manuscripts, and you use a VAX in the office, I can't think of why you would avoid purchasing MASS-11. The graphics and its ability to import graphics from other sources into your text are features other word processors are still struggling with.

CONCLUSIONS

By all indications the products I've reviewed and those still to be released will continue to improve in performance and versatility.

In fact, I'm confident that many of these programs may blossom into fully integrated packages which would do away with the whole moniker technical and scientific word processor. Imagine, just one system that processes words -- technical or otherwise -- with a minimum amount of keyboard calisthenics. Programs such as T3, Spellbinder Scientific and MASS 11 are on that track, and they bear watching. As it stands, no one system seems to address all the myriad needs and possible applications. But they're getting close, real close.

LAST MINUTE ADDITION

Arriving just in time for a studied peek was Hockney's Egg by Perrigrin Falcon, Inc. That's right, Hockney's Egg...seems this Professor Hockney wanted to type equations in an orderly manner and see them displayed on his PC screen the way they'd print. And the Egg is the product.

The Egg installed fast, no questions asked other than if the installation would take place on a hard or floppy disk. The program itself, including font sets, is right at 200K. And an amazing 200k it is.

Besides being slim, the Egg is fast -- fast to read and write, fast to get running, easy to use. I didn't even crack open the documentation until I wanted to know about the embedded print commands. A card detailing all the major commands was all I needed to enter, edit, move and block move, skip through the 700+ character fonts and get to the character generator and editor.

The Egg may be light on some regular word processing features such as auto indexing, a spelling checker or mail merge, but if you want a word processor for your technical text that takes virtually an hour or two to learn, has all the features you want to edit, enter and modify alternate character sets, Hockney's Egg is a good choice. Macros make entry of large math or chemical constructions as effortless as typing up a Function key. And if you don't like the sets that are pre-defined when you purchase the Egg, it's a simple matter to reprogram all thirty keys to whatever you wish -- including regular text editing functions.

Perhaps the most exciting and innovative aspect of Hockney's Egg is what is to be released in the next version: the ability to directly drive the Line-a-tron 202 or 101 composing machines. Technical and scientific manuscripts are among the most costly to typeset ($45-$70 per page, as I understand). By having a word processor that can directly address the manuscript's own typeset ting needs, costs would be drastically reduced.

The font sets which are provided are more than adequate for a variety of technical, engineering, and scientific manuscripts. If not, the font editor is available to create whatever character you might need or dream up.
Disk Sale

**WHO KILLED ROGET?**

**WORD FINDER did.**

Now you can have the perfect word for your thought in seconds without taking your fingers off the keyboard. WORD FINDER is the electronic thesaurus that becomes an extra function _inside_ your word processor. Whenever you give the command, WORD FINDER reads the word the cursor is on and pops open a window of synonyms. Pick one, and the new word replaces the old, automatically.

WORD FINDER provides 90,000 synonyms for over 9,000 words, nearly twice that of other thesaurus programs. Yet, WORD FINDER uses only 27K of RAM and keeps all the synonyms in a 157K file that can be kept on any disk drive. (A new, even larger database with 150,000 synonyms for 15,000 words will also be available in December, 1985.)

WORD FINDER will quickly help you make your point eloquently, clearly and concisely. So put pizzazz (punch, sizzle, flair) in your writing by putting WORD FINDER in your word processor, today.

**WORD FINDER works inside most popular word processors,* and it's fast and easy to install. Just ask a writer:**

WORD FINDER is an affordable adjunct for whipping (flogging, thrashing) your prose into shape quickly and efficiently, without leaving the document file you're working on.

—David Obregon, PC Magazine


**Writing Consultants**

Call 1-800-828-6293

Available from your local dealer through SOFTEL.

30 DAY, MONEY-BACK GUARANTEE.
ATTENTION: $100 BUS USERS!

Lomas Data Products Presents:
IBM-PC Compatibility for Only

$1749

*User supplied keyboard and monitor required.

IBM Compatible S100 Bus Color Graphics

Lomas Data Products presents Color Magic, the most complete compatibility solution for $100 bus computer products. Color Magic includes three major hardware subfunctions which allow it to emulate the IBM-PC: an entirely compatible video function, an IBM-PC keyboard interface and an IBM-PC compatible timer interface including IBM-PC sound compatibility. Color Magic is a benefit or requirement LDP offers the only viable solution.

THUNDER 186, COLOR MAGIC SET ........ $1749.00

IBM-PC Compatible S100 Bus System

Lomas Data Products offers IBM-PC compatible systems with performance far exceeding that available from IBM. You can purchase systems offering performance of an eight Mhz 8086 or up the performance of an 8Mhz 80286. Each system is capable of supporting 8Mhz math coprocessors. Our 8Mhz 80286 system offers IBM-PC compatibility while offering up to twice the performance of the IBM-PC-RT. For applications where PC compatibility is desirable but higher performance is a benefit or requirement LDP offers the only viable solution.

Who is Lomas Data Products...

Lomas Data Products offers IBM-PC compatible systems with performance far exceeding that available from IBM. You can purchase systems offering performance of an eight Mhz 8086 or up the performance of an 8Mhz 80286. Each system is capable of supporting 8Mhz math coprocessors. Our 8Mhz 80286 system offers IBM-PC compatibility while offering up to twice the performance of the IBM-PC-RT. For applications where PC compatibility is desirable but higher performance is a benefit or requirement LDP offers the only viable solution.
The Egg was effortless to learn and use, the documentation disarmingly frank and genuinely witty, at times even giving you unobtrusive insights into the inner workings of the Egg.

If you have assistants or typists or fellow workers who are complaining about the overly complicated technical processor they are now using, show them the Egg and stand back.

If you want bells, buzzers, and lots of frills with your technical word processor, don't look here -- but if your technical manuscripts demand a great deal of work with alternate character sets, the paper fairly boiling over with equations, try the Egg.

Steven Bosak lives in Evanston Illinois and is a freelance writer/editor who has worked on everything from technical writing to fiction. His novel, titled "Gammon" has just been published by St. Martins/Marek Press.

In addition to the above sets, all characters can be displayed in boldface. A number can also be shown in the small set or script. Egg also has a number of macros (shown to the right) for math functions (F1-F30) and chemical blocks and symbols.

---

**FORTRAN GRAPHICS**

MicroGlyph/SciPlot™ graphics library provides FORTRAN routines for scientific and engineering applications.

- Presentation quality high resolution graphics with 240 by 216 dots/inch.
- Software generated text fonts, data centered symbols, uniform or proportional spacing, string plotting with rotation and size scaling capability, 14 sets.
- Linear or Log scaled axes.
- Vector graphics, point to point plotting with user selectable origin and data scaling.
- Multiple plot frame processing.

**SYSTEM REQUIREMENTS:**
- IBM PC/XT/AT 256k RAM, DOS 2.0, MS FORTRAN-77
- IBM Professional FORTRAN, and graphics printer or color card.

Trademarks: IBM of IBM, MS of Microsoft.
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.

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.

TO ORDER, CALL TOLL FREE: 1-800-367-5134, ext. 804
For information or technical assistance: (808) 623-6361
Specify desired 5¼" or 8" format. Personal check, cashier's check, money order, VISA, MC, or COD welcomed.
P. O. Box 2151
Honolulu, HI 96805

FOR EVERYONE WITHOUT A DISPENSATION FROM MURPHY'S LAW
CUSTOM PRODUCTS
DESIGN • LAYOUT
MANUFACTURING

ECT-100-F
RACKMOUNT CARD CAGES

8080 CPU
CENTRAL PROCESSING UNITS

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

RM-10
CARD CAGE & POWER SUPPLY

64K RAM
FULLY STATIC MEMORY

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.
BUILD AN S-100 EPROM EMULATOR

by Robert A. Rioja

INTRODUCTION
Computers are often used to create new computers and circuits. One way to do this is to use an existing computer (the host) to emulate all or part of the one being developed (the target). One such technique is called In-Circuit Emulation, or ICE. This involves building into the host a circuit (the emulator) which emulates a part of the target, such as its processor, memory, etc., and connecting it to the target via cables. The ends of the cables are terminated with connectors that are physically identical to the components being emulated.

For example, if a particular ICE emulates a 40-pin processor such as a Z80, its cable would end in a 40-pin connector which plugs into the Z80 socket in the target computer. The target system thinks that the component being emulated is actually plugged in, is the cable from the emulator. Because this circuit resides in the host, it can be controlled completely by the host computer and is guaranteed to work, even if the target is incomplete or malfunctioning. Such ICE circuits are readily available for some computers but are not as common for S-100 systems.

OUR EMULATOR
The circuit described in this article is an S-100 board that emulates a number of EPROMs. This means that if you are developing software designed to reside in an EPROM, you can load the software into the emulator and connect its cable to the EPROM socket in your circuit. Now you can test the software on your target, debug it, and make changes or corrections any number of times without having to program, erase with ultraviolet light, and reprogram an EPROM. Once you are satisfied that your software is correct, you can program it into the EPROM, unplug the emulator, plug in the EPROM, and enjoy the fruits of your labor. If you design a lot of EPROM based software, the time you spend building this emulator may be miniscule compared to the time you would have to spend erasing and reprogramming EPROMs while debugging your software.

The emulator consists of up to two memory components each of which emulates one EPROM. Thus you can emulate up to two EPROMs simultaneously! The memories used are eight x 8-k RAMs of eight kilobyte capacity each. This will allow you to emulate any combination of 2708 (1K), 2716 (2K), 2732 (4K), and 2764 (8K) EPROMs. Since the two memories work independently, they can emulate two EPROMs of the same type or of different types. For example, one memory might be used to emulate a 2764 while the other a 2716. The only restriction is that both EPROM sockets in the target MUST share common address and data buses. This is normally the case and should not be a problem. More than one emulator can be used at the same time to emulate more EPROMs if needed. The emulator has two cables coming from it to plug into the EPROM sockets in the target.

The circuit is connected to the S-100 bus through six consecutive output ports. No input ports are used, since data is only sent to the emulator and never read from it. Only the target reads the data from the emulator, but it does so through the cables. The actual port assignments are set by an on-board DIP switch. They are:

<table>
<thead>
<tr>
<th>PORT</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>SET EMULATOR TO LOAD MODE</td>
</tr>
<tr>
<td>1</td>
<td>SET EMULATOR TO EMULATE MODE</td>
</tr>
<tr>
<td>2</td>
<td>LEAST SIGNIFICANT BYTE OF ADDRESS</td>
</tr>
<tr>
<td>3</td>
<td>MOST SIGNIFICANT BYTE OF ADDRESS</td>
</tr>
<tr>
<td>4</td>
<td>WRITE DATA TO FIRST MEMORY</td>
</tr>
<tr>
<td>5</td>
<td>WRITE DATA TO SECOND MEMORY</td>
</tr>
</tbody>
</table>

BASIC OPERATION
The operation of the emulator can be summarized by the following four steps:
1. Set emulator to "load" mode.
2. Load into the emulator the data that would normally reside in the EPROM.
3. Set the emulator to "emulate" mode.
4. Reset and run the target system.

To perform step 1, a byte is output to port 0. The actual data sent is not important, the only thing that matters is that port 0 be addressed. This step disconnects the emulator from the target and connects the memory to the S-100 bus.

In step 2, for each byte of data you want to load into the memory, three bytes must be output to three ports. First the address of the byte as it would reside in the EPROM must be sent. This is done by outputting the low byte of the address to port 2, and the high byte of the address to port 3. The order in which this is done is not important. Then the actual data byte must be sent to either port 4 or 5 depending on which of the two memories you want to access.

Step 3 is similar to step 1 and is performed by outputting a byte to port 1. Again, the actual byte is not important as long as port 1 is addressed. This step disconnects the emulator's memories from the S-100 bus and connects it to the target.

SOFTWARE CONSIDERATIONS
Listing 1 is a short Z80 sample program for loading three bytes into the first memory. Let us assume that the program to be loaded into the emulator resides at host address 1000H and is to be loaded into the first EPROM starting at EPROM location 200H.

A more complete program appears in listing 2. It was written in Z80 assembly language and assembled with the Microsoft M80 assembler. Any monitor that can out-
HARDWARE CONSIDERATIONS

The emulator board is addressed by the S-100 host through the first six of eight consecutive output ports. Since the board ignores the last two ports, they could be used by another board in the system. The emulator board contains a six switch DIP module. Switches 1 to 5 set the high-order bits of the port address.

The address and data lines to the target are brought through the cable for memory 1 only. This is why the two EPROMS in the target must have common address and data buses. Memory-1 shares its address and data buses with memory-2. If each memory was connected to the address and data buses of the target, these buses would have two cables connected to them instead of one. This would mean twice the loading and much more noise on the buses, as cables are actually antennae capable of picking up stray electrical noise and bringing it to the bus.

The cable for memory-2 carries only the control signals for chip select. Although both cables must have a ground connection, the power line, Vcc, is not needed since the board receives power from the S-100 bus. If only one EPROM is being emulated, memory-1 should be used and memory-2 should be turned off. This is done by closing DIP switch 6.

CONSTRUCTION

The circuit contains only thirteen ICs and can easily be wire-wrapped. Each IC should have a 0.01-uF capacitor connected across 5V and ground. Assemble the two cables to emulate the EPROMs you want to emulate. One end of the cable is plugged into the emulator, while the other end is soldered to a header and plugged into the EPROM socket.

Cable 1 can be wired point-to-point or it can be assembled with a mass-termination header. Cable 2 is best wired point-to-point since it only needs three wires as shown in Table 1. Table 2 lists the point-to-point wiring for cable 1 for the different EPROMs.

Robert A. Rioja is an electrical engineer and educator with over 15 years in electronics and computers. An active consultant, he is the founding president of Superior Control Systems. He has designed and built custom-made computers used by several government agencies, as well as private industry. Most of his designs are based on the S-100 bus and a proprietary CP/M-compatible, multi-processing operating system. He can be contacted at 166 Wright St, Staten Island, NY 10304, (718) 442-0665.
Increase PC / XT Compiler Throughput 300%

Benchmarks using MS Pascal 3.3 show compile & link times reduced from 1:16 to 0:23!

