The Official Heath/Zenith Computer Users Magazine

REMark®

October 1990

New ZUG Secretary
Page 5

30% Off All ZUG Software
Pages 2 & 3

HADES II

It's HOTTER than ever! Jam-packed with new features, HADES II still remains the easiest-to-use disk editor ever! Just look at some of the features:

- Sector Display/Editing
- Sector HEX/ASCII String Search
- File Display/Editing
- Physical and Logical Cluster Display
- File HEX/ASCII String Search
- Drive Parameter Display
- 512 MegaByte Drive Size Limit
- File Attribute Display/Edit
- Automatic Erased File Recovery
- Manual Rebuild File Recovery
- Works with Headerless MS-DOS Disks
- PC-Compatible or H/Z-100

HADES II is still only $40, and original HADES owners can upgrade their distribution disk for only $15. Call HUG today at: (616) 982-3463.

October 1990

REMark

The Official Heath/Zenith Computer Users Magazine

PC Compatibles

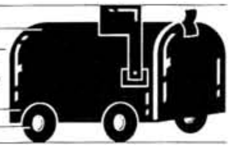
H/Z-100 and PC Compatible

General

Resources

"I WON'T BE CONTRIBUTING DUE TO A COMPUTER ERROR."

MOVING?



Don't Miss A Single Issue!

Let us know 3-4* weeks before you move!

Reader Service No.		*Page No.*
104	*FBE Research Co., Inc.*	*48*
136	*Lindley Systems*	*5*
114	*Micronics Technology*	*31*
117	*Payload*	*26*
193	*Quik Data*	*32*
149	*W.S. Electronics*	*16*

Managing Editor
Jim Buszkiewicz
(616) 982-3837

Software Engineer
Pat Swayne
(616) 982-3463

Production Coordinator
Lori Lerch
(616) 982-3794

Secretary
Lisa Cobb
(616) 982-3463

COM1 Bulletin Board
(616) 982-3956
(Modem Only)

HUG
Software Orders
(616) 982-3463

Hardware Questions
(616) 982-3309

Printer
Imperial Printing
St. Joseph, MI

Contributing Editor
William M. Adney

Contributing Editor
Robert C. Brenner

Advertising
Rupley's Advertising Service
Dept. REM, 240 Ward Avenue
P.O. Box 348
St. Joseph, MI 49085-0348
(616) 983-4550

	U.S. Domestic	APO/FPO & All Others
Initial	$22.95	$37.95*
Renewal	$19.95	$32.95*

* U.S. Funds

Limited back issues are available at $2.50, plus 10% shipping and handling - minimum $1.00 charge. Check HUG Product List for availability of bound volumes of past issues. Requests for magazines mailed to foreign countries should specify mailing method and appropriate added cost.

Send Payment to: Heath/Zenith Users' Group
P.O. Box 217
Benton Harbor, MI 49022-0217
(616) 982-3463

Although it is a policy to check material placed in REMark for accuracy, HUG offers no warranty, either expressed or implied, and is not responsible for any losses due to the use of any material in this magazine.

Articles submitted by users and published in REMark, which describe hardware modifications, are not supported by Heath/Zenith Computers & Electronics Centers or Heath Technical Consultation.

HUG is provided as a service to its members for the purpose of fostering the exchange of ideas to enhance their usage of Heath/Zenith equipment. As such, little or no evaluation of the programs or products advertised in REMark, the Software Catalog, or other HUG publications is performed by Heath Company, in general, and HUG, in particular. The prospective user is hereby put on notice that the programs may contain faults, the consequence of which Heath Company, in general, and HUG, in particular, can not be held responsible. The prospective user is, by virtue of obtaining and using these programs, assuming full risk for all consequences.

HUG

H8 - H/Z-89/90

PRODUCT NAME	PART NUMBER	OPERATING SYSTEM	DESCRIPTION	PRICE
ACCOUNTING SYSTEM	885-8047-37	CPM	BUSINESS	20.00
ACTION GAMES	885-1220-[37]	CPM	GAME	20.00
ADVENTURE	885-1010	HDOS	GAME	10.00
ASCIRITY	885-1238-[37]	CPM	AMATEUR RADIO	20.00
AUTOFILE (Z80 ONLY)	885-1110	HDOS	DBMS	30.00
BHBASIC SUPPORT PACKAGE	885-1119-[37]	HDOS	UTILITY	20.00
CASTLE	885-8032-[37]	HDOS	ENTERTAINMENT	20.00
CHEAPCALC	885-1131-[37]	HDOS	SPREADSHEET	20.00
CHECKOFF	885-8010	HDOS	CHECKBOOK SOFTWARE	25.00
DEVICE DRIVERS	885-1105	HDOS	UTILITY	20.00
DISK UTILITIES	885-1213-[37]	CPM	UTILITY	20.00
DUNGEONS & DRAGONS	885-1093-[37]	HDOS	GAME	20.00
FLOATING POINT PACKAGE	885-1063	HDOS	UTILITY	18.00
GALACTIC WARRIORS	885-8009-[37]	HDOS	GAME	20.00
GALACTIC WARRIORS	885-8009-[37]	CPM	GAME	20.00
GAMES 1	885-1029-[37]	HDOS	GAMES	18.00
HARD SECTOR SUPPORT PACKAGE	885-1121	HDOS	UTILITY	30.00
HDOS PROGRAMMERS HELPER	885-8017	HDOS	UTILITY	16.00
HOME FINANCE	885-1070	HDOS	BUSINESS	18.00
HUG DISK DUPLICATION UTILITIES	885-1217-[37]	CPM	UTILITY	20.00
HUG SOFTWARE CATALOG	885-4500	VARIOUS	PRODUCTS THRU 1982	9.75
HUGMAN & MOVIE ANIMATION	885-1124	HDOS	ENTERTAINMENT	20.00
INFO. SYSTEM AND TEL. & MAIL SYSTEM	885-1108-[37]	HDOS	DBMS	30.00
LOGBOOK	885-1107-[37]	HDOS	AMATEUR RADIO	30.00
MAGBASE	885-1249-[37]	CPM	MAGAZINE DATABASE	25.00
MAPLE	885-8005	HDOS	COMMUNICATION	35.00
MAPLE	885-8012-[37]	CPM	COMMUNICATION	35.00
MISCELLANEOUS UTILITIES	885-1089-[37]	HDOS	UTILITY	20.00
MORSE CODE TRANSCEIVER	885-8016	HDOS	AMATEUR RADIO	20.00
MORSE CODE TRANSCEIVER	885-8031-[37]	CPM	AMATEUR RADIO	20.00
PAGE EDITOR	885-1079-[37]	HDOS	UTILITY	25.00
PROGRAMS FOR PRINTERS	885-1082	HDOS	UTILITY	20.00
REMARK VOL 1 ISSUES 1-13	885-4001	N/A	1978 TO DECEMBER 1980	20.00
RUNOFF	885-1025	HDOS	TEXT PROCESSOR	35.00
SCICALC	885-8027	HDOS	UTILITY	20.00
SMALL BUSINESS PACKAGE	885-1071-[37]	HDOS	BUSINESS	75.00
SMALL-C COMPILER	885-1134	HDOS	LANGUAGE	30.00
SOFT SECTOR SUPPORT PACKAGE	885-1127-[37]	HDOS	UTILITY	20.00
STUDENT'S STATISTICS PACKAGE	885-8021	HDOS	EDUCATION	20.00
SUBMIT (Z80 ONLY)	885-8006	HDOS	UTILITY	20.00
TERM & HTOC	885-1207-[37]	CPM	COMMUNICATION & UTILITY	20.00
TINY BASIC COMPILER	885-1132-[37]	HDOS	LANGUAGE	25.00
TINY PASCAL	885-1086-[37]	HDOS	LANGUAGE	20.00
UDUMP	885-8004	HDOS	UTILITY	35.00
UTILITIES	885-1212-[37]	CPM	UTILITY	20.00
UTILITIES BY PS	885-1126	HDOS	UTILITY	20.00
VARIETY PACKAGE	885-1135-[37]	HDOS	UTILITY & GAMES	20.00
WHEW UTILITIES	885-1120-[37]	HDOS	UTILITY	20.00
XMET ROBOT X-ASSEMBLER	885-1229-[37]	CPM	UTILITY	20.00
Z80 ASSEMBLER	885-1078-[37]	HDOS	UTILITY	25.00
Z80 DEBUGGING TOOL (ALDT)	885-1116	HDOS	UTILITY	20.00

H8 - H/Z-89/90 - H/Z-100 (Not PC)

PRODUCT NAME	PART NUMBER	OPERATING SYSTEM	DESCRIPTION	PRICE
ADVENTURE	885-1222-[37]	CPM	GAME	10.00
BASIC-E	885-1215-[37]	CPM	LANGUAGE	20.00
CASSINO GAMES	885-1227-[37]	CPM	GAME	20.00
CHEAPCALC	885-1233-[37]	CPM	SPREADSHEET	20.00
CHECKOFF	885-8011-[37]	CPM	CHECKBOOK SOFTWARE	25.00
COPYDOS	885-1235-37	CPM	UTILITY	20.00
DISK DUMP & EDIT UTILITY	885-1225-[37]	CPM	UTILITY	30.00
DUNGEONS & DRAGONS	885-1209-[37]	CPM	GAMES	20.00
FAST ACTION GAMES	885-1228-[37]	CPM	GAME	20.00
FUN DISK I	885-1236-[37]	CPM	GAMES	20.00
FUN DISK II	885-1248-[37]	CPM	GAMES	35.00
GAMES DISK	885-1206-[37]	CPM	GAMES	20.00
GRADE	885-8036-[37]	CPM	GRADE BOOK	20.00
HRUN	885-1223-[37]	CPM	HDOS EMULATOR	40.00
HUG FILE MANAGER & UTILITIES	885-1246-[37]	CPM	UTILITY	20.00
HUG SOFTWARE CATALOG UPDATE #1	885-4501	VARIOUS	PRODUCTS 1983 THRU 1985	9.75
KEYMAP CPM-80	885-1230-[37]	CPM	UTILITY	20.00
MBASIC PAYROLL	885-1218-[37]	CPM	BUSINESS	60.00
NAVPROGSEVEN	885-1219-[37]	CPM	FLIGHT UTILITY	20.00
SEA BATTLE	885-1211-[37]	CPM	GAME	20.00
UTILITIES BY PS	885-1226-[37]	CPM	UTILITY	20.00
UTILITIES	885-1237-[37]	CPM	UTILITY	20.00

Price List

PRODUCT NAME	PART NUMBER	OPERATING SYSTEM	DESCRIPTION	PRICE
X-REFERENCE UTILITIES FOR MBASIC	885-1231-[37]	CPM	UTILITY	20.00
ZTERM	885-3003-[37]	CPM	COMMUNICATION	20.00

H/Z-100 (Not PC) Only

PRODUCT NAME	PART NUMBER	OPERATING SYSTEM	DESCRIPTION	PRICE
ACCOUNTING SYSTEM	885-8048-37	MSDOS	BUSINESS	20.00
CALC	885-8043-37	MSDOS	UTILITY	20.00
CARDCAT	885-3021-37	MSDOS	BUSINESS	20.00
CHEAPCALC	885-3006-37	MSDOS	SPREADSHEET	20.00
CHECKBOOK MANAGER	885-3013-37	MSDOS	BUSINESS	20.00
CP/EMULATOR	885-3007-37	MSDOS	CPM EMULATOR	20.00
DBZ	885-8034-37	MSDOS	DBMS	25.00
DUNGEONS & DRAGONS (ZBASIC)	885-3009-37	MSDOS	GAME	20.00
ETCHDUMP	885-3005-37	MSDOS	UTILITY	20.00
EZPLOT II	885-3049-37	MSODS	PRINTER PLOTTING UTILITY	25.00
GAMES (ZBASIC)	885-3011-37	MSDOS	GAMES	20.00
GAMES CONTEST PACKAGE	885-3017-37	MSDOS	GAMES	25.00
GAMES PACKAGE II	885-3044-37	MSDOS	GAMES	25.00
GRAPHIC GAMES (ZBASIC)	885-3004-37	MSDOS	GAMES	20.00
GRAPHICS	885-3031-37	MSDOS	ENTERTAINMENT	20.00
HELPSCREEN	885-3039-37	MSDOS	UTILITY	20.00
HUG BACKGROUND PRINT SPOOLER	885-1247-37	CPM	UTILITY	20.00
KEYMAC	885-3046-37	MSDOS	UTILITY	20.00
KEYMAP	885-3010-37	MSDOS	UTILITY	20.00
KEYMAP CPM-85	885-1245-37	CPM	UTILITY	20.00
MAPLE	885-8023-37	CPM	COMMUNICATION	35.00
MATHFLASH	885-8030-37	MSDOS	EDUCATION	20.00
ORBITS	885-8041-37	MSDOS	EDUCATION	25.00
POKER PARTY	885-8042-37	MSDOS	ENTERTAINMENT	20.00
SCICALC	885-8028-37	MSDOS	UTILITY	20.00
SKYVIEWS	885-3015-37	MSDOS	ASTRONOMY UTILITY	20.00
SMALL-C COMPILER	885-3026-37	MSDOS	LANGUAGE	30.00
SPELL5	885-3035-37	MSDOS	SPELLING CHECKER	20.00
SPREADSHEET CONTEST PACKAGE	885-3018-37	MSDOS	VARIOUS SPREADSHEETS	25.00
TREE-ID	885-3036-37	MSDOS	TREE IDENTIFIER	20.00
USEFUL PROGRAMS I	885-3022-37	MSDOS	UTILITIES	30.00
UTILITIES	885-3008-37	MSDOS	UTILITY	20.00
ZPC II	885-3037-37	MSDOS	PC EMULATOR	60.00
ZPC UPGRADE DISK	885-3042-37	MSDOS	UTILITY	20.00

H/Z-100 and PC Compatibles

PRODUCT NAME	PART NUMBER	OPERATING SYSTEM	DESCRIPTION	PRICE
ADVENTURE	885-3016	MSDOS	GAME	10.00
ASSEMBLY LANGUAGE UTILITIES	885-8046	MSDOS	UTILITY	20.00
BACKGROUND PRINT SPOOLER	885-3029	MSDOS	UTILITY	20.00
BOTH SIDES PRINTER UTILITY	885-3048	MSDOS	UTILITY	20.00
CXREF	885-3051	MSDOS	UTILITY	17.00
DEBUG SUPPORT UTILITIES	885-3038	MSDOS	UTILITY	20.00
DPATH	885-8039	MSDOS	UTILITY	20.00
HADES II	885-3040	MSDOS	UTILITY	40.00
HELP	885-8040	MSDOS	CAI	25.00
HEPCAT	885-3045	MSDOS	UTILITY	35.00
HUG EDITOR	885-3012	MSDOS	TEXT PROCESSOR	20.00
HUG MENU SYSTEM	885-3020	MSDOS	UTILITY	20.00
HUG SOFTWARE CATALOG UPDATE #1	885-4501	VARIOUS	PROD 1983 THRU 1985	9.75
HUGMCP	885-3033	MSDOS	COMMUNICATION	40.00
ICT 8080 TO 8088 TRANSLATOR	885-3024	MSDOS	UTILITY	20.00
MAGBASE	885-3050	VARIOUS	MAGAZINE DATABASE	25.00
MATT	885-8045	MSDOS	MATRIX UTILITY	20.00
MISCELLANEOUS UTILITIES	885-3025	MSDOS	UTILITIES	20.00
PS's PC & Z100 UTILITIES	885-3052	MSDOS	UTILITY	20.00
REMARK VOL 8 ISSUES 84-95	885-4008	N/A	1987	25.00
REMARK VOL 9 ISSUES 96-107	885-4009	N/A	1988	25.00
REMARK VOL 10 ISSUES 108-119	885-4010	N/A	1989	25.00
SCREEN DUMP	885-3043	MSDOS	UTILITY	30.00
UTILITIES II	885-3014	MSDOS	UTILITY	20.00
Z100 WORDSTAR CONNECTION	885-3047	MSDOS	UTILITY	20.00

PC Compatibles

PRODUCT NAME	PART NUMBER	OPERATING SYSTEM	DESCRIPTION	PRICE
ACCOUNTING SYSTEM	885-8049	MSDOS	BUSINESS	20.00
CARDCAT	885-6006	MSDOS	CATALOGING SYSTEM	20.00
CHEAPCALC	885-6004	MSDOS	SPREADSHEET	20.00
CP/EMULATOR II & ZEMULATOR	885-6002	MSDOS	CPM & Z100 EMULATORS	20.00
DUNGEONS & DRAGONS	885-6007	MSDOS	GAME	20.00
EZPLOT II	885-6013	MSDOS	PRINTER PLOTTING UTILITY	25.00
GRADE	885-8037	MSDOS	GRADE BOOK	20.00
HAM HELP	885-6010	MSDOS	AMATEUR RADIO	20.00
KEYMAP	885-6001	MSDOS	UTILITY	20.00
LAPTOP UTILITIES	885-6014	MSDOS	UTILITIES	20.00
PS's PC UTILITIES	885-6011	MSDOS	UTILITIES	20.00
POWERING UP	885-4604	N/A	GUIDE TO USING PCS	12.00
SCREEN SAVER PLUS	885-6009	MSDOS	UTILITIES	20.00
SKYVIEWS	885-6005	MSDOS	ASTRONOMY UTILITY	20.00
TCSPELL	885-8044	MSDOS	SPELLING CHECKER	20.00
ULTRA RTTY	885-6012	MSDOS	AMATEUR RADIO	20.00
YAUD (YET ANOTHER UTILITIES DISK)	885-6015	MSDOS	UTILITIES	20.00

The following HUG Price List contains a list of all products in the HUG Software Catalog and Software Catalog Update #1. For a detailed abstract of these products, refer to the HUG Software Catalog, Software Catalog Update #1, or previous issues of REMark.

Now Available!

HUG software is now available on 2" disks. Just put a "-90" at the end of the part number (i.e., 885-6014-90). Also add $3.00 to the purchase price of the software (i.e., $20.00 + $3.00 = $23.00).

LAPTOP OWNERS . . . don't feel left out! All of HUG's MSDOS software is available on 3-1/2" micro-floppies too! When ordering, just add a "-80" to the 7-digit HUG part number. For the standard 5-1/4" floppy, just add a "-37".

Make the no-hassle connection with your modem today! ***HUGMCP*** doesn't give you long menus to sift through like some modem packages do. With *HUGMCP*, YOU'RE always in control, not the software. Order ***HUG P/N 885-3033-37*** today, and see if it isn't the easiest-to-use modem software available. They say it's so easy to use, they didn't even need to look at the manual. "It's the only modem software that I use, and I'm in charge of the HUG bulletin board!" says Jim Buszkiewicz. *HUGMCP* runs on ANY Heath/Zenith computer that's capable of running MS-DOS!

ORDERING INFORMATION

For VISA and MasterCard phone orders, telephone the Heath Users' Group directly at (616) 982-3463. Have the part number(s), descriptions, and quantity ready for quick processing. By mail, send your order, plus 10% postage and handling ($1.00 minimum charge, up to a maximum of $5.00) to: Heath Users' Group, P.O. Box 217, Benton Harbor, MI 49022-0217. VISA and MasterCard require minimum $10.00 order. No C.O.D.s accepted.

Questions regarding your subscription? Call Margaret Bacon at (616) 982-3463.

BUGGIN' HUG

Heathkit RS-232 Breakout Box

Dear HUG:

Dual-type D connectors, testing points on all 25 lines, switch disabling capability on 23 lines and LED indicators on 12 key signal lines make the Heathkit RS-232 Breakout Box a versatile piece of test equipment. To make it even more versatile, I recommend adding 9- to 25-pin serial adapters. There are numerous personal computers with built-in 9-pin RS-232 serial ports. These adapters will enable the Heathkit RS-232 Breakout Box to access these serial ports and the peripheral equipment designed to connect to them.

9- to 25-pin serial adapters are available through many electronics suppliers (such as Radio Shack) or you can construct your own using Table 1 as a guide. Two adapters are needed when using the Heathkit RS-232 Breakout Box to interconnect devices with 9-pin RS-232 serial connectors. Conventionally, one adapter with a DB9 female connector is needed to connect to the DTE device and another adapter with a DB9 male connector is needed to connect to the DCE device. The gender of each adapter's DB25 connector is not relevant since the Heathkit RS-232 Breakout Box has dual-type D connectors. If you can not construct or purchase adapters with the required DB9 connector gender, many electronics suppliers also sell gender changers.

Make your Heathkit RS-232 Breakout Box even better with 9- to 25-pin serial adapters.

John B. Wood
2720 Palmer Drive, Apt. N1
Gulfport, MS 39507

Table 1

Pin Number		
DB9 Connector	DB25 Connector	Line Name
1 <————	————> 8	DCD
2 <————	————> 3	RD
3 <————	————> 2	TD
4 <————	————> 20	DTR
5 <————	————> 7	SG
6 <————	————> 6	DSR
7 <————	————> 4	RTS
8 <————	————> 5	CTS
9 <————	————> 22	RI

Hard Drives for the ZW-241

Dear HUG:

I'd like to enlist the aid of your readers in a search I've been conducting. I have a ZW-241 with two hard disk drives, a 20 Meg. and a 40 Meg., both in good working order. I'd like to find a way of adding some additional hard drives to the machine without spending $1600 for one of the special controllers designed for network file servers.

Cards to add additional floppy drives to a computer have been available for some time. In fact, I have a CompatiCard IV in the machine now to operate the pair of 8" drives I salvaged from my Z-100. Since this is a 4-drive controller which carries a D37 connector for supplying power, data, and control signals to an external drive, I plan to add an external ED 3.5" floppy drive to the machine in the near future. This controller supports a 1 Meg. data transfer rate and the new 2.9 Meg. 3.5" floppy drives.

The hard disk controller on my ZW-241 has a jumper which permits its address to be set to 170H instead the customary 1F0H. However, MS-DOS does not support more than two hard drives; a suitable device driver would therefore have to be added and invoked by the CONFIG.SYS file or, possibly, through a combination of device driver and BIOS chip the way the CompatiCard IV does. The CPU card in the ZW-241 has two empty ROM sockets which could be used for additional BIOS. (The ZDS MS-DOS manual indicates that DRIVER.SYS can be set to support an external hard drive, but I don't know if there is actually a controller and drive which will work with this driver. I know that some "secondary" floppy controllers or drives will not work with DRIVER.SYS.

An XT-type controller will work in an AT-type machine if the CMOS setup is told that no hard drive is present. XT-type controllers contain their own BIOS at address C800H, an address which is not in use in my machine. (MFM controllers for AT-compatibles seem to read the hard drive BIOS from part of the system ROM and/or MS-DOS. Because of the differences in the way ESDI drives are setup, they must have their own BIOS. Controllers for these drives therefore require an on-board ROM which is often addressed at C800H and sometimes at D800H.) One possible problem is that (as nearly as I can discern) hard drive 0 is addressed at 320H and hard drive 1 is addressed at 322H on both XT-compatibles and AT-compatibles. Further, adding an XT-type hard disk controller to a fully-populated AT-compatible does not get around the two drive MS-DOS limit. A suitable device driver will still be required.

I am frequently asked why I don't just substitute a larger drive for one of the existing drives. My answer is that I don't see any reason to discard a perfectly functional drive; further, 40-80 Meg. drives are currently very attractive pricewise and thus represent an addition that doesn't have to be agonized over to the extent that, say, a 100 Meg. drive would have to be.

To complicate matters, I would like to consider a SCSI host adapter as my "secondary hard drive controller". This would permit me to connect as many as seven additional hard drives or other mass storage/backup devices to the system in an external daisy chain. So far, I have not found one adapter for which there is also an OS/2 driver. Everyone I talked with at COMDEX in June said "real soon now". I've heard that refrain for a year. While William Adney doesn't think much of OS/2, it's just the thing I've been looking for to port mainframe FORTRAN code to PCs. The Unix compilers I've worked with are error-prone or not very informative about what they don't like or both.

Speaking of COMDEX, one of the interesting things I saw was a 3.5" floppy drive able to read and write 20 Meg. disks, as well as 720K and 1.44 Meg. disks. It has an on-board SCSI controller and thus can be connected to a system through a host adapter. I understand that prices for these drives will be on the order of $600 to $700 when they are available early next year. Given the capability such a drive provides, this is something I might spring for.

Something else I saw at COMDEX was a card which carried a 386SX, some additional logic, 64K of cache RAM, a 387SX socket, a 16 MHz or 20 MHz clock crystal, and a cable to plug into the 286 socket of AT-compatibles (PGA, LCC, or PLCC). The card itself is a reasonable $495 for the 16 MHz version and $595 for the 20 MHz version, but with $165 extra for the cable, that puts it into the price range of a complete SX-clone. ZDS' new 286LP+ is a very interesting approach to this upgrade dilemma. When 32-bit software becomes available, the 286 CPU daughter card in the machine can be replaced by a 386SX daughter card for an educational price of $359.

I have added a number of boards to my machine so I've had to learn something about the IRQs, DMAs, and addresses used by AT-compatibles. One thing that surprises me is that there do not seem to be any boards that make use of DMA channels 5, 6, and 7 which have 16-bit data paths. Most machines also have IRQ 10, 11, and 12 available as well as many unused device addresses. Instead of taking advantage of these capabilities, all the available boards I'm aware of seem to try to be XT-compatible as well so they use DMA channel 1 (the only one available on an XT with hard drive) and IRQ 2-5. Unfortunately, I've already got a board on channel 1 and use IRQ3 and 4 with devices on COM4 and COM3.

I'd be interested in hearing from any reader – including vendors – who has a good suggestion for me.

Sincerely,
Frederick O. Smetana
5425 Parkwood Drive
Raleigh, NC 27612

The New ZUG Secretary is Here!

There's a new voice on the phone, and a new signature at the bottom of most of the correspondence you will see. We finally found someone willing to fill Margaret's shoes.