Nucleus Overthruster increases clock speed from 4.77Mhz to 7.38Mhz, providing 60% more throughput for CPU bound programs. No slot required. High/Normal speed switch with indicator light. CPU reset button. Only $295.

Talltree JRAM-3 provides 2MB RAM-disk in one slot. Increase throughput 1 to 5 times for disk bound programs. Conforms to Lotus/Intel extended memory spec. Includes RAM-disk, Spooler, and Jet-copy software. Only $649 (fully loaded) -- save $50 from list.

Order both for only $895 -- save $99 from list!

TSF Source Locator provides cross-references showing how data/procedures are used. Combine any number of Assembler, Basic, C, and Pascal modules into one report. Requires 256K, MSDOS 2+. $75.

Call Toll Free
For orders or information.
Ask for Operator 2053
800-543-6277
CalHi 800-368-7600

TSF • Dept. B-1 • 649 Mission Street
• San Francisco, CA 94105

The Software Family: Programmer Productivity
<table>
<thead>
<tr>
<th>CONNECTOR</th>
<th>2708</th>
<th>2716</th>
<th>2732</th>
<th>2764</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>NC</td>
<td>NC</td>
<td>NC</td>
<td>NC</td>
</tr>
<tr>
<td>2</td>
<td>NC</td>
<td>NC</td>
<td>NC</td>
<td>NC</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>9</td>
</tr>
<tr>
<td>10</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>10</td>
</tr>
<tr>
<td>11</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>11</td>
</tr>
<tr>
<td>12</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>12</td>
</tr>
<tr>
<td>13</td>
<td>11</td>
<td>11</td>
<td>11</td>
<td>13</td>
</tr>
<tr>
<td>14</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>14</td>
</tr>
<tr>
<td>15</td>
<td>13</td>
<td>13</td>
<td>13</td>
<td>15</td>
</tr>
<tr>
<td>16</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>16</td>
</tr>
<tr>
<td>17</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>17</td>
</tr>
<tr>
<td>18</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>18</td>
</tr>
<tr>
<td>19</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>19</td>
</tr>
<tr>
<td>20</td>
<td>18</td>
<td>18</td>
<td>18</td>
<td>20</td>
</tr>
<tr>
<td>21</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>21</td>
</tr>
<tr>
<td>22</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>22</td>
</tr>
<tr>
<td>23</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>23</td>
</tr>
<tr>
<td>24</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>24</td>
</tr>
<tr>
<td>25</td>
<td>23</td>
<td>23</td>
<td>23</td>
<td>25</td>
</tr>
<tr>
<td>26</td>
<td>NC</td>
<td>NC</td>
<td>NC</td>
<td>NC</td>
</tr>
</tbody>
</table>

Table 2

<table>
<thead>
<tr>
<th>CONNECTOR</th>
<th>2708</th>
<th>2716</th>
<th>2732</th>
<th>2764</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>NC</td>
<td>NC</td>
<td>NC</td>
<td>NC</td>
</tr>
<tr>
<td>2</td>
<td>NC</td>
<td>NC</td>
<td>NC</td>
<td>NC</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>9</td>
</tr>
<tr>
<td>10</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>10</td>
</tr>
<tr>
<td>11</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>11</td>
</tr>
<tr>
<td>12</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>12</td>
</tr>
<tr>
<td>13</td>
<td>11</td>
<td>11</td>
<td>11</td>
<td>13</td>
</tr>
<tr>
<td>14</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>14</td>
</tr>
<tr>
<td>15</td>
<td>13</td>
<td>13</td>
<td>13</td>
<td>15</td>
</tr>
<tr>
<td>16</td>
<td>14</td>
<td>14</td>
<td>14</td>
<td>16</td>
</tr>
<tr>
<td>17</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>17</td>
</tr>
<tr>
<td>18</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>18</td>
</tr>
<tr>
<td>19</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>19</td>
</tr>
<tr>
<td>20</td>
<td>18</td>
<td>18</td>
<td>18</td>
<td>20</td>
</tr>
<tr>
<td>21</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>21</td>
</tr>
<tr>
<td>22</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>22</td>
</tr>
<tr>
<td>23</td>
<td>21</td>
<td>21</td>
<td>21</td>
<td>23</td>
</tr>
<tr>
<td>24</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>24</td>
</tr>
<tr>
<td>25</td>
<td>23</td>
<td>23</td>
<td>23</td>
<td>25</td>
</tr>
<tr>
<td>26</td>
<td>NC</td>
<td>NC</td>
<td>NC</td>
<td>NC</td>
</tr>
</tbody>
</table>
PUBLIC DOMAIN SOFTWARE

Our public domain software library:
- now in excess of 1000 disks
- over 30,000 programs
- includes the complete CP/M UG, SIG/M, PCSIG libraries
- 20-40 new volumes added per month
- supporting over 200 CP/M computers (including Northstar and Apple) and all major IBM type computers (including DEC)

Available In One of Two Ways

Via Modem
- online directly from one of our 11 remote RCP/M, RBBS or FIDO Host systems
- 24 hours per day, 7 days per week
- limited access, restricted membership
- access fee only $35 Can ($30 US) covers one year access
- supporting 300/1200 and 2400 baud modems

By Mail
- all software available by mail on over 200 different disk formats
- priced by disk, from $10 Can ($8 US) per disk
- all major user groups available

Canadian Users
- a full line of commercial software available
- 12 different brands of modems including the US Robotics 300/1200 baud Password and the US Robotics 300/1200/2400 baud Courier
- Echelon and Night Owl software available directly from us

For a free minicatalog, call (416) 239-2835 or write:

CANADA REMOTE SYSTEMS LTD.
4691 Dundas Street West
Islington, Ontario, Canada
M9A 1A7

For a free minicatalog, call (416) 239-2835 or write:

CANADA REMOTE SYSTEMS LTD.
4691 Dundas Street West
Islington, Ontario, Canada
M9A 1A7

Micro/Systems Journal January/February 1986
STARTER SYSTEM SPECIALS

WOW! Z-148-21
A drive, 128K
FCC-SYS148-02 only $1195.00
148, 2 drives
640K RAM
1220 or 1230 Monitor
FCC-SYS148-02 only $1595.00
158, 2 drives
640K RAM
1220 or 1230 monitor

WOW! POWERFUL WINCHESTER SYSTEMS

FCC-SYS158-11
only $2475.00
158, one floppy
11 meg half high winchester
640K RAM
1220 or 1230 monitor
With ZVM-135 color monitor only $2899.00
FCC-SYS158-25
only $2849.00
158, one floppy
25 meg half high winchester
640K RAM
1220 or 1230 monitor
With ZVM-135 color monitor only $3299.00

WOW! FREE FRAMEWORK!
with purchase of 36 meg SuperSystem!
Z158, one floppy
36 Meg winchester,
full height, 40ms access time
640K RAM
ZVM-135 color monitor
only $3875.00

AUTO CAD SYSTEM
Complete workstation with plotter and digitizer

Z-158 8 Mhz
one floppy, 25 meg H/H internal
8087-2 8 Mhz
ZVM-135 monitor
AutoCAD w/ ADE-1, 2, & 3
GTCO 12 x 12 4 button digitizer
ZVM-136 w/ Techmar Graphics Master
only $9599.00
640 x 400 Color, add $785.00

FIRST CAPITAL — THE ZENITH EXPERTS!
Order direct from First Capitol Computer.
All prices pre-paid (cash, check, Mastercard or Visa).
Add 2% Shipping charge.
Missouri residents add 6.225% sales tax.
Call for purchase order terms, government, and quantity pricing.

WOW! Printers with above systems:

Epson LX-90, only $259.00
Epson FX-85, only $399.00
Toshiba P-1340, only $596.00
Epson FX-185, only $549.00
Toshiba P-351 only $1225.00

We also have the two hot new machines from Zenith, the 171
and the 200! Call for our latest price and specials, as well
custom configurations on the 200, and the new 158.

Call for pricing on tape backup systems and cartridge winchester
Intersystem Processing

by Hank Volpe

Ever wondered what makes a terminal emulator program go? Some of these programs can be rather expensive, but if you know a few hardware and software tricks, you can open up a new horizon for yourself and your computer system.

My "computer-room" consists of an IMSAI-8080 (which is more of a North Star Horizon) using the CP/M 2.2 operating system and an IBM-PC. My interest in developing emulator software emerged quite suddenly when my 10 year old Hazeltine 1500 went up in a pretty blue puff of smoke one winter evening. I figured this was the perfect opportunity to try to make the PC run the S-100 system. With a little more work, I wrote this emulator to do file transfers between the two systems, and some multiprocessing. The program consists of two parts. Part one is the Hazeltine emulator. This section handles all the communications and hardware "handshaking" between the two systems. Part two is the file transfer section which requires a program be loaded in the S-100 system computer in order to make the file transfer process as simple as possible.

Before we get started on the software, it is important to explain some of the hardware involved and how the software routines control the hardware.

LOOKING AT THE HARDWARE

Although parts of this section deal specifically with my IMSAI system, you should realize that all serial data equipment communicates in a similar manner. I know that the North Star Horizon computers and many other S-100 systems use a similar configuration. But even if your hardware is different, it still uses the method described here to communicate. For a device to talk to another device, it must know three things: 1) that the device it is trying to talk to is on-line, 2) that the device it is trying to talk to knows that it is on-line, and 3) when it can or can't transmit or receive data. This is not very complex or exotic, but it does require a peek into how a computer communicates with a terminal.

A Terminal Emulator & File Transfer Program Between CP/M & MS-DOS Systems

Most S-100 computers use a serial interface to CRT's. My IMSAI is equipped with an IMSAI S-I/O board. This board supports two serial I/O channels using twin Intel 8251A USARTS. Each serial channel uses 2 ports. One port, in each channel, is used for Data I/O, and the other to program the 8251A and read its line status.

When the IMSAI is cold-booted, a terminal initialization routine in the BIOS module programs this chip for the correct start/stop bits, parity, and baud rate factor. Whenever a program or operating system requests communication with the system terminal, the CPU reads the program/channel port to test the status of certain data line conditions. If you are familiar with implementing CP/M on your system, you have probably been exposed to the software polling techniques that examine the line status registers to see if data is ready to be received or transmitted.

In the PC/terminal communications occur in a similar manner, except for the fact that the keyboard is not an external device, but an interrupt driven device that uses program memory for its buffering and a memory mapped display for its visual display. Therefore, unless already furnished, you will have to buy a RS-232 Asynchronous Adapter. Some multifunction boards already come with a serial port, or you can buy a stand-alone unit for under $100.

Although the hardware itself is installed simply by inserting the board in the PC and hooking up the cable to the CRT port of the IMSAI, the way the software controls this hardware needs to be understood. "Handshaking," i.e., communications between the terminal and the computer are sometimes visible to the software (as in CP/M BIOS modules) or invisible (done by the terminal hardware). If a serial port is linked to a modem, a standard group of control signals are used which inform the modem that the computer wishes to transmit or is ready to receive data over the serial link. Serial devices communicate using the standard modem handshaking signals; DSR (Data Set Ready), DTR (Data Terminal Ready), RTS (Request To Send), and CTS (Clear To Send). When these signals are used by serial data terminals, they are used in a slightly different manner.

DSR and DTR are considered as one pair of handshaking signals, RTS and CTS as another pair. When used with a modem, a terminal will receive the Data Set Ready (DSR) signal from the modem as an indication that the modem has received a Data Terminal Ready (DTR) signal from the terminal. When used with a terminal, the DTR output of one device is connected to the DSR input of the other. This satisfies conditions 1 & 2 above.

Next is the RTS/CTS signal pair. When used with a modem, the Clear To Send (CTS) signal is supplied to the computer to indicate that it is permissible to begin the transmission of a message. When used with a terminal, the Request To Send (RTS) output signal of one unit is connected to the Clear To Send input of the other.

The thing to watch out for here is exactly which pair in your computer handle the "handshaking." There is no reason why the DSR/DTR pair or the RTS/CTS pair separately or together could not handle the handshaking process. There are several ways to find out.

First, if you are good at reading logic diagrams, check if a valid RTS signal is needed to enable your S-100 USART. This is the way my 8251A USARTS operate. Whenever a valid CTS signal is received, it enables the 8251A to transmit. Your system, however, may look for a valid DSR/DTR handshake instead.
If you can’t read a logic diagram, take a look at the CP/M BIOS module for your S-100 computer. Most vendors provide a BIOS listing. When the BIOS performs its terminal polling routines, look to see if it “masks” for a valid DSR signal (comments regarding the mask should be either in equates or in the comments of the assembler routine). If no test is made, then your system probably uses RTS/CTS. If you really want to be safe, write the software to control both!

THE PC CONTROLLING PROGRAM

When used as a terminal emulator, the PC replaces my existing terminal. Although it is a little like using an M-16 to go rabbit hunting, it allows me to continue to run and develop programs on the IMSAI that for reasons of specific hardware needs cannot be run on the PC.

To exchange files requires both systems to run a file exchange program. The goal of this part of the program is to make the exchange as invisible to the user as possible. All the user should have to do is select which way the files are to be transferred, type in the file name, and then sit back and watch the show!

Now that we have mapped out what to do, how its done, and what we need, it is time to take a look at the actual program and how it works.

The PC program is called COMM.COM (no puns intended). It is written in Turbo Pascal. I selected this as the language (over an assembler) for two reasons: 1) Turbo Pascal allows the programmer access to the computer hardware in a manner similar to an assembler while offering a high-level file access system that would make it easier to do file exchanges, and 2) I have both Z-80 and 8088 versions of Turbo Pascal in the IMSAI and PC computer systems. If you feel that you would be more comfortable using an assembler, you can easily translate the procedures dealing with communications between the two systems.

The CON$T section of the program equates program names with hex values (similar to an assembler). If you are using a system with different base or register addresses, just substitute the correct hex value. If your system is a PC compatible, check your system Technical reference manual and make sure that the base address of the COM1 port is 03F8.

The type declaration describes a record called regpack. When a variable is assigned the type regpack, it can be used in the Turbo Pascal interrupt procedure. This procedure allows access to the MS-DOS and ROM-BIOS interrupt vectors.