Lisa Cobb began her ZUG secretarial duties on September 10th. She "escaped" from the ZDS parts department where she was the ZDS customer "interface" for several large commercial clients. Her duties included taking parts orders and ensuring that the correct parts were shipped in a timely manner. Her knowledge of the parts operation will be a big help to all of us in the ZUG group. It means that she knows who to talk to for the right information and to get some action.

Lisa's other full-time job is taking care of a husband and four month old daughter. Her whole family is into sports, especially football. Seems one of her husband's best friends plays for some team called the "Redskins", whoever they are. (This is Bear, Lion, Wolverine, Spartan, and Irish country.)

Lisa will need some time to get into the operation of our little group. So when you call or write, please make her feel welcome. And give her some slack already; she has a lot to learn.

Zenith Data Systems NEWS COLUMN

Henry Fale
Quikdata, Inc.
2618 Penn Circle
Sheboygan, WI 53081-4250

In this installment, you will read about the finalization of the Groupe Bull purchase, an enchanced MinisPort with a better screen and a 20MB hard drive, University of California MBA students being given MinisPort computers, Windows 3 and Asymetrix Toolbook being installed on computers at the factory, top executives being shifted, price reductions on computers and memory, a new 286 LP with instant 386SX upgrade CPU capability, MS-DOS 4 and Windows 3 upgrades, and good news about memory expansions for the forgotten EaZy PC and how to use a modem on the mouse port of the EaZy PC.

Groupe Bull Purchase Finalized

If you remember our last column back in May, you may recall that there was a price dispute in the final closing price of the Groupe Bull purchase of Zenith Data Systems. Bull was asking for a $49 million refund on the ZDS buy, and Zenith Electronics Corporation said that Groupe Bull owed them $49.5 million.

It has been settled. Zenith Electronics Corporation will receive an additional payment of $15 million (plus interest from December 28, 1989) from Groupe Bull for Zenith's computer products business.

The additional payment, the result of 30 days of negotiations under provisions of the sale agreement, brings the final sale price for Zenith's computer business to about $511 million. Zenith Electronics Corporation sold its computer products business to Paris-based Groupe Bull on December 28, 1989, and received an advance payment of $496.4 million in cash.

Because the negotiated final payment was less than the net receivable that Zenith Electronics Corporation had recorded on its books at year-end, Zenith's first-quarter 1990 results will reflect a charge to discontinued operations of about $11 million.

Well I'm glad that's settled. Now they can get down to business and avoid the costs and problems of arbitration.

ZDS Enhances MinisPort 20MB Hard Drive, Brighter Screen

Zenith Data Systems (ZDS) introduced an enhanced model of its MinisPort notebook personal computer (PC) that features a fast 20-megabyte (MB) 2.5-inch hard disk drive, a new Crystal Bright screen and a faster processor.

The MinisPort HD is based on a 10-megahertz 80C88 processor, which may be slowed to 4.77 megahertz to increase battery life.

"We've made the MinisPort HD the thinnest notebook PC with a hard disk drive by eliminating all unnecessary size and weight," said ZDS President and Chief Executive Officer John P. Frank. "As a result, this system is ideal for even the most mobile computer user, and is powerful enough for both business and educational applications."

The MinisPort HD features a 2.5-inch, 20MB hard disk drive developed by Connor Peripherals, San Jose, Calif. This hard disk is no larger than a deck of playing cards and offers quick data accesses rated at 23 milliseconds with 1:1 interleave. "This level of performance is typically found only on high-performance desktop systems," Frank said.

The system's optional 3.5-inch, 720-kilobyte (K) external floppy disk drive, which runs off the system's power, gives users the option of carrying a floppy drive with them or leaving the added weight behind.

The 6-pound (with battery) MinisPort HD has a new Crystal Bright screen that displays black characters against a backlit bright white background. This new CGA screen can display eight shades — from light grey to black — and is brighter than the display of the other MinisPort models for easy readability (a major complaint of the previous MinisPort models). The screen's viewable area is 3.3 inches high by 8.5 inches wide and measures 9.5 inches diagonally.

The MinisPort HD operates for about three hours using its easily removable nickel-cadmium (NiCad) battery pack, which recharges in less than 2.5 hours. Battery life can be increased dramatically and programs will run faster while working from the silicon disk rather than the hard or floppy disk drive. When the system is turned off, data in the silicon disk is backed by the NiCad battery.

The system has 1MB of memory, of which 360 kilobytes can be allocated as an ultra-fast "silicon disk drive." Rupp Corp.'s *FastLynx LX*™ (an easy-to-use, self-uploading data transfer program) and an abridged version of MS-DOS™ 3.3 Plus is stored in the system's non-erasable read-only-memory. A complete copy of MS-DOS also is factory-installed on the hard disk. Also featured are standard I/O ports, including serial, parallel, floppy disk drive and color monitor ports.

Other MinisPort HD features include:

- 9-pin RS-232-C serial port
- Centronics-compatible parallel port
- 9-pin RGB video output connector
- Slot for optional 2400-baud AutoSync or 1200-baud modem
- 80-key, full-size keyboard with embedded numeric keypad
- 110/240 VAC power supply and battery charger
- 3-headed serial cable for FastLynx LX data transfers
- One-year warranty

MinisPort HD options include:

- External 5.25-inch 360K floppy disk drive
- External 2-inch 720K floppy disk drive
- 2400-baud AutoSync modem
- 1200-baud modem
- Extra NiCad battery packs
- Monitor stand
- Nylon carrying case with pocket
- Leatherette slipcover.

The suggested retail price for the MinisPort Model 20 is $2,799. ZDS has begun shipping the MinisPort HD, which measures 12.4"W × 9.8"D × 1.3"H.

University of California: Outfits MBA Students With Notebook PCs

The rustling of pages soon will be eliminated by a new type of notebook. This fall, graduate students entering the University of California, Irvine, Executive Master's of Business Administration program will be outfitted with a Zenith Data Systems (ZDS) MinisPort™ HD notebook computer to accompany them to classes.

"We view the MinisPort HD as an instructional tool, equal in importance to books and course materials," said Ann Jennings, director of the Executive M.B.A. program. "Our well-rounded curriculum assures that each graduating student has the skills necessary to make prudent business decisions. Personal computers are an essential aid to high-level business decisions."

For the next three years, each of the M.B.A. candidates admitted annually to the two-year Executive program will be issued a MinisPort HD. The price of the notebook computer, as well as books, and other education materials are included in the tuition. Faculty members also will be issued a system.

According to Jennings, "Of the systems we tested, the MinisPort HD was the only hard disk notebook PC that allowed us to get through a three-hour class on one battery charge."

ZDS Installs Microsoft Windows 3.0 Asymetrix Toolbook on Desktop PCs

Advancing the definition of the personal computer (PC), Zenith Data Systems (ZDS) announced that it is now factory-installing Microsoft Windows 3.0 on the hard disk drives of all '286- and '386-based desktop PCs. ZDS also is installing ToolBook 1.0, Asymetrix Corp.'s new Windows-based software construction set, on all '386-based desktop systems.

"These tools are the dashboard which allows users to drive the most powerful PCs and software without ever opening a manual," said John P. Frank, ZDS president and chief executive officer. "We're immediately inviting our customers into a computing environment where all software responds to the user in the same way and taps the hardware's full capabilities."

We're delighted with Zenith Data Systems' commitment to Windows 3.0 and graphical user interface computing," said Bill Gates, Microsoft chairman and chief executive officer. "ZDS' strategy to make Windows 3.0 a standard feature is a dramatic extension of the company's long-standing relationship with Microsoft, which has made ZDS the largest provider of Windows worldwide."

ZDS has shipped more copies of Windows than any other PC manufacturer by bundling it with all hard-drive desktop systems since 1985.

ZDS also is bundling a Microsoft Mouse and installing MS-DOS 4.0 on all '286 and '386 desktop PCs. For systems already in dealer inventories, ZDS will provide its resellers with copies of Windows 3.0 and ToolBook, as well as a mouse, to bundle.

ZDS Realigns Top Execs

Groupe Bull's Zenith Data Systems unit last week reshuffled its senior management roster, shifting several top executives to new jobs and reducing the direct control of John Frank, president and chief executive, over such areas as engineering, customer service and business planning.

The management changes come five months after Groupe Bull bought ZDS from Zenith Electronics Corp. and follow a 78 percent decline in operating profits and a 24 percent dip in sales at the PC vendor during 1989, as reported recently by Zenith Electronics (EN, April 9).

Much of the tail-off in ZDS business came in late 1989, following the unit's November loss of the $700 million Desktop III Air Force PC contract. The PC maker, which gets more than a third of its sales from the government, has since also had a $534 million PC upgrade contract from the Navy thrown out by the General Services Administration.

A ZDS spokesman insisted that the reorganization "was not a reactionary move" related to the 1989 showing. He described it as "more of a pro-active step to strengthen our position in the market" and push more decision-making responsibilities to management levels below Mr. Frank.

He would not comment on current business conditions at ZDS beyond noting that executives are expecting changes implemented at the start of the year in the company's reseller program and discount schedule to have a positive effect on sales during 1990.

Mr. Frank has one less person reporting to him following the reorganization, ZDS said. However, the executives in charge of engineering and customer service no longer report to him, while a new vice president of planning who does report to Mr. Frank holds duties Mr. Frank had handled.

ZDS combined product development and engineering under Andrew Czernek, who was named vice president of product strategy and development. He was vice president of marketing.

ZDS Reduces Prices on Desktop SX, Portable '286 Systems
Memory Prices Lowered 20 Percent

On July 12, 1990, Zenith Data Systems (ZDS) reduced the suggested retail prices of its desktop Z-386SX and portable SupersPort 286e personal computers (PCs) by $300, or up to 9 percent. ZDS also lowered its expansion memory prices by about 20 percent.

"We expect 386SX-based PC sales to continue their record-breaking pace, as the market grows increasingly competitive with promotions and pricing actions," said Arthur Lambert, ZDS vice president of sales and marketing. "This price drop, combined with our recent strategy to bundle a mouse and factory-install Microsoft Windows 3.0 and Asymetrix ToolBook on the hard drive models, has allowed us to add almost $1,000 of value to these systems within the last two months."

The Z-386SX Model 1, which has a 16-megahertz Intel 386SX processor, 2 megabytes (MB) of memory, a VGA video card and a 1.44MB 3.5-inch floppy drive, now has a suggested retail price of $2,999, down from $3,299. The prices of the Model 40, which has a 40MB hard drive, and the Model 80, which has an 80MB hard drive, are now $3,699 and $4,399, down from $3,999 and $4,699, respectively.

In response to growing demand for 80286-based portable PCs with VGA video, ZDS lowered the prices of its SupersPort 286e Model 20 and Model 40 to $4,399 and $4,699, from $4,699 and $4,999, respectively.

In addition, ZDS has reduced the suggested retail prices of expansion memory for all portable and desktop PCs by 20 percent. Prices of 1- and 2-megabyte memory chips have been lowered to $399 and $799, from $499 and $999, respectively.

New Desktop '286 Upgrades to SX Processor

On July 10, 1990, An 80286-based personal computer (PC) that can be upgraded with an Intel 386SX processor was introduced by Zenith Data Systems (ZDS).

The new Z-286 LP Plus is ZDS' first system designed to be easily upgraded with a faster processor.

Based on a 12-megahertz (MHz) 80286 processor, the Z-286 LP Plus is a low-cost, all-purpose computing platform designed for word processing, spreadsheet analysis, desktop publishing and simple graphics applications. The system's upgradeable design allows it to keep pace with users' advancing needs while preserving their PC investment.

"Although '386-based systems clearly represent the best long-term PC investment, '286-based systems still offer significant value and performance," said Arthur Lambert, ZDS' vice president of sales and marketing.

When the user demands faster performance and 32-bit compatibility, the Z-286 LP Plus can be upgraded with a 16MHz 386SX processor by the user in less than five minutes. The system's 12-megahertz 80286 processor is mounted on a removable card that can be replaced by an upgrade card featuring a 16-megahertz 386SX. The processor cards also have coprocessor sockets. This SX upgrade will increase performance by more than 25 percent in typical DOS applications. Coprocessor sockets are included on the cards. The upgrade card has a manufacturer's suggested retail price of $599.

The new graphical user interface, Microsoft Windows 3.0, which we've preinstalled on the hard disk systems, will further aid a swift and seamless migration to 32-bit computing," he said.

Standard features of the Z-286 LP Plus include VGA video, 1 megabyte (MB) of memory (expandable to 16MB, 8MB on system board), three open slots and a 1.44MB 3.5-inch floppy disk drive. The Z-286 LP Plus also is available with a 20MB or 40MB hard disk drive, which both have an access time of 28 milliseconds and 1:1 disk interleave.

Other Z-286 LP Plus features include:

- Two 9-pin serial ports
- 25-pin parallel port
- Real-time clock with battery backup
- 101-keyboard
- Microsoft Mouse with hard drive models
- Microsoft Windows 3.0 pre-installed on hard drive models
- MS-DOS 4.0 pre-installed on hard drive models
- One-year warranty

The Z-286 LP Plus offers three security features:

- Cabinet cover key lock to prevent unauthorized opening of the cabinet
- Lock-down loop to secure the system to a work surface
- Password protection to prevent unauthorized access to the system.

The Z-286 LP Plus' small cabinet measures 14 inches wide, 15 inches deep and 3.8 inches high. The suggested retail price for the Model 20 with a 20MB hard disk drive is $2,399, and the Model 40 with a 40MB hard disk is $2,699. For network node applications, the Model 1 with a 1.44MB floppy disk drive is $1,999. The suggested retail price for the 386SX upgrade card is $599.

The Z-286 LP Plus Models 20 and 40 come with a Microsoft Mouse, as well as Microsoft Windows 3.0 and MS-DOS 4.0 factory-installed on the hard disks. The system measures 14 inches wide by 15 inches deep by 3.8 inches high, offering a small footprint.

Upgrade Time From ZDS *MS-DOS 4.0*

ZDS customers can now upgrade to MS-DOS version 4.0 (OS-105-MS) for $49.00. This offer is good until December 31, 1990 If you are a registered owner of a previous version you should be getting an update card from Zenith Data Systems with full instructions.

Version 4 has some new features, most which have been available in other software packages such as PC TOOLS and stand alone programs for a long time. These include the following:

DSKSCAN — Scans the entire disk searching for bad sectors trying to recover data contained in any bad sectors.

EMM386.SYS — Allows MS-DOS programs and applications to use extended memory, as well as expanded memory (EMS vers. 4). Only for '386 systems.

HIMEM.SYS — Allows MS-DOS programs on computers with extended memory to access the extended memory (XMS vers. 2.0).

INSTALL — Executes a command during CONFIG.SYS processing.

MEM.EXE — Displays the amount of memory and a list of programs that are loaded.

REM — Allows for comments to be placed in the CONFIG.SYS files.

Include $4.50 shipping and handling plus your state tax. If you don't have the coupon, read on.

No Coupon Upgrade to MS-DOS 4.0

Zenith Data Systems' normal procedure for handling its software upgrades is for its Software Registration department to mail an upgrade coupon to registered owners of its software. Software Registration will also accept upgrade orders according to the following procedure:

1. Photocopy your distribution disk(s) of the relevant software product.
2. Complete the form below. Be sure to specify whether you want 5.25" or 3.5" diskettes. If you do not specify, you will be shipped 5.25" diskettes.
3. Mail the completed form, the photocopy, and full payment to:

   ZDS Software Registration
   Attn: Kathryn Tollas
   P.O. Box 1000
   St. Joseph, MI 49085

Please do not use this coupon until after July 7, 1990, and use it only if you have not received a printed coupon in the mail by then.

Payment: Purchase Orders are not accepted. You may pay by check or by ZDS Credit Card, Visa, or MasterCard. You must include $4.50 shipping for each item, and any applicable sales tax.

Sales Tax: Include local sales taxes if you are ordering from these States: AL, AR, AZ, CA, CO, DC, FL, GA, HI, IL, IN, KS, LA, MA, MD, MI, MN, MO, NC, NJ, NM, NY, NH, OH, OK, PA, RI, TN, TX, UT, VA, WA, WI.

FAX orders: As an additional courtesy, you may order by FAX if you pay by ZDS Credit Card, Visa, or MasterCard. FAX the completed form below and the photocopies of your distribution disks to: (616) 982-5099. Please send payment in U.S. dollars only.

This offer expires December 31, 1990. Not valid in combination with other offers. Prices, availability, and specifications subject to change without notice. Void where prohibited or taxed by law. MS-DOS is a trademark of Microsoft.

The Zenith Data Systems COM1 Upgrade Order Form

Product	**Price**	**Quantity**	**Amount**
MS-DOS 4.0 OS-105-MS (5.25") OS-103-MS (3.50")	$49.00		$
Shipping	$ 4.50		$
Sales Tax			$ ======
TOTAL			$

[ ] Check Enclosed
[ ] Charge To:

[ ] ZDS Credit Card
[ ] Visa
[ ] MasterCard

Card Number: ____________________

Expiration Date: ____________________

Your Signature: ____________________

Ship To:

Name: ____________________

Company: ____________________

Street: ____________________

City: ____________________ State: ____ Zip: __________

* Note that ZDS cannot ship to a Post Office Box. You must use a street address.

Zenith Data Systems Upgrade to Windows Version 3.0

I received several upgrade coupons for DOS 4, but what also surprised me is that I received an upgrade order form for Windows Version 3.0 for $49. If you have ever had a registered previous version of Windows:

Imagine, only $49, for all the benefits of Windows v.3.0, including:

- Program Manager, an organizational life-saver.
- A graphical directory tree called File Manager that makes file maintenance easy and lets you connect to and disconnect from network servers while inside Windows.
- Special productivity tools. Like Microsoft Windows *Write, Paintbrush, Terminal, Recorder, Clipboard, Program Information File Editor, and Control Panel.* Plus, Windows v.3.0 shatters the 640K memory barrier, giving you access to up to 16MB of memory.

Minimum configuration requirements to run Windows 3.0, and, more importantly, optimal configurations to achieve expected operating performance depends on the mode you will be running it in. Real mode, standard mode, or 386TM enhanced mode. The mode that you use depends on the equipment you have and the software applications you want to run under the Windows environment.

REAL MODE is the operating mode that provides maximum compatibility with previous versions of Windows applications (2.x versions). Real mode is also the only mode available for computers with less than 1 megabyte of available memory.

STANDARD MODE is the normal operating mode for running Windows. This mode provides access to extended memory on 80286 systems and above, and also lets you switch among non-Windows applications.

386 ENHANCED MODE provides access to the virtual memory capabilities of the Intel 80386 family of processors. Virtual memory capabilities let Windows applications use more memory than is physically available by using disk space. Enhanced mode also allows for multi-tasking of non-Windows applications.

Based on the mode you want to run in, you can determine if your system hardware has the appropriate requirements to perform effectively. The minimum software and hardware requirements your computer system needs to run Windows 3.0 successfully are:

- MS-DOS 3.1 or later
- A hard disk with 6 to 8 megabytes of free disk space and at least one floppy drive (to load the software initially)
- A minimum of a CGA monochrome monitor

For Real Mode:

- A system with an Intel 8088 processor
- 640K of conventional memory available

For Standard Mode:

- A system with the Intel 80286 processor
- 1 megabyte of available memory (640K conventional memory and 256K extended memory). Note that expanded memory (EMS) only available above the 640K will OT suffice.

For '386 Enhanced Mode:

- A system with an Intel 80386 SX processor or above
- 2 MB of available memory (640K conventional and 1024K extended)

Although a mouse is not required, it is highly recommended so that you can take full advantage of the easy-to-use graphical interface.

Obviously, the more powerful your system, the more effective Windows 3.0 will be. But the single most important hardware feature you may want to upgrade will be memory. Zenith Data Systems recommends that although Windows 3.0 will run on an 8088-based system with 640K of memory and a monochrome monitor, it is not an appropriate system to take advantage of Windows capabilities. Zenith Data Systems highly recommends that Windows be run at least at a minimum configuration to support Standard Mode — that is an 80286-based system with 1MB of available memory (640K convetional, 256K extended). Be sure to note that most systems only offer extended memory beyond the first 1MB of installed memory so if your system has ONLY 1MB total memory it is unlikely that you have available extended memory to run Standard Mode. Additional memory is highly recommended if you expect to be running multiple programs simultaneously.

Problems With Windows Version 3 Reported

Problems have been reported with the Z-386/16 and Microsoft Windows 3. Some problems exist which will prevent the windows from booting properly in the standard or EMS mode. There evidently was a problem wit the Windows not being designed for drive partitions over 32 megs. I guess a "swapfile.exe" file is supposed to fix that. Problem is that Windows uses the hard drive to swap info back and forth like an electronic disk, and it was not set up properly for larger drives. I also understand some swapfile.exe files do not work properly I heard that Zenith Data Systems' versions contains the proper file, but I have a customer who said the Microsoft version still did not work on his ZDS system with an 80MB hard drive. Rumors also have it that this problem may exist on the SupersPort 286e version.

Another problem existed where disk fragmentation could occur if the hard drive had over 1023 cylinders.

There was still another which supposedly could be cured by upgrading an I/O bus pal to a version 5, which seems almost impossible to get. It was first rumored that the problem may have been the cache card, but has been confirmed to be a PAL on the 386/16 CPU card. Chips prior to -4 MAY not have a problem, but -4 does and -5 fixes it. At this point, you can not buy this chip through Heath Parts but MUST buy it through a Zenith dealer or service center (Quikdata is stocking this part). Call 1-800-877-7704 for your nearest dealer or service center.

EaZy PC Memory Expansion

As you EaZy PC computer owners may know, Heath and ZDS have long ago discontinued the 128K RAM upgrade for the 512K EaZy PC computer, leaving no possible way of upgrading your EaZy PC to the 640K needed for lots of software.

I spoke to scientist Dave Brockman of FBE Research about it, and he pondered quite awhile about it, then finally broke down and purchased an EaZy PC from a liquidation specialist. He went to work and designed a way of adding the 128K.

Quikdata, Inc., is now handling this upgrade. Following is the blurb from Dave Brockman:

Expand the 512K base memory of the ZDS EaZy PC computer to 640K with the EZM-128 memory expansion. The EZM-128 consists of a memory circuit board, a protective cover and installation documentation. The circuit board plugs into the option connector on the back of the computer. The cover encloses the circuit board and attaches to the rear of the computer with two screws. The EZM-128 is priced at $95 (retail price is $125).

In development for the EaZy PC is the EZCLOCK board. This is (the prototype is completed and working) a calendar/clock circuit board that installs piggyback on the EZM-128 memory board. The clock is quartz controlled and backed up with a long life (many years) lithium battery. It is based on the same technology used in the SmartWatch module and avoids the need to disassemble the EaZy PC for installation as is required by the SmartWatch. The EZCLOCK will be priced at $33 (retail price is $35).

Quikdata is carrying both of these products.

Using a Modem on the EaZy PC Mouse Port

ZDS produced many EaZy PC computers thinking it would be a great low-cost computer. Problem is, it was too expensive, too limited, and came too late. The product line died quite rapidly, with ZDS discontinuing the expansion memo-

Continued on page 21

dBASE III
Part 7

D. R. Cool
7421 Troy Manor Drive
Huber Heights, OH 45424

Procedure Files

In the last two articles, a more integrated approach to the editing of the PROJECTS and PROJXREF data bases was developed in the form of the PROJUPD2 program. This program called five subprograms using the DO command. In some instances, a subprogram was invoked more than once. These subprograms could be thought of as subroutines, much like a GOSUB in BASIC, the difference being that a subprogram must be loaded into memory each time it is called. It would obviously be better if the subprograms could be part of the main program. The procedure file allows for precisely this.

A procedure file is a dBASE III program that may consist of as many as 32 separate programs or procedures. Such a file is loaded into memory with the command

```
SET PROCEDURE TO procedure file name
```

Once the procedure file is loaded into memory, any of the procedures within the file may be called without any additional disk access.

Each procedure must begin with a PROCEDURE statement which identifies the name of the procedure:

```
PROCEDURE procedure name
```

Procedure names may be up to nine characters long (the dBASE III manual says eight characters, but I know for certain the system can distinguish up to nine characters) and may contain an embedded underscore (_). Just as with any dBASE III program, procedures are called with the DO command. A procedure may call another program or another procedure; however, if a program exists with the same name as a procedure, the DO command will execute the procedure rather than the program. To avoid possible confusion, procedure names should not duplicate program names, at least not within the same directory.

Procedure files are edited the same way as program files. Since most procedure files exceed 5000 bytes in length, you will have to use an external editor or word processor. As you are testing a procedure file, you will no doubt find errors that need correction. Here is one fact that the dBASE III manual (or any other book I've read) doesn't mention - if you make corrections in a procedure file and then re-run the procedure, the system will act as though the corrections were never made. When I first started writing procedure files, this just about drove me crazy. The reason the system ignores your corrections is fairly simple. The command

```
SET PROCEDURE TO procedure file name
```

loads a copy of the file into RAM where it can be accessed. You run a procedure and find an error. You then use MODIFY COMMAND which loads your editor or word processor. You make corrections to the file and exit the word processor, writing the corrected version to disk. However, the original version still resides in RAM. In order to test the corrected version, you must re-issue the SET PROCEDURE TO command. Obviously, if you exit dBASE whenever you edit a procedure file, you will never encounter this.

We will now set up a procedure file called PROJPROC.PRG. The main procedure will be called MAIN_UPDT. The MAIN_UPDT procedure will be a slightly modified version of the PROJUPD2.PRG of Part 5. In addition, seven additional procedures called by the main procedure will be included in the procedure file. Six of the procedures are identical to programs that were developed in Parts 4 and 5. A seventh procedure - ERR_MSG - is a short procedure designed to display an error message on row 23.

The PROJPROC procedure file is shown in Listing 1. Because of space limitations, the six procedures - SCREEN, GET_PD, DISP_ST, DISP_DEL, GET_DTD and VALIDATE - are not reproduced in Listing 1. If your editor or word processor includes a "block read" or "file read" feature, you can import these files directly into the PROJPROC document and save re-typing (assuming you typed them previously!). Make sure each procedure starts with the PROCEDURE statement followed by the procedure name and ends with the RETURN command.