In order to keep the program as universal as possible, all access to system function calls (except window calls) are done through the MS-DOS interrupt 21. There are several minor exceptions. Procedure Setup programs the IBM-PC Async adaptor using ROM-BIOS interrupt 14. Procedures that use windows implement them using Turbo Pascal window procedures. References to Background and Text colors use Turbo Pascal procedures.

If your MS-DOS computer is not compatible with Interrupt 14, programming the Async board could be a problem. You can solve it by either setting up the board with a communications program before running COMM.COM, or finding out what the codes would be to set up your Async board and placing them in the Setup Procedure of the program. If your MS-DOS computer is not compatible with Turbo Pascal windows, you can delete any references to windows and the procedures that setup status windows for file transfers. If your MS-DOS computer is not compatible with Turbo Pascal color procedures, you can eliminate all of them. This is also possible for those who do not have a color display. If left in, there is no problem with running on a Mono display, but if for some reason memory is a problem, you can save a little on the text and code that is generated. These are all of the problems that I can see if you are using a more "incompatible" PC-compatible.

THE EMULATOR - KEY TO THE PROGRAM

I have been told that one of the reasons that programs like this are hard to write is because it is hard to make a computer act like a terminal. I feel this is partially true, because most programmers don’t try to understand how the hardware in the terminal operates. If you sit down and block out exactly what goes on, you will see that it is really quite easy.

A terminal spends 99% of its time looking for an input. If it doesn’t see one, it then checks to see if a key has been struck. If so, it transmits it. If the terminal receives a character (and it is a “smart” terminal), it checks to see if it has been sent a command (i.e. clear screen) from the computer system. If it has, it performs the function. If it hasn’t, then it assumes the character received is to be displayed, and it does so.

To implement this from software, especially from a high-level language, is very tricky. You might try to write machine-code routines if you like, however, this program is able to do all direct I/O controls using conventional Turbo Pascal procedures.

The main program section begins with a menu. You select to use the program for either file transfers or as a terminal to run the slave computer system. Since we are looking at the Terminal emulator, we will select option #1.

Selecting option #1 sets up the terminal emulator mode. The first thing the program does upon entering the emulator is send a control-C character over to the slave system. This serves two purposes. First, it performs a CP/M warm-boot. Second, if a program (such as the file transfer program XFER.COM) was running, it stops that program and returns the slave to the CP/M command mode. From that point on, the program simply uses two procedures, receive and transmit, to control the slave system. Procedure RECEIVE simply calls another procedure named IMSAI. IMSAI is a look-forence routine that scans the Async Modem control register to see if the slave is on-line, and if so, scans the Line Control Register to see if a character has been received. If a character was ready, IMSAI performs two important routines. First, it drops the RTS line which prevents the slave computer from transmitting any more data until the Master is ready. This little trick allows the program to operate without the need for a communications buffer area in the Master Computer system’s memory. Secondly, procedure IMSAI reads the Async adaptor’s base port and gets the data that was transmitted. If a character was received, IMSAI sets RXRDY true.

When the program returns to procedure RECEIVE and RXRDY is true, it checks to see if the code that was received was a lead-in code for the implementation of the Hazeltine 1500 remote commands, CONTROLCODE, which is a local procedure to Receive, is the only hardware dependent module in the program. If, for example, you wish to emulate another type of terminal, then substitute its lead-in code and change the values scanned in procedure CONTROLCODE to make your emulator work. Before RECEIVE returns to the main program, it repeats to make sure no other characters are ready to be received. When the slave has finished transmitting, the program then branches to see if a key was struck on the Master computer.

Procedure IBMKEY detects if a key was struck by using MS-DOS function call #6. This is necessary because it is the only character input routine for the PC that does not detect a control-break
Z SETS YOU FREE!

Z Operating System, an 8-bit OS that flies! Optimized HD64180/Z80 assembly language code — full software development system with proven linkable libraries of productive subroutines — relocating (ROM and RAM) macro assembler, linker, librarian, cross-reference table generator, debuggers, translators and disassemblers — ready to free you!

High performance and flexibility! Productivity results from dynamically customized OS environments, matching operator, tasks and machine.

Real-time control kernel option allows quick software development for industrial control applications, other tools and utilities for office desk-top personal computing functions, local area networks to Ethernet, AppleTalk, Omninet, PC-Net (Sytek) — from micro to mainframe command, control and communications. Distributed processing application programs are easily developed.

• Extreme organizational flexibility, each directory another environment
• Multiple Commands per line
• Aliases (complex series of commands known by simple names) with variable passing
• Named Directories with absolute password security
• Full-screen command line editing with previous command recall and execution
• Shells and Menus Generators, with shell variables
• Command-file search Paths, dynamically alterable
• Screen-oriented file manipulation and automatic archiving and backup
• 512 megabyte file sizes, 8 gigabyte disks handled
• Auto disk reset when changing floppy's
• TCAP database handles characteristics of over 50 computers and terminals, more easily added
• Free-structured online help and documentation subsystem
• 76 syntax-compatible support utilities

Your missing link has been found — Z! Now fly with ease! Fast response, efficient resource utilization, link to rest of computing world — shop floor to executive suite, micro to corporate mainframe. Call 415/948-3820 for literature.

Echelon, Inc. 101 First Street • Suite 427 • Los Altos, CA 94022 • 415/948-3820

FILE EXCHANGE • MASTER TO SLAVE COMPUTER

There are several ways (file capture, memory buffering, etc.) to do one-way transfers from the S-100 system to the PC. However, two-way traffic requires both systems to create and save files and signal the other when they are done.

The easiest way I found was to write another program for the IMSAI called XFER.COM. This program would be loaded by the PC file transfer sections and remain resident until the Terminal Emulator section of the program was given control. The basic function of XFER.COM would be to see which way the PC was requesting file transfer. If it was going from the IMSAI to the PC, the IMSAI would find the file, flag the PC that its ready to send and wait for the PC to give permission to transmit. If file transfers were from the PC, the IMSAI would create a file of the correct name and signal the PC to transfer, and signal the PC when it had finished.

The PC is also required to do the same functions. These, however, are incorporated inside of COMM.COM and do not need to be loaded externally.

When you enter the IMSAI-to-PC loop, the program opens two windows. One reflects what is going on with the PC, the other displays what is going on in the S-100 system. The program queries you for a file name and allows one disk user to put another disk in at file-transfer time. From here on out, the two computers handle all file transfer procedures.

Using procedure LOADISP, the PC commands the IMSAI to load the XFER.COM program. When this is loaded, the IMSAI transmits the ASCII character ‘#’ back. If a ‘?’ is received, it indicates the CP/M system could not find the file, and the program is aborted.

When XFER.COM is loaded, the PC sends a ‘R’ to the IMSAI, instructing it that the file transfer will be an IMSAI system ‘read’ of the terminal. Again the IMSAI signals its ready with the ‘#’ symbol. At this point, if the user had indicated that the program was on another disk, the procedure allows you to change disks on the S-100 system. When you have done so, or if the program was on the same disk as XFER.COM, the actual file transfer begins.

The PC sends the filename to the IMSAI. The IMSAI creates a file and if successful, signals back with an ‘@’ symbol. If there was a problem in creating the file, the IMSAI will send a ‘!’ symbol, which halts the process and return to the main menu.

(control-c) code. This makes it possible to transmit the control-C (warmboot for CP/M) to the slave computer without halting the Master computer’s program. Function call #6 will return with a zero in register AL if no key was struck or with the character in register AL that was entered on the Master computer.

If a character was detected, TXRDY is set true.

If TXRDY is true, the main program now checks to see if the user desires to break the link between the computers. This is necessary because control-C is masked out to allow it to warmboot the slave. Therefore some kind of control codes are needed to stop the program when it is running in the emulator loop. I picked control-u then control-f as my break code. Again, the selection is entirely up to you. If a control-u was pressed on the keyboard, the program checks if the next code is a control-f. If so, the program returns to the Main menu. However, if any other key has been pressed, the program calls procedure SEND. Procedure SEND checks to see that the slave is on-line. It checks for a DSR signal, and if valid, sends the character to the slave computer.

Before leaving the emulator, I would like to explain why there are two variables, BUFFER and IBM DATA, declared for the character handling routines. BUFFER is declared as a byte. IBM DATA is declared as a character with the same address as BUFFER. This was done in order to allow the program to sometimes treat the variable as an eight bit value and at others as an ASCII character. If you write programs that need to do special things like the emulator, you might want to try this to allow you some programming flexibility.
Next, the PC opens the desired file and transfers the file until a Control-Z (hex lA) is read, indicating end-of-the PC file. When detected, the PC sends a Control-Z to the IMSAI, which upon reception closes the file and returns to the XFER.COM menu loop. XFER.COM stays resident until the emulator portion of the program is re-entered.

FILE EXCHANGE - SLAVE TO MASTER
In a similar manner as above, the PC program asks for the name of the file to be transferred from the S-100 system. The PC commands XFER.COM to be loaded (if not already resident in the IMSAI), sends a '\W' telling the IMSAI that it will write the file to the terminal (PC), and sends the name of the file to transfer. When located, the S-100 system sends a '\z' symbol saying its ready to transfer. The PC now creates a file with the desired filename on the PC default drive and signals the IMSAI to start. When the IMSAI sees an end-of-file condition, it sends the PC a Control-Z character. Upon reception, the PC closes the file and returns to the main menu. The IMSAI returns to XFER.COM menu and remains resident.

For those only interested in one-way transfers, I wrote another option of transferring files using the CP/M "TYPE" command. Using this allows transfers without writing or loading XFER.COM on the S-100 system. No handshaking (except terminal hardware control) occurs between the two systems. The PC simply creates a file with the correct name on the PC default disk, commands the S-100 system to "TYPE" the file, and then captures into the file all characters transmitted by the IMSAI up to the CCP "A>" prompt.

THE FUTURE OF THE INTERSYSTEM PROCESSOR
I mentioned at the beginning of this article that this could open up new horizons for you. With a terminal emulator and file transfer framework installed, you could now begin to write multi-system processing jobs, spoolprinters, remote acquisition of data via the S-100 system, and many other tasks that are left to your imagination. You could make the whole program interrupt driven instead of software polled, or make certain multi-system tasks use interrupts to the MS-DOS PC that you are using. I used the S-100 to spoolprint this article while I worked on a spreadsheet on the PC. I hope that you give it a try. You will find it well worth your time!

Hank Volpe has been hooked on computers since the mid '70's when he built his IMSAI S-100 system from a kit. This system now has 56k of RAM (24k built and etched by hand), a Ithaca Inter systems Z-80 card, and a North Star MDS SSDD disk controller. Languages include Turbo Pascal, Fortran, Pilot, North Star Basic, and Microsoft Macro-assembler. The IBM-PC supports a Mono & Color monitor, 256K ram, an Async Adaptor, and its languages are Basic, Turbo Pascal, LOGO, and a Microsoft Macro-assembler. He is Assistant Chief Engineer for WKBW-TV, where he uses computers in a wide variety of tasks from Character Generators and Graphic "Paint-Boxes" to controlling an Automated Satellite TVRO system. Readers with questions about hardware and control systems, color TV, graphic displays, or almost anything electronic, can write to him at WKBW-TV, 7 Broadcast Plaza, Buffalo NY 14202.
These window procedures can be deleted if your MS-DOS system does not support IBM window function calls. If you delete them, make sure all calls for these procedures are also deleted in the source code.

procedure clrwindow;
begin
window(1,1,60,25);
end;

procedure mainwindow;
begin
clrwindow;
window(1,1,60,24);
end;

procedure ibmbanner;
begin
window(1,6,80,14);
end;

procedure imsaibanner;
begin
window(1,14,80,23);
end;

procedure ibmwinfo;
begin
window(1,7,60,13);
end;

procedure imsaibmwindow;
begin
window(1,15,60,23);
end;

procedure center (holder: scr_buf; ypos: integer);
begin
xpos := integer;
ex := integer
end;

procedure online;
begin
if offline in [true] then
begin
window(1,25,80,25);
gotoxy(1,25); textbackground(red); textcolor(white);
clearwindow;
clearscreen;
end;
end;

procedure online;
begin
if offline in [true] then
begin
window(1,25,80,25);
gotoxy(1,25); textbackground(red); textcolor(white);
clearwindow;
clearscreen;
end;
end;

procedure online;
begin
if offline in [true] then
begin
window(1,25,80,25);
gotoxy(1,25); textbackground(red); textcolor(white);
clearwindow;
clearscreen;
end;
end;

procedure online;
begin
if offline in [true] then
begin
window(1,25,80,25);
gotoxy(1,25); textbackground(red); textcolor(white);
clearwindow;
clearscreen;
end;
end;

procedure online;
begin
if offline in [true] then
begin
window(1,25,80,25);
gotoxy(1,25); textbackground(red); textcolor(white);
clearwindow;
clearscreen;
end;
end;

procedure online;
begin
if offline in [true] then
begin
window(1,25,80,25);
gotoxy(1,25); textbackground(red); textcolor(white);
clearwindow;
clearscreen;
end;
end;

procedure online;
begin
if offline in [true] then
begin
window(1,25,80,25);
gotoxy(1,25); textbackground(red); textcolor(white);
clearwindow;
clearscreen;
end;
end;

procedure online;
begin
if offline in [true] then
begin
window(1,25,80,25);
gotoxy(1,25); textbackground(red); textcolor(white);
clearwindow;
clearscreen;
end;
end;

procedure online;
begin
if offline in [true] then
begin
window(1,25,80,25);
gotoxy(1,25); textbackground(red); textcolor(white);
clearwindow;
clearscreen;
end;
end;
procedure imsa;
    [this is a look only once routine]
    (get character from imsa, 8251 usart and flag caller loop if ready)
    (if not ready, rdy returns false)
begin
    rdy:=false;
    port[control]:=000; [tell imsa to send a character]
    modem:=port[modemstat];
    modem:=modem and dtr;
    if modem = dtr then
    begin
        port[control]:=hold; [inhibits imsa from transmitting any more]
        line:=port[linestat]; [tell this character is processed]
        line:=line and dr;
        if line = dr then
        begin
            buffer:=port[base];
            rdy:=true;
        end;
    end;
end; (imsai)

procedure getchar;
    [routine loops waiting until imsa is ready to transmit character]
begin
    rdy:=false;
    while rdy=false do
    begin
        imsa;
    end; (getchar)
end; (getchar)

procedure receive;
    [this procedure contains only terminal dependent data in system]
    [Control codes for Hazeltine 1500 terminal are implemented in]
    [routine Controlcode which is called only if character was transmitted]
    (by imsa) that was = 125 (126 is leading code for control routines)
begin
    getchar;
    if buffer > 125 then controlcode
    else
        write(char(buffer));
    end;
end; (receive)

procedure send;
    [procedure takes character and sends it to imsa]
begin
    port[control]:=hold;
    modem:=port[modemstat];
    modem:=modem and imsa_rdy;
    if modem = imsa_rdy then
    begin
        port[control]:=hold;
        line:=port[linestat];
        line:=line and dtr;
        if line = dtr then port[base]:= buffer;
    end;
end; (send)

procedure loadisp;
    [procedure loads and communicates with isp module on imsa]
begin
    loadstring:=loadstring+char(13);
    imsa;
    if rdy in [true] then receive;
    for x:=1 to length(loadstring) do
    begin
        ibmdata:= loads tr ing(x);
        send: delay(10); [delay so UART can turn around and receive]
    end;
end; (loadisp)