If you compare the MAIN_UPDT procedure to the PROJUPD2 program from which it was adapted, you will note some significant changes. First, the SET commands which set the environment were deleted. These commands will be included in a master control program which I will talk about later in the article.

Secondly, the MAIN_UPDT procedure displays data from the first PROJECT record immediately on start-up; whereas the PROJUPD2 program required the user to execute a search before any data was displayed. This is primarily a personal preference. It also eliminates the need for the "You must search first" error messages of the PROJUPD2 program.

A third change has to do with an idiosyncracy of dBASE III that I have found difficult to work around. This has to do with the End-of-File (EOF) function and the use of this function to prevent a user from trying to move beyond the last record with "N" for "Next". You can verify this yourself using the PROJUPD2 program. First, determine the drawing number of the last record in the PROJECTS data base by issuing the following commands:

```
USE PROJECTS INDEX PROJDWRV
GO BOTTOM
EDIT
```

Make note of the drawing number. Next, if you have the PROJUPD2 program available, type DO PROJUPD2. Press "S" to SEARCH and enter the drawing number noted previously. The record pointer is now sitting on the last logical record. Now, press "N" for "next record". The program responds by displaying what appears to be a blank record. Press "N" again and you will finally get the error message "This is the last project". Finally, press "S" and enter a non-existant drawing number such as "99999". The program will now bomb with the error message "Record is out of range".

What caused this sequence of events? On the first search, when you entered the drawing number of the last project, even though the record pointer is sitting on the last record, the EOF function will evaluate as FALSE. You then pressed "N" to advance to the next record. Since an end-of-file condition did not exist, the SKIP command was executed. The EOF function now evaluates as TRUE and the system brings up a non-existant "blank" record. Pressing "N" at this point causes the program-generated error message to display. Next, you pressed "S" for Search. The program saves the current record number. However, the current record number is a false number because of the end-of-file condition. Then, when you entered the non-existant drawing number, another EOF condition occurred. Line 51 of PROJUPD2 says "goto RECNO". RECNO is the memory variable containing the record number stored prior to the search. In this particular sequence, the record number in RECNO is a non-existant record, thus causing dBASE III to respond with "Record is out of range".

The whole problem stems from what dBASE III calls "end-of-file". End-of-file doesn't actually occur until you SKIP past the last record. What is needed in this application is an indication of the last record, not the end of file. This is what lines 8 to 11 accomplish. The command GO BOTTOM places the record pointer on the last record. Line 9 stores the record number of the last record to the memory variable LAST_REC. In a similar manner, lines 10 and 11 save the record number of the first record. How do I know that I am storing the first and last record numbers of PROJECTS and not PROJXREF? The answer is that the PROJECTS data base was opened after PROJXREF (see lines 3-6).

Having stored the first and last record numbers, when the user selects "N", thr program first compares the current record number with LAST_REC (line 80). Only if the two records are not equal is the SKIP executed. Likewise, if the user selects "P", the program first compares the current record number with FIRST_REC (line 102).

The remainder of the MAIN_UPDT procedure is pretty much the same as the PROJUPD2 program which was analyzed in Part 6. There is one slight change to the SEARCH routine. The SEEK command (line 56) uses the TRIM function which removes any trailing spaces from the (MDWGNR + MREV) expression. This allows a partial input. For example, suppose a record existed for drawing number 5962-85525, revision "A". If you input "5962-85525" to MDWGNR and " " to MREV, the TRIM function will remove the trailing space. Since EXACT is not set ON, the SEEK command will find a matching record.

The one remaining change in the PROJUPD2 program involves the new procedure ERR_MSG which is invoked by the MAIN_UPDT procedure wherever an error message is to be displayed. This procedure is shown in full at the end of Listing 1. Wherever this procedure is called (line 59, for example), it is always followed by the word "with" and an error message in quotes. If you look at the DO command in the dBASE III manual, you will see the full syntax as

```
DO <filename> |WITH <parameter list>
```

The WITH portion is optional and has not been used until now. This option provides a method of passing various parameters to another program or procedure. This makes it possible for a program or procedure to operate in a general way using specific parameters. The ERR_MSG procedure is designed to do the following:

1. Sound the system bell.
2. Clear row 23.
3. Determine the starting column of a message based on the length of the message.
4. Display the message on row 23
5. Wait for the user to press a key

The variable in this procedure is the message. The message is passed to the procedure using the DO...WITH command and is received into the memory variable MESSAGE. The memory variable MESSAGE is defined as the "receiving" parameter with the PARAMETER command. This command must be the first executable command in the receiving program or procedure. The receiving parameter list assigns local variable names to the parameter list in the "sending" DO command. The number of items in both lists must match.

As stated previously, before you can run any procedure within a procedure file, you must load the file into memory with the SET PROCEDURE TO command. Once this is done, the main procedure can be executed with the command DO MAIN_UPDT. As I stated earlier, we will be using a master control program to set the system environment. This control program will also generate a main menu and will be responsible for loading any procedure files required for the system. This master program is included here as Listing 2. As usual, the line numbers are for reference only. With the master control program on disk, a single command - DO PROJECTS - is all that is necessary to activate the entire system.

The SET commands that were included in the PROJUPD2 program are now part of the master program (lines 1 to 5). Line 5 - SET SCOREBOARD OFF - is used here for the first time. SET SCOREBOARD determines whether dBASE III messages appear on the status line, row 0. With SET SCOREBOARD ON, you see messages such as "DEL" for a deleted record, "INS" if you are in insert mode, etc. Since it is desired that these messages do not appear on the screen, SCOREBOARD is SET OFF.

The master program includes two DO WHILE loops; one which begins by clearing the screen (line 8) and a nested loop which begins with line 13. The nested loop is not actually necessary; however, without it, whenever the user makes an invalid selection, the screen would be cleared and the menu re-displayed. By using a nested loop, whenever the user makes an invalid selection the LOOP command at line 39 returns program control back to line 14.

In the case of SELECT = "1", the program executes the MAIN_UPDT procedure. When the user exits from this procedure, the program executes the EXIT command at line 21. This causes the program to exit from the nested loop. The program returns to the top of the main loop and redraws the main menu.

The second item - Monthly Report - will be fore the purpose of printing a monthly report which will summarize the projects completed for the month. In the next article, I will discuss the report form generator and its advantages and disadvantages. The procedure for printing the monthly report will also be developed and added to the PROJPROC procedure file.

Listing 1

```
* PROJPROC.PRG
* PROCEDURE FILE FOR PROJECTS
* WRITTEN BY:  D.COOL       02/27/90

*------------------------------------
* Procedure       Purpose
*------------------------------------
* MAIN_UPDT       Main procedure for edit or inquiry of
PROJECTS and
*                 Projects and PROJXREF data bases.
* SCREEN          Paints screen template for MAIN_UPDT
* GET_PD          Gets the project data
* DISP_ST         Displays the project status
* DISP_DEL        Displays project's "deleted" status
* GET_DTD         Gets device type data
* VALIDATE        Validates the project data
* ERR_MSG         Displays an error message on row 23
*------------------------------------
procedure MAIN_UPDT

* OPEN DATABASE FILES:
select 2
use PROJXREF index PROJXPJN
select 1
use PROJECTS index PROJDWRV

* Find first and last record of PROJECTS:
go bottom
store recno() to LAST_REC
go top
store recno() to FIRST_REC
* Initialize variables:
store " " to PRUP_OPT
store " " to CRUP_OPT
ROW = 1

do SCREEN

* The following command initializes MAIN_OPT to "S".
  If this is not
* done, the procedure GET_PD will exit with an error
  message.
store "S" to MAIN_OPT

do GET_PD
clear gets
do DISP_ST
do DISP_DEL
store PROJNR to MPROJNR
select PROJXREF
seek MPROJNR
if .not. eof()
    do GET_DTD
    clear gets
else
    do ERR_MSG with "Device type data does not exist
      for " +; "project - " + PRESS_MSG
endif

do while .T.      && MAIN PROGRAM LOOP

    select PROJECTS
    set intensity off
    store " " to MAIN_OPT
    @ 22, 0 clear
    @ 22,35 say "MAIN MENU"
    @ 23,19 say "(S)earch  (N)ext  (P)rev  (U)pdate
            e(X)it";get MAIN_OPT picture "!"
    read
    set intensity on

    do case       && MAIN_OPT

      case MAIN_OPT = "X"
        return
      case MAIN_OPT = "S"
        store space(10) to MDWGNR
        store " " to MREV
        @ 23, 0
        @ 23,10 say "Enter drawing number and revision
            letter: ";get MDWGNR picture "9999-99999"
        @ 23,66 get MREV picture "@! A"
        read
        if MDWGNR = " "
            loop
        endif
        store recno() to RECNO
        seek trim(MDWGNR + MREV)
        if eof()
            goto RECNO
            do ERR_MSG with "Project not found - " +\
                    PRESS_MSG
            loop
        endif
        do GET_PD
        clear gets
        do DISP_ST
        do DISP_DEL
        store PROJNR to MPROJNR
        select PROJXREF
        seek MPROJNR
        if .not. eof()
            do GET_DTD
            clear gets
        else
            @ ROW+6,53 say space(2)
            @ ROW+7,53 say space(15)
            @ ROW+8,48 say space(11)
            do ERR_MSG with "Device type data does not
                exist for " +;"project - " + PRESS_MSG
        endif
        select PROJECTS

      case MAIN_OPT = "N"
        if recno() <> LAST_REC
            skip
            do GET_PD
            clear gets
            do DISP_ST
            do DISP_DEL
            store PROJNR to MPROJNR
            select PROJXREF
            seek MPROJNR
            * The IF...ENDIF which follows anticipate
            * the possibility that:  (1) the PROJNR
            * field is blank, or (2) no device
            * type record exist for this project.
            if .not. eof()
              do GET_DTD
              clear gets
            else
              @ ROW+6,53 say space(2)
              @ ROW+7,53 say space(15)
              @ ROW+8,48 say space(11)
            endif
            select PROJECTS
        else
            do ERR_MSG with "This is the last project -
                    " + PRESS_MSG
        endif

      case MAIN_OPT = "P"
        if recno() <> FIRST_REC
            skip -1
            do GET_PD
            clear gets
            do DISP_ST
            do DISP_DEL
            store PROJNR to MPROJNR
            select PROJXREF
            seek MPROJNR
        if .not. eof()
          do GET_DTD
          clear gets
        else
          @ ROW+6,53 say space(2)
          @ ROW+7,53 say space(15)
          @ ROW+8,48 say space(11)
        endif
        select PROJECTS
  else
    do ERR_MSG with "This is the first project - " +
PRESS_MSG
  endif

case MAIN_OPT = "U"     && UPDATE PROMPT
  set intensity off
  store " " to UPDT_OPT
  ? chr(7)
  @ 23, 0
  @ 23,25 say "Projects or Crossref? (P/C)";
        get UPDT_OPT picture "!"
  read
  set intensity on

  do case      && UPDT_OPT

    case UPDT_OPT = "P"
        do while .T.          && PROJECT UPDATE
          @ 22, 0 clear
          @ 22,33 say "PROJECT UPDATE"
          @ 23,15 say "(C)hange  (D)el  (R)ecover
                 (A)dd  "+; "(S)tatus  e(X)it"
          set intensity off
          store " " to PRUP_OPT
          @ 23,66 get PRUP_OPT picture "!"
          read
          set intensity on

          do case      && PRUP_OPT

            case PRUP_OPT = "X"
              exit

            case PRUP_OPT = "C"
              do GET_PD
              read
              if dtoc(APPDATE) <> "  "
                replace STATUS with "C"
              endif
              do VALIDATE
              do DISP_ST

            case PRUP_OPT = "D"
              set intensity off
              store " " to CONFIRM
              do while .not. CONFIRM $ "YN"
                ? chr(7)
                @ 23, 0
                @ 23,14 say "Are you sure you want to " + ;
                        "delete this project? (Y/N)";
                        get CONFIRM picture "!"
                read
              enddo
              if CONFIRM = "Y"
                delete
                do DISP_DEL
                @ 23, 0
                @ 23,15 say "Project has been deleted - "
                        + PRESS_MSG
                wait ""
              endif

            case PRUP_OPT = "R"
              if PRUP_OPT = "R"
                recall
                do DISP_DEL
                @ 23, 0
                @ 23,15 say "Project has
                    been recovered - " +;
                   PRESS_MSG
                wait ""
              endif

            case PRUP_OPT = "S"
              do case      && PRUP_OPT = "S"

                case STATUS = "C"
                  do ERR_MSG with "Status
                    of a completed " +;
                    "project cannot be
                    changed - " +;
                    PRESS_MSG

                case STATUS $ "AD"
                  if STATUS = "A"
                    store "D" to NEW_STAT
                    store "DISCONTINUED"
                         to NS_WORD
                  else
                    store "A" to NEW_STAT
                    store "ACTIVE" to
                         NS_WORD
                  endif
                  replace STATUS with
                         NEW_STAT
                  do DISP_ST
                  do ERR_MSG with "Status
                    has been changed to "
                    +;NS_WORD +  " - " +
                    PRESS_MSG

                  endcase   PRUP_OPT = "S"

                case PRUP_OPT = "A"
                  store space(10) to
                         MDWGNR
                  store " " to MREV
                  @ 23, 0
                  @ 23,10 say "Enter
                  drawing number and " +;
                  "revision letter: " ;
                  get MDWGNR picture
                  "9999-99999"
                  @ 23,68 get MREV picture
                         "@! A"
                  read
                  if MDWGNR = " "
                    loop
                  endif
                  seek MDWGNR + MREV
                  if .not. eof()
                    do ERR_MSG with "This
                         project already
                         exists - "+;
                         PRESS_MSG
                  else
                    append blank
                    replace DWGNR with
                         MDWGNR, REV with
                         MREV,;STATUS with
                         "A"
                    * Blank out any previ-
                         ous device type
                         data:
                    @ ROW+6,53 say
                         space(2)
                    @ ROW+7,53 say
                         space(15)
                    do GET_PD
                    read
                    do VALIDATE
                    do DISP_ST
                    do DISP_DEL
                    store PROJNR to
```

```
                            MPROJNR
                      select PROJXREF
                      do while .T.
                        append blank
                        replace PROJNR with MPROJNR
                        do GET_DTD
                        read
                        set intensity off
                        store " " to ANOTHER
                        ? chr(7)
                        @ 23, 0
                        @ 23,31 say "Add another?
                            (Y/N) "; get ANOTHER
                            picture "!"
                        read
                        set intensity on
                        if ANOTHER = "Y"
                          @ 23, 0
                          loop
                        else
                          @ 23, 0
                          exit
                        endif
                      enddo
                    endif

                  otherwise
                    ? chr(7)
                endcase         PRUP_OPT
              enddo       PROJECT UPDATE

            case UPDT_OPT = "C"     && CROSSREF UPDATE
              do while .T.      && CROSSREF UPDATE LOOP
                select PROJXREF
                @ 22, 0 clear
                @ 22,32 say "CROSSREF UPDATE"
                @ 23,11 say "(N)ext  (P)rev  (C)hange
                    (D)el   " +;"(R)ecover  (A)dd
                    e(X)it"
                set intensity off
                store " " to CRUP_OPT
                @ 23,70 get CRUP_OPT picture "!"
                read
                set intensity on

                do case      && CRUP_OPT

                  case CRUP_OPT = "N"
                    if .not. eof()
                      skip
                    endif
                    if PROJNR <> MPROJNR
                      skip -1
                      do ERR_MSG with "No further dev.
                            type exists " +;"for this
                            project - " +;PRESS_MSG
                    else
                      do GET_DTD
                      clear gets
                    endif

                  case CRUP_OPT = "P"
                    if .not. bof()
                      skip -1
                    endif
                    if PROJNR <> MPROJNR
                      skip
                      do ERR_MSG with "This is the
                            first dev. " +;"type for
                            this project - " +;
                            PRESS_MSG
                    else
                      do GET_DTD
                      clear gets
                    endif

                  case CRUP_OPT = "C"
                    if .not. eof()
                      do GET_DTD
                      read
                    else
                      do ERR_MSG with "No device record
                            exists " +;"for this pro-
                            ject - " + PRESS_MSG
                    endif

                  case CRUP_OPT = "D"
                    set intensity off
                    store " " to CONFIRM
                    do while .not. CONFIRM $ "YN"
                      ? chr(7)
                      @ 23, 0
                      @ 23,19 say "Are you sure you
                            want to " +;"delete this
                            device? (Y/N) ";get CON-
                            FIRM picture "!"
                      read
                    enddo
                    if CONFIRM = "Y"
                      delete
                      @ ROW+8,48 say "* DELETED *"
                      ? chr(7)
                      @ 23, 0
                      @ 23,19 say "Record has been
                            deleted - " +;PRESS_MSG
                      wait ""
                    endif

                  case CRUP_OPT = "R"
                    recall
                    @ ROW+8,48 say space(11)
                    do ERR_MSG with "Record has been
                          recovered - " +;PRESS_MSG

                  case CRUP_OPT = "A"
                    append blank
                    replace PROJNR with MPROJNR
                    do GET_DTD
                    read

                  case CRUP_OPT = "X"
                    exit

                  otherwise
                    ? chr(7)

                endcase      CRUP_OPT

              enddo       CROSSREF UPDATE LOOP

            otherwise
              ? chr(7)

          endcase     UPDATE SELECTION

        otherwise
          ? chr(7)

      endcase     MAIN_OPT

enddo    MAIN PROGRAM LOOP
return

***************************************
procedure SCREEN
      .
      .

{ Insert SCREEN.PRG (Listing 2 of Part 5) }
      .
      .

***************************************
```

```
procedure GET_PD
       .
       .
{ Insert GET_PD.PRG (Listing 3 of Part 5) }
       .
       .


*************************************
procedure DISP_ST
       .
       .
{ Insert DISP_ST.PRG (Listing 4 of Part 5) }
       .
       .


*************************************
procedure DISP_DEL
       .
       .
{ Insert DISP_DEL.PRG (Listing 5 of Part 5) }
       .
       .


*************************************
procedure GET_DTD
       .
       .
{ Insert GET_DTD.PRG (Listing 6 of Part 5) }
       .
       .


*************************************
procedure VALIDATE

{ Note: This program was unintenionally omitted from Part 5,
 for which I apologize.  It is included here in its entirety.}

do while .T.
   if DWGNR = " "
      ? chr(7)
      @ 23, 0
      @ 23,28 say "Drawing number is blank"
      @ ROW+3,20 get DWGNR picture "9999-99999"
      read
      loop
   endif
   if PROJNR = " "
      ? chr(7)
      @ 23, 0
      @ 23,28 say "Project number is blank"
      @ ROW+3,54 get PROJNR picture "@! 9999-A9999"
      read
      loop
   endif
   if .not. ENGINEER $ "ABC/JAD/JZB/TTT"
      ? chr(7)
      @ 23, 0
      @ 23,26 say "Engineer entry is not valid"
      @ ROW+3,74 get ENGINEER picture "@! AAA"
      read
      loop
   endif
   if dtoc(STARTDATE) = '  /  /  '
      ? chr(7)
      @ 23, 0
    @ 23,30 say "Start date is blank"
      @ ROW+5,23 get STARTDATE
      read
      loop
   endif
   if dtoc(STARTDATE) <> '  /  /  ' .and. dtoc(ESTDATE) <> '  /  /
';
                                            .and. STARTDATE > ESTDATE
      ? chr(7)
      @ 23, 0
      @ 23,17 say "Start date is greater than est.
completion date"
      @ ROW+5,23 get STARTDATE
      read
      loop
   endif
   if dtoc(ESTDATE) = '  /  /  '
      ? chr(7)
      @ 23, 0
      @ 23,26 say "Est. completion date is blank"
      @ ROW+6,23 get ESTDATE
      read
      loop
   endif
   if dtoc(EDITIN) <> '  /  /  ' .and.
dtoc(EDITOUT) <> '  /  /  ';
                                       .and. EDITIN >
EDITOUT
      ? chr(7)
      @ 23, 0
      @ 23,19 say "Edit-in date is greater than
edit-out date"
      @ ROW+7,23 get EDITIN
      read
      loop
   endif
   if dtoc(EDITOUT) <> '  /  /  ' .and.
dtoc(APPDATE) <> '  /  /  ';
                                       .and. EDITOUT >
APPDATE
      ? chr(7)
      @ 23, 0
      @ 23,19 say "Edit-out date is greater than
approval date"
      @ ROW+8,23 get EDITOUT
      read
      loop
   endif
   exit
enddo
return


*************************************
procedure ERR_MSG
parameters MESSAGE

CENTER = 40 - len(MESSAGE)/2
? chr(7)
@ 23, 0
@ 23,CENTER say MESSAGE
wait ""
return
```

Listing 2

```
   * PROJECTS.PRG
   * MASTER PROGRAM FOR PROJECTS
   * WRITTEN BY:  D.COOL       02/27/90
         *
   * SET ENVIRONMENT:
  set talk off
  set bell off
  set deleted off
  set intensity off
  set scoreboard off

  store "Press any key to continue" to PRESS_MSG
  set procedure to PROJPROC

  do while .T.
     * Paint screen:
     clear
```

```
@ 3, 0
text
                ****************************
                *   ENGINEERING PROJECTS   *
                *                          *
                *        MAIN MENU         *
                ****************************

            Select:
                    1. Update/inquiry.
                    2. Monthly report.
                    3. (Reserved)
                    4. Exit to dBASE.
                    5. Exit to system.
endtext

do while .T.
   set intensity off
   store " " to SELECT
   @ 10,29 get SELECT picture "9"
   read

   do case

      case SELECT = "1"
         do MAIN_UPDT
         exit

         case SELECT = "2"
            do REPORTS
            exit

         case SELECT = "4"
            close databases
            set talk on
            set bell on
            set intensity on
            set scoreboard on
            set procedure to
            clear
            return

         case SELECT = "5"
            clear
            quit

         otherwise
            ?? chr(7)
            loop

      endcase
   enddo
enddo
```

Reader Service #149

A REPLACEMENT FOR COMMAND.COM

MIKE WOLFSON
1125 MASTERS AVENUE
ASHLAND, OH 44805

Everyone who uses DOS compatible computers seems to have an opinion about the operating system. Some users love it, some hate it, most tolerate it.. Many people look at the MacIntosh interface and sigh with envy. Unix also has its supporters, as does OS/2. I'm one of those who is basically satisfied with DOS, especially if some minor gripes would be fixed.

Most of my complaints with DOS weren'twith the operating system itself, as much as with the options and capabilities it gave me. In essence, if the command processor, "COMMAND.COM", would have been changed and given more capabilities, I would have beeen more satisfied with it. Someone must have been listening to my thoughts because a substitute for "COMMAND.COM" is available. A shareware program, 4DOS, by Rex C. Conn and Tom Rawson and distributed by J.P. Software is a big improvement over "COMMAND.COM".

OVERVIEW

As a replacement for COMMAND.COM", 4DOS has a number of features that make it a strong competitor. In its normal configuration, it allows you to edit the command line and gives you the ability to use wildcards in filenames with more flexibility than a user normally has with "COMMAND.COM". For example, a user can issue the command:

```
dir *c*.*
```

and get a list of files that have a "c" anywhere in their names. A user can also enter multiple commands on one line by inserting a "^" between each command. If you don't like the default command separator, you have the option of redefining it to one you like better. Additional standard features include: support for EGA/VGA graphics, the ability to redirect the output of error messages, the ability to prevent the overwriting of existing files and the option of executing conditional commands. Finally, 4DOS is compatible with networks that operate under MS-DOS.

The flexibility of the program, though, does not stop there. It offers a number of installation options. If it is installed totally in regular memory, it takes up about 55k of RAM. You have, however, the option of swapping out a portion of the program, either to disk or to EMS memory. If either of these options are selected, only 5K of regular memory is needed. If desired, an on-line help can be installed that includes assistance with 4DOS's internal commands, as well as MS-DOS's external commands.

A number of features can be set to suit your own requirements. A command history list is automatically maintained by the software. The buffer used to keep these commands can be configured to hold from 512 to 8192 characters. Once in the buffer, commands can be reviewed or edited by using the "cursor" keys and the "insert" and "delete" keys.

Another user-definable option is the DOS environment. The size of the environment can be set from 512 to 32000 bytes and the contents of the environment can be edited as desired. Variables used in the environment can also be redefined.

One interesting capability is the ability to add a description of up to 40 characters to each one of your files. The description can be displayed with a normal "DIR" command. One other nice touch is the ability to define a long command into a simple string called an alias. This feature, along with the buffered list of commands, is found in the program CED. CED, though, is an external TSR program that takes up additional memory and, thus, introduces the possibility of interfering with another program. Here, both capabilities are integrated into the command processor as normal features.

Special consideration has also been given to those of us with 286/386 based systems. Two versions of 4DOS are available. The basic version, "4DOS88.EXE", will run on a system with a 8088, 80286, or 80386 processor. A special version, "4DOS286.EXE", specifically for 80286 and 80386 systems, is smaller and faster and is written to take advantage of the additional capabilities of these CPUs.