(Main program loop starts here)
begin
    clrscr;
    transfer:=false;
    offline:=true;
    setup;
    buffer:=407;
menu:
    mainwindow;
textbackground(black);
circon;
online;
textcolor(lightblue);
screen:='INTERSYSTEM PROCESSOR PROGRAM';
center(screen,1);
screen:='Version 1.1 ... from Hank Volpe Computers (c) 1985';
lowide;
center(screen,2);
screen:='Select from the following menu';
center(screen,7);
gotoxy(20,9);
write(1) Emulate Hazeltine 1500 Terminal!;
gotoxy(20,11);
write(2) Transfer ASCII files from IMSA to IBM-PC';
gotoxy(20,13);
write(3) Transfer ASCII files from IBM-PC to IMSA!';
gotoxy(20,15);
write(4) Transfer CP/M files to the IBM-PC via TYPE Command';
gotoxy(20,17);
write(5) End Program Loop';
selection:
gotoxy(1,19);clrscr;
write('Enter Selection : '); read(answer);
case answer of
'1': begin
    clrscr;
    goto 1;
end;
'2': goto 2;
'3': goto 3;
'4': goto 4;
'5': halt;
end;
goto selection;

[Main program loop for Hazeltine 1500 emulation]
1:
buffer:=#03; send;
delay(10);
receive;
transfer:=false;
online;

loop:
receive;
ibmkey;
if tready = true then
begin
if ibmdata=char(27) then
[procedure breaks link between computers but does not stop]
[any processing being done by slave unit]
[break codes can be modified to any desired by user]
begin
repeat
ibmkey;
until tready=true;
end;

else
send;
end;
goto loop;

[routine transfers files from IM SA to PC. IM SA must have]
[disk containing INTERSYSTEM software in default drive. If disk]
[change is necessary, program will prompt user when necessary]

[When Xfer.com program is loaded on IM SA, it will signal]
[PC with an '##'. PC will send Filename to IM SA]
[If file is found, IM SA will transmit '##'
[indicating it is ready to send file data. When finished, IM SA will]
[transmit 1AH (Ctrl-Z) saying file has been transferred]

2:
clrscr;
gotoxy(1,1); textcolor(yellow);
write('INTERSYSTEM XFER PROGRAM');
write('Version 1.0.0');
write('Hank Volpe Computers (c) 1985');
lowvideo;

/set up the display windows]/
imbanner; textbackground(red);
clar; textcolor(yellow);
textbackground(red);
write('IBM-PC FILE STATUS');
imwindow; gotoxy(1,1);
write('Enter the name of the file :');
readin(filename);
write;
procedure transfers a file from IBM-PC to IHS to in similar manner as documented above.

(set up the display windows)

[ procedure transfers a file from IBM-PC to IHS to in similar manner as documented above. ]

(cirsr; gotoxy(1,1); textcolor(yellow); writeln('INTERSYSTEM XFER PROGRAM');
   writeln('IBIll-PC to IHS');
   writeln('Version 1.0');
   writeln('Hank Volpe Computers (c) 1985');
   lowvideo;

(set up the display windows)

[lmbanner; textbackground(black);
cirsr; textcolor(yellow);
 writeln('IBIll-PC FILE STATUS ');

[Next send filename and wait for acknowledge ($) or no file (!) ]

delay(10);
loadstring:='$'; loadisp; [send ibm ready]
delay(10);
repeat
   getchar;
   writeln(ibmdata);
   write(disk,buffer);
   until ibmdata = ' $';
close(disk);
   writeln('File Has been transferred');
repeat
   imsa;
   until ibmdata in ["$","!"];
if ibmdata = '! ' then begin
   writeln('ERROR');
goto menu;
end;

[routine transfers files from IHS to PC using type cp/llll command. This frees user from having to have xfer.com present on] [5-100 system computer. This is for one-way transfers only !!!]

cirsr;
gotoxy(1,1); textcolor(yellow);
 writeln('INTERSYSTEM XFER PROGRAM');
 writeln('IHS to IBM-PC');
 writeln('Version 1.0');
 writeln('Hank Volpe Computers (c) 1985');
 lowvideo;

(set up the display windows)

[End send file name and wait for acknowledge ($) or no file (!) ]

[Next, branch into File transfer program on IHS] [If #, then program in loop, if > then back in CP/M ]
delay(10);
transfer :=true;
ONCE
imsaiwindow;
cirsr;
loadstrings:='#'; loadisp; [Send command to send file]
repeat
   imsa;
   write(char(buffer));
   until ibmdata in ['#','>'];
if ibmdata = '>' then begin
   writeln('Program loop terminated');
goto menu;
end;
write('ENTER FILE STATUS');

loadstring:='type '+filename:
loadisp;

(PC now waits for '#' to indicate program is running or ? to indicate Program was not found by CP/M)

imsaiwindow;
gotoxy(1,1); {Now Create file on PC and signal IMSAI ready to transfer}

assign(disk,filename);
write(disk,buffer);
rewrite(disk,buffer); gotoxy(1,1);

delay(10);
repeat
getchar;
loadstring:='previous'+randdata;
write(disk,buffer);
previous:='previous'+randdata;
until loadstring = 'A';
close(disk);
write('File Has been transferred');
repeat
imsa;
until randy in [false];
goto menu;
end.

******************************************************************** LISTING 2 ********************************************************************

program intersystem_processor;
{program must be on default drive of S-100 system when}
{Master MS-DOS computer enters file transfer mode}

{allows 2 way transfer of ASCII files between systems}
{Program written by Hank Volpe Computers (C) 1985 and released to}
{Public domain use}

label 1;

var
  x,y,z : integer;
  disk : text;
  buffer : char;
  ibytes : byte absolute buffer;
  command : char;
  more : boolean;
  answer : char;
  filename : string[12];

procedure sendfile;
begin
  write(9); sift signal PC that sendfile is active
  load(filename);
  load(bios(13));
  ($1-) assign(disk, filename);
  reset(disk); ($1+) if resolult <> 0 then
  begin
    write('I');
    halt;
  end;
  write('@');
  repeat
    read(command);
    until command = 'A';
  delay(10);
  while not(disk in [false]) do
  begin
    read(disk, buffer);
    write(buffer);
    write(disk, buffer);
  until ibytes = $1a;
  close(disk);
  end;
end;

procedure getfile;
begin
  write(9); sift signal PC that sendfile is active
  load(filename);
  load(bios(13));
  ($1-) assign(disk, filename);
  reset(disk); ($1+) if resolult <> 0 then
  begin
    write('I');
    halt;
  end;
  write('@');
  repeat
    read(command);
    until command = 'A';
  delay(10);
  while not(disk in [false]) do
  begin
    read(disk, buffer);
    write(buffer);
    write(disk, buffer);
  until ibytes = $1a;
  close(disk);
  end;
end; (getfile)

{Main menu starts here...note screen printing is not necessary except}
{when debugging in Emulator mode, however # prompt must be}
{printed if all other messages are not}

begin
  clrscr;
  writeln('INTERSYSTEM PROCESSOR');
  writeln('Version 1.0 Imsa/CMS-80');
  writeln('Hank Volpe Computers (C) 1985');
  writeln;
  1:
  writeln('Enter Command #'); {all above except write ('9') can be deleted}
  read(command); command := upcase(command);
  case command of
    'W':sendfile;
    'G':getfile;
  end;
  goto 1;
end.
3002T
5" Floppy/Winchester
10 Cards $565
$100

3307
8" Floppy/5" Winchester
7 Cards $494
$100

3010
MAIN FRAMES & DISC ENCLOSES
FROM $100

LASER 300 DISC/COVERS (not shown)

3916F
5" Floppy $100
2 ea. 5" Winchester $199

3915
5" Floppy $100
2 ea. 5" Winchester $199

1 piece; prices lower in quantity.

**Integrado**

RESEARCH CORPORATION
8620 Roosevelt Ave./Visalia, CA 93291 209/651-1203

Write or call for our brochure which includes our application note:
"Making micros, better than any of' box computer"
YOU CAN'T BUY HIGHER QUALITY, MORE RELIABLE S-100 STATIC MEMORY BOARDS AT ANY PRICE!

STANDARD FEATURES:
- 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

OPTIONS:
- Military MIL-STD-883B type burn in
- Faster ram chips available

NO HIDDEN CHARGES! Performics Inc., prices include dynamic burn in, insurance and UPS ground delivery within the continental U.S.

TERMS: Certified check, Visa/MasterCard
Volume discounts available.

Performics continues to deliver high performance products and excellent service at realistic prices.

S-RAM 128 and S-RAM 256 are trademarks of PERFORMICS INC.

SPECIAL OFFER
ORDER NOW!
(603) 881-8334

11 Morning Dove Rd. • Kingston, New Hampshire 03848
ATTENTION Z-80* CP/M* USERS
NOW A NEW IMPROVED VERSION OF
ZDM AND ZDMZ

ZDM/ZDMZ is a complete Z-80 debugger and monitor designed to replace DDT in CP/M systems for Z-80 and 8080* software development and general program debugging. ZDMZ displays 100% ZILOG mnemonics for disassembled code. ZDM is identical except for displaying extended 8080 mnemonics (similar to TDL). ZDM/ZDMZ include the following features:

- Automatic terminal installation file created
- Will not crash when loading large files
- All DDT commands implemented except "A"
- Ten additional commands including:
  - Alternate register display/alter/exchange
  - Hex and ASCII string search
  - I/O port addressing
  - Trace option to break at branch points only
  - User interrupt control
  - Thirty-four unique command functions
  - Upper/lower case keyboard input
  - Complete manual including ZILOG/ZDM mnemonics table
  - Guaranteed immediate delivery and customer support

ZDM and ZDMZ are available together on standard 8-inch disk and most 5¼-inch disk formats. Entire package still only $50.00 shipped postpaid in U.S. California residents please add 6.5% sales tax.

S.C. DIGITAL INC.
1240 N. Highland Ave., Suite 4 • P.O. Box 506, Aurora, Illinois 60507
Phone: (312) 897-7749

RP/M2™ creates
CP/M® 2.2 compatible IBM PC

1. Remove the 8088
2. Install the NEC µPD70108
3. Boot PC RP/M2

MSDOS and CP/M 2.2 capability coexist in your IBM PC. The NEC µPD70108 CPU chip is a fast 8088 that can also execute 8080 machine code. PC RP/M2 is standard 2.2 compatible RP/M2 with our CBIOS for the IBM PC. System disk with manual and NEC µPD70108 $129. Shipping $5 ($10 non-US)

118 SW First St. - Box G
Warrenton, OR 97146
(503) 861-1765

Micro/Systems Journal January/February 1986
The UNIX File
by Ian F. 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 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!

UNIX Public Domain Software

Finally - an archive of UNIX public-domain software. There are two main 'source' newsgroups on the UNIX network called USENET, namely net.sources and mod.sources. Net.sources is unmoderated, that is, anybody can post anything to it, and they do. Mod.sources is 'moderated', that is, it has a moderator who gets to approve or veto postings. There is a growing trend to use of moderated newsgroups on USENET as the number of sites coming online explodes and the number of ignorant postings continues to rise. Thus postings about your Aunt Petunia's gardening secrets might well appear in an inappropriate unmoderated group, but will be excluded from the moderated newsgroups. Being moderator is strictly a volunteer position, in keeping with USENET tradition. John Nelson moderates the USENET newsgroup mod.sources, and in Volume 3, Issue 26, posted the index of what's available.

John has set up an archive of all the software posted to mod.sources. Unfortunately, archive access is not available automatically - he has to mail files out electronically. He is hoping to have multiple sites maintain and distribute the mod.sources archive in the future. There is currently no mechanism for uucp access to the archive, such as allowing you to give a uucp command to get the programs without manual intervention by John. Some of the new archive sites may be able to provide this service. To retrieve one or more modules from the Archives, John writes, "send me uucp mail - I will mail the files back to you. If multiple modules are requested, then I usually ask for a 9 track tape along with a self-addressed stamped mailer, and a list of what sources you want). I will put the sources onto the tape in 1600bpi tar format. I am willing to consider alternative methods.

"Note that I sometimes get backlogged, but if you send mail to me, and don't get a reply in a reasonable time (say two weeks) then you can assume that either I never received your note, or my reply got eaten (mail by uucp does not feature guaranteed delivery). Also, if you expect a reply via uucp mail, please include a return address starting at some well known site - especially if you are on a local network that uses unusual addressing conventions. More than once I have not been able to reply (and the apparent source sitename was not in the uucp map)."

I think John understates the case. He is doing this strictly on a volunteer basis; if you don't get an answer immediately, please be patient. And if he directs you to another site closer to you, or requests that your site call his site directly, or asks you in return to provide a copy of the full archive to other local sites, please understand that John is doing this on his own and can only do so much. Public domain software efforts must be supported by the public. To get in touch with him, try one of these uucp addresses: [linus, decvax, mit-eddie!genrad!panda!sources-request or seismo!harvard!wjh12!panda!sources-request. If you have a program that you want distributed to mod.sources, send it to ...panda!sources; John will either pass it along to mod.sourses, or suggest an alternative forum if there's a more appropriate newsgroup. Source programs should be of interest to the UNIX community; there are better distribution means for CP/M, DOS and other non-UNIX software - see the columns on SIG/M and PC/BLUE elsewhere in Micro/Systems Journal. To reach John by hardcopy mail, write to him (or send a tape with a return paid mailer and a list of what you want) to John P. Nelson, GenRad MS 6, 300 Baker Avenue, Concord, MA 01742, U.S.A. And thanks to John for undertaking this!

Access to the software archive in Australia is handled by Robert Elz (seismo!'munnari!krex); readers down under should contact him for details. In the future I'll try to list other sites that maintain full archives.

Table One lists the software currently available. Remember that it's public domain; although John Nelson has taken pains to reject obvious trash and copyright violations, it's the user's responsibility to determine the usefulness of the software. You'll see an interesting variety of coding styles, to say the least, and some obvious errors here and there. But if the program you want is listed, go for it.

Join - a mystery command

Join is one of those programs that people are too rushed to read about, but not too rushed to reinvent. It's one of those tools whose idea sometimes seems easy to grasp only after you've already grasped it. But it is worth grasping. A "join" is a technical term in database circles for the operation of selecting all records that have the same field value in two or more tables or files. Say you have two files, one with name and salary and another with name and department, to wit

<table>
<thead>
<tr>
<th>j.salaries:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smith, John:$27,000</td>
</tr>
<tr>
<td>Smith, Mary:$30,000</td>
</tr>
<tr>
<td>Stone, V:$15,000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>j.depts:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smith, John:Physical Plant</td>
</tr>
<tr>
<td>Smith, Mary:Mental Plant</td>
</tr>
</tbody>
</table>

To start with a simple example, let's produce a file showing name, salary, and department. All you have to do is

```
$ join -t $ j.salaries j.depts
```

The input files are stored with ':', delimiting the fields, so the output appears the same way; most any printable character could be used if it doesn't appear in the data. If the fields didn't have any spaces in them, you could use the default separator, which is white space (blanks and tabs).

What happened to $2? Gone. By default, join only outputs the fields that match in both files. This is easy to change with the
1.1 Try it and think about what you see.

(1 part) Try it and think about what you see.

The -0 option, though, it works fine. The -e option takes a list of 'x.y' fields; each x is a file number (1 or 2), and each y is a field number within that file.

This probably should work, but doesn't, at least on my system. When I explicitly mention the fields I want printed (with the -o option), though, it works fine. The -e option takes a list of 'x.y' fields; each x is a file number (1 or 2), and each y is a field number within that file.

After some initialization, the first command puts the full who output into file $F. Then we use cut to get just the first field, i.e., the names of the users currently logged on, into file $H. Then we egrep these users from the password file, and from them exclude internal staff (on our system, all internal staff have "zz" at the beginning of a field in the accounting entry in the password file), and put the word 'paying' at the end of each line. So file $G contains lines like

mary paying
auuer paying

Then we simply join the who output with this file, and get a list of who is paying to use the machine. You can probably see how to modify this command (use '-a') to print everybody’s information along with flagging the paying customers, and adapt it to your own system’s localisms.

Despite its power, join remains underutilized in the UNIX community. Look at the /bin/lsorder shell file for another example. Join is not perfect. It is picky about options: -c can not take a space between the two fields (and the character, while -o must have a space (presumably future versions of join will use the standard getopt argument parser). But it’s a powerful tool for merging parallel files. Knowledge of its use belongs in every UNIX programmer’s tool kit.

That’s all for this month. I’d be glad to get letters and electronic mail on these topics. Suggestions for topics for future columns are always welcome. Cheers!

---

Table 1 - UNIX Public Domain SOFTWARE (mod. sources archive)

<table>
<thead>
<tr>
<th>Volume 1:</th>
<th>Volume 2:</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANSI.C</td>
<td>access (1 part) Kernel Hacks for access control lists</td>
</tr>
<tr>
<td>ANSI.bug</td>
<td>basic (1 part) KSH Interpreter in C (needs work)</td>
</tr>
</tbody>
</table>
| Bourne | bgrep (1 part) Boyer-Moore based grep like program (
| | bm (1 part) much faster Boyer-Moore |
| Small | bmk (1 part) various bm updates |
| UK-1.1 | compress 4.0 (1 part) Compression program better than pack or compact |
| Ximpl.4 | cshar (1 part) update to C shar (in volume 1) |
| bed | cshari (1 part) software "kit" generation script |
| | csharp (1 part) multiple dump per tape utility |
| | csharp2 (1 part) pathalias data |
| | cshar3 (1 part) New Release-mod.map database |
| cshar | progpath (1 part) path optimiser |
| csharX | remote (1 part) Remote tape routines |
| cpg | remote2 (1 part) small patch to remote tape library |
| cpgX | rtar (1 part) Diff to use remote system's tape drive |
| cp | runtime (1 part) runtime memory allocation for multi-dimensional arrays |
| cpgXdep | tools (6 part) Software Tools in Pascal |
| cpgXdepX | uroff (1 part) uroff - nroff unliner |
| eg | uhosts (1 part) uhosts - automatically grab mod.map data for later use |
| egf | wire (2 part) Wire news programs |
| ex | wm (4 part) BSD 4.2 window manager + Patches to Curses |
| diff | Mod.sources: Volume 3: (so far) |
| dynamic | Hey (1 part) Hey! (from This/World, Oct 85) |
| getopt | LaserJet (2 part) Ditroff HP LaserJet drivers |
| lib | bm 1.1 (1 part) Kon Yap's changes to bm (in volume 2) |
| libmg | dircex 4.2 (1 part) dircex (directory hierarchy display) - 4.2 |
| newlib | idiom (1 part) yet another idiom (BSD 4.2 only) |
| news | ieee (6 part) IEEE Floating Point Calculator (Pascal) |
| patch | laserjet (1 part) BSD 4.2 ld printc/comment/spooler for LaserJet printer |
| patch 1.3 | match (1 part) match - faster than bm (VAX only!) |
| pss | mdump (1 part) Revised mdump (multiple dump per tape utility) |
| rc | modnotes (1 part) Notes (1.7 or later) updates - |
| rcf_862 |pretty (1 part) Pretty printer in libp *( |
| rz | match (1 part) match - faster than bm (VAX only!) |
| rz 4.3 | Revized mdump (multiple dump per tape utility) |
| rz | modnotes (1 part) Notes (1.7 or later) updates - |
| rpc | pretty (1 part) Pretty printer in libp * |
| sendmail.cf | wc-colonizer in CLU |
| uucp | selectable C preprocessor |
| uucpman.7 | selectc (2 part) Selective C preprocessor |
| uucpman.V | uucpinfo (2 part) uucpinfo from LOGFILE (awk script) |
| uucpman2 | selectc (2 part) Selective C preprocessor |
| uucpman2 | uucpinfo (2 part) uucpinfo from LOGFILE (awk script) |
| uucpman2 | selectc (2 part) Selective C preprocessor |
| uucpman2 | uucpinfo (2 part) uucpinfo from LOGFILE (awk script) |
| vax | selectc (2 part) Selective C preprocessor |
| vax | selectc (2 part) Selective C preprocessor |
| xfer | selectc (2 part) Selective C preprocessor |
| xfer | selectc (2 part) Selective C preprocessor |
| xfer | selectc (2 part) Selective C preprocessor |
| xfer | selectc (2 part) Selective C preprocessor |
| being (2 part) Getaway patches/Configuration |
| files-Georgia Tech |
CSD PROVIDES SOLUTIONS IN REALTIME...

Intelligent 80186 Based Hi-Power Communications Controllers

Single Channel Controllers
- SDLC/HDLC or ETHERNET
  - RS-422/485 or RS-232
- Multi Channel (star) Controllers
  - SDLC/HDLC or ETHERNET
  - RS-422/485 or RS-232

Super Hi-Speed Communication Subsystems (use above controllers)
- 600 MHZ Fiber Optic Gateway, accepts Multi Protocols and up to 60 10MBit or 100 5MBit inputs
- 300 MHZ Fiber Optic Token Ring LAN, accepts Multi Protocols

Fault Tolerant Dual 8086 Processor computer board - add in parallel

Intelligent Digital Filtered A/D Controller, Programmable

Array Processor - 16 or 32 bit fixed point
- 32 bit floating point
- 80 bit fixed/ floating parallel processing node
- optimized for FFT processing

CSD :: BUILDING AND SUPPORTING FINE CUSTOM HARDWARE/SOFTWARE/SYSTEMS SINCE 1978

If you need a product but cannot find it call us with your requirements and we will provide a solution at production quantity prices.

OTHER FINE S-100 PRODUCTS TO SUPPORT YOU...

Contact us for pricing on any product not listed
... including printers, terminals, enclosures, software, etc

WE WILL INTEGRATE ANY HARDWARE TO MEET YOUR SYSTEM REQUIREMENTS. CALL FOR DETAILS.
WE ARE YOUR S-100 SOURCE ... THE SOURCE IS WITH YOU
UPGRADE YOUR IBM PC!
REPLACE THE MOTHER BOARD
BUILD THE FASTEST CLONE AROUND

WITH THE

CHALLENGER XT-186
SINGLE BOARD
COMPUTER

• Fully IBM PC-XT Compatible
• Four Times Faster
• Two & One Half Times the Memory Capacity

Standard Features Include:
• Intel 80186 CPU
• On-board 640K, parity checked RAM memory
• 8 slots for I/O Interface Board
• 4 channel 8237 DMA
• 8 channel 8259 Interrupt
• 3 channel 8253 Timer
• IBM PC-AT-like Bus with true 16 bit data path
• Optional 8087 Numerical Processor support

The Challenger XT-186 offers full IBM PC-XT compatibility. It supports industry standard operating systems: MS/DOS, PC-DOS 1.0 through 3.1, UNIX and CP/M and runs all existing software with higher performance. The XT-186 mother board can directly replace any existing PC or PC-XT mother board, as it has the same physical dimensions and mounting holes. The Intel 80186 and 640K on-board 16 bit path RAM makes the XT-186 the fastest PC currently available.

Add-On Cards:
Color Graphic Board: RGB and composite, 320X200 Color/B&W, 640X200 B&W only, light pen input......................... $120.00
Multifuntion Card: Real Time Calendar Clock with Battery Backup, 2 RS-232 ports built-in, 1 parallel port.............................. $120.00
Floppy Diskette Controller Card: Double-sided, double density, can support 4 floppy diskette drives ................................................. $70.00
Monochrome Display Card: Supports 256 character codes on 4K ROM, 4K display buffer on static RAM with TTL monochrome video output ................... $110.00
Monochrome Graphic/Printer Card: Two graphic pages with a resolution of 720X348, software supplied to allow the use of the basic graphic command, centronic parallel printer interface ....................... $140.00

MOTHER BOARD, 8MHz, 640K, RAM, 8 I/O slots
$795.00

HOLLISTON COMPUTER
P.O. Box 615, Chepachet, R.I. 02814
401-568-0522

Visa and Mastercard Accepted
Please Add $3.00 Shipping

SM P(-XT·AT, PC-DOS are trademarks of International Business Machines Corp., MS/DOS is a registered trademark of Micro-Soft Corp., CP/M is a registered trademark of Digital Research, Inc.
The Data Base Forum

by Nelson T. Dinerstein

This column will be a regular feature in MIS. It will discuss problems associated with the application of database managers such as dBase-II/III, Framework, and Rbase 5000. Suggestions, comments and questions are welcomed.

In Part I of this two-part series, I introduced rules for speeding up dBase-II applications. In this concluding part, I present five more rules and examples.

CHANGING RECORDS IN A FILE

In the following discussion, the term "a field upon which an index is based" will be used. If a file has a field named FIELDA and there is an index file, say A, that uses this field, then we say that index A is based upon the field, FIELDA. If a file is indexed on the field SSN (for social-security number) with the statement:

INDEX ON SSN TO SSNINDEX

then SSNINDEX is said to be based upon the field, SSN. If a file is indexed upon a combination of fields, then it is based upon each of the fields involved in the combination. For example, if a file is indexed upon the combination of LASTNAME and FIRSTNAME with the statement:

INDEX ON LASTNAME,FIRSTNAME TO NAMEINDEX

then NAMEINDEX is said to be based upon the field, LASTNAME and upon the field, FIRSTNAME. This terminology is introduced here because we will be interested in identifying fields in records that, when changed, affect the indices based upon these fields.

I recommend that you change records in an indexed file with a program that uses "@.. SAY .. GET" statements. This will allow you to display the current field values before you make changes, giving an extra measure of protection. Again, you will want to control the number of indices in use as carefully as possible. When you make any changes in an indexed file, dBase-II will update every index currently in use, whether or not you have made changes to a field upon which an index is based. In many cases, this is a terrible waste of time. I recommend that you use the following:

**RULE 5: When changing the contents of a record in an indexed file, update only the indices that are affected by the changes.**

Rule 5 can be applied in a variety of situations.

1. **Displaying a record on the screen.**
   Never use the "@.. SAY .. GET" statement for the records to be displayed in this type of program. Use only the "@.. SAY .. GET" statement. This will insure that updating of index files will not take place when the user wishes only to view the data.

2. **Changing only numeric fields.**
   Either remove (deactivate) the indices in use prior to the update or use a special program for the update. In the first case, we will need to verify, before the update of the data file, that no fields upon which indices are based have been changed. To do this, use the following technique:
   (a) save the original record values in memory variables,
   (b) capture the new record values in new memory variables,
   (c) compare the new values with the old values,
   (d) verify that there have been no changes to fields upon which indices are based, then
   (e) update the actual record fields using the REPLACE statement.

In the second case, merely write a special program for the update. For example, in an inventory system, one frequently wishes to update the numeric fields, QUANTITY ON HAND, REORDER POINT, and REORDER AMOUNT. A special program to accomplish this will be the fastest and easiest since:
   (a) at most one index is needed to FIND the appropriate record, and this index is easily removed (deactivated) prior to the actual update, and
   (b) you will not need to verify whether changes have been made to fields upon which indices are based, because this type of program does not allow changes to such fields.