The 4DOS package includes the following programs:

4DOS.DOC - The 4DOS documentation
4DOS.COMloader for EMS or disk swapping
4DOS88.EXE - The 4DOS program file for any PC- compatible system
4DOS286.EXE - The 4DOS program file for 80286 / 80386 systems ONLY
ALIASES.BAT - Sample aliases
ANSI.SYS - Enhanced ANSI driver
ARC-E.COM - A de-archiving program
ARC-E.DOC - Documentation for ARC-E
DOS.HLP - 4DOS help text file
FIXSTACK.DOC - Documentation for FIXSTACK
FIXSTACK.EXE - The FIXSTACK program
HELP.EXE - The 4DOS help program
INSTALL.BAT - Installation batch file
KEYSTACK.SYS - 4DOS KEYSTACK driver
README.DOC - Last minute revisions
README.1ST - Special README file (on shareware disk only)
REGISTER.FRM - The 4DOS license and registration form
TSRCOM.ARC - Version 2.6 of the TSRCOM utilities from TurboPower Software

The documentation included with the package is better than what I've seen included with many commercial programs, let alone that which normally come with public domain or shareware software. The "4DOS.DOC" file is over 200k in size and yields a set of documentation of about 120 printed pages. The registered copy comes with a professionally printed, 119 page, spiral-bound 6.5"x8.5" manual. This manual includes a table of contents, an index and two appendices. The "README.DOC" file amounts to another 20 pages of last minute details and notes. This documentation is in addition to the instructions for Vernon Berg's ARC-E and TurboPower's TSRCOM utilities. Together, these programs form a flexible and powerful substitute for DOS's normal command processor.

4DOS is written in Microsoft C 5.1 and assembly language. Because at least part of the program is written in C, it will not run as fast as a program written entirely in assembler. I tested this software on a Z386 equipped with 1 meg of RAM, a Seagate 225 and a Seagate 251 hard disk, a Paradise VGA+ card, and a Z1490 monitor. Using the MAPMEM program to check on the amount of free memory I had after booting up, I noted that I had 570,352 bytes of free main memory when I used MS-DOS 3.3+ and "COMMAND.COM". When I substituted "4DOS.COM" and "4DOS286.EXE", I had 568,592 bytes of free main memory.

I configured the 256k of memory available above the 640K boundary as expanded memory using "EMM.SYS". Part of this area, 64k, was devoted to a swap area for use by 4DOS. The rest, 192k, was configured as a disk cache by using "ZCACHE.SYS". For those of you who don't know, the remaining 128k is used by the computer to load the system ROM into fast, random access memory so that these commands will run faster. Zenith calls this "Slushware".

I later added a 4 megabyte memory board to my system, with all memory above the first megabyte set as extended memory and specified that 4DOS use its disk swapping mode. This new configuration worked without a hitch.

Installation

4DOS can be installed either manually or by using a batch file. I chose to install the software using the batch file. I, of course, backed up my C drive first. The batch file is easy to use and checks a number of different things to make sure the setup is correct and to make sure that all the necessary files are copied to the proper directories. The batch file prompts the user as to where to install the working files. It also asks if you are using a computer with a 286 or 386 processor. If you want, you can load some of the support files to a separate subdirectory. I chose to load all of the files to my boot directory.

During installation, your "CONFIG.SYS" file is displayed and you are asked if certain files are already listed in it for loading at bootup. If these files are not listed, the installation first saves your existing "CONFIG.SYS" file as "CONFIG.SAV" and then writes a new "CONFIG.SYS" file with the necessary statements added to it. I particularly liked this feature because it made no assumptions about whether I wanted it done or not, it amended my existing "CONFIG.SYS" file and saved the old one in case I wanted to reuse my old configuration.

Commands

The program includes most of DOS's internal commands, adds additional features to many of them and supplements existing commands with a number of entirely new ones. To illustrate the differences, I am including a list of the commands and a breakdown of what category they fall into. These commands are equivalent to their counterparts in "COMMAND.COM":

BREAK	CALL	CD	CHCP
CHDIR	CLS	CTTY	DATE
EXIT	FOR	GOTO	TIME
VER	VERIFY	VOL	

These commands have had features added to them:

ATTRI	COPY	DEL	DIR
ERASE	MD	MKDIR	PATH
PROMPT	RD	REN	RENAME
RMDIR	SET	SHIFT	TYPE

These commands are new:

?	ALIAS	BEEP	CANCEL
CDD	DESCRIBE	DIRS	ENDLOCAL
ESET	EXCEPT	FREE	GLOBAL
GOSUB	HELP	HISTORY	INKEY
INPUT	KEYSTACK	LIST	MEMORY
MOVE	POPD	PUSHD	QUIT
RETURN	SCREEN	SELECT	SETDOS
SETLOCAL	TEE	TEXT	TIMER
UNALIAS	Y		

For an explanation of how to use the first set of commands, those that are equivalent to their DOS counterparts, I suggest that you consult your DOS manual. I intend to limit my comments to the ways that 4DOS differs from its more well known competitor. I'm going to do that by briefly describing each command.

First of all, I'll cover the commands that have extended capabilities compared to the DOS alternatives. They are:

ATTRIB: This command allows you to set or reset ALL attribute bits (hidden, system, archive, and read only).

COPY: Employing this command allows you to copy more than one file at a time to a target directory. A number of options are also supported. They are:
"/P"prompt user before copying.
"+"combine files just like you normally can.

DEL/ERASE: This command allows you to delete files (including hidden or system files) or subdirectories. It also includes several options. The alternatives are:
"/P" prompt the user before the file is deleted.
"*" wildcard character.

DIR: This is a very versatile command and has many options. They are:
"/1" one column display with file name, size, date, time and description. This is the default.
"/2" two column display with name, size, date and time.
"/A" display name and attributes only (R-read-only, H-hidden, S-system, A-archive).
"/D" display current directory and all of its subdirectories.
"/F" display files only; no subdirectories.
"/H" display hidden files.
"/J" justify the output.
"/W" display in multiple columns.
"/P" stop at end of each page.

These flags can also be combined. For example, you can display all files, including hidden files, in wide format and stop at the end of each page by typing the command:
DIR/H/W/P

MD/MKDIR: This command can use ".." to designate the directory to start at the parent directory; otherwise, the syntax is the same.

PATH: This command is similar to its DOS counterpart. It does have one useful difference from its DOS counterpart, though. When this command is used to search for a file in a particular path, it will disregard invalid path names. The command can also be used to display current path.

PROMPT: In addition to the normal characters that DOS can use, 4DOS adds the following:
"a" - the "&" character.
"c" - open parenthesis.
"f" - close parenthesis.
"s" - space character.

The default prompt at bootup is "PG". If a small "g" is specified the path is displayed in lower case letters.

RD/RMDIR: This command is used to erase directories. It is similar to the DOS command. Wildcard characters are permitted in the directory names.

REN/RENAME: This command is similar to its DOS alternative. Wildcard characters are allowed in the name of the file to be renamed. Files may be renamed to another directory on the same drive. One option is permitted. It is:
"/P" prompt to confirm action before the file is renamed.

SET: This command is used to display, create, modify or delete environment variables. Variable names are limited to 80 characters and variable values are limited to 255 characters.

TYPE: This directive is similar to the DOS command. Two flags can be invoked in the command line, though. They are:
"/L" prints a line number before each line.
"/P" stops output at the end of each page.

If you only consider the enhancements that 4DOS has made to normal DOS commands, you would be wise to consider it as a replacement for what Microsoft has to offer. These enhancements, though, are only a taste of the capabilities that J.P. Software has added in their package. Just as I've already done, I'll review each new command by explaining what it does.

?: By issuing this command, a list of 4DOS commands is displayed.

ALIAS: An "alias" is a new name for an existing command, or an entirely new command created by combining existing commands. Issuing the "ALIAS" command, causes a list of user-defined aliases to be displayed on the screen, including replaceable parameters. Aliases can also be nested to create new aliases.

CDD: Using this command, allows the user to change the default drive and directory to the one indicated in the command line.

DESCRIBE: This command adds a 40 character description to DOS filenames. The description can be listed by use of the "DIR" command with the single column option or with the "SELECT" command.

DIRS: Command used to display the current directory stack.

SET: Command used to set the environment variables in aliases or to edit the command line.

EXCEPT: This command allows one file, or a group of files, to be excluded during the execution of a DOS command.

FREE: This directive is used to display on the screen the amount of free space on the disk specified. If no disk is designated, free space on the currently logged drive is shown. This alternative works faster than CHKDSK.

GLOBAL: When used in a command, it executes the command on the current directory and all subdirectories.

HELP: The user is provided with an online help for MS-DOS and 4DOS commands. "HELP" will display the proper command syntax and a description of the command.

HISTORY: This command displays on the screen a list of all commands in the current command history list. One option is available:
"/F" clears the command history list.

KEYSTACK: The user may input keystrokes to a program through an alias or a batchfile. This command is also described in the section of this article dealing with "Batch Files".

LIST: This command generated similar output to Vern Buerg's LIST program. It has the capability to scroll or page forward or backward. This command also has the ability to search for a text string.

MEMORY: This command is used to display system memory status, including the status of any expanded or extended memory present in the system. It will also show the amount of free environment space.

MOVE: This command allows the user, in one operation, to copy a file (or files) from one drive/directory/subdirectory to another drive/directory/subdirectory and then delete the original file in one operation.

POPD: This command is used to allow the user to change to the drive and directory at the top of the directory stack. It is used in conjunction with the "PUSHD" command. One option is permitted. It is:
"*" input after the "POPD" command, clears the directory stack.

PUSHD: When this command is used, it saves the current drive and directory to the top of the directory stack. The directory stack can hold up to 255 characters (about 10 or 20 drive/directory entries). It is used in conjunction with the "POPD" command.

SELECT: This command lets the user execute a command on a file or a number of files by marking the files to be affected from a full-screen listing of all the files in the specified directory/subdirectory. Several options are available:
"/e" sort files by extension.
"/t" sort files by time created or last changed (oldest first).
"/z" sort by file size.
"/u" files are not sorted.
"/r" reverse the sort order.

SETDOS: This command is used to set or reset 4DOS configuration variables. Several options can be set by the user:
"/C" can be used to define the character used to separate multiple commands on the same line. Default is " ^ ". The one limitation of this command is that you cannot use the redirection "< >" or piping "|" characters.
"/E" can be used to define alternative escape argument. Default is "X". A limitation of the command is that it cannot use redirection or piping characters.
"/H" defines the minimum command length to be saved in the history list.
"/M" defines the default line editing mode.
"/N" defines the status of the "NOCLOBBER" variable. If set to active, then existing files will not be destroyed by the output of redirection. When set and redirection is used to append data, the file to be appended must already exist..
"/V" defines the VERBOSE variable and affects whether the input of batch files will be echoed.

TEE: This directive copies the standard input to the standard output and saves a copy in the specified file. One option is available:
"/A" indicates to the user that the input is to be appended to an existing file rather than having the existing file be overwritten.

TYPE: This command is used to send the output of a command to a file. Two user-selectable options are available to enhance the use of this command. They are:
"/L" numbers each line of output displayed on the screen.
"/P" causes the output to pause at the end of each page.

UNALIAS: When used, it removes an alias from the alias list..

Y: This command will copy the input to a specified output file or device and copy the file identified in the command to the specified output. After executing a command, DOS returns a result called an exit code.

Conditional commands allow the user to perform tasks based on the exit code returned by the previous command. There are two conditional commands in 4DOS. They are:

&&: If the "&&" conditional command is used, the second command will only be executed if the first command returns an exit code of "zero".

||: If the "||" conditional command is used, the second command will only be executed if the first command returns a "non-zero" exit code.

As in "COMMAND.COM", some of these commands (for example, "DIRS", "MEMORY" and "FREE") are used internally by 4DOS. Other commands (i.e., "CANCEL", "GOSUB" or "SCREEN") are external to the program and support enhanced capabilities of the package.

Batch File Commands

The commands and options available for use in batch files are a welcome addition to anyone who has struggled with the limitations imposed by DOS. Besides the normal batch language commands, 4DOS adds the following:

BEEP: When used, this causes the system's speaker to beep.

CANCEL: When issued, this command will terminate nested batch files.

GOSUB: This command is used to call a subroutine from within a batch file.

INPUT/INKEY: This command provides the user with the capability to input variables from the keyboard while in a batch file.

KEYSTACK: This command will send keystrokes to a program just as if they had been entered from the keyboard.

QUIT: This command causes the batch file to terminate and returns the user to the command line.

RETURN: This command is used to exit a subroutine and return to the batch line on the line following the "GOSUB" that invoked the subroutine.

SCREEN: A command to allow the user to position the cursor on the screen and if desired, to display some text..

SETLOCAL/ENDLOCAL: In combination, these commands are used to define a local environment from within the batch file by saving the original environment, disk, and directory and by restoring them when the batch file has finished executing.

SHIFT: Instead of being limited to DOS' 10 parameters, 4DOS supports 128 parameters. It is also possible to reverse direction by giving a negative value to the parameter.

TEXT/ENDTEXT: This command combination allows the user to display a block of text on the screen.

TIMER: When used, this command turns the system timer on and off.

Compatibility

The "README.DOC" file included with the software contains a section addressing the compatibility of 4DOS with other software and with "COMMAND.COM". This section is extensive and is very informative. In my opinion, the incompatibilities are not serious and can be worked around. As with many programs, there will be problems when working with some software packages. These incompatibilities, however, are not major enough to deter me from using 4DOS.

APPEND Command

One area of incompatibility lies with DOS' "APPEND" command. 4DOS currently does not support the internal portion of the DOS "APPEND" command. This means that you can boot up with the "APPEND" command in your "AUTOEXEC.BAT" file or from the command line and have it execute properly. If, however, you want to change the "APPEND" path you must run "APPEND" from "COMMAND.COM" instead of from 4DOS. The "/X" switch can be used, and it will affect 4DOS directory searches for many 4DOS commands (as it does for "COMMAND.COM"). To use the "APPEND" "/E" switch, 4DOS must be loaded with the "/M" command line switch to set up a standard master environment.

TIME and DATE Commands

Some manufacturers distribute versions of DOS that automatically set the hardware clock date and time (this information is maintained in battery-backed RAM while the system is turned off). This particular time and date is different from the DOS date and time when the corresponding DOS command is executed. This method is hardware-specific and may not function as you might expect under 4DOS. If you normally set the hardware clock's date and time with DOS commands, you should either run a secondary copy of "COMMAND.COM" to set them or invoke your system's "SETUP" or other utility program to do so. While testing 4DOS, I did not notice any problems with this command.

SELECT and MOVE Commands

J.P. Software also recommends that the user exercise some caution when specifying the target in a "SELECT MOVE" command. If multiple files are being acted on using the "SELECT ... (filename) ..." syntax and the target is not an existing directory name, each file will be "MOVEd" to the same target name so that the last file relocated will be overwritten by the next file. The result will be that only the last file moved will remain; the rest will have been consecutively overwritten. This problem can be overcome by invoking "SELECT" with square brackets instead of parentheses. All file names are placed on one line and the built-in protection in "MOVE" will then prevent this error from occurring.

ENVIRONMENT Variable and Stack Problems

Environment variables are not stored by 4DOS in the same way as they are in "COMMAND.COM". 4DOS normally stores the master copy of the environment by swapping it out of memory when in memory-swapping mode. Programs which search for the master environment, in order to change it, will not be able to find it. Even when 4DOS is loaded in memory-resident mode, some programs may have trouble finding the master environment. If you want to run software that modifies the master environment, the size of the 4DOS environment can be set with the "/M" switch instead of the "/E" switch and a standard master environment will be created.

This change only needs to be done when running programs that modify the MASTER environment. This change is not needed to run programs that modify the environment, but that leaves the master environment unaltered.

When starting a secondary shell, the 4DOS loader searches for the "4DOS.EXE" file on the drive and path specified in the "COMSPEC" environment variable. Therefore, if you set "COMSPEC" manually, you must include the full drive, path and directory name for "4DOS.COM". If you fail to do this, 4DOS will not be able to find its files. In addition, the "4DOS.EXE" files must be located in the directory specified by "COMSPEC".

Because of the way 4DOS stores command aliases in the environment, some programs compiled with certain popular C compilers can create stack overflow problems. These compilers cause the environment to be copied to the stack before executing the program. If you have created a lot of aliases, and thus are using a lot of environment space, the combination can cause a number of problems when you try to load or execute these programs. These problems take the form of "Stack Overflow", "Insufficient Memory", or "No Room for Environment" messages. By allocating a larger environment at startup with the "/E" installation option, most of these problems can be solved.

If this problem persists, there are two other solutions. The first solution is to use the "FIXSTACK" program included with 4DOS to modify the affected program's stack size. This program, however, can only be used with "EXE" type files. In some cases, modifying the stack size will not solve the problem. The second alternative is to reduce the environment size by removing all aliases before the affected program is run. This alternative will always work.

ANSI Command Sequences

ANSI command sequences also are handled a little differently with 4DOS. Just like "COMMAND.COM", ANSI escape sequences can be sent to the screen, or to the printer, to control its operation. The main difference is, that except for defining the prompt, you cannot send ANSI escape sequences to the screen using the "PROMPT" command in a batch file as you can with "COMMAND.COM" . The reason for this is that 4DOS does not display the prompt for each line of the batch file like "COMMAND.COM" does. If you have to send ANSI escape sequences to the screen from inside a batch file (for example, to set function key values), J. P. Software suggests that you use the "ECHO" command to transmit them, not the "PROMPT" command.

Networks

DOS-compatible networks should work properly with 4DOS. According to the 4DOS manual, network users should be sure that more than one user is not using the same physical drive and directory as a disk swap file. If multiple users are using the same drive/directory, the files will overwrite each other and cause problems for some of the users. In addition, earlier versions of 4DOS did not work correctly with Novell's "NET3". The problem has been corrected and "NET3" should work properly provided that you specify use of a standard master environment as described above. If problems do occur with "NET3", J.P. Software suggests that you try loading "COMMAND.COM" as your primary shell, next loading "IPX" and "NET3" from "AUTOEXEC.BAT", and finally starting 4DOS.

Multitasking Software

Most multitasking systems, such as Windows, DESQview, Software Carousel and DoubleDos can be used with 4DOS. Usually, 4DOS will work properly when it is loaded as the root shell loaded "under" the operating environment or when run inside a window.

To start 4DOS from inside a window in any of these environments, you can define it as a program to be run directly or you can have the task switcher load it automatically as the command processor by leaving "COMSPEC" set to 4DOS. If run directly, you can also set these parameters on the 4DOS command line.

In most cases, the "COMSPEC" environment variable can be set to control whether 4DOS or "COMMAND.COM" is used to run batch files and to open windows that are not closed when exiting to DOS. As an alternative, setting the "COMSPEC" variable to "COMMAND.COM" before starting the operating environment and then loading 4DOS in its own window can be slightly more efficient. In most cases, however, it is suggested that you leave the "COMSPEC" set to 4DOS.

When 4DOS is loaded in low memory before starting the multitasking software, it will monitor the copies of 4DOS in each window and will assign each a unique "shell number" which is used as the extension for the disk swap file ("4DOSSWAP.000", "4DOSSWAP.001", etc.). No conflicts will occur using this method of operation.

If, however, 4DOS is loaded after the multitasker, then 4DOS will not be able to monitor the copies running in each window and each copy of 4DOS will assume it is the only copy being run and each will use the same swap file name, "4DOSSWAP.000". To avoid conflicts, each copy of 4DOS started in a window in disk swapping mode has to be run from a different directory, or with a specific "/S:D" switch that specifies a unique swapping directory for that window. This problem will only occur if disk swapping is used in more than one window as described above. There is no similar conflict with EMS swapping.

To make operating under DESQview easier, J.P. Software has included a version 2.24. In order to install 4DOS under DESQview, you should refer to the 4DOS and DESQview manuals. The 4DOS manual, for example, includes detailed information about configuring DESQview to recognize 4DOS.

Software Carousel and DoubleDos are compatible with 4DOS and will run without difficulty as long as you run 4DOS from within either program. 4DOS can only be run from inside a Software Carousel partition or inside DoubleDos. Both multitaskers expect "COMMAND.COM" to be the system command processor and both rely on logic found in "COMMAND.COM" and the way it is written. This makes it impossible to write a program which works properly as an alternate command processor loaded underneath these products.

The best method to run 4DOS inside a Carousel partition is to leave the "COMSPEC" set to "COMMAND.COM" when Carousel is loaded; then set the "CAROUSEL.OPT" file using the "PROGRAMn" lines to load 4DOS into a partition.

Because 4DOS can only be loaded in a partition when running Software Carousel, using 4DOS disk swapping in multiple partitions can cause the same problems noted when using DESQview.

J.P. Software has also identified incompatibilities with Bourbaki's 1DIR+ shell. The program can be used when the system is started under 4DOS, but you can't run 1DIR+ directly from 4DOS or utilize 4DOS functions easily from 1DIR+. The 4DOS manual suggests that the user set COMSPEC to "COMMAND.COM" and then load 1DIR+. After exiting 1DIR+, type "EXIT" to return to 4DOS.

Sidekick and Sidekick Plus

As with a lot of software, Sidekick and Sidekick Plus have compatibility problems when used with 4DOS. You cannot use Sidekick's internal unload command or you may hang your system. The manual suggests that you use at least version 2.6 of the "MARK" and "RELEASE" utilities to unload Sidekick and Sidekick Plus. Earlier versions are not compatible with 4DOS.

Operation

After having used 4DOS for several months, I can say with confidence that I am entirely satisfied with it. In fact, during the course of preparing this review, I sent in my registration and am now running a fully registered copy of the software. I find that operation of the program is transparent to the user and satisfies 95% of the shortcomings I felt "COMMAND.COM" had. Best of all, the software seems to be pretty "bullet proof". In short, I am quite happy with the software and don't have any plans of going back to the alternative from Microsoft. I have not run into any compatibility problems with the registered copy of the program. When using the shareware copy, 4DOS sometimes had problems finding "Q.COM", the text editor I use from within PROCOMM+. The registered copy did not have the same problem. The other software I normally run includes GENERIC CADD, PC-FILE+, WORD PERFECT, QUICKBASIC 4.5, PKZIP/PKUNZIP, Fastback, WindowDos" and various utilities. All run without problems.

Final Notes

Based on my experience with 4DOS, I feel that I can recommend it without reservation. I wouldn't be saying this if I wasn't very satisfied with the program. I am so satisfied that, as I said, I bought a registered copy. I now run 4DOS exclusively on my Z-386 as well as on my Z-183. In my opinion, 4DOS is an excellent alternative to Microsoft's "COMMAND.COM" and should be considered by anyone wanting an enhanced command processor.

4DOS can be downloaded from many bulletin boards*, or directly from J.P. Software at the following address:

J.P. Software
P.O. Box 1470
E. Arlington, Massachusetts 02174
617-646-3975

If you want to try the shareware version of the program, but don't have a modem or can't find it on a bulletin board, send me a disk (either 5.25" or 3.5"; 360k, 1.2 meg, 750k, 1.4meg capacity) and a stamped, self-addressed, disk mailer and I'll make you a copy. Please make sure the mailer has adequate postage and that it is clearly addressed. My name and address is at the beginning of this article.

* Including "COM1", the official ZDS Bulletin Board.

Continued from Page 9

ry and modem accessories far too early in the life cycle. They liquidated thousands of them and they are now being sold in liquidation catalogs at far below the original dealer cost.

Above we gave you news on the memory expansion. Now we bring you the rest of the story, how to use a modem with the mouse port. The combination of these two items will help those who have the EaZy PC systems. Many thanks to H/Z guru Dave Brockman for this info.

The mouse port appears to be standard 8250-type serial port with an AT-type 9-pin serial port minus a few input signals. In reality, there is no 8250 serial chip in the EaZy PC which is the standard UART used for serial port/modem communication. It is simulated instead using the serial port built into the V40 microprocessor used by the EaZy PC.

The mouse port is physically addressed as COM2, but because of the way things work, it will be recognized by DOS as COM1 if it is the only serial port present. This means that is you want to use the DOS MODE command to set up the port, use COM1. However, if you have a program (like QMODEM) which directly access the serial port, use COM2. The mouse port supports only the standard baud rates (110, 150, 300, 600, 1200, 2400, 4800, 9600 and 19200.) Non-standard rates will not work.

The missing signals are Carrier Detect (pin 1), Data Set Ready (pin 6) and Clear To Send (pin 8). Because of the way the EaZy PC emulates the serial port, these signals will always appear to be true when read by software. Whether this will have any effect depends on the modem program being used. Some programs won't mind; some will.

The mouse port has been used with the popular shareware modem program, QMODEM, and a 1200 baud Hayes compatible modem successfully. To do this, a special cable was made up and wired as follows:

EaZy PC DB9S	Modem DB25P
2	3
3	2
4	20
5	7

Note: The cable can be purchased from Radio Shack as part number 26-269.

The first time that QMODEM was run, it was configured via the Setup menu for a generic Hayes clone (Modem, Quick Modem Config!), the COM2 port (Modem, Communications Parameters, J, 2) and the 16550 UART Enable was set off (Options, Runtime Parms, 16550 UART Enable).

It has been found that a public domain modem program which can be downloaded from many bulletin boards under the archieved name of BOYAN40A.ZIP works best with the EaZy PC and the above cable.

Graphics Printer or Epson FX - Part 7

John A. Day
5 Rue Sauer
77500 Chelles, France

In the July issue of REMark, we inadvertently left out two programs from John's article. So, we have reprinted the Listings below. We're sorry for any inconvenience this may have caused.