3. **Changing one or more fields upon which indices are based.**
   I recommend that you use the following technique, if you have to change fields upon which indices are based:
   (a) identify all of the indices affected by the changes,
   (b) save the current record number,
   (c) activate only the indices affected by the changes,
   (d) GOTO the record whose number you have saved in step (b), then
   (e) update the appropriate fields using the REPLACE statement.

The reason for saving the record number and then using the GOTO is as follows. Whenever you use the SET INDEX statement, either to establish (activate) new indices or to remove (deactivate) existing indices, the pointer to the current record of the file is lost. The value of # is now 0. To get back to the desired record, we use the GOTO. Example 4 illustrates this point:

**Example 4.**

```
1. USE X
2. ERASE
3. STORE * TO MONEY
   0.9.1 SAY "ENTER THE KEY" GET MONEY
4. SET INDEX TO A
5. FIND MONEY
   IF # = 0
      0.9.1 SAY "NO RECORD FOR KEY=" MONEY
      0.9.1 SAY "TYPE ANY KEY TO CONTINUE"
5.1 RETURN
6. END
7. STORE # TO INDIC R
   0.9.1 SAY "GET THE CHANGES HERE"
   0.9.1 SAY "DELETE/MAP INDICES TO USE SET INDEX TO ...."
   0.9.1 SAY "QUIT MENU"...
   0.9.1 SAY "MAKE REPLACEMENTS AS NECESSARY"
```
RULE 6: Never use the ".. SAY .. GET" statement for records when you wish only to view the contents of the record.

RULE 7: If you frequently change a group of or just one numeric record field(s), use a program for this task. Use an index to FIND the appropriate records, but remove (deactivate) the index before the actual update.

RULE 8: When REPLACING fields, make sure that only the required indices for that specific update are activated. The required indices may vary from record to record.

Be careful of the manner in which you use the REPLACE statement. Each REPLACE statement updates both the data file and all of the active index files. If you have N REPLACE statements in a row, it will take N times as long to update the files as one REPLACE statement will. It is therefore recommended that you substitute:

```
REPLACE A WITH A1, B WITH B1, C WITH C1
```

whenever it is reasonable to do so. But be careful, there is a limit on the length of any single statement allowed in dBASE II (254 characters). You absolutely must not exceed this limit. If you try to substitute one REPLACE statement for a group of them and you find that the statement exceeds the maximum allowable length, then you must use two or more statements.

RULE 9: Try not to have a group of REPLACE statements. If several REPLACE statements appear one right after the other, substitute with as few REPLACE statements as you can.

SEQUENTIALLY UPDATING NUMERIC FIELDS IN INDEXED FILES

There are occasions when you will need to access a data file sequentially through an index file and make changes in the data file. For example, in an inventory file you may wish to sort through the file by part number, changing specific values (Example 5). This is the case when you wish to update the QUANTITY ON HAND for a large percentage of inventory parts after performing a physical inventory count. We will use the appropriate index to sort through the file by part number, but we will remove the index before the update and will re-establish both the index and position within the index file after the update.

Example 5.

```
USE INVENTORY INDEX INVFILE
DO WHILE NOT EOF
    READ & TO INVNO
    SET INDEX TO G00D INVNO
    ENTR
    DO WHILE NOT EOF
        READ  & TO OLDAMOUNT
        ERASE & TO NEWAMOUNT
        ENTR
    DO
```

Instant-C™:
The Fastest Interpreter for C

Runs your programs 50 to 500 times faster than any other C language interpreter.

Any C interpreter can save you compile and link time when developing your programs. But only Instant-C saves your time by running your program at compiled-code speed.

Fastest Development. A program that runs in one second when compiled with an optimizing compiler runs in two or three seconds with Instant-C. Other interpreters will run the same program in two minutes. Or even ten minutes. Don't trade slow compiling and linking for slow testing and debugging. Only Instant-C will let you edit, test, and debug at the fastest possible speeds.

Fastest Testing. Instant-C immediately executes any C expression, statement, or function call, and display the results. Learn C, or test your programs faster than ever before.

Fastest Debugging. Instant-C gives you the best source-level debugger for C. Single-step by source statement, or set any number of conditional breakpoints throughout your program. Errors always show the source statements involved. Errors always show the source statements involved. Once you find the problem, test the correction in seconds.

Fastest Programming. Instant-C can directly generate executable files, supports full K & R standard C, comes with complete library source, and works under PC-DOS, MS-DOS, or CP/M-86. Instant-C gives you working, well-tested programs faster than any other programming tool. Satisfaction guaranteed, or your money back in first 31 days. Instant-C is $495.

Rational Systems, Inc.

P.O. Box 480
Natick, MA 01760
(617) 653-6194

Nelson Dinerstein is an Associate Professor of Computer Science at Utah State University and an active consultant in database applications on micros. He has a Ph.D. in mathematics from the University of Utah. And, he is the author of 8 books on dBase-II/III, Framework, and Rbase 5000 applications.
S-100 EPROM PROGRAMMER

EPROM-32

- Held-proven board meets IEEE-600 standard.
- Programs 1K through 32K (byte) EPROMs.
- Textool zero-insertion-force programming socket.
- EPROM is programmed through ID ports and can be verified through 110 ports or located in memory space for verification.
- Programming voltage generated on-board.
- Personality Modules adapt board to EPROMs: PM-1-2508, 2758 PM-2-2716, 2516, 2716 PM-3-2732 PM-4-2564 PM-5-2764 PM-6-2732A PM-7-68764 PM-8-27128 PM-9-27256
- Feature-packed CP/M-compatible control software includes fast programming algorithm.
- One year warranty.

$269.95* (A & T)

MicroDynamics
Corporation
P.O. Box 17577 • Memphis, TN 38187
(901)-755-0619

* Price includes EPROM-32, documentation and two Personality Modules (specify). Additional Modules—$17.95. Central software on 8" SSD—$29.95. UPS ground—$2.00. UPS air—$4.00. COD—$2.00. Foreign add $15.00. VISA & MASTERCARD welcome.


C LANGUAGE
PROGRAMMER'S
REFERENCE CARD

A concise comprehensive reference to C language. Saves time and avoids cumbersome manuals. Organized to keep train of thought while programming and enhance productivity.

Includes: Data Types, Conversions, Statements, Expressions, Operators, Preprocessor Statements, Constants, examples, utility usage, and lots, lots more.

An 8-1/2" x 11" card, on laminated plastic, printed in two colors, printed both sides.

$595.81 (U.S. Canada & Mexico); foreign $3
Shipping & Handling. Send check (drawn on U.S. bank) or postal order to Micro/Systems Journal, Box 1192, Mountainside NJ 07082.

tsp/XEDIT

INTRODUCING a very powerful and affordable version of XEDIT, IBM's mainframe editor for the VM/CMS Operating system, for the IBM/PC and true compatibles:

Syntax is highly compatible with XEDIT, IBM's editor for the VM/CMS Operating System
Most editing commands are supported
Full PREFIX AREA and COMMAND LINE support
A "User Profile" allows you to customize options
Edit files much larger than memory, records up to 500 bytes, FIXED- and VARIABLE-length records
New commands have been added to:
justifying text (left, center and right)
edits in PAGES (pagesize can be changed or turned off at any time)
License to copy for $25 per copy
All future upgrades guaranteed to be of nominal cost

$49.95 + $7 P&H

ORDERS/INFO: Triangle Software Products, P.O. Box 1614, Raleigh, NC 27602-1614 (checks, MC, VISA, PO) ORDERS ONLY: 1-800-824-7866, Operator #103 (except Alaska) (MC, VISA only)

PROGRAMMER'S GUIDE TO CP/M
EDITED BY SOL LIBES

A collection of all the CP/M oriented articles which appeared in Microsystems magazine from Vol. No. 1 (January 1980) up through Vol. 3 No. 1 (February 1982) .... an absolute wealth of in-depth information on CP/M. Covers topics such as: Structure and Format, Interfacing, Utilities and Enhancements, Assemblers, Applications and lots more. Also includes a CP/M Programmer's Reference Guide. 200 pages. $12.95 + $2.00 shipping & handling.

U.S., Canada and Mexico.
$5 for other countries. Send check for full amount payable in U.S. funds by a U.S. bank. We cannot bill or take credit cards.

Send orders to: Micro/Systems Journal, Box 1192, Mountainside NJ 07082.

Micro/Systems Journal January/February 1986
Of course, POWER!™ saves your Bad Disk.
NOW! WINDOWS FOR IBM!

It also does 54 other things to keep your disk in line.

EVERYTHING YOU ALWAYS WANTED TO DO, BUT WERE AFRAID TO TRY

Unlike some utility programs that are a headache to use, POWER! is engineered to spoil you with 55 features, simple and uniform commands, and utter simplicity of use. POWER! automatically alphabetizes and numbers your files. You select by the number and never type file names again. Need to (COPY), (RENAME), (ERASE), or (RUN) programs? Just type in their menu number! POWER! also locks number and never type file names again. Need to sensitive data from prying eyes with (PASS) word protection, or compare files (CHECK). POWER! also makes easy work of patching, (DISPLAY/REPLACE), customizing software (LOAD/SAVE). Among the other commands are (SIZE), (STAT), (LOG), (DUMP), (TYPE), (DUMP), (FILE), (SET), and the CP/M version lets you restore erased files—even when you don’t remember the filename—at a flick of the POWER! (RECLAIM) command. (Still 50 commands to go!) POWER! NOW FOR IBM’s PC-DOS AS WELL AS CP/M

POWER! ONE PROGRAM DOES IT ALL!

You may own a few utility programs for your computer, each with its own commands to memorize. POWER! has all the programs rolled into one 16K integrated package, so you do things you’ve never tried before—every day. Save sensitive data from prying eyes with (PASS) word protection, move a block of memory (MOVE), look for data (SEARCH) or compare files (CHECK). POWER! also makes easy work of patching, (DISPLAY/REPLACE), customized software (LOAD/SAVE). Among the other commands are (SIZE), (STAT), (LOG), (DUMP), (TYPE), (DUMP), (FILE), (SET), and the CP/M version lets you restore erased files—even when you don’t remember the filename—at a flick of the POWER! (RECLAIM) command. (Still 31 commands to go!) Call or write for our free report on truly advanced debugging technology—which explains DSD86’s design and why it is superior to the debugger you are currently using.

Take the DSD challenge: secure a money back guarantee with any of our competitors. Buy both debuggers and use them for a month. Send the one you like best back for a refund.

ONLY $69.95!

Soft Advances
P.O. Box 49473
Austin, Texas 78765
512-478-4763

“Programming for Productivity and Profit”
Please include $4 shipping

GET ORGANIZED!
with TURBO REF
a Cross-Reference and Lister utility for Pascal source programs.

“TURBO REF from Gracon Services produces the most complete, professionally formatted cross-reference listings we’ve seen, with more formatting and control options than you can shake a symbol table at.”
Computer Language Review, Nov. 85

POWER/ONE PROGRAM DOES IT ALL!

We will do whatever it takes to make DSD86 the best debugger available for the IBM PC.

For starters, we have by far the best design, a superior base to build from. While the competition adds new “modes” for every feature, we have a pure, consistent and expandable design. While the competition forces you to accept their particular philosophy, we offer maximized flexibility. If you already have a debugger or are looking for your first, look no further because you can’t do any better. We invite you to compare our debugger, DSD86, with any other on the market.

- Recursive Command Macros & Files
- Bind Macros to any key
- Multi-segment Symbol Support
- Symbolic Register & Stack Displays
- User Customizable Screen Layout
- Superior Mode-less Design
- Source Window for MS Languages
- User Writable Commands & Displays
- Fast Screen Update
- Unique Breakpointing Facilities
- 30 Day Money Back Guarantee

Call or write for our free report on truly advanced debugging technology which explains DSD86’s design and why it is superior to the debugger you are currently using.

Take the DSD challenge: secure a money back guarantee with any of our competitors. Buy both debuggers and use them for a month. Send the one you like least back for a refund.

ONLY $69.95!

Soft Advances
P.O. Box 49473
Austin, Texas 78765
512-478-4763

“Programming for Productivity and Profit”
Please include $4 shipping
Do you know where your bugs are?

This C programmer is finding his bugs the hard way...one at a time. That's why it's taking so long. But there's an easier way. Use PC-Lint

PC-Lint* analyzes your C programs (one or many modules) and uncovers glitches, bugs, quirks, and inconsistencies. It will catch subtle errors before they catch you. PC-Lint resembles the Lint that runs on the UNIX* O.S., but with more features and some awareness of the 8086 environment.

- Full K&R C
- Supports Multiple Modules—finds inconsistencies between declarations and use of functions and data across a set of modules comprising a program.
- Compares function arguments with the associated parameters and complains if there is a mismatch or too many or too few arguments.
- User-modifiable library description files for most major compilers.
- All warning and information messages may be turned on and off globally or locally (via command line and comments) so that messages can be tailored to your programming style.
- All command line information can be furnished indirectly via file(s) to automate testing.

- Use it to check existing programs, programs about to be exported or imported, as a preliminary to compilation, or prior to scaling up to a larger memory model.
- All one pass with an integrated preprocessor so it's very fast.
- Has numerous flags to support a wide variety of Cs, memory models, and programming styles.

- Price: $139.00 MC, VISA
  (Includes shipping and handling) PA residents add 6% sales tax. Outside USA add $10.00.
  Runs under MS-DOS® 2.0 and up, with a minimum of 128K of memory. It will use all the memory available.

Trademark* PC-Lint (Gimpel Software), UNIX AT&T, MS-DOS (Microsoft).