Annexe 1
Font Design Program

```
10 REM Input/Modification of download fonts
20 REM Demonstration version, February 1989
30 REM Author: John Day, Paris (France)
40 DEFINT A-Z
50 KEY OFF
60 DIM BLOC(3,10),W(10) ' blobs for pins on screen
70 DIM MSK(7) ' weights from 1 to 128
80 DIM T(255,10,1),TA(255),TD(255) ' nearly 255 printable characters
90 DEF SEG =&H40
100 S1$ = "78946123Ff" + CHR$(13) + CHR$(27)
110 S2$ = CHR$(71) + CHR$(72) + CHR$(73) + CHR$(75) + CHR$(77)
120 S2$ = S2$ + CHR$(79) + CHR$(80) + CHR$(81)
130 WGT = 1
140 FOR I = 7 TO 0 STEP -1
150   MSK(I) = WGT
160   WGT = 2 * WGT
170 NEXT I
180 SCREEN 2
190 XBS = 190 ' screen area for drawing character
200 YBS = 20
210 DB$ = CHR$(9)+"DB"+CHR$(9)+"27,38,0"
220 ' CMV$: characters which can't be shown on the screen
230   CMV$ = CHR$(7) + CHR$(9) + CHR$(10) + CHR$(11) + CHR$(12) + CHR$(13)
240   CMV$ = CMV$ + CHR$(21) + CHR$(28) + CHR$(29) + CHR$(30) + CHR$(31)
250 LINE (XBS,YBS)-(XBS+7,YBS+5),,BF
260 GET (XBS,YBS)-(XBS+7,YBS+5),W ' white blob
270 FOR I = 0 TO 10
280   BLOC(1,I) = W(I)
290 NEXT I
300 PUT (XBS,YBS),W,PRESET
310 GET (XBS,YBS)-(XBS+7,YBS+5),W ' black blob
320 FOR I = 0 TO 10
330   BLOC(0,I) = W(I)
340 NEXT I
350 LINE (XBS+2,YBS+1)-(XBS+5,YBS+4),,BF
360 GET (XBS,YBS)-(XBS+7,YBS+5),W ' black blob with cursor
370 FOR I = 0 TO 10
380   BLOC(2,I) = W(I)
390 NEXT I
400 PUT (XBS,YBS),W,PRESET
410 GET (XBS,YBS)-(XBS+7,YBS+5),W ' white blob with cursor
420 FOR I = 0 TO 10
430   BLOC(3,I) = W(i)
440 NEXT I
450 CLS ' blank and surround drawing box
460 LINE (XBS-1,YBS-1)-(XBS+88,YBS+54),,B
470 LOCATE 1,1
480 INPUT "Input file";FI$
490 IF FI$<>"" THEN GOSUB 1010
500 LOCATE 7,1
510 PRINT "Ctrl-<--¦ to end"
520 GOSUB 600
530 LOCATE 2,1
540 INPUT "Output file";FI$
550 IF FI$<>"" THEN GOSUB 1060
560 KEY ON
570 LOCATE 23,1
580 END
590 ' main program loop
600 LOCATE 3,1
610 PRINT "Character:     ";
620 LOCATE ,12
630 C$ = INKEY$
640 IF C$="" THEN 630
650 IF C$="CC" THEN C$ = "C" ' known BASIC/keyboard bug
660 IF LEN(C$)>1 THEN BEEP: GOTO 630 ' not in ASCII set
670   IF (C$=chr$(10) OR C$=chr$(27)) AND (PEEK(23) AND 4) THEN RETURN
680 C = ASC(C$)
690 IF INSTR(CMV$,C$) THEN PRINT " "ASC(C$) ELSE PRINT C$;ASC(C$)
700 GOSUB 1610 ' blank work zone and draw existing pattern
710 DBL = 0
720 GOSUB 2220 ' draw new pattern
730 GOSUB 2010 ' check if exists and set attribute
740 GOTO 600
1000 '
1010 OPEN "I",#1,FI$ ' read ASM file
1020 GOSUB 1120
1030 CLOSE 1
1040 RETURN
1050 '
1060 OPEN "O",#1,FI$ ' write ASM file
1070 GOSUB 1340
1080 CLOSE 1
1090 RETURN
1100 '
1110 ' read 1 line of ASM DBs
1120 IF EOF(1) THEN RETURN
1130 INPUT# 1,D$,D$,D$,D$ ' skip Esc & 0 n
1140 INPUT# 1,I,TA(I)
1150 IF TA(I) AND 128 THEN GOSUB 1200 ELSE GOSUB 1260
1160 TD(I) = -1
1170 GOTO 1120
1180 '
1190 ' straight read of font description, pins 1 thru 8
1200 FOR J = 0 TO 10
1210  INPUT# 1,T(I,J,0)
1220 NEXT J ' pin 9 not used
1230 RETURN
1240 '
1250 ' read and shift down to pins 2 thru 9
1260 FOR J = 0 TO 10
1270  INPUT# 1,T(I,J,0)
1280  IF T(I,J,0) AND 1 THEN T(I,J,1) = 128
1290  T(I,J,0) = T(I,J,0) \ 2
1300 NEXT J
1310 RETURN
1320 '
1330 ' write characters which have been defined
```

```
1340 FOR I = 1 TO 255 ' loop thru font table
1350  IF TD(I) THEN GOSUB 1400
1360 NEXT I
1370 RETURN
1380 '
1390 ' print (tab)DB(tab)27,38,0,n,n,a or DB Esc,"&",0,n,n,a
1400 I$ = RIGHT$(STR$(I),LEN(STR$(I))-1)
1410 PRINT# 1,DB$","I$","I$","RIGHT$(STR$(TA(I)),LEN(STR$(TA(I)))-1);
1420 IF TA(I) AND 128 THEN GOSUB 1470 ELSE GOSUB 1530
1430 PRINT# 1,"" ' add cr/lf
1440 RETURN
1450 '
1460 ' add pin data
1470 FOR J = 0 TO 10
1480  PRINT# 1,","RIGHT$(STR$(T(I,J,0)),LEN(STR$(T(I,J,0)))-1);
1490 NEXT J
1500 RETURN
1510 '
1520 ' shift pin data left 1 bit and add to line
1530 FOR J = 0 TO 10
1540  TC = T(I,J,0) * 2
1550  IF T(I,J,1) THEN TC = TC + 1
1560  PRINT# 1,","RIGHT$(STR$(TC),LEN(STR$(TC))-1);
1570 NEXT J
1580 RETURN
1590 '
1600 ' draw character on screen
1610 LINE (XBS,YBS)-(XBS+87,YBS+53),0,BF
1620 IF NOT TD(C) THEN TF = 0: LOCATE 5,1: PRINT "  ": N = 2: GOSUB 1880:
     RETURN
1630 FOR I = 0 TO 10
1640  W(I) = BLOC(1,I)
1650 NEXT I
1660 X = XBS ' left margin
1670 C2 = 0
1680 FOR J = 0 TO 10 ' 11 columns
1690  Y = YBS ' top margin
1700  FOR K = 0 TO 8 ' 9 pins
1710    L = K \ 8 ' 1 if pin 9
1720    M = K MOD 8 ' bits 0 thru 7
1730    IF T(C,J,L) AND MSK(M) THEN PUT (X,Y),W,PSET
1740    Y = Y + 6
1750  NEXT K
1760  X = X + 8
1770  IF T(C,J,0) OR T(C,J,1) THEN C2 = J
1780 NEXT J
1790 IF T(C,0,0) AND 128 THEN N = 3 ELSE N = 2
1800 GOSUB 1880
1810 LOCATE 5,1
1820 TF = 0 ' suppose auto width
1830 IF T(C,0,0)=0 AND T(C,0,1)=0 AND (TA(C) AND 112)=0 THEN TF = -1:
     GOTO 1850
1840 IF (TA(C) AND 15)=11 AND C2<8 THEN TF = -1
1850 IF TF THEN PRINT "Fix" ELSE PRINT "  "
1860 RETURN
1870 ' set cursor for new character
1880 FOR I=0 TO 10
1890  W(I) = BLOC(N,I)
1900 NEXT I
1910 X = XBS
1920 Y = YBS
1930 PUT (X,Y),W,PSET ' cursor
1940 XC = 0
1950 YC = 0
1960 YCA = 0
1970 YCB = 0
1980 RETURN
1990 '
2000 ' set attribute byte
2010 C1 = 11 ' first non-blank column
2020 C2 = 11 ' last ditto
2030 P9 = 0 ' pin 9 not used
2040 FOR J = 0 TO 10
2050  FOR K = 0 TO 1
2060    IF T(C,J,K) THEN C2 = J: IF C1=11 THEN C1 = J:
2070  NEXT K
2080  IF T(C,J,1) THEN P9 = -1:
2090 NEXT J
2100 C2 = C2 + 3 ' add 3/120" gap
2110 IF C2 > 11 THEN C2 = 11 ' full 1/10"
2120 ' if no points and auto width, mark no character
2130 IF (C1 = 11) AND NOT TF THEN TD(C) = 0: RETURN
2140 TD(C) = -1
2150 IF C1>7 THEN C1 = 7 ' max for 3-bit field
2160 ' attribute '11' [or 139] is full-width, cols 0 to 11
2170 IF TF THEN TA(C) = 11 ELSE TA(C) = (16 * C1) OR C2
2180 IF NOT P9 THEN TA(C) = TA(C) OR 128
2190 RETURN
2200 '
2210 ' main drawing routine
2220 Z$ = INKEY$
2230 IF Z$ = "" THEN 2220
2240    IF LEN(Z$)=1 THEN Z = INSTR(S1$,Z$) ELSE Z = INSTR(S2$,RIGHT$(Z$,1))
2250 IF Z=0 THEN 2220
2260 IF Z>10 THEN RETURN
2270 IF Z>8 THEN LOCATE 5,1: IF TF THEN TF = 0: PRINT "  ":
     GOTO 2220 ELSE TF = -1: PRINT "Fix": GOTO 2220
2280 IF PEEK(23) AND 2 THEN IF DBL THEN DBL = 0 ELSE GOSUB 2450 ELSE DBL = 0
2290 IF PEEK(23) AND 1 THEN GOSUB 2780
2300 IF T(C,XC,YCA) AND MSK(YCB) THEN N = 1 ELSE N = 0
2310 FOR I=0 TO 10
2320  W(I) = BLOC(N,I)
2330 NEXT I
2340 PUT (X,Y),W,PSET ' erase old cursor
2350 ON Z GOSUB 2870,2870,2870,3030,3080,2950,2950,2950
2360 ON Z GOSUB 3030,2850,3080,2850,2850,3030,2850,3080
2370 IF T(C,XC,YCA) AND MSK(YCB) THEN N = 3 ELSE N = 2
2380 FOR I = 0 TO 10
2390  W(I) = BLOC(N,I)
2400 NEXT I
2410 PUT (X,Y),W,PSET ' new cursor
2420 GOTO 2220
2430 '
2440 ' add a blob
2450 Q9 = 0
2460 GOSUB 2570
2470 IF Q9 THEN BEEP: RETURN
2480 T(C,XC,YCA) = T(C,XC,YCA) OR MSK(YCB)
2490 DBL = -1
2500 FOR I = 0 TO 10
```

```
2510  W(I) = BLOC(0,I)
2520 NEXT I
2530 IF XC>0 THEN GOSUB 2690
2540 IF XC<10 THEN GOSUB 2730
2550 RETURN
2560 ' check pins 1 and 9
2570 IF YC = 0 THEN GOSUB 2600
2580 IF YC = 8 THEN GOSUB 2640
2590 RETURN
2600 FOR J = 0 TO 10
2610  IF T(C,J,1) THEN Q9 = -1
2620 NEXT J
2630 RETURN
2640 FOR J = 0 TO 10
2650  IF T(C,J,0) AND 128 THEN Q9 = -1
2660 NEXT J
2670 RETURN
2680 ' check pin position to immediate left
2690 IF T(C,XC-1,YCA) AND MSK (YCB) THEN PUT (X-8,Y),W,PSET
2700 T(C,XC-1,YCA) = T(C,XC-1,YCA) AND NOT MSK(YCB)
2710 RETURN
2720 ' check pin position to immediate right
2730 IF T(C,XC+1,YCA) AND MSK(YCB) THEN PUT (X+8,Y),W,PSET
2740 T(C,XC+1,YCA) = T(C,XC+1,YCA) AND NOT MSK(YCB)
2750 RETURN
2760 '
2770 ' rub out
2780 T(C,XC,YCA) = T(C,XC,YCA) AND NOT MSK(YCB)
2790 RETURN
2800 '
2810 ' cursor move routines
2820 '
2830 ' dummy second shift for orthogonal moves
2840 ' (diagonal moves are first up/down, second left/right)
2850 RETURN
2860 ' move up 1
2870 DBL = 0
2880 IF YC = 0 THEN RETURN
2890 Y = Y - 6
2900 YC = YC -1
2910 YCA = YC \ 8
2920 YCB = YC MOD 8
2930 RETURN
2940 ' move down 1
2950 DBL = 0
2960 IF YC=8 THEN RETURN
2970 Y = Y + 6
2980 YC = YC + 1
2990 YCA = YC \ 8
3000 YCB = YC MOD 8
3010 RETURN
3020 ' move left 1
3030 IF XC = 0 THEN RETURN
3040 X = X - 8
3050 XC = XC - 1
3060 RETURN
3070 ' move right 1
3080 IF XC = 10 THEN RETURN
3090 X = X + 8
3100 XC = XC + 1
```

Annexe 2
Bar Charting Program

```
10 REM Quick bar charting
20 REM Demonstration version, September 1989
30 REM Author: John Day, Paris (France)
40 DEFINT A-U,W-Z
50 KEY OFF
60 DATA 139,56,68,0,130,0,130,0,68,56,0,0
70 DATA 169,0,0,66,0,254,0,2,0,0,0,0
80 DATA 139,66,128,6,128,10,128,18,128,98,0,0
90 DATA 139,132,0,130,0,162,0,210,0,140,0,0
100 DATA 139,8,16,40,64,136,0,254,0,8,0,0
110 DATA 139,228,2,160,2,160,2,160,2,156,0,0
120 DATA 139,12,18,32,82,128,18,0,18,12,0,0
130 DATA 139,128,0,130,4,136,16,160,64,128,0,0
140 DATA 139,108,146,0,146,0,146,0,146,108,0,0
150 DATA 139,96,144,0,144,2,148,8,144,96,0,0
160 DATA 167,0,0,6,0,6,0,0,0,0,0,0
170 DIM T(10,11),TA(10) ' pin data, attributes, 0-9 .
180 DIM L1(780),L2(780) ' line buffers, 6-1/2" x 120 dpi
190 FOR I = 0 TO 10
200   READ TA(I) ' attribute byte
210   FOR J = 0 TO 10
220     READ T(I,J) ' data bytes
230   NEXT J
240 NEXT I
250 DEF SEG = &H40
260 LP1 = PEEK(8) + 256 * PEEK(9)
270 OPEN "I",#1,"BAR.DTA"
280 GOSUB 400
290 Z = 12 ' Form Feed
300 GOSUB 1260
310 KEY ON
320 END
380 '
390 ' main program loop
400 IF EOF(1) THEN RETURN
410 INPUT# 1,V$
420 V = VAL(V$)
430 W = V * 12 ' convert to 1/120"
440 IF W>780 THEN W = 780: OFLW = -1 ELSE OFLW = 0
450 L1(0) = 255 ' vertical rule
460 L2(0) = 255
470 FOR I = 2 TO W STEP 2
480   L1(I) = 64 ' pin 2
490   L2(I) = 8 ' pin 5
500 NEXT I
510 IF NOT OFLW THEN L1(W-1) = 0: L1(W) = 127: L2(W-1) = 0: L2(W) = 248
520 PW = 5 ' sum proportional width
530 FOR I = 1 TO LEN(V$)
540   IF MID$(V$,I,1)="." THEN C = 10 ELSE C = VAL(MID$(V$,I,1))
550   PW = PW + (TA(C) AND 15) + 1 - (TA(C) AND 112) \ 16
560 NEXT I
570 IF PW > W THEN L1(W) = L1(W) AND 224: L2(W) = L2(W) AND 31: W = PW
580 C1 = W + 4 - PW ' start column
590 FOR I = 1 TO LEN(V$)
600   IF MID$(V$,I,1)="." THEN C = 10 ELSE C = VAL(MID$(V$,I,1))
610   FOR J = ((TA(C) AND 112)\16) TO (TA(C) AND 15)
```

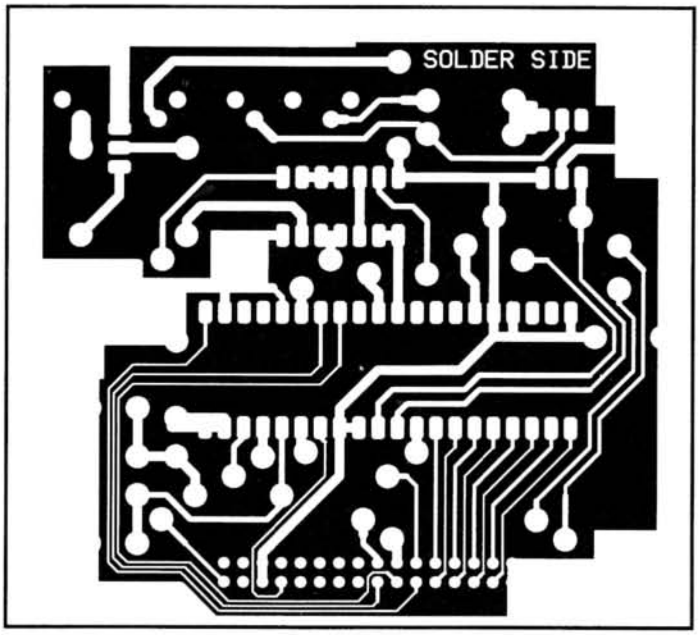
```
620    L1(C1) = L1(C1) OR T(C,J) \ 8
630    L2(C1) = L2(C1) OR (T(C,J) AND 7) * 32
640    C1 = C1 + 1
650  NEXT J
660 NEXT I
670 GOSUB 1000
680 FOR I = 0 TO W ' output top half
690  IF I = 0 THEN Z = L1(I) ELSE Z = L1(I) AND NOT L1(I-1)
700  GOSUB 1260
710  L1(I) = 0
720 NEXT I
730 GOSUB 1150
740 GOSUB 1000
750 FOR I = 0 TO W ' output bottom half
760  IF I = 0 THEN Z = L2(I) ELSE Z = L2(I) AND NOT L2(I-1)
770  GOSUB 1260
780  L2(I) = 0
790 NEXT I
800 GOSUB 1150
810 GOTO 400
980 '
990 ' set up for W graphics bytes with 1" margin
1000 Z = 27
1010 GOSUB 1260
1020 Z = 89 ' Esc Y: double speed, double density graphics
1030 GOSUB 1260
1040 Z = (W + 120) MOD 256 ' low-order byte
```

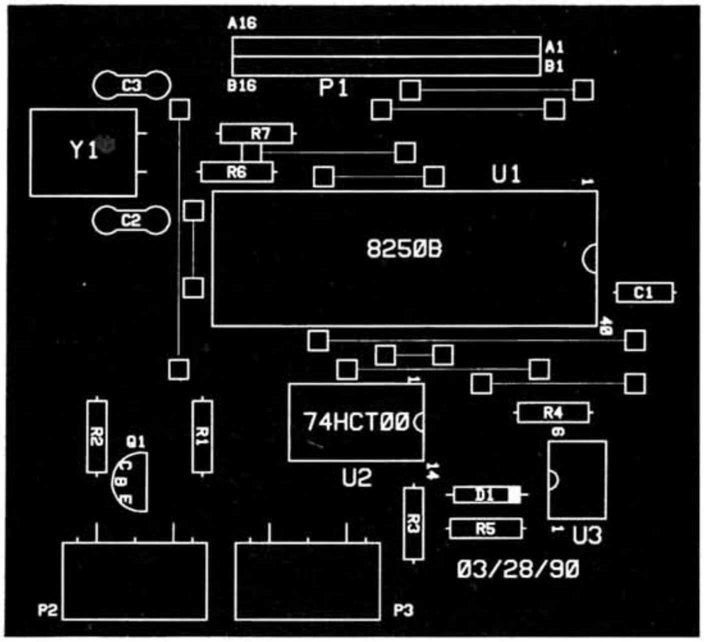
```
1050 GOSUB 1260
1060 Z = (W + 120) \ 256 ' high-order byte
1070 GOSUB 1260
1080 Z = 0
1090 FOR I = 1 TO 119
1100  GOSUB 1260 ' 1" left margin
1110 NEXT I
1120 RETURN
1130 '
1140 ' advance 24/216"
1150 Z = 13 ' CR
1160 GOSUB 1260
1170 Z = 27
1180 GOSUB 1260
1190 Z = 74 ' Esc J n
1200 GOSUB 1260
1210 Z = 24
1220 GOSUB 1260
1230 RETURN
1240 '
1250 ' direct printer output of character Z to lpt1
1260 PST = INP(LP1+1) AND &HF8
1270 IF PST = &H58 THEN 1260 ' wait for previous character
1280 OUT LP1, Z ' latch character on to data line
1290 OUT LP1+2, &HD ' drop strobe low
1300 OUT LP1+2, &HC ' put strobe high
1310 RETURN
```

Add MIDI to Your SupersPort

Terry Perdue
3646 Crestview Drive
St. Joseph, MI 49038

For ease in duplicating the MIDI board for the SupersPort described in the August issue of REMark, here are full-sized foil patterns that may be photocopied.





FREE MICROSOFT WINDOWS * DETAILS BELOW

Seagate HARD DRIVES

MODEL	CAPACITY/FORMAT/SPEED/SIZE	DRIVE ONLY	XT KIT
* ST-125	21 MEG / MFM / 40 MS / 3.5"	$229.00	$279.00
* ST-138	32 MEG / MFM / 40 MS / 3.5"	$277.00	$327.00
* ST-138-1	32 MEG / MFM / 28 MS / 3.5"	$307.00	$357.00
* ST-151	42 MEG / MFM / 24 MS / 3.5"	$353.00	$403.00
* ST-138R	32 MEG / RLL / 40 MS / 3.5"	$258.00	$313.00
* ST-157R	49 MEG / RLL / 40 MS / 3.5"	$286.00	$341.00
* ST-225	21 MEG / MFM / 65 MS / 5.25"	$199.00	$254.00
* ST-251-1	42 MEG / MFM / 28 MS / 5.25"	$312.00	$362.00
* ST-4096	80 MEG / MFM / 28 MS / 5.25" FH	$582.00	$633.00
* ST-238R	32 MEG / RLL / 65 MS / 5.25"	$218.00	$273.00
* ST-277R-1	65 MEG / RLL / 28 MS / 5.25"	$348.00	$403.00
* ST4144R	122 MEG / RLL / 28 MS / 5.25" FH	$623.00	$678.00

* IDE, SCSI, ESDI AND OTHER SEAGATE MODELS AVAILABLE. PLEASE CALL.

*** ZENITH PC COMPUTER UPGRADES ***

SmartWatch from FBE RESEARCH

⇨Installs in ROM Socket on the CPU Board in Zenith computer series Z-100/138/148/150/160 and most all other XT computers. This clock/calendar contains a ten year battery and keeps your computer informed of both date and time at each boot-up. Instructions and software included. $35.00

Z-150/160 MEMORY UPGRADE

⇨This kit includes a replacement memory decoder PAL chip for the standard Z-150/160 memory card (not for the Z-157/58). With this PAL and the 18 pieces of 256K RAM chips (included), you will expand the memory on the card to 640K or 704K. ZP640+/18 Kit.....$59.00. PAL chip only ZP640+....$18.00

Z-150 SERIES HARD DISK DRIVE KIT

⇨These kits include high speed Seagate drives with autopark heads. Each kit includes all cables, hardware and instructions to mount the hard drive under your two floppy drives in your Z-150 series computer.

* ST-125/Z150 Kit	21 Meg, 40 MS,	$283.00
* ST-138/Z150 Kit	32 Meg, 40 MS,	$331.00
* ST-151/Z150 Kit	42 Meg, 24 MS,	$409.00

Z-148 HARD DISK DRIVE KIT

⇨Includes the hard disk drive and a Z-148 compatable controller together with the Z-148 Expansion Card described below. All required cables, hardware and instructions are included for you to replace one floppy with a Seagate Hard Drive in your Z-148. Add only $30.00 the the following price if your would like us to include a SmartWatch.

* ST-125/Z148 Kit	21 Meg, 40 MS,	$354.00
* ST-138/Z148 Kit	32 Meg, 40 MS,	$402.00
* ST-151/Z148 Kit	42 Meg, 24 MS,	$478.00

Z-148 EXPANSION CARD

⇨Adds one full length and one half length IBM expansion slot to your Z-148 for hard drive controller, video card, modem, etc. ZEX-148......$79.00

Z-150 VIDEO ELIMINATOR

⇨For the Z-150 or Z-160 only. Not required for the Z-157/158/159 computers. A small piggyback board which replaces the scratch pad memory on your current video card. This allows the removal of the original Zenith video card and replacment with an EGA, VGA or any other 8 bit video card. Order VCE-150 .... $54.00

2400 BAUD MODEMS

⇨Fully Hayes compatible 2400/1200/300 Baud with Software. Internal $99.00 External Model $128.00 Cable for External Modem $8.50

15718 SYLVAN LAKE * HOUSTON TX 77062

FLOPPY DISK DRIVES

⇨MITSUBISHI MF501	5.25" 48 TPI DS/DD 320K/360K	$ 68.00
⇨MIITSUBISHI MF504	5.25" High Density 360K/1.2 MEG	$ 81.00
⇨ MITSUBISHI M-353	3.5" in 5.25" frame 720K	$ 84.00
⇨MITSUBISHI M-355	3.5" in 5.25" frame 1.44 MEG	$ 94.00
⇨TOSHIBA ND352	3.5" with 5.25" frame 720K	$ 74.00
⇨TOSHIBA ND356	3.5" with 5.25" frame 1.44 MEG	$ 79.00

⇨M-355 and ND356 run on AT compatible or special controller only.

PAYLOAD CUSTOM ASSEMBLED COMPUTERS

⇨We assemble 8088 XT, 80286 AT, 80386SX or 80386 IBM compatible computers to your specifications. Please write or call for a work-up sheet showing items available and prices.

VIDEO MONITORS

⇨ZCM-1492	ZENITH Color Flat Screen VGA	$ 679.00
⇨MA2565	SAMSUNG Amber TTL 720x350	$ 89.00
⇨CW4656	SAMSUNG Color RGB 640x200	$ 223.00
⇨CM4592	SAMSUNG Color EGA 640x350	$ 339.00
⇨CJ4681	SAMSUNG VGA 720x400	$ 360.00
⇨CVB4581	SAMSUNG Multi-sync VGA 1024x768	$ 469.00
⇨CM1440	SEIKO VGA 1024x768 .25 dot	$ 609.00

VIDEO CARDS

⇨EGA480	PARADISE AUTOSWITCH 640x480	$ 99.00
⇨VGAPLUS	PARADISE AUTOSWITCH 800x600	$ 168.00
⇨VGAPLUS16	PARADISE AUTO 16 BIT 800x600	$ 207.00
⇨VGA 1024-256k	PARADISE AUTO 1024x768	$ 249.00
⇨VGA 1024-512k	PARADISE AUTO 1024x768	$ 309.00

MEMORY CHIPS, ETC.