---

NEWI AFFORDABLE ENGINEERING SOFTWARE

<table>
<thead>
<tr>
<th>Product</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>LORIPRO Root Locus</td>
<td>$69.95</td>
</tr>
<tr>
<td>ACTPL-US Active Filter Design</td>
<td>$69.95</td>
</tr>
<tr>
<td>STAP Static Thermal Analysis</td>
<td>$69.95</td>
</tr>
<tr>
<td>MATRIX MAGIC Matrix Manipulation</td>
<td>$69.95</td>
</tr>
<tr>
<td>RIGHTWRTER Proofreader &amp; Writing Analyzer</td>
<td>$74.95</td>
</tr>
<tr>
<td>ACNAP2 AC Circuit Analysis</td>
<td>$69.95</td>
</tr>
<tr>
<td>DCNAP DC Circuit Analysis</td>
<td>$69.95</td>
</tr>
<tr>
<td>SPP Signal/Systems Analysis</td>
<td>$69.95</td>
</tr>
<tr>
<td>PLOTTE Scientific Graph Printing</td>
<td>$69.95</td>
</tr>
<tr>
<td>PCPLOT2 High Resolution Graphics</td>
<td>$69.95</td>
</tr>
</tbody>
</table>

Engineering Professional Software

2200 Business Way, Suite 207 • Riverside, CA 92501
(714) 781-0252
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

TERMINAL EMULATION $49

With XTERM your PC can emulate the DEC VT100/102,
VT52, IBM3101, HP2624, ADDS, and ADM3A. Site
licenses available for all products.

BASICA WINDOWS $49

Enhance your basica programs with a dozen new window
commands for both interpreter or compiler. Add
$399 for complete “ASM” source code.

DES ENCRYPTION $59

Government approved data encryption standard with
compression and telecommunications formatting. Add
$399 for complete “C” source code.

REMOTE OPERATIONS $69

Operate your PC/DOS remotely via telephone using
CONSOLEX. Great for LAN or dBase II applications. Add
$399 for complete “ASM” source code.

HAWKEYE GRAFIX
23914 Mobile Street
Canoga Park, CA 91307
(818) 716-5220
THE SLICER
Real 16 Bit Power on a Single Board—
Featuring the Intel 80186 (C) Step CPU
  • Complete 8 MHz 16-bit micro-
    processor on a 6" x 12" board
  • 256K RAM, plus up to 64K EPROM
  • SASI port for hard disk controller
  • Two full function RS232C serial ports
    with individually programmed
    transmission rates—50 to 38.4K baud
  • Software compatibility with the 8086
    and 8088.
  • 8K of EPROM contains drivers for
    peripherals, commands for hardware
    checkout and software testing
  • Software supports most types and
    sizes of disk drives
  • Source for monitor included on disk
  • BIOS supports Xebec 1410 and
    Western Digital WD 1002 SHD
    controller for hard disks

Fully assembled and tested only $895

THE SLICER 1-MByte EXPANSION BOARD
For expanded memory, additional ports,
and real time clock
  • 1-Mbyte additional dynamic RAM
  • 2 RS232C asynchronous ports
    with baud rates to 38.4K for
    serial communication
  • 2 additional serial ports for asynch
    (RS 232) or sync (Zilog 8530 SCC)
    communication
  • Real Time Clock with battery backup
  • Centronics type parallel printer port

Fully assembled and tested only $750

THE SLICER 1-MBYTE EXPANSION COMBO
The Slicer (without RAM or RAM
controllers) with Full 1-MByte
Memory Expansion
Kit form only $1295

THE SLICER PC EXPANSION BOARD
Gives your Slicer high performance
video capability
  • IBM compatible monochrome video
  • Video memory provides 8 pages of
    text or special graphics capability
  • 2 IBM type card slots for color video,
    I/O expansion, etc.
  • IBM type keyboard port

Fully assembled and tested only $600

All boards available in kit forms
The SLICER Bulletin Board (300/1200 baud)
612/788-9599
Runs MS DOS generic software; PC DOS
program operation not guaranteed

Also available: The µSlicer 186 $700;
8087 Math Co-Processor Bd. (call); 10 MB
Hard Disk $700; W.D. 1002-SHD M.D.C. Bd.
$200, Enclosures, Power Supply, and
Support Hardware.
CP/M 86 $85, CCP/M $250 (Digital
Research, Inc.); MS DOS $175
(Microsoft Corp.)
MasterCard, Visa, Check, Money Order, or C.O.D.
Allow four weeks for delivery. Prices subject to change
without notice.

QUALIFIED DEALER INQUIRIES INVITED

Slicer Computers Inc.
2543 Marshall St. N.E.
Minneapolis, MN 55418
612/788-9481
Telex 501357

SLICER UD

C Programmers:
File System Utility Libraries
Source Code Included, No Royalties,
Powerful & Portable.

BTREE Library
$75.00
  • High speed random and sequential access.
  • Multiple keys per data file.
  • Up to 16 million records per file.
  • Full documentation and example programs included.

ISAM Driver
$40.00
  • Works with the BTREE Library.
  • Greatly speeds application development.
  • Combines ease of use of database manager with flexi-
    bility of programming language.
  • Supports multi key files and dynamic index definition.
  • Very easy to use; fully documented; example pro-
    grams included.

Both products
Are written entirely in K&R C.
Come with complete source code.
Are free of any royalty charges.
+ $3.00 Shipping & Handling Charge.

For more information call:
softfocus
1277 Pallatine Drive
Oakville, Ontario, Canada
L6H 1Z1
(416) 844-2610

TARBEIL'S DUAL
CPU 80186/Z80H
DOES A LOT MORE!

• Runs PC-DOS, CP/M-80, CP/M-86 and
  CCP/M-66
• Floppy interface, 8 and 5-inch
• Two CPU's: 80186 and Z80H, both at
  8 MHz speed
• Two serial ports: RS-232
• PROM: 4 kilobyte with monitor
• Memory Management: 16 Megabytes
• Interrupt Handler
• S-100 bus

Price: $1100 including manual and CP/M 2.2

 Concurrent CP/M-86, CP/M-88, CP/M-2.2, and MP/M-11 are trademarks of
 Digital Research, Inc. PC-DOS is a trademark of IBM.

Micro/Systems Journal January/February 1986
HIGH PERFORMANCE RAM

64K CMOS STATIC RAM
256K CMOS STATIC RAM
LITHIUM BATTERY BACKUP

✓ COMPARE

<table>
<thead>
<tr>
<th>RAM</th>
<th>8-BIT DATA</th>
<th>RAM DESELECTS</th>
<th>WRITE PROTECTION</th>
<th>BATTERY BACKUP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compro</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>Octagon</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>Commodore 256K</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>Commodore 256K II</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>Dynamic</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>BG-Bank</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
</tbody>
</table>

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 696/1-100 • 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.... Battery Backup ......
BG BANK 64S....... Battery Backup ......

call for new lower prices
MEGABYTE MADNESS
FOUR 256S CARDS
$229.50

call for new lower prices

BG COMPUTER APPLICATIONS, 206 Brookside, Bryan, Texas 77801. International orders add 30%. (409) 775-5009

Sidekick for CP/M!
Write-Hand-Man
Desk Accessories for CP/M
NEW! Now with automatic screen refresh!
Suspend CP/M applications such as WordStar, dBase, and SuperCalc, with a single keystroke and look up phone numbers, edit a notepad, make appointments, view files and directories, communicate with other computers, and do simple arithmetic. Return to undisturbed application! All made possible by Write-Hand-Man. Ready to run after a simple terminal configuration! No installation required.

Don't be put down by 16 bit computer owners. Now any CP/M 2.2 machine can have the power of Sidekick.
Bonus! User extendable! Add your own applications.

$49.95 plus tax (California residents), shipping included! Volume and dealer discounts.
Available on IBM 8 inch and Northstar 5 inch disks. Other 5 inch formats available with a $5.00 handling charge. CP/M 2.2 required; CP/M 3 not supported.

COD or checks ok, no credit cards
Poor Person Software
3721 Starr King Circle
Palo Alto, CA 94306
tel 415-493-3735

Write-Hand-Man trademark of Poor Person Software, CP/M trademark of Digital Research, Sidekick trademark of Borland International, dBase trademark of Ashton-Tate, WordStar trademark of Micropro, SuperCalc a trademark of Sorcim.

Announcing a New Product...
OPEN SPOOL PRINTER RIBBON
THAT RE-INKS ITSELF

FOR USERS OF
Okidata and Other
Open Spool Printers
Including Teletype

OUR SELF-INKING DEVICE WILL LAST 15 TIMES THE RIBBON YOU ARE NOW USING.
• 15 TIMES LESS RIBBON CHANGES
• NO MORE MESSY HANDS

MONEY BACK GUARANTEE

CONTROLLED PRINTOUT DEVICES, INC.
P.O. BOX 869, BALDWIN ROAD
ARDEN, NC 28704
(704) 884-9044 • TELEX: (FILMON-AREN) 577454

CONTACT US BY MAIL, PHONE OR TELEX AND WE WILL FORWARD YOU A BROCHURE
NOW AT THE SBC MART

HARD DISKS
- complete 10MB Kit
- special, only $460
- complete internal 1/2-high 10MB kit, incl. controller and cables, special $460
- complete 10MB kit with low-power 3-1/2" half-high drive, fits 5-1/4" slots, $495
- 3-1/2" 10MB Winchester drive alone HD3 just $395
- 20MB half-high drive HD-2 now $455

SOFTWARE SPECIALS
- Microsoft Macro assembler BEST PRICE $107; MS BASIC interpreter regularly $340, now $280; BASIC compiler reg $385, just $265
- Ellis/Utah COBOL or PASCAL compilers, BASIC interpreter, or EDIT each with bound tutorial textbook only $37 each
- Open Systems Business BASIC interpreter, reg $180, Sale $155
- Lattice C compiler (from Lattice) regularly $475, $349
- Borland Turbo Pascal (3.0) list $70, reg $49, now $22

SPECIAL BONUS Norton Utilities Ver 3.0 Reg $90, only $60 w/purchase of any disk drive, IBM PC-DOS 3.1 (the real thing) Reg $65, only $40, w/purchase of motherboard.

MOTHERBOARDS
- full IBM PC/XT compatibility
- 8 I/O slots
- runs IBM's PC-DOS 3.1
- BASIC interpreter available
- great foundation for business or personal system
- 1 year warranty

ADD-ON CARDS
- multifunction 384K, clock, S, P, with OK save $50 Model MFC-4 $145
- monochrome graphics card runs 1-2-3 graphics, w/printer port, 720 x 348, hi-res monochrome TTL output Model MGC-1 now $145
- floppy controller for 1-4 48 or 96tpi drives, w/cable save $25 Model PFC-2 now $74
- color adapter w/light pen port, RGB and composite outputs reg $140 Model CC-1 now just $116
- hard disk controller for 1-2 ST-506/412 drives reg $220 Model HDC-1 only $180

MOTHERBOARDS
- full IBM PC/XT compatibility
- 8 I/O slots
- runs IBM's PC-DOS 3.1
- BASIC interpreter available
- great foundation for business or personal system
- 1 year warranty

ADD-ON CARDS
- multifunction 384K, clock, S, P, with OK save $50 Model MFC-4 $145
- monochrome graphics card runs 1-2-3 graphics, w/printer port, 720 x 348, hi-res monochrome TTL output Model MGC-1 now $145
- floppy controller for 1-4 48 or 96tpi drives, w/cable save $25 Model PFC-2 now $74
- color adapter w/light pen port, RGB and composite outputs reg $140 Model CC-1 now just $116
- hard disk controller for 1-2 ST-506/412 drives reg $220 Model HDC-1 only $180

POWER SUPPLY
- 135 Watts
- side switch
- top quality
- one year warranty
- standard cables for 4 disk drives
- Model PS-135 power supply regularly $129, now only $95

Shipping and handling: drives/motherboards/cases/power supplies $4.50 ea, software $2.50 ea, cards $2 ea, keyboards $4. CA residents add 6% sales tax.

The SBC Mart, 821 Commercial St., P.O. Box 1296, Ridgecrest, CA 93555

Visa/MC/AmEx ORDERS: (619) 375-5744

The ideal foundation for complete business, personal, and workbench PC's. Complete with a manual that will please novice thru expert.

* motherboard w/256k RAM installed save $97 now ONLY $248, w/OK Reg $345, now $220

Csharp Realtime Toolkit

Csharp Realtime Toolkit can do it! Get the tools without operating system overhead. Cut development time with C source code for realtime data acquisition and control. Csharp includes: graphics, event handling, procedure scheduling, state system control, and interrupt handling. Processor, device, and operating system independent. Csharp runs standalone or with: MSDOS, PCDOS, or RT11. Csharp runs on: PDP-11 and IBM PC. Csharp includes drivers for Hercules and IBM graphics boards, Data Translation and Metabyte IO boards, real time clock, and more. Inquire for Victor 9000, Unix, and other systems. Price: $600

Systems Guild, Inc., P.O. Box 1085, Cambridge, MA 02142
(617) 451-8479
The following selected books can be ordered thru Micro/Systems Journal. Prices include shipping (NJ residents add sales tax) with prices shown for U.S., Canada and Mexico. Add $5 for other countries. Send check for full amount payable in U.S. funds by a U.S. bank. We cannot bill or take credit cards. Allow 15-20 days for order to be processed.

Send orders to: Micro/Systems Journal, Box 1192, Mountainside NJ 07082.

C

C: A Reference Manual
by S Harbison & G Steele. Without doubt the best reference manual for C. Don Libes (C Forum author) uses this as his reference. $20.95

Learning To Program In C
by Thomas Plum. Considered one of the best books on the C language. $26.00

C Made Easy
by Herbert Schildt. For programmers looking to make the transition from Basic to C. $16.95

The C Library
by Kris Jamsa. 125 routines from macros to UNIX utilities. $15.95

TURBO PASCAL

Using Turbo Pascal
by Steve Wood. For the new Turbo Pascal (Version 3.0) programmer. Contains lots of code examples. $20.95

8086/8088

The 8086 Book
by R Rector & G Alexy. A guide to 8086 and 8088 hardware and software. Includes examples of interfacing and applications. $19.95

68000

68000 Assembly Language Programming
by G Kane, D Hawkins and L Leventhal. Covers the topic in explicit detail and contains many programming examples. Also useful as a reference...very complete. Even good for programmers new to assembler. $19.95

CP/M

Those wanting to customize their BIOS and add CP/M enhancements will find the following books useful.

CP/M Techniques
by Barbier. $20.95

The Programmer's CP/M Handbook
by Johnson & Laird. $22.95

The following books, by Dave Cortesi, are recommended for those looking to learn more about CP/M.

A Programmer's Notebook: Utilities For CP/M Plus
$18.95

Inside CP/M $27.50

Inside CP/M-86 $18.95

Inside Concurrent CP/M $18.95

OTHER BOOKS OF INTEREST

Getting The Most From Wordstar & Mailmerge
by M David Stone. $15.95

Para Dynamics builds a variety of modern, efficient, trouble-free expandable housing systems for most S-100 BUS configurations. Whether a rack mount, desk top, or stand alone, our patented super-efficient heat dissipation system can end your board-level failures due to high temperatures. Please call today for full details. (602) 991-1600

Para Dynamics Corporation • 7895 East Acoma • Scottsdale, AZ 85260
Sol Libes is doing it again!

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.

TAKE ADVANTAGE OF SPECIAL LIMITED TIME OFFER.
Offer expires December 31, 1985 so hurry and mail your subscription now.

[ ] Send a sample issue (enclose $4.00, $6.00 other foreign)
[ ] YES! Start my subscription!
[ ] 1 YEAR* ....................... $20.00
[ ] 2 YEARS** ................... 35.00
[ ] 1 YEAR (Canada & Mexico)* .... 28.00
[ ] 2 YEARS (Canada & Mexico)** 52.00
[ ] 1 YEAR (other foreign)* ........ 35.00
[ ] 2 YEARS (other foreign)** ..... 64.00
* 6 issues ** 12 issues

Checks must be payable in U.S. funds by a U.S. bank

Name: ____________________________
Address: __________________________
City: ____________________________ State: _______ Zip: _______

make check payable to: MICRO/SYSTEMS JOURNAL
Box 1192
Mountainside, NJ 07092
Some of the issues of the old Microsystems magazine are still available in very limited quantities. They are $4 per copy ($5.50 foreign, cash, Postal Order or U.S. Bank check) including shipping. If ordering 3-9 copies deduct 10%, 10 or more copies deduct 15%. Orders can be combined with back issues of Microsystems for discount. Send to: M/SJ, Box 1192, Mountainside, NJ 07092.

1984

MARCH: MS-DOS Overview Part-1, Enhancing MP/M-II Part-2:adding login and date and time functions, Julian Date Conversions, I/O changes for Fortran-80, CP/M Mass Renaming by Filetype, CP/M BIOS public domain enhancements, Power Failure Backup for S-100 systems; REVIEWS: MS-DOS for CompuPro From Computer House, Lomas Lightning One, Dual Systems SIO4, MagicBind, SAL/80K

JANUARY: Enhancing MP/M - Part 1; Installing MP/M; Add Concurrency to MP/M; Two Users on CP/M; Relocating Assemblers & Linkage Editors - Part 3; S-100 Wait States; REVIEWS: MP/M-8/16, ProComp-8, Paragraphs Game Board, ProLog.

1983

NOVEMBER: Intro to 80286, 68000, and 16032 Microprocessors; Intro to Local Area Networks - Part 2; Extended Memory Management for older S-100 Systems; Notes on Microsoft Forthran-80; Building S-100 Parallel Ports; REVIEWS: CompuPro CPU-68K, System 8/16, Xenith Z-100, Nevada & Ellis Computing Fortran.

OCTOBER: Intro to Local Area Networks, Part-1: Build Low-Cost LAN; Build S-100 Bubble Memory Card; Use Radio Shack Model 100 portable with a CP/M system; Write Menu-Driven Utility for Setting Printer Options; North Star Improvement; True Z-80 Random Number Function; Hide Code in Basic REM statements; Machine Code loader for MBasic; Increase Single-Density Disk Formatting; Relocating Assembler & Linkage Editors, Part-2; Run MX-80 with North Star; User Group Directory; CP/M-86 Versus CP/M-80; REVIEWS: CP/M-86 Versus CP/M-80; REVIEWS: CP/NET, QBAX, S-Basic.

AUGUST: XERA Program; Logging-On CP/M; WordStar Date/Time Patch; Find Location of Variable in NorthStar Basic; Prevent System Crashes During Warm Boot; Enhance Spreadsheet Print Files; Plotting Package-Part 3; Run WordStar under TP/M; 50-line Text Formatter; Using the LU Utility; User Areas under CP/M; REVIEWS: Stiff Upper Lisp, MuLisp-80, Supersoft Lisp, Cromenco C-10, Access Manager, Fancy Font, Computime SBC-880 S-100 card.

JULY: Using RCPMs; RCPM Directory; PIP Data Between Computers; Toward Smarter Modem Programs; Interface MX-80 via Parallel Interface; Digital Audio On CP/M System; Customize CP/M CBiOS; Plotting Package Part-2; REVIEWS: DRI PL/I-86 and PL/I-80, S-100 PMMI MM-VT1.

JUNE: Plotting Package Part 1; Drive HP Plotter; Laboratory Graphics Applications; Console Keyboard interrupts; Customize Wordprocessor Keyboard; WordStar Patch for H-19/Z-19 Terminal; Relocatable Code; REVIEWS: Graffitalk, JES S-100 Graphics Controller, ZCPR2.
HARDWARE FEATURES:
• Lightweight (4.5 lb.) battery powered 3.5 inch micro-floppy system. Attaches/detaches easily.
• 360K bytes/diskette (same as on the IBM PC). 348K per file, due to CP/M.
• Approximately 20 hours on a single battery charge assuming normal disk access.
• 32K bytes additional RAM in the PICDISC - converting your portable to a 64K byte CP/M 2.2 micro-computer.
• Includes all cables, mounting hardware and battery charger.

SOFTWARE FEATURES:
• Filer - A menu driven LOAD/STORE program for transferring individual files (or total memory) to disc.
• CP/M 2.2 Command Line Interpreter system, provides access to all popular CP/M based software (including dBASE II, SuperCalc, and T/Maker).
• Modem - Public domain communications package (both terminal and disc file transfer modes)
• Utilities - DISKCOPY, COPY, FORMAT, and CP/M.

SAVE ON POWERFUL INTEGRATED SOFTWARE - $200 when purchased with PICDISC:
T/Maker includes:
• Word Processing
• Spreadsheets
• Spelling Checker with a 50,000 word Dictionary
• Relational Database Manager

Call Us About Our Free Software Offers
For Literature/Catalog or to order,
Call or write: PERSONAL INTEGRATED COMPUTERS (PIC)
Toll Free 1-800-421-6300
18013 SkyPark Circle, Ste. D., Irvine, CA 92714 (714) 261-0503
CP/M, dBASE II, SuperCalc, & T/Maker are trademarks of Digital Research, Ashton-Tate, Sorcim, & T/Maker respectively.

Microsystems Back Issues continued

APRIL: IEEE-488 Tutorial; Interfacing to Lab Instruments; CP/M-86 System in Lab; Implementing CP/M + Part II; Build Simple S-100 Card Extractor; Macros & Macro-Assemblers; REVIEWS: Pickles & Trout S-100 Controller; CP/M Utilities; Morrow Decision I.

1982
NOVEMBER/DECEMBER: CP/M Vs MS/DOS; CP/M-86 Vs MS-DOS; Intro to ADA Part 2; Virtual Disk for NorthStar; CP/M Program Auto-execute; Macros & Macro-Assemblers; REVIEWS: Janus, Aztec-C, C/80, Morrow S-100 M26 Hard Disk System, Teleram S-100 Bubble Memory Card, Jade S-100 Bus Probe.

1981
JULY/AUGUST: 16-Bit Disk Operating Systems; Input Queuing For NorthStar; Variable Speed Automatic Slow Step; Build S-100 Clock/Calendar Card; REVIEWS: TEC-86 System, Seattle Computer 8086 System, AlphaMicro, Godbout Dual Processor, CP/M-86, Televideo 920-C Terminal.

1982
Micro/Systems Journal will accept Classified Ads. The charge is $6/line (3 lines minimum); 50 characters maximum per line. Three times Frequency $5/line; six times $4/line; non-profit clubs $2/line. Logos, special type, etc., are an extra charge. Check must accompany ad copy. Send to Micro/Systems Journal, POB 1192, Mountainside NJ 07092.

S-100 USERS!!!

S-100 BOARD BANK buys, trades, sells, consigns ancient, modern S-100 cards, etc. SELLERS: send full description, BEST price. BUYERS: savings on S-100 cards from only $49; send $4.37 or more to subscribe to our whimsical, official irregular.

FOR SALE: CompuPro CPU-Z 6MHz $125, Disk 1A and CP/M-80 $475, 64K static RAM $97. ALL equipment is in new condition. James Walker, 1205 North Main, O'Fallon MO 63366.

DISK CONVERSION SERVICE

Over 300 CP/M and MS/Dos (PCDos) formats. Originals and copies returned in 48 hrs. Reasonable rates. Personal service. For information and list of available formats, write: RH Associates, Box 7035, Silver Spring MD 20907, or call (301) 587-6230.

JONTEL BUSINESS SYSTEM

The Jontel Business System is a five journal accounting package based on a dual organ and ad extraordinaire, THE STATE BOARD BANK, Box 344, Mailorder (707 Edge Hill Rd., New Bern, N.C. 28560 Source code available for custom installations, compiled CB80.

$995 LAP-TOP CP/M MACHINE

NEC 8401A with serial and parallel interfaces, 300b modem built-in, battery powered, and 16 line x 80 col. display. Interface Technology, Inc., Box 3040, Laurel MD 20708, (301) 490-3608.

Micro/Systems Journal January/February 1986
INTERCONTINENTAL MICRO is shipping solutions today for all your S-100 BUS 16-bit and PC network needs. Of course, we also offer a complete line of 8-bit and interface/controller products as well as the sophisticated TurboDOS™ multi-user operating system.

SYSTEM FLEXIBILITY. With 8 MHz 16-bit and 4 or 6 MHz 8-bit Single Board Computer (SBC)/Masters; 10 MHz 8086 and 8 MHz 8086 16-bit Slaves; ARCnet™ links between systems; and a complete line of memory, interface and controller boards, INTERCONTINENTAL MICRO SYSTEMS gives you and your customers flexibility—to grow and change.

TurboDOS™ allows you to construct true multi-user systems with CP/M™, PC-DOS™, MPM™, MS-DOS™, and CPM-86™ compatibility. ARCnet™ made possible with our exclusive TurboLAN™ software, make local area networks with up to 4000 users possible—and very cost effective.

SPEED AND POWER. Features are great—but they don't mean much unless they help you do your job better and faster.

CPZ-186—
8 MHz 8086 SBC Master Processor, 2 sync or async serial I/O channels, 20 parallel I/O lines, 256K RAM expandable to 1 megabyte, on-board floppy disk controller.

CPS-186—
10 MHz 8086 SBC Slave Processor, 4 sync or async serial I/O channels, 20 parallel I/O lines, 256K RAM expandable to 1 megabyte.

CPS-16X—
8 MHz 8086 SBC Slave Processor, 256K RAM expandable to 1 megabyte, 2 sync or async serial I/O channels, 20 parallel I/O lines.

LANPC—
Allows IBM PC's™, XT's™, AT's™, or compatibles to integrate into TurboLAN™ ARCnet™ and S-100 bus networks.

LANOIO—
Permits networking of S-100 Bus systems to other S-100 Bus systems, Zenith Z-100™ and IBM PC™ or PC compatible machines.

LANjr™—
Provides cost effective true multi-user PCjr™ networks with simple installation.

GET IT ALL. A comprehensive product line loaded with benefits for you and your customers. Flexibility to grow. And support that sets industry standards. Call INTERCONTINENTAL MICRO SYSTEMS today—we can help.
We've just done something we know you'll like. We've made the SemiDisk far more affordable than ever before. With price cuts over 25% for most of our product line. Even our new 2 megabyte units are included.

**COMPARE WITH THE OTHERS.**

SemiDisk Systems builds fast disk emulators for more microcomputers than anyone else. S-100, IBM-PC, Epson QX-10, TRS-80 Models II, 12, and 16. Up to 2 megabytes per board, standard. Up to 8 megabytes per computer, using only four bus slots, max! Software drivers are available for CP/M 80, MS-DOS, ZDOS, TurboDOS, VALDOCS 2, and Cromix. SemiDisk turns good computers into great computers.

**GREAT NEWS FOR IBM PC AT USERS.**

New MS-DOS 3.0 software drives take full advantage of the 80286's expanded instruction set, moving data four times faster than can be done on PC or XT.

**BATTERY BACKUP, TOO.**

At 0.7 amps per 2 megabytes, SemiDisk consumes far less power than the competition. And you don't have to worry if the lights go out. The battery backup option gives you 5-10 hours of data protection during a blackout. Nobody else has this important feature. Why risk valuable data?

**IT'S EXPANDABLE TO 8 MEGABYTES.**

You can start with as little as 512K bytes, and later upgrade to 2 megabytes per board. At your own pace, as your needs expand.

In an IBM PC, XT, and AT you can have multiple drives on a single system. And the SemiDisk gives you mainframe performance on the top of your desk.

**AND THE BEST NEWS IS SAVED FOR LAST:**

<table>
<thead>
<tr>
<th></th>
<th>512K</th>
<th>1MB</th>
<th>2MB</th>
</tr>
</thead>
<tbody>
<tr>
<td>SemiDisk I, S-100</td>
<td>$695</td>
<td>$1395</td>
<td>$1995</td>
</tr>
<tr>
<td>SemiDisk II, S-100</td>
<td>$995</td>
<td>$1795</td>
<td>$2395</td>
</tr>
<tr>
<td>IBM PC, XT, AT</td>
<td>$695</td>
<td>$1395</td>
<td>$1995</td>
</tr>
<tr>
<td>QX-10</td>
<td>$595</td>
<td>$1195</td>
<td>$1795</td>
</tr>
<tr>
<td>TRS-80 II, 12, 16</td>
<td>$695</td>
<td>$1395</td>
<td>$1995</td>
</tr>
<tr>
<td>Battery Backup Unit</td>
<td>$150</td>
<td>$300</td>
<td>$450</td>
</tr>
</tbody>
</table>

SemiDisk Systems, Inc.
P.O. Box GG, Beaverton, Oregon 97075
503-642-3100