⇨Memory chips are once again at reasonable prices. The market prices have been changing daily, therefore we are only able to list estimated prices. Please call for the current price before placing your order. We buy in large quantities and work on the smallest of margins in order to bring you great values.

⇨41256 256x1 80 ns.....$2.95		SIP 1Mx9 80 ns $80.00
⇨41256 256x1 100 ns $2.65		SIM 1Mx9 80 ns $79.00
⇨41256 256x1 120 ns $2.30		SIM 256x9 80 ns $24.00

Z-100 SERIES COMPUTER UPGRADES

High Density 1.2 Meg Drives

⇨External floppy drive set-up complete with drive, power supply, case and cables. Ready to connect to your 8" floppy controller. Single Drive Unit $217.00

⇨Dual Drive Unit $309.00 Bare drive and cable for internal mount $127.00

ZMF100A by FBE Research

⇨A modification package which allows 256K chips to be used on the old-style motherboard (part number85-2653-1) to reach 768K. Simple assembly with no soldering or trace cutting. Compatible with Easy PC and Gemini Emulator. Order 27 256x1 RAM chips to complete this kit. ZMF100A$60.00

UCI Memory Upgrade Pal Chip Set

⇨For the Z-100's with the newer motherboard part number 181-4918 or greater. Allows the installation of 256K RAM chips on the motherboard. With the addition of 27 256K 150 ns RAM chips (sold separately) a total memory of 768K is obtained. Chip Set....... $64.00

UCI EasyWin HARD DRIVE SYSTEM

⇨Complete Hard Disk System for mounting inside your Z-100. Includes S-100 bus board, matched XT hard disk controller, EasyWin software, manual and Misc installation hardware. Order a hard disk (ST-125 or ST-138 recommended) under the SEAGATE HARD DISK DRIVE ONLY listing to complete the kit. $288.00

Z-100 SERIES SOFTWARE

⇨PART NUMBER /DESCRIPTION	LIST PRICE	SALE PRICE
⇨CD-463-2 Condor File Manager	$175.00	$10.00
⇨MS-253-1Basic-80 (8-bit)	$175.00	$10.00

MICROSOFT WINDOWS version 1.04 for PC (not Z-100)

⇨FREE with any order from this ad over $30.00. Just ask for it and add $5.00 for
⇨shipping and handling. Includes 5 disks and 300+ page manual. Offer good until
⇨existing inventory given away (about 300+packages).

⇨Please Mail, Phone or FAX your order. All hardware carries the manufacturers warranty plus the PAYLOAD 90 day guarantee. No surcharge on credit card orders. COD shipments on request. Add $5.00 to all prepaid orders for handling and shipping in the Continental USA, we pay the balance. Actual shipping costs for foreign, overseas and net billing orders. We accept purchase orders from schools, government and approved accounts. Mail or phone your order for prompt friendly service. Texas residents please add 8.0% state sales tax.

PHONES: ** ORDERS AND INFO **(713) 486-0687** ** FAX: **(713) 486-8994** **

PAYLOAD * PAYLOAD * PAYLOAD * PAYLOAD

Reader Service #117

Sophisticated Setups:
Enhanced Batch File and Menuing Programs

Harold C. Ogg
357 W. Diversey Avenue
Addison, IL 60101-3508

This is the Year of the Batch File! As MS-DOS has matured into versions 4.xx of its evolution, PC support literature has presented dozens of source listings of reasonably complicated bootup routines and batch file labor savers. For casual and applications-only users, these macros eliminate the tedium of learning the intricacies of MS-DOS and its many pitfalls.

Although MS-DOS is not technically a "language" per se, a clever programmer can use it as such. The batch "language" portion of MS-DOS has only seven operators–ECHO, FOR, GOTO, IF, PAUSE, REM, AND SHIFT. Version 4.0 of MS-DOS adds an eighth operator–CALL–which allows one batch file to invoke another. But to perform any depth of nesting or conditional jumping within a system setup environment, a programmer must either get very creative or use a high-level language such as C or Pascal. There are some aids to uphold this creativity, as the following will evidence.

The Motivation

To a non-programmer, MS-DOS is like boot camp–you have to get past the drill sergeant before you're allowed to do more interesting things. The "drill" to get into word processors, databases, and spreadsheets–for some office workers, data inputters, and technophobics–is a round of sparring with MS-DOS commands that (hopefully) will set the proper environment, video mode, and numbers of file handles and buffers. Complicating the matter are circumstances where more than one person works on any given PC; there usually just isn't any one setup that will please everybody or fill everybody's needs.

Even for the more technically minded PC user, the set-it-and-forget-it mentality is a thing of the past. Memory hungry programs that demand nearly all the base 640 kilobytes of RAM dictate that only the essential applications, TSR's (terminate and stay resident programs) which you need must be reloaded, or unneeded ones must be taken from memory. A resetting, or in some cases, a rebooting of the machine is mandated. The drudgery comes in the form of lists of MS-DOS commands that must be reentered at each system reset, time after time as each new working environment comes to bear.

It would be nice, then, to have built-in lists of setups for the users' various needs, typed once and saved to command files for instant access. This is the situation that begs for intelligently cultivated batch files and menuing systems. Success in such endeavors requires that a programmer have two tools: the ability to write complicated files (or access to reference sources from which to copy them), and the means to maintain, augment, and streamline such lists. This article attempts to provide both.

Sources for Batch File "Code" and Utilities

For collections of batch source files, the bookstores contain several useful items. A good reference is the Microsoft *MS-DOS Batch Files*[1], quite handy for quick lookups and some detailed explanation of the basic functions. *MS-DOS Batch File Programming... Including OS/2*[2] and *Advanced MS-DOS Batch File Programming*[3] are companion volumes that offer dozens of lines of code for menuing, environment handling, use of ERRORLEVEL, and the mastery of passing replaceable parameters. The code in these books makes ideal input for the batch file compilers (described below).

MS-DOS Utility Programs[4] is a treasure trove of software. Many of the tools described therein are not to be found through ordinary channels, and there is an entire section on "Batch File Aids." Lengthy annotations and ordering information are given for such items as Batutil, a control mechanism for the DOS environment; EBL, an Extended Batch Language; and Stackey, a utility for placing redundant keystrokes in a buffer and having a batch file automatically pass keystrokes for you. *Tips and Techniques for Using Low-Cost and Public Domain Software*[5] also contains chapters on extended batch language and menuing, and offers the bonus of advising you how to derive maximum benefit from freeware and shareware utilities so readily available.

Serious programmers would also benefit from copies of the catalogs from The Programmer's Shop[6] and the Programmer's Connection[7]. Even if no software is ordered from these vendors, the procurement of their buyers' guides provides a wealth of descriptive narratives on hundreds of developmental programs (including batch file and menuing programs) which are not available in retail outlets. The Programmers' Shop additionally offers the FastFaxts service, in which anyone with a telefacsimile machine can, by sending touch tone telephone signals to the Shop's computerized fax setup in Boston, receive telefacsimile copies of product descriptions of most items in their catalog.

Public Brand Software offers two special collections of miscellany, named (appropriately) Batch Utilities 1 and Batch Utilities 2[8]. While these collections contain some redundancies of programs available on their other disks referenced elsewhere in this article, PBS has attempted to round up the free and least expensive ones in the Batch Utilities. Write also to PBS and request a copy of their current catalog.

We Shall Overcome: Efficient Juggling of TSRs and .SYS Drivers

Zenith, and all the other 8086/80286/80386-based clones, still must contend with the 640K memory limitation under MS-DOS. The use of software that invokes the Lotus/Intel/Microsoft (LIM) 3.2 or 4.0 standard, i.e., expanded memory, alleviates the problem somewhat. However, it is not practical for us to limit our work environment only to those "well behaved" programs. There are many common programs that crowd the lower 640K region and refuse to be sociable in the presence of much else.

Exacerbating the problem is the variety of device drivers that must reside in that same 640K area. These memory-resident .SYS modules are essential to running any of our sophisticated peripherals, and cannot be ignored if we are to perform any "power" programming. Our favorite TSR's compete for space as well. Often we must sacrifice utility or choose and eliminate from a laundry list of drivers and popup software those luxuries we think we need. The latter action is the only way we can avoid the dreaded "insufficient memory" slogan across the screen.

The heart of the problem is that we don't want the PC set up exactly the same way each time it is powered on or rebooted. At times we want a RAM disk, other times not. I decided some time ago that I could live with just two setups at boot time. My Z248 has an extended memory board which requires a utility (Above DISC) to emulate the LIM 4.0 expanded memory specification. At other times, all I need is linear memory for my dual boot OS/2 operating system, with no EMS driver in place. Using batch files, the quick-and- dirty way I set up a toggle between different versions of the CONFIG.SYS file is with the familiar swapout method that assembly language programmers employ to load registers. My short CHBOOT.BAT file looks like (Figure 1):

```
ECHO OFF
C:
CD \
REM swap CONFIG.SYS files
REM CONFIG.TMP is a placeholder
COPY CONFIG.SYS CONFIG.TMP
ERASE CONFIG.SYS
REN CONFIG.BAK CONFIG.SYS
REN CONFIG.TMP CONFIG.BAK
CD \ABOVE
REM ABOVE is directory holding
REM the Above Disk LIM
REM memory manager
REM ABOVE.CFG is the
REM parameters file
COPY ABOVE.CFG ABOVE.TMP
ERASE ABOVE.CFG
REN ABOVE.BAK ABOVE.CFG
REN ABOVE.TMP ABOVE.BAK
CD \
```

Figure 1
CHBOOT.BAT

To switch between versions of CONFIG.SYS, type CHBOOT at the prompt and perform a system reset. Since a placeholder file for the dormant versions of CONFIG.SYS and ABOVE.BAK have the extension .BAK, I have to be careful not to erase all the .BAK files in the root directory during routine housekeeping.

There are utility programs which will allow you to switch between several versions of an AUTOEXEC.BAT and CONFIG.SYS files. Dynaboot[9] was written especially for PC's that require multiple (up to 100, in fact) different bootup settings, as when several users share the same PC. The program's value lies in its ability to resolve unreferenced interrupt vectors that aren't cleared when applications programs remove TSR's from memory. Dynaboot paves the way for activation of "new" system drivers by "cleaning up" de-allocated memory fragments that can inhibit memory usage by the most recently loaded programs.

...And a Few Slick Control Devices

An activity which has become fashionable of late is the accessing of one's office or home computer while on the road, usually with a modem-bearing laptop. At first, this practice was confined largely to the uploading and downloading of files. As software became more sophisticated and modems faster, users demanded a longer reach, in that they wanted remote control of their desktop PC's from any location that provides a hookup for a modular telephone jack.

There exist several programs which will allow the aforementioned. Face-2-Face[10], pcAnywhere III[11], and Carbon Copy Plus[12] all work in a similar fashion, allowing a remote keyboard to run most of the host PC's programs. There are two problems: first, some graphics-or video-intensive programs can lock the host machine to the point where a hard reset is required of both the remote and the host. Second, the remote machine must be left running twenty-four hours a day, always ready for access. There has always been the situation that, if something goes wrong at the host PC's location, a human must be present to set things right and reset the communications devices to receive input.

(Face-2-Face, however, allows a "Reboot on Disconnect" when its Host module is loaded into the PC using AUTOEXEC.BAT. This provision is made available automatically inasmuch as an end user must purchase one Host and at least one Remote Face-2-Face software module to make the system function.)

It is possible to circumvent the aforementioned problems with a few ingeniously written batch files and a couple of tricks. The easiest workaround is to place all the system startup commands, including the parameters for the modem's invocation script, into the AUTOEXEC.BAT file. When the PC is powered on, everything–including the modem protocols–is set without operator intervention. The trick is to connect the host PC to a programmable timer, which can be purchased at a hardware store for a few dollars. The operator can then set the PC to be on and off at specific times during the day. If the system "hangs," the remote operator need only wait for the next powerup interval for the PC to be reset, initialized up with the same AUTOEXEC.BAT file as at the previous bootup.

Thus using a mechanical timer, if the remote operator desires a different host working environment, he or she can upload a new AUTOEXEC.BAT to the host at the end of a dialup session. This will replace the AUTOEXEC.BAT file on the host, such that the new configuration will be invoked at the next powerup.

The obvious disadvantage of the preceding is that the remote operator is stuck with whatever uptime intervals he or she has programmed before leaving the host machine. Once set, a programmable timer must be physically manipulated to change its intervals. A better solution is to use one of the so-called "monitor modems" now available. An example of this is the Unison UniPower[13] box, which turns the host PC on (i.e., powers it up) after so many rings. The system is turned off in the same fashion–by hanging up the telephone on the remote end. After 2-3 minutes, the host can be recalled and repowered (with a new environment, if necessary) as though it had never been accessed.

There are several Unison models from which to choose, including the UniPower PS4.5 at $699.00, the PS6.0 at $799.00, and the DP800 at $995.00, the higher numbers representing greater output power loads. These devices are used primarily as uninterruptable power supplies, targeted for local area networks and distributed processing environments. Each model supports a standard RJ-11C telephone jack, performing double duty as a dialup watchdog. Albeit a more expensive proposition, this hardware gives the remote operator full control over a host PC, at times and intervals of a random nature.

Batch File Extenders and Enhancers

The Standard Edition of the Norton Utilities includes a well-defined batch language extender. Norton's contribution is in the realm of interactive batch commands, such as ASK, similar to INKEY() in BASIC; Time Mark (TM), an electronic stop watch; Screen Attributes (SA), to set the colors of screen display text; and BEEP, for audio prompts. These commands are used with BE.EXE (the batch enhancer executable module), and you can program the batch file to play music at appropriate times, if you like. It is possible, using methods described in the official manual of the Norton Utilities[14], to set up a simple menuing system with these commands, as illustrated in Figure 2.

The resulting menu screen is shown in Figure 3. SCREEN.BAT uses ERRORLEVEL to determine the condition for the jump to

```
REM  SCREEN.BAT
ECHO OFF
CLS
REM Call the screen overlay file
  TYPE TEMPLATE.TXT
  BE ASK "     Select a Program-", 1234
  IF ERRORLEVEL 4 GOTO BAILOUT
  IF ERRORLEVEL 3 GOTO APP3
  IF ERRORLEVEL 2 GOTO APP2
  IF ERRORLEVEL 1 GOTO APP1
  REM call WordStar
  :APP1
  C:
  CD \WS5
  WS
  CD\
  C:
  SCREEN
  REM Call dBASE
  :APP2
  CD \DBASE
  DBASE
  CD \
  SCREEN
  REM Call SuperCalc
  :APP3
  CD \SC
  SC
  CD \
  SCREEN
  REM Return to MS-DOS
  :BAILOUT
  CLS
```

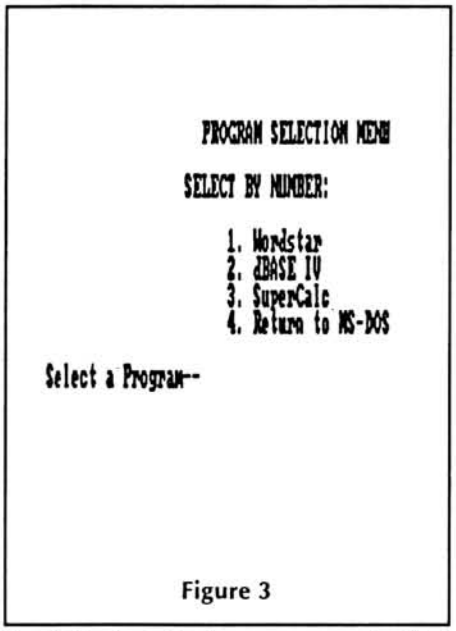
PROGRAM SELECTION MENU

SELECT BY NUMBER:
1. Wordstar
2. dBASE IV
3. SuperCalc.
4. Return to MS-DOS

Figure 2B: TEMPLATE.TXT

Figure 2A
SCREEN.BAT

Figure 2
Menu Maker and Overlay for Norton Batch Extender



PROGRAM SELECTION MENU

SELECT BY NUMBER:

1. Wordstar
2. dBASE IV
3. SuperCalc
4. Return to MS-DOS

Select a Program--

Figure 3

the desired applications program, signified by :APPx. TEMPLATE.TXT is an overlay file called by SCREEN.BAT to paint the information portion of the menu. Pressing the '4' key returns the user to DOS; any other keystroke sounds a warning beep and returns nothing. With some modification, plus the addition of a few parachutes, it is possible to add more program selections (up to nine) to the list and make the menu more sophisticated.

XTree Pro Gold[15] provides many of the same menuing capabilites as Norton. Additionally, XTree Pro Gold acts as a DOS overlay to enable a user to interact with the operating system in a logical fashion, visually, by manipulating a displayed "tree" diagram of a disk's contents. By causing the PC to call XTree from the AUTOEXEC.BAT file, an end user can invoke executable files and applications programs by pointing to them in a list, menu style, and the DOS prompt may never have to be seen.

Beyond BAT[16] is an example of a commercially available standalone batch file extender. Its target audience is the developmental programmers who must write detailed but extremely user friendly screen handling shells for non-technical PC users. Beyond BAT provides its own text editor to create a batchlike script language for providing full screen input, incorporating point-and-shoot screens and string handling routines which can eliminate the necessity for end users to learn DOS prompts.

Some of the batch file utilities are inexpensive and available in the public domain or as shareware. One popular extender is DOSPlus[17]. This program adds, among other functions, a BEEP command to manipulate the PC's speaker, a call to determine the status of free memory and disk space, and a command to change the video page. There are also several useful single-function batch enhancers on the market, among them PDaily[18], ANSI-Chk[19], and EchoF[20]. PDaily is loaded in the AUTOEXEC.BAT file and causes the PC to run any chosen programs one time per day; its validity remains even if the computer is booted multiple times in a 24-hour period. ANSI-Chk allows a user to determine from a batch file whether ANSI.SYS is resident. The program also includes the assembly language (.ASM) source code for programmers who wish to include the routine in other programs. EchoF is used in batch files to allow output of control (Ctrl) and escape (Esc) sequences. This is a most useful routine where a printer or video screen must be preset to particular output conditions in advance of loading an applications program.

Menuing Programs

If your intended audience consists of users who ordinarily function on a nontechnical level, these persons will interact best with their programs if there is a minimum of fuss to invoke their applications software. The simplest way to overlay a shell to access several programs is to provide a menu screen at boot time. Until recently, menu programs were proprietary to each particular application, or, if they existed as DOS shells, were created only from elaborate, graphics intensive language compilers at high cost.

But no more. One exploitable area for the use of extended batch files is in menu screen setups, accomplished by one of several available menu generator programs. Simply stated, such programs generate ASCII code in the form of batch commands and embed screen manipulation provisions that provide such tools as pulldown and popup menus, file (program) selection through reverse video bars, and even mouse support.

A number of menu generation programs are commercially available. The Menu Works[21] provides an operating system shell that invokes one or more of 1,000 common applications programs with minimal user intervention. The Menu Works program also provides mouse support and password protection for sensitive programs. Direct Access[22] is more of a code generator, allowing the customization of menu screens for any program callable under DOS, resulting in a highly flexible program management system that is user friendly and easy for the novice to follow.

There is a good, minimum level menu manager described in PC Magazine. MENU.COM[23], provided as an executable file (or in .ASM and .BAT source code), is available over the magazine's PC MagNet bulletin board. The download procedure is described at the end of the MENU article. MENU operates as a single command line, with the format (Figure 4) where switch /D displays any screen/ASCII file, /E allows editing of a screen display file, /N prevents MENU from restoring the original DOS screen upon exit, 'screen' is the file containing display information, 'template' the

MENU [/D/E/N] screen template out1 [out2 ...]

Figure 4
MENU.COM Entry Format

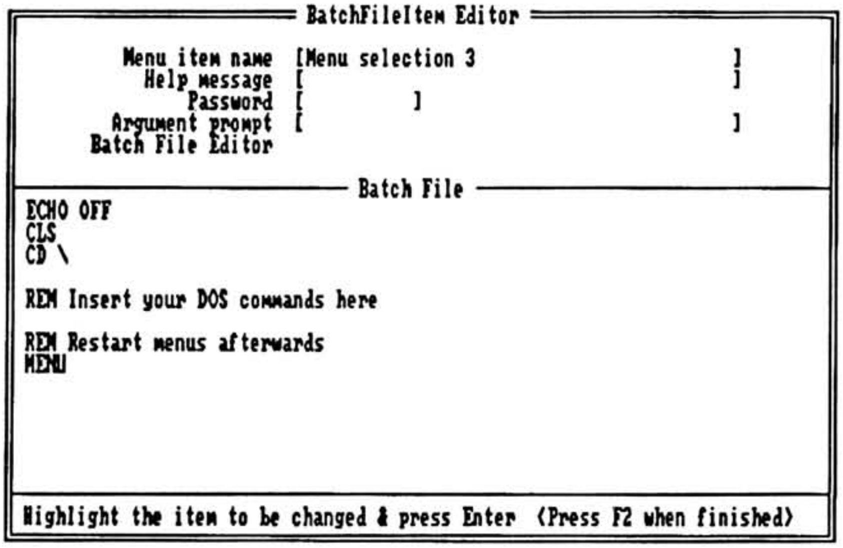
controller for the format files, and out1, out2, etc., the operations filename designates. You can create up to 64 outfiles of up to 300 lines each. There is an example FAX.BAT file that addresses the currently popular trend of using internal PC telefacsimile boards, along with the author's description of how he uses the utility for office fax management.

More recently, PC Magazine made available a utility named PAN.COM[24], which combines the fundamentals of several of the features of other programs reviewed in this article. Run as a script language, PAN, or Program Animator, allows you to power up into operating menus, schedule automatic operations for specific times of the day, and set up program demonstrations. A special feature is the expansion of the DOS MORE.COM command, allowing MORE to page rather than scroll, clearing the screen before updating the file display. The complete .ASM source code listing is included with the article.

Le Menu[25]adds the capabilities of mouse and network support to its menu building functions. The resulting menu programs show end users the disk structures in a treelike format, and, with its capacity to tag files as read only, read/write, etc., Le Menu is particularly valuable on PC's where accurate hard disk management by end users is a must. Le Menu allows for 26 menu choices per screen at five levels of nested submenus, and requires only one kilobyte of RAM of overhead to use concurrently with applications programs.

Perfect Menu[26] offers the feature of hot key command selection to make the menu generator almost self-documenting. The menu creator can opt for memory-resident (TSR) or non-memory resident screens in case RAM usage by the called applications program would cause a conflict. Perfect Menu also provides the end user with a popup on-screen calculator, an ASCII table, and a calendar, all in a similar manner to Borland's Sidekick, as well as a clock, keyboard lock, and file management system conveniences. Mouse, EMS memory, and extended memory support, EGA and VGA accessibility, and file password protection are also included.

The Menu Construction Set[27] is a screen painter type of utility aimed at developers and programmers who must write screen interfaces in quantity. It works in a similar manner to program code generators, providing templates for menu generation which result in batch files usable as shells for applications programs and DOS environments. A sample of its batch file building screen is shown in Figure 5. This vendor offers a companion program RECONFIG, similar to Dynaboot (above), which allows easy swapping between as many as 100 different versions of the AUTOEXEC.BAT and CONFIG.SYS files. The RECONFIG utility eliminates the necessity of retyping these two files (or stepping through an involved RENAME/ERASE procedure) each time a different operating environment is desired. RECONFIG provides for automatically rebooting your machine for each new AUTOEXEC or CONFIG file.



```
═══════════════════ BatchFileItem Editor ═══════════════════
           Menu item name  [Menu selection 3                ]
             Help message  [                                ]
                 Password  [          ]
          Argument prompt  [                                ]
        Batch File Editor
──────────────────────────── Batch File ─────────────────────
ECHO OFF
CLS
CD \

REM Insert your DOS commands here

REM Restart menus afterwards
MENU

Highlight the item to be changed & press Enter  (Press F2 when finished)
```

Figure 5

Those programmers who must contend with non-English language command sets would do well to examine Menu-IX[28]. In addition to support of foreign language input characters, there is a mechanism for creating pulldown and popup windows. Another feature is a help dialogue generator for the screen templates which you construct. Menu-IX can run under DOS, Xenix, and OS/2.

Batch File Compilers

If you want to coax every last microsecond of speed from your setup files, consider promoting them to .COM or .EXE status. Batcom[29]creates binary, executable files from ASCII batch file statements, and adds sixty commands to the standard seven (or eight, depending on the version of DOS used) of the DOS batch file command list. The program also provides the ability to execute subroutines and include user-defined variables along with the DOS command line parameters and environment variables.

The Builder[30] is a similar program, and it gives the programmer a choice of generating .COM or .EXE files to replace ASCII files of batch commands. As of this writing, it offers all batch file features of DOS version 3.3. It, too, extends the batch language and features a wide variety of operations to manipulate directories and subdirectories within a batch file. It further allows the creation of dropdown menus, popup menus (as dialogue boxes), and lightbar selection boxes, all with choices of colors. The Builder also provides for subroutine manipulation with CASE/END, IF/ELSE, and REPEAT structures.

It should be noted that batch file compilers provide the greatest value in handling longer batch files (fifty or more lines). For short, minimum setup files, the compilers generate modules that actually waste space. For instance, the SCREEN.BAT and TEMPLATE.TXT files in Figure 2, when compiled with The Builder, result in a .COM plus overlay file totaling 6,948 bytes. The corresponding ASCII files require only 593 bytes, and do not require a significantly longer amount of time to execute as ordinary batch files.

An alternative to batch file compilers, for those who prefer an interactive setup, is OPAL[31]. This program features a menu builder and batch language extender/enhancer as do the previously mentioned utilities, but OPAL performs as an interpreter. It acts like GW-BASIC, executing commands as typed, adding some proprietary verbs and action words to the DOS batch language. As an interpreter, OPAL is a powerful tool to use in an applications prototyping setting, allowing the programmer instant feedback and error checking when creating end user frontend screens and shells.

Footnotes and Sources of Products Mentioned

Some of the following programs are available as demonstration disks from Public Brand Software (see footnote 8 below). In such cases, the notation PBS XXx.x is made after the entry. The cost is $5.00 per disk, plus $5.00 shipping and handling; many of the PBS disks include the full length versions of the programs cited.

[1] Jamsa, Kris. MS-DOS Batch Files. Redmond, WA: Microsoft Press, 1989. Paper, 166 pages, $6.95.

[2] Richardson, Ronny. MS-DOS Batch File Programming... Including OS/2. Blue Ridge Summit, PA: Windcrest, 1988. 300 pages, $25.95; paper, $17.60.

[3] Gookin, Dan. Advanced MS-DOS Batch File Programming. Blue Ridge Summit, PA: Windcrest, 1989. Paper, 385 pages, $24.95 (companion disk available for $24.95).

[4] Richardson, Ronny. MS-DOS Utility Programs: Add-On Software Resources. Blue Ridge Summit, PA: Windcrest, 1989. Paper, 665 pages, $24.95.

[5] Gliedman, John. Tips and Techniques for Using Low-Cost and Public Domain Software. New York: McGraw-Hill, Inc., 1989. Paper, 387 pages, $24.95.

[6] Programmer's Catalog, available from The Programmer's Shop, Five Pond Park Road, Hingham, MA 02043. Issued quarterly.

[7] The Connection: Programmer's Connection Buyer's Guide, available from the Programmer's Connection, 7249 Whipple Avenue NW, North Canton, OH 44720. Issued quarterly.

[8] Batch Utilities 1 and Batch Utilities 2, available from Public Brand Software, P.O. Box 51315, Indianapolis, IN 46251. (PBS UA3.1 and UA6.1)

[9] Dynaboot, available from Matthew J. Palcic, MJP Enterprises, 1030 Dayton-Yellow Springs Road, Xenia, OH 45385-9508. $15.00. (PBS UA2.1)

[10] Face-2-Face, available from Modem Controls, 432 North Clark, Suite 202, Chicago, IL 60610. Host module, $149.00; Remote module(s), $99.00 each, plus shipping and handling. Demonstration/evaluation copies available upon request.

[11] pcAnywhere III, available from Dynamic Microprocessor Associates, 770 Middle Country Road, P.O. Box Y, Selden, NY. $110.00.

[12] Carbon Copy Plus, available from Meridian Technology, Inc., Seven Corporate Park, Suite 100, Irvine, CA 92714. Version 5.1, $195.00 each; two or more copies are required.

[13] Manufactured by Unison Technologies, Inc., 23456 Madero, Mission Viejo, CA 92691.

[14] Krumm, Rob. Inside the Norton Utilities. New York: Brady (division of Simon & Schuster, Inc.), 1988. Paper, 481 pages, $19.95.

[15] XTree Pro Gold, available from XTree Company, 4330 Santa Fe Road, San Luis Obispo, CA 93401. Version 1.31, $129.00.

[16] Beyond BAT, available from VMpersonal Computing, 41 Kenosia Avenue, Danbury, CT 06810. $99.00.

[17] DOSPlus, available from Steven A. Georgiades, SMG Software, 701-H South Hayward Street, Anaheim, CA 92804. Version 1.03, $15.00. (PBS LB9.0)

[18] PDaily, available from Norm Patriquin, P.O. Box 8263, San Bernardino, CA 92412. (PBS LB7.0)

[19] ANSI-Chk, see footnote 16 (above) for availability. Version 1.11. (PBS LB9.0)

[20] EchoF, see footnote 16 (above) for availability. Version 1.02, $10.00. (PBS LB 9.0)

[21] Menu Works, available from PC Dynamics, Inc., 31332 Via Collins, Suite 102, Westlake Village, CA 91362. $24.95.

[22] Direct Access, available from Data Technology International, Inc., 1621 Westgate Road, Eau Claire, WI 54703. $90.00.

[23] Smith, Ron Q. "Create Colorful, Interactive Menus for Your Applications." PC Magazine, October 31, 1989, pp. 319-333.

[24] Maclean, Pete. "Introducing PAN.COM, the Program Animator." PC Magazine, May 15, 1990, pp. 253-290.

[25] Le Menu, by Bartel Software, available from The Programmer's Shop (see footnote 6 above). $80.00 for single-user version.

[26] Perfect Menu, available from International Computer Group, Inc., 18520 Office Park Drive, Gaithersburg, MD 20670. $50.00.

[27] The Menu Construction Set, available from Optimal Solutions of NH, Six Salem Road, Atkinson, NH 03811. Single user version, $47.50; LAN server license, $147.50; Unlimited run-time license, $147.50; RECONFIG utility version 4.0, $39.50. (PBS UM11a.0- UM11c.0)

[28] Menu-IX, available from System IX Networks, Ltd., 55 Bedford Court Mansions, Bedford Avenue, London WC1B 3AG, UK. £49.95. Also available domestically from The Programmer's Shop (see footnote 6 above) for $99.00.

[29] Batcom, available from Wenham Software Co., Five Burley Street, Wenham, MA 01984. $59.95.

[30] The Builder, available from Hyperkinetix, Inc., 666 Baker #405, Costa Mesa, CA 92626. $100.00.

[31] OPAL, available from The Software Factory, Inc., 15301 Dallas Parkway, Suite 750, LB 44, Dallas, TX 75248. $129.00.

QUIKDATA - 13 YEARS OF H/Z SUPPORT!

For all your H/Z 8-bit and PC/XT/AT needs

ACCELERATE YOUR PC!

From Sota Technologies, Inc., the fastest and most proven way to breath new life in your Heath/Zenith PC/XT computer, giving it -AT compatible speeds! Turn your turtle into a -286 rabbit with a 12.5 Mhz 80286 accelerator board. Complete with 16K on-board CACHE.

There are other ways to speed up your H/Z PC/XT computer, but you spend too much and get too little. Our EXP12 **286i** is the effective solution, making your H/Z150/160/150/158/159 series of computers, or any general PC/XT computer faster, in many cases, than a standard IBM AT type computer! Simple half-card plug-in installation with switchable speed control. **You won't believe your stop watch!**

EXP-12 - $295
EXP386 - $395 Much faster 16Mhz 80386 SX version

MEMORY UPGRADES

Note: All memory upgrades come without memory chips. 150ns 256K DRAMs are $1.95 as of this printing.

Z150MP - $17 Will allow you to **upgrade your H/Z150/160 to up to 704K** on the main memory board, using up to 18 256K DRAM chips.

MEGARAM - $43 **Upgrades your H/Z150160 series** with up to 704K of main memory, and about 512K for RAMDRIVE memory. Includes documentation, software RAMDRIVE disk, PAL and jumper wire. For the full **1.2 megs total memory**, 45 256K DRAM chips are required.

ZMF100 - $53 Will allow you to **upgrade your H/Z110/120** (old motherboards; with p/n less than 181-4918) **to 768K system RAM**. Requires 27 256K DRAM chips.

Z100MP - $76 Similar to ZMF100 above, but for new motherboards with p/n 181-4918 or greater.

Z159 EMS LOGIC UPGRADE includes the PAL chip, logic chips and one bank of 256K (9) chips. Up to two banks (18) can be installed for a total system RAM of 1.2 meg on the Z159 main board.
Z315-1 - $79!!

WINCHESTER UPGRADE KITS

PCW20 - $259 Complete winchester setup for a H/Z150, 148, 158, 159, 160, PC etc. Includes **21 meg** formatted half-height Segate ST-125 37ms drive, WD (DTC for Z148) controller, cable set, doc.

PCW30	$329	32 meg with 38ms Segate ST-138.
PCW40	$359	42 meg with 28ms Segate ST-251-1.
PCW80	$629	80 meg with Segate ST-4096 full size drive.

We also have the DTC controllers ($59) and daughter board expansions ($65) to place a hard drive in the **H/Z148** computers

BARE WINCHESTER DRIVES

ST-125	$219	21 meg 37ms autopark 3.5" drive in 5" frame
ST-138	$275	32 meg 38ms autopark 3.5" drive in 5" frame
ST-251-1	$309	42 meg 28ms autopark HH drive
ST-4096	$575	80 meg 28ms autopark FH drive
ESDI	$949	**160MB ESDI** drives - several types - call or write

FLOPPY DRIVES

T28M	$159	2.8MB 3.5" floppy
MF501	$75	5" 360K DS/DD drive
MF504	$83	96 TPI 1.2 meg AT/Z100 drive
MF353	$79	720K 3.5" drive in 5" frame
MF355	$89	1.4 meg 3.5" AT drive in 5" frame
SIEM-R	$39	40 track SS refurb (H8/H89 type)
TM100-4R	$75	90 trk DS refurb (H8/H89 type)

DOES YOUR COMPUTER KNOW THE TIME?

We carry the SMART CLOCK for any PC/XT computer, including the Z100 series. Simply plug it in the system ROM socket, and plug the ROM into the clock module. Self contained sealed lithium battery for 5-10 year life. Includes software for the clock/calendar.
SMARTCLK - $33 (add $2 for Z100 installation for spacers)

ANY DRIVE IN YOUR PC/XT/AT

With the CompatiCard, you can install up to four additional drives, of any type in your PC/XT/AT computer. Add a 1.2 meg 5" floppy, or a 1.44 meg 3.5" floppy, or any other drive, including 8" to your system. The CompatiCard (CCARD) will handle up to four drives, and the CompatiCard II (CCARD2) will handle up to 2 drives. CCARD4 has boot ROM to allow it to be used as primary boot controller in systems that allow you to remove floppy controller in some systems. Also handles 2.8MB 3.5" floppy drives. Additional cables and external enclosures may be required.

CCARD2 - $79 **CCARD** - $99 **CCARD4** - $125

OUR QD LINE OF COMPUTERS

Excellent quality bare bones PC/XT/AT 0-wait state clone systems (our name is on them - we sell them with pride!) with 101 keyboard and one year warranty. Buy the basic system and choose from our product line to add what you need for a complete system. FCC certified. Pick any system, order other items you want installed from us and we'll install the whole works, test it, and ship it out pronto! No extra charge.

QD88/12 is our high-powered entry level complete starter system with 640K RAM, keyboard, 360K floppy drive, 12Mhz V20 CPU, co-processor socket, clock/calendar, 4 bay cabinet with turbo/reset button and keylock, 150 watt switching power supply, 8 card expansion slot, parallel port, serial port, game port. Add TTL video monitor and go, or add hard drive, additional floppy drives, color video, etc. $475

QD286/12 is a solid 12Mhz 80286 AT compatible work-horse with a 200 watt switching supply, 1 meg RAM expandable to 4 megs on motherboard, 80287 socket, clock/calendar, 5 bay cabinet with turbo/reset button and keylock, 8 card expansion slot. Roll your own! Add controller, ports, video, drives and go. $619

QD386/25 is a fast top of the line 25Mhz 80386 system with the same specs as 286/12 except expandable to 16 megs on main board, 80387 or Weitek socket. $1195

QD386SX 16Mhz 28ms 40MB 1:1 interleave hard drive/controller, 1.2 meg 5", 1.4 meg 3.5", 1MB RAM, 2 serial, 1 parallel, 1 game port, 101 keyboard, baby case (8 slots total). Add video and go! $1495

8-BIT/Z100

We carry a full line of replacement boards, parts and power supplies for the H/Z89/90 and Z100. We also have some H8 boards available. We continue to fully support and carry a full line of hardware and software products for the H8/H89/90 and Z100 computers including almost the full line of Software Toolworks items. Other items we carry include diskettes of all types, printers, modems, hard drives, etc.

OTHER STUFF

Quikdata also carries spike protection filters, backup power supplies, modems, printers, disk drives and diskettes of all types, drive enclosures, cables and connectors, laptop batteries, CMOS batteries, video monitors and video cards, memory cards, memory chips and ICs, joysticks, accessory cards, serial and bus mouse, a variety of useful and most popular software and much more!

Call or write in to place your order, inquire about any products, or request our **new** free no obligation catalog. VISA and Master Card accepted, pick up 2% S&H. We also ship UPS COD and accept purchase orders to rated firms (add 5% to all items for POs). All orders under $100 add $4 S&H. Phone hours: 9AM-4:30PM Mon-Thu, 9AM-3PM Friday. Visit our bulletin board: (414) 452-4345. FAX: (414) 452-4344.

QUIKDATA, INC.

2618 PENN CIRCLE
SHEBOYGAN, WI 53081-4250
(414) 452-4172

Reader Service #193

Powering Up
Volume 2

Choosing a Programming Language

William M. Adney
P. O. Box 531655
Grand Prairie, TX 75053-1655

Virtually all computer users, at one time or another, do some kind of programming on their systems. That may seem like an extremely broad statement, but nearly all users set up and/or modify an AUTOEXEC.BAT file, which is nothing more than a form of programming. In general terms, *PROGRAMMING* is nothing more than defining a series of logical steps (or commands) that are used to perform a specific task. For microcomputers, there are four commonly used programming languages. These include assembler, BASIC, C, and Pascal; and this article will review the general advantages and disadvantages of each. But an AUTOEXEC.BAT file is nothing more than a logical series of commands (the "program") that are used to start up (the "task") a computer. To understand what programming really is, let's take a look at a simple program.

The AUTOEXEC.BAT Program

Any batch file, including the AUTOEXEC.BAT file, is nothing more than a series of logical steps that are repetitive. And nearly anything that is repetitive can be programmed in one way or another with a programming language. Programming languages have a command form (usually consisting of reserved keywords) and a command syntax, which defines how the various commands can be used. For example, batch file programming has some very special keywords (or commands like GOTO, IF ERRORLEVEL, etc, although you can generally enter any command in a batch file just like you would enter it on the DOS command line. Programming need not be a complex and burdensome task, but there are "good" programs and "bad" programs. To illustrate the point, take a look at a "good" AUTOEXEC.BAT file as shown in Figure 1.

Figure 1 is a very simple example of an AUTOEXEC.BAT file that is used to set the command prompt, PATH, and start WordStar (using the WS command) for word processing. A good program contains comments about what is going on, and that is the purpose of the REM command in a batch file. Also note that the PROMPT, PATH, and WS commands are entered in the AUTOEXEC.BAT file just like you would type them on the DOS command line. If you want to learn more about batch files in general, you should refer to Chapter 7 in the original *Powering Up* book on the subject of "Using Batch Files and CONFIG.SYS." This AUTOEXEC.BAT program will work correctly, but what happens if it is not programmed properly? To find out, consider the bad program in Figure 2.

If you compare Figure 2 with Figure 1, you will find that the last path *C:\WORDSTAR* has been deleted, which causes two things to happen. First, the system cannot find the WS command, and it will display an error message like "Bad command or file name." And second, the program will not work properly as a whole because it does not accomplish its intended task, namely to start WordStar. This kind of error, known in programming circles as a "bug", is one of the problems that all programmers must deal with. If you have worked with your computer very long, you have probably found at least one bug in the software you use.

The whole point of using this example to introduce programming is to illustrate how important it is to break a task into logical steps and make sure that each step is complete. And if you have difficulty understanding what constitutes a good and bad program as discussed here, then you probably want to avoid learning any programming language.

The First Decision

The most important decision you must make is to determine what you want to do with a programming language once you learn it. Do you want to create games, do you want to create "system" programs or do you want to write major applications, such as a word processor? Do you want to create large programs where speed of execution is important or are you more interested in learning a language so that you can write the program quickly? In some cases, answers to those questions are less important than they once were because of the flexibility of today's languages.

For example, it used to be that BASIC was employed almost exclusively to create games because most versions of BASIC supported a wide variety of graphics that were necessary. Today, most of the popular languages (C and Pascal) also provide significant graphics support, and there are additional graphics libraries that can be purchased which will provide nearly any-

```
REM Set the command prompt to look like C:\ ==
PROMPT $P $Q$Q$G
REM Set path to search root directory, DOS, WORDSTAR for commands
PATH C:\;C:\DOS;C:\WORDSTAR
REM Start WORDSTAR
WS
```

Figure 1 — Good AUTOEXEC.BAT File

```
REM Set the command prompt to look like C:\ ==
PROMPT $P $Q$Q$G
REM Set path to search root directory and DOS for commands
PATH C:\;C:\DOS            (C:\WORDSTAR is deleted)
REM Start WORDSTAR
WS
```

Figure 2 — Bad AUTOEXEC.BAT File

thing you want, even for assembler. In short, consider what kind of programming you want to do because that will help you choose the language that is best for you.

There are a wide variety of considerations that are necessary to choose the best programming language for you. To some extent, choosing a programming language and vendor is a very personal and subjective thing, just like any software. Let's begin by looking at the two different types of programming languages.

Types of Programming Languages

Understanding how a programming language really works can be complex, but the general process is not. Simply stated, all programming languages must provide some means of converting that language into instructions that can be processed by a computer. What that really means is whatever program you write must eventually be "converted" to binary (zeros and ones) form so that the computer can run the program. Many people seem to believe that a computer actually works with hexadecimal numbers (base 16) because virtually all technical computer software (e.g., the DEBUG command) displays byte values in hexadecimal. As it turns out, the hexadecimal form is used as a matter of ease and convenience because it is much easier to remember and work with a hexadecimal representation of **9A** than the binary string of **10011010**. By definition, a computer *ONLY* understands how to work with zeros and ones in the final analysis, regardless of what the display form is.

There are really two types of programming languages: interpreted and compiled.

Interpreted languages have been around for years, and the first one I learned was the CLIST (Command List) language on a mainframe computer. BASIC was originally developed as an interpreted language for microcomputers, but that has changed in recent years with the introduction of a number of BASIC compilers. Data base software, such as dBase, also uses an interpreted language, but compilers are also available for those languages too. Compiled languages have also been around for years, but before we get into too much detail, let's take a look at the major difference between the two types of languages beginning with an example of GW-BASIC that is available for nearly any PC compatible computer.

For an interpreted language, there must be a special program called an "Interpreter." In most cases, it generally does not matter whether you are using GW-BASIC on a Heath/Zenith computer or IBM's "Advanced Basic" on an IBM computer because in most cases, the interpreter is usually identified in the disk directory as BASICA.COM, sometimes followed by BASICA.EXE. When you write a BASIC program, it is called the source or source code. That source code must be converted to something that the computer can process, and that is the job of the BASIC interpreter, which generally works as shown in Figure 3.

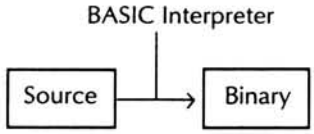


Figure 3 — Interpreted Language Conversion

The Source code is converted into Binary by the BASIC interpreter as shown in Figure 3. More importantly, the BASIC Interpreter is not particularly intelligent, and it has to convert each "instruction" as it finds it in the program. For example, if the interpreter finds the same instruction 25 times in a program, then the interpreter must convert that instruction from the Source to Binary 25 times, instead of converting it one time. This kind of programming construct is quite common in many different kinds of programs and is usually called a LOOP. For example, one can create a simple program to clear the screen using the BASIC PRINT command to print a blank line 25 times on the screen, which will essentially perform the same function as the CLS command. Because the interpreter must convert the PRINT command to binary 25 times, instead of just once, the program will be quite slow.

Another problem with an interpreted language is that the interpreter, such as the BASICA program(s), *MUST* be available to run the program. In the ZDS GW- BASIC version 3.2 that I have, the two programs take over 86,000 bytes of disk space. If you write some programs that you want to give to a friend, you must make sure that your friend has the equivalent version of a BASIC interpreter or your program may not run on that system. Even if you write a small program that only takes a few kilobytes of disk space, your friend STILL must have the correct version of the BASIC interpreter to run it, which may take over 86,000 bytes of disk space. And it is illegal to give your friend a copy of your BASIC interpreter because that is copyrighted software, even if your program is not. The point is that interpreted BASIC requires "support software" in the form of an interpreter, and that adds to the disk space requirement. Compiled (or assembled) programs do not need that support software.

The process for creating a compiled program is slightly different and is illustrated in Figure 4.

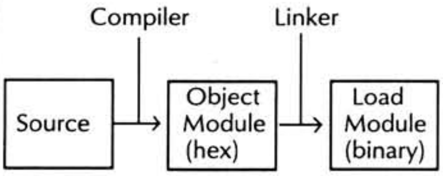


Figure 4 — Compiled Language Conversion

As you can see in Figure 4, converting a program to its final binary form is a two-step process. First, you take the Source code (in C, Pascal, assembler, etc.) and "compile" it to create an Object Module. For a computer that uses the Intel series of CPUs (Central Processing Units), the object module is converted to an Intel standard format, usually defined as hexadecimal. The second step is to "link" the program (e.g., using the LINK command), which essentially converts (or links) one or more object modules into a single, binary Load Module (sometimes called an executable file) that becomes an EXE file if you look at a disk directory with the DIR command. Compiled programs must be "linked" because the linking process essentially defines how and where the program runs in a computer's memory. And if you want to create a COM file, then you must also run the EXE2BIN (a third step, not shown) command on the resulting EXE file. Which leads to a common question that many users ask...

What is the difference between a COM file and an EXE file? If you look at the various files on a DOS distribution disk, you will find that some have a file type (or extension) of COM and some have a file type of EXE. You may have noticed that you have various programs, such as FORMAT.COM or FIND.EXE, and wondered why there was a difference. In general, the answer is that a program with a COM file type is coded a little differently than a program with an EXE file type. There is also a restriction that a COM "program" cannot exceed 64 K of *memory*, which is kind of left over from the old 8-bit computer days, but an EXE "program" can be much, much larger. I should also mention that you can find COM programs that exceed 64 K of *disk space*, but you will generally find that these programs were created with special program development software. It's not uncommon to find EXE programs, such as a word processor, that exceed several hundred kilobytes. For purposes of this discussion then, the only difference between a COM and an EXE file is the way they are programmed.

Interpreted versus Compiled

In the "old" days (just a few years ago), the discussion of an interpreted versus a compiled language usually amounted to interpreted BASIC versus all other languages which were compiled. That's not true any more because there are a wide variety of BASIC compilers which are quite good, but one can still look at the advantages and disadvantages of each.

The biggest single advantage of an interpreted language is that the programmer can get almost immediate feedback on how a program is working or not working. That is, one does not have to wait for the compiler and linker to perform its magic. This particular feature is quite useful when one is trying to learn a programming language because you do not have to wait for program compilation, even though it may take as little as 15-30 seconds for a small program. In general, an interpreted language is somewhat easier for some people to learn because of the immediate feedback feature.

The real disadvantage of an interpreted language is that it is slow to run as I mentioned earlier. And of course an interpreted language also requires some kind of support software that takes up additional disk space.

For compiled languages, there are several advantages. First, a compiled program can be copied without any need for support software, which reduces disk space requirements. Compiled programs are also faster to run because they are already in machine-readable (i.e., binary) form.

Compiled languages also have a disadvantage in that a program must compile and link a program each and every time a change is made to that program. That takes time, although most of today's compilers perform the compile/link process very quickly for small programs. As you might guess, the time required for a compile and link depends on the size of the program, so if you begin by writing small programs, this particular disadvantage is not significant.

In summary, I think you will find that there is no real disadvantage in learning one of today's compiled languages, and there are many advantages. Many languages —BASIC, dBase, and even C— are available in both interpreted and compiled versions, so the choice is pretty much up to you. In this article, we will only look at general-purpose programming languages which means that a language specific to a given application, like a data base, will not be discussed. That's because nearly every major data base has its own programming language that is considerably different. And a data base programming language rarely lends itself well to the kinds of programs a beginning programmer is likely to create, unless that beginner is working with a data base application of course.

Popular Programming Languages

Instead of trying to decide if there is some special order of presenting this discussion of programming languages, each one is discussed in alphabetical order: assembler, BASIC, C, and Pascal.

Assembler

There are as many different assembly languages, sometimes called just assembler, as there are different *SERIES* of computers. For example, the Intel 80xxx series of CPUs (e.g., 8088, 80286, 80386, 80486, etc.) has one "kind" of assembly language, usually called 8088 assembly language. As each new CPU was developed, new assembly language instructions were added, which means that an 80286 CPU can run 8088 assembly instructions, in addition to the new instructions added for the 80286 CPU features. That means a program written in 8088 assembly language will run on an 80286, 80386 or an 80486, because each of the newer CPUs "includes" the 8088 instruction set. In technical terms, this is called Upward Compatibility. Conversely, a program written specifically for an 80386 computer that includes the new instructions added for that CPU *CANNOT* be run on an 8088 or even an 80286 CPU. In technical terms, we say that this kind of program is *NOT* Downward Compatible. This particular concept is important to all computer users because some software, such as Windows 3.0, specifically requires at least an 80286 CPU or later (i.e., an 80386 or 80486) because some of the programming instructions used simply did not exist in the original 8088 instruction set. Most of you probably know that the Intel 80xxx series (or compatible, such as the NEC V series) of CPU chips is used in all virtually all current models of Heath/Zenith computers.

In contrast, the Apple Macintosh uses the Motorola 68000 series of computer chips including the 68020, 68030 or 68040. Again, the assembly language used by this series of chips is upward compatible, but is not downward compatible for the same reason that the Intel chips are not. More importantly, the assembly language used by the Motorola 68000 series of CPUs is considerably different than that used in the Intel 80xxx series of CPUs. If you want to write an assembly language program for the Motorola 68000 series, then you will need to learn a different assembly language if you want to write a program for the Intel series of CPUs.

Because an assembler language is specifically dependent on the brand and type of CPU that a computer uses, we generally say that assembly language is "machine-dependent", although it is perhaps more accurate to say that assembly language is "CPU-dependent." That dependency applies to virtually all microcomputers that have different brands and types of CPUs, including mainframes. In the mainframe world, I learned the series 360 assembler, and when the 370 series was released, then I had to learn that. When I got my first microcomputer (an 8-bit H-89 kit that used a Z-80 CPU and ran the CP/M operating system), I spent the time to learn the assembler for that. Then I bought the Z-100 kit and began learning 8088 assembler for that. The point is that I spent a lot of time learning these assemblers because I needed to write various programs for each of these different kinds of computers.

The whole point of this discussion is that assembly language is CPU-dependent, which means that one needs to learn a different language for each different type of CPU. Moreover, it is generally agreed that assembly language is difficult to learn, and it is NOT recommended for beginners. Programming in assembly language takes a lot of time, and it can be difficult to "fix" a program. In short, there are lots of disadvantages to learning assembler that runs on any computer. For Heath/Zenith computers and most other PC compatibles, the usual standard is to use the Microsoft Assembler (MASM) for the Intel series of CPUs.

If assembler is so difficult to learn, program, and debug, why would one choose it as a programming language? One reason is due to the fact that assembler is CPU-related, which means that a program will generally run much faster in assembler compared to any other language. Assembler programs are nearly always much smaller in size requirements, including disk space, which is another reason for using assembler.

BASIC

BASIC is an acronym for Beginner's All-purpose Symbolic Instruction Code. Although the original BASIC was designed and developed at Dartmouth College in the late 1960's as an educational introduction to programming, it is available in a wide variety of versions and "dialects" that run on many different kinds of computers. Although BASIC was originally designed as an interpreted language, there are many different compiled "versions" available today, and nearly all of them are quite good. Indeed, the BASIC programming language was one of the reasons that microcomputers became so popular, and there are many reasons for that.

The fact that BASIC is generally an interpreted language is one reason many people think it is much easier to learn. All things considered, BASIC is an easy language to learn, which is probably its greatest advantage. Although you can do many different kinds of programming in BASIC, the language itself has a very primitive Input/Output (read/write) capability, which means that it is not generally suitable for reading and writing complex information to the screen and disk. That particular problem has been reduced to some extent with the introduction of newer and more powerful versions of BASIC by all vendors, but the language still has some limitations. Still, I have found that BASIC lacks the power of some of the other languages, and it is generally not suitable for many complex programming tasks, including large programs.

BASIC has its advantages, including the fact that it is easy to learn and is quite appropriate for small programs. Its near-English commands, statements, and syntax make it perhaps the most popular of all of the high-level programming languages. Unfortunately, the capability to define variables "on the fly" generally leads to sloppy programming by nearly everyone (including me). Virtually all other programming languages require variables to be specifically defined before they are used, and this requirement sometimes makes it difficult for a BASIC programmer to learn another language. When I teach BASIC, I always require my classes to define all variables at the beginning of the program (with REM statements) which helps reduce the problem of learning another language. Early versions of BASIC also required the use of line numbers (e.g., 100, 120, etc.) instead of labels, and using line numbers can make a program nearly unreadable. Most current versions of BASIC allow the use of labels which make program coding much easier and are more consistent with other high-level languages. I admit that I am not much of a BASIC fan, but I do use the language occasionally, primarily for checking out printer control commands to make sure they work as they are supposed to.

For a beginner, I recommend either Spectra's PowerBASIC (originally Borland's TurboBASIC) or Microsoft's QuickBASIC. These languages are generally inexpensive and are well supported by reference books and support libraries, which are the primary reasons I recommend them.

C

The C language was originally designed and developed for the UNIX operating system on the DEC PDP-11 at Bell Laboratories by Dennis Ritchie. For all practical purposes, the language was essentially defined in a book called *The C Programming Language* by Brian W. Kernighan and Dennis M. Ritchie, and this book is customarily referred to as "K & R."

C is certainly one of the most popular of today's languages because it combines the power of assembler's "bit-fiddling" capabilities with an ease of use common to many other high-level languages. Perhaps a large part of C's popularity is that compilers are available for a remarkably wide number of CPUs, ranging from microcomputers to mainframes. And because there is a generally accepted standard for this language, a program can be written once (with a little care) and can be run on just about any size computer, assuming that it is compiled on that computer of course. To accomplish this, the program must be written using the standard C functions, and the compiler takes care of any CPU-dependent changes that must be made.

I think that most C programmers would agree that C is a "terse" language, which simply means that a single statement can be extremely powerful. Indeed, the terse feature of the C language has caused some observers to say that C is a "write only" language because it can be written, but not read after the original coding is done. I disagree. A knowledgeable programmer can make most any program completely unreadable in any programming language, including BASIC. The whole secret of writing a good program in any language is to make sure that you include comments within the program as you are writing it.

Because of the popularity of the C language, one can find several versions of an interpreter for C. The only one that I have actually seen is the RUN/C interpreter by Lifeboat, and it seems to be a fairly standard and complete C implementation. Some people find it easier to learn an interpreted language, which is the reason I mentioned the availability of C interpreters. In general, they may not be totally compatible with compilers, but it is an alternative if you want to learn C.

For a beginner, I recommend either Borland's TurboC (including TurboC Professional) or Microsoft's QuickC. Again, both of these languages are generally inexpensive and are well supported by reference books and support libraries, which are the primary reasons I recommend them. My personal preference is TurboC Professional, and I use it for nearly all of the programs I write.

Pascal

Pascal is also a very popular language. It was named after a man named Pascal, and the language was developed in the mid-1970s by Nicholas Wirth. In general, it is quite powerful, but some of its restrictions make it difficult to use for some low-level "bit-fiddling" programming tasks. Pascal is an excellent procedural language that is quite suitable for beginners because it is fairly easy to learn, which no doubt accounts for a large part of its popularity.

For a beginner, I recommend Borland's Turbo Pascal, and although Microsoft also has a similar package, called QuickPascal, it is not nearly as popular or as well supported as Turbo Pascal. TurboPascal is especially well supported by reference books and available libraries, which are the primary reasons I recommend it.

Other Languages

There are a large number and wide variety of vendors, languages, and additional libraries that are available. In this article, I have not tried to discuss each and every possible combination and alternative for learning a programming language because that is more of a book-length discussion. Moreover, new languages and acronyms seem to pop up every day. For example, you may want to learn about artificial intelligence (commonly called AI) programming which includes some of the 4th Generation Languages (frequently referred to as 4GL) such as LISP or PROLOG. Modula 2 is also a good programming language and Ada is used by the Department of Defense. And there is no way to omit FORTRAN (Formula Translation) that is commonly used for scientific programming or COBOL (Common Business Oriented Language) that is used primarily on mainframes in the business world. Compilers of one kind or another are available for all of these languages, and you may want to check out a couple of my favorite sources for programming software.

Buying Programming Software

I have purchased a variety of programming software from two sources: The Programmer's Connection and The Programmer's Connection. The technical support provided by both firms is excellent, and they may be able to help you select a specific vendor's product based on your requirements. Just be sure that you know what your requirements really are. Addresses and phone numbers are listed at the end of this article. I recommend that you give one or both of them a call to obtain their catalogs, which will give you an idea of the "real" purchase price of software discussed in this article. For comparison purposes in this article, I have only provided the common list price of each software package, so I suggest obtaining at least one of these catalogs.

Using a Programming Language

Once you decide to do some programming, there is much more to it than simply choosing a vendor and a language. I have written several articles on various types of languages and compilers, and I inevitably receive letters to the effect that a certain language, vendor or documentation is not "good." A few letters have criticized a vendor's documentation of a language because a new user says it is not "complete," which in the various letters I have seen means that a beginning programmer did not know enough about the language to use it effectively. There are several things to consider about learning and using a programming language.

If you decide to learn to program, remember that it will take some time for you to learn the language, as well as the mechanics of using it. Have patience, and do not expect to learn how to program in a day or two. Remember that the purpose of virtually all of the vendors' products discussed here is to provide the programming tool, not teach you how to program. If you need additional information on programming, and most beginning programmers do, there are lots of good reference books for various languages. Look around, and find one that suits your needs. You may even want to check your local bookstore to find out what reference books are available to help you make the decision on which vendor's language to buy.

Buying a and learning how to use a specific language and compiler will probably not be the end of the learning process. Depending on what kind of programming you want to do, you may have to learn about the actual hardware in your computer, and you may have to learn some of the technical intricacies of DOS as well. For example, you may need to know exactly how your video display card works because you will find that some kinds of programming are unique to the display, such as CGA, EGA or VGA. Or, you may need to know the details of how DOS works with a disk drive if you want to write that kind of program.

The Last Decision

Perhaps the actual choice of a programming language and a vendor is not quite as important as the reference information available. For example, there are any number of good reference books you can buy that are specific to a given vendor's programming language, and I recommend that you choose the vendor and language based on information availability. In particular, I suggest that you choose a well-know vendor, such as Borland or Microsoft, because there are any number of reference books available for all of their languages. This is especially important for a beginning programmer because I believe that examples of working, functional programs are critical to the learning process.

If you want to learn the C language for example, then I specifically recommend either Borland's Turbo C or Microsoft's QuickC because there is so much reference information available on both. Moreover, there are any number of special "libraries" that you can buy for these popular languages which have support for special requirements, such as screen design or memory-resident programming. These libraries can make your programming tasks easier and faster because you do not have to "reinvent the wheel" when you want to try something new. Whatever you choose, make it easy on yourself by selecting a vendor and a language for which you can find reference information. Also, it is quite reasonable to choose a vendor and a language based on the fact that you have a friend who is knowledgeable and experienced in that particular language.

One other point about buying any programming software: be aware that one vendor's implementation of a language is NOT completely identical to another vendor's implementation. Each vendor attempts to add bells and whistles that are expected to differentiate and help "promote" that specific package. What that means is that you cannot expect a program written in TurboC to work perfectly with QuickC without at least some minor changes. That is especially important if you are considering learning BASIC.

I have intentionally omitted a discussion of GW-BASIC and IBM's BASICA from this discussion because I think PowerBASIC and QuickBASIC are far superior for a beginner because they can be compiled. The GW-BASIC and BASICA interpreters are virtually identical for programming purposes (assuming the corresponding versions of each are used), which means that a program written in one will run just fine under the other. It is important to note that a program written for either GW-BASIC or BASICA *MAY NOT* run under either PowerBASIC or QuickBASIC without at least some minor changes. For this reason, keep in mind that the vendor, language, and version are critical if you expect to run BASIC programs of any kind.

Powering Down

This article was not intended to be a review of programming languages and vendors; rather, it was intended to be an overview of some of the popular languages that are generally available for microcomputers. Moreover, this article was designed to be an introduction to programming languages for a beginner, and I did not try to discuss some of the technical points that might make one vendor's software preferable to another because that depends on one's needs and experience.

In the next article, we will look at: "How to Use COMMAND.COM and the DOS Environment." The introduction of hard drive partitions larger than 32 megabytes in ZDS MS-DOS 3.3 Plus has introduced some new requirements for proper DOS handling of these partitions. You will find out how to use the DOS environment with COMMAND.COM and see how it works with the SET, PATH, PROMPT, and SHARE commands, as well as the SHELL= command in CONFIG.SYS

If you have any questions about anything in this column, be sure to include a self-addressed, stamped envelope (business size preferred) if you would like a personal reply to your question, suggestion or comment.

PRODUCTS MENTIONED

Assembler

MASM Macro Assembler (Microsoft)	$150.00

BASIC

QuickBASIC (Microsoft)	$ 99.00
PowerBASIC (Spectra)	129.00

C

QuickC (Microsoft)	$ 99.00
QuickC/Quick Assembler (Microsoft)	199.00
TurboC (Borland)	150.00
TurboC Professional (Borland)	250.00
RUN/C Interpreter (Lifeboat)	250.00

Pascal

QuickPascal	$ 99.00
TurboPascal	150.00

SOFTWARE SOURCES

The Programmer's Connection
7249 Whipple Avenue NW
North Canton, OH 44720-7137
(800) 336-1166

The Programmer's Shop
5 Pond Park Road
Hingham, MA 02043
(800) 421-8006

Unix on a Shoestring

Exploring the MKS Toolkit

Brad Rylander
328 Hanover Avenue #10
Oakland, CA 94606

Curious about Unix? If you have one of Zenith's 80286 or 80386 machines, at least four megabytes of memory, a BIG hard disk, and a spare $500 to $1000 for software you can begin to experiment. You'll surely be well rewarded, but most of us will find the investment pretty steep.

If you simply want to learn Unix from a user's point of view and add a great deal of Unix's power to your DOS or OS/2 machine, the MKS Toolkit from Mortice Kern Systems is hard to resist. It lets you run all your favorite programs in a Unix-like environment even on a PC or XT compatible and gives you access to a large number of standard Unix tools.

Rightly or wrongly, Unix has been called cryptic and difficult to use. The MKS Toolkit will let you decide for yourself. You'll probably find it no more cryptic than DOS. At one time, Microsoft touted DOS as being Unix-like. You're already familiar with many of the Toolkit's commands, the new ones that you find useful will soon become second nature.

Difficult to use? Yes and no. In any field of endeavor, complex problems require complex solutions. The Toolkit won't make difficult tasks easy, it just may make them possible.

The everyday tasks you do in DOS; copying and deleting files, moving about in directories, all the general housekeeping chores are greatly simplified by the Toolkit. Here's some examples...

The cd command works the same as the DOS cd, but typing cd by itself brings you to a predefined "home" directory and cd - returns you to your previous working directory.

The mv (for move) command works the same as the DOS ren, you "move" a file from one name to another. However, mv also lets you rename directories and move (copy and delete original) one or more files from one directory or drive to another.

Wild cards, called "regular expressions" in Unix, can be as simple as the DOS equivalents but can also be made quite complex and highly specific.

You'll find all of the Toolkit's commands have these "nice little features." While none of them alone is worth raving about, taken together they'll give you new dimensions of control over your machine — power you'll appreciate.

The Toolkit is not cheap. A list price of $249 (and no heavy discounts) hardly qualifies it for the "shoestring" category. However, if you consider that the Toolkit consists of a complete implementation of the Korn shell as well as more than one hundred other programs, the price seems almost cheap.

To be honest, a few of these new commands have limited use under DOS. The command fg will hardly excite you with its ability to bring a DOS process back into the foreground and displaying the user's name with id won't dazzle anyone on a single user system.

On the other hand, you'll put many of the tools to immediate use and at least two of them, vi and awk, qualify as major applications. Vi is an exceptionally powerful full screen editor. Awk is billed as a "data transformation, report generation language." It's this and much more. But first let's look at...

The Korn Shell

Before we can discuss the shell we'd better define the term itself. Over the years, journalists have heard the word "shell" and not having any idea what it meant, applied it to menu programs than communicate with COMMAND.COM.

In computerese a shell is that part of the operating system that the user communicates with. The normal shell for MS-DOS is COMMAND.COM. Even if you're running from within a menu program such as XTREE, COMMAND.COM is your shell.

The MKS Korn shell is a full implementation of AT&T's latest release as defined by David Korn's book [2]. Although you can use the Toolkit with COMMAND.COM and run the shell as a separate program when desired, this program makes an excellent replacement for COMMAND.COM.

Why replace COMMAND.COM? We've all heard the maxim 'if it ain't broke, don't fix it.' Good advice. The shell won't give you multiuser or multitasking ability, your machine will still be MS-DOS at heart, so why bother?

There comes a time when all good things become outmoded. COMMAND.COM has changed little since version 1.0 of DOS. In fact, it still shows its CP/M heritage — it has few features that weren't present in the 1976 version of CP/M's CCP.

Not many of us would tolerate a 1976 word processor today, the spreadsheet wasn't even conceived back then, yet habit and the lack of an obvious alternative have kept us using trusty old COMMAND.COM and binding us to its limitations. Let's look at the added features the Korn shell (I'll just call it the shell in the future) brings us.

Command Line Editing

While COMMAND.COM does provide "template" editing of the last command via function keys, it's pretty difficult to take this feature seriously. The shell lets you choose between two built-in sets of editing commands, emacs or vi. The "real" emacs and vi are powerful programming editors, the versions built into the shell provide all the commands of the "real" versions that operate on a single line and horizontal scrolling is automatic.

This may not seem like an important feature; most DOS commands are but a few characters long and retyping them is no burden. The Unix commands however, can be hooked together in pipelines and each command can have a variety of switches that modifies its behavior.

These shell commands can easily exceed the 128 byte DOS limit. Recalling these long command lines, editing them until they do what you want and perhaps saving them for future use is an important part of the Unix game.

Command History

The shell automatically stores each command you give it in a file named sh_histo. Retrieving past commands is as easy as hitting <ESC> and scrolling up and down the list one command at a time with the arrow keys, pressing <RET> to rerun the chosen command or editing it with one of the built-in editors.

The shell limits the size of a command line to a generous 8k bytes. If this seems excessive compared to the DOS 128 byte limit, I'll soon show you how such a huge command line can be put to good use. As handy as the built-in editors are, once you start developing really large commands there's no substitute for your favorite full screen editor.

Full screen editing? Sure. A Unix command "line" can extend over many lines, even many screens up to an 8k byte limit. The shell provides a feature fc (for fix commands). Typing fc from the prompt will run the editor of your choice on the selected command. When you exit your editor the file will be deleted, the edited command will be passed to the shell, saved in the history file and executed.

The Shell as a Programming Language

While COMMAND.COM will execute batch files, it's perhaps too kind to say the usefulness of these files is somewhat limited. A cottage industry has grown around extending the abilities of batch files but even with this help, COMMAND.COM still shows its CP/M heritage—some of you may still remember the submit files from that venerable old operating system.

The shell is, in effect, a complete high level programming language. It has variables, functions, flow control statements, integer arithmetic, even a random number generator. Complete applications can be written in this language as readily as in BASIC or C.

This shell language can send its output to other programs in the Toolkit and can read and act upon the output of those other tools. Features that the shell lacks, floating point arithmetic for instance, can be handled by other tools, all within a shell program. And it's fast.

Shell programs, or scripts as they are called, can be written on the command line, tested and with fc (the fix command feature) modified and tested until they work. Then they can be saved in a file with a .ksh extension and used just like batch files. For really big programs (over the 8k byte command line limit), you would write directly into a file just as you would in BASIC or Pascal.

You don't need to be a programmer to write simple shell scripts. In fact you can change the extension on many of your batch files from .BAT to .KSH and they will run just as they are.

If you have access to a Unix machine, perhaps at a university, or access to a Unix bulletin board, you'll find many "precanned" scripts available. Then there's always the challenge of extending your knowledge of the language and writing some tricky scripts of your own.

The Tools

As with a true Unix system, the strongest feature of the MKS Toolkit is its large collection of programs. One-hundred-seven in all, these commands (or tools as they are called) perform functions as varied as counting the number of words in a file to maintaining an appointment reminder system.

The real strength of these tools lies not in their individual functions, but rather in the ease and elegance with which they can be combined to perform far more sophisticated functions. Let's take a look at some of the more important ones.

awk

This is an easy to use programming language. It gets its strange name from its authors; Aho, Weinberger and Kernighan. If the phrase "programming language" fills you with dread, relax. The emphasis is on the phrase "easy to use." Ok, I'll admit it, computer programming will never be truly easy but awk is about as easy as it gets.

How easy? Here's an example. This program counts the words in a text file and displays the total on the screen:

```
{count += NF}
END {print count}
```

That is the entire program! Because truly useful work can be done with a minimum of effort, awk is ideal for writing little "throwaway" programs — enter them on the command line and when they've done their job let them disappear.

Don't get the idea that awk is a "toy" language. Far from it. It has a full compliment of control-flow statements, arrays, math functions and powerful string handling abilities.

While awk is an interpreted language like GWBASIC, it's reasonably fast and can take advantage of a math co-processor. Awk programs typically run at about 60% of the speed of equivalent C programs. That's not much of a penalty to pay when you realize that the program's development time might be one percent of its C counterpart.

If you'd like to learn programming, awk is an ideal learning tool. This statement isn't my own idle speculation, but comes from the program's authors [1]. These guys should know. They not only wrote awk but have all been involved in the development of C as well as Unix itself. The language shares most of its syntax with C and the transition from awk to C should be gentle.

vi

This editor alone is worth the price of the Toolkit. Already I can hear howls of protest from vi haters — editors are such personal things. Vi has been called, among other things, vile. It's not easy to learn, no editor is. The real problem with vi is that until recently there has been no decent documentation for this gem, but you'll soon see that is no longer a problem.

While any powerful editor will require some effort to learn, many remain difficult to use. Vi is without question, simple to use. There is no groping for function keys or arrow keys. The function keys are there for your macros, the arrow keys work if you really insist on using them.

All commands are issued from the "home" position. If you can type, you can fly with vi. Commands are seldom longer than three keystrokes and the most commonly used ones are right under your fingertips. If it takes you more than six keystrokes to put the cursor anyplace on the screen you're really not trying.

Vi's only weakness is its lack of windows. However it does have provisions to move easily between files. While vi lacks windows, the program provides a host of useful features found in few other editors. Here's just a hint...

Abbreviations can save a lot of keystrokes but also allow you to create flow control templates for your favorite programming language. Undo and redo, plus automatically saving the last nine deletions limits any damage caused by "big" mistakes. The tags command lets you go directly to functions defined in other files. A line editor, ex, is contained within vi. More on this in a moment.

Vi is available on virtually all Unix systems. This alone makes it worthwhile learning. Unfortunately, the current MKS version of vi (ver. 3.1) has a couple of aggravating bugs that sometimes mess up word wrapping. These are the only bugs I'm aware of in the entire Toolkit.

The company knows of this and has hinted at a fix "real soon now." I believe them. I'm writing this with a vi from a long defunct firm. It has its own bugs but is more suitable for text, while the MKS vi is ideal for programming.

The Toolkit also provides three other editors. In order of complexity and power there are: ex (a line editor actually contained within vi), ed (another line editor) and sed (a "stream" editor).

Line editors such as EDLIN as we enter the ninteen-ninties? A good question, but they do have their place. While no one but a dedicated nostalgia buff would ever voluntarily use a line editor to enter text, editing that text is quite a different matter.

For really complex editing jobs, these "old fashioned" line editors can perform tricks that no word processor or full screen programming editor can touch. All of these editors can read a list of editing commands from a file and automatically apply the commands to your text file. In fact sed, the "stream" editor can only be used this way.

I've written several little command files, or scripts for ed that go through files I've down loaded from my favorite bulletin boards, extract messages addressed to me, delete multiple blank lines and generally clean up all of the nonessential text that clutter up my message files.

Here's a more serious and much more complex application. It's quite possible to write a script that will transform a bibliography written according to one style, let's say the Chicago Manual of Style, into another style, say the Government Printing Office Style Manual.

Try that with WordStar. While writing such a script might be almost as much fun as making the changes manually, the script probably already exists. Ask your friends or explore one of the many Unix bulletin boards.

The examples I've given barely provide a clue to the utility of the many programs in the Toolkit. Unix is famous for providing the user with a rich set of tools for manipulating text — a job we're all involved in. Awk may be the only free-form database manager you'll ever need. I've hinted at the power of the editors but I'll leave it to you to discover the usefulness of the dozens of tools that I haven't mentioned.

Documentation

MKS has done a fine job here. They provide two spiral bound manuals, one a 450 page Reference Manual, the other a 250 page User's Guide. The Reference Manual describes each of the hundred-fifty or so commands in typical Unix fashion. That is to say the descriptions are thorough, detailed and utterly terse, although bits of humor do pop up.

While this book is not exactly light reading, it is well organized. The information you need is easy to find. A brief appendix explains some of the more difficult points and it has an excellent index.

The User's Guide clears up most of the mysteries left by the Reference Manual. It clearly explains the subtle and possibly confusing differences between DOS and Unix terminology. It has excellent tutorials on the shell, awk and vi. In fact, only recently has a better manual for vi been published[3].

There is also an on-line help facility. Typing help command brings up an abbreviated version of the Reference Manual's pages for command. Even experienced users will find help indispensable.

Most commands have numerous options and it's far better to let help keep track of them than to try to memorize them all. It's easy to add your own "manual pages" to this system and to customize the existing ones. You'll give this feature a lot of use.

Further Reading

Although MKS has done a good job documenting their Toolkit, the topics are so vast that you'll welcome additional information. There are numerous Unix books that pertain directly to the Toolkit. Here are some of the best.

The Kornshell Command and Programming Language[2]. This book is the official AT&T specifications of the shell. As you might guess, this is some pretty heavy reading. The first two-thirds of the book covers each shell command in detail. Frequently, I find that I completely understand a command yet have no idea how to use it. Fortunately, the last third of the book is filled with examples — useful programs explained in detail.

Unix Shell Programming[4]. Probably the best book on this subject for beginners. It assumes practically no previous knowledge of Unix. It **starts** by teaching the most basic commands like ls to display a directory and cat to display a file. Starting with these basics, the book leads a step at a time into building an interesting Rolodex program. Highly recommended.

The Awk Programming Language[1]. The MKS awk tutorial is excellent. However if it piques your interest, then you deserve this book. As you might guess from the authors' names, this book is the definitive book on awk. Surprisingly, this book also makes the difficult concepts of programming accessible to beginners. It stands as a model of what language manuals could and should be.

It starts by emphasizing awk's simplicity, developing useful one and two line programs. It assumes no previous programming experience and leads the reader from the most basic concepts (what variables are) through topics delving deep into computer science (recursive decent parsing and topological sorting).

The book's scope is broad and this information is packed into two-hundred pages. Never-the-less, each new concept is clearly explained. There's no abstractions here, each concept is demonstrated with useful programs. Well worth your while.

The Ultimate Guide to the VI and EX Text Editors[3]. Until this book arrived on the scene, the MKS vi tutorial was by far the best vi material in print. Its grandiose title is truly descriptive. While the MKS tutorial is a far more than adequate introduction, if you get bitten by the vi bug you'll want this book.

Tools on the Z-100

While almost all of the programs in the Toolkit work perfectly well on the Z-100, there are two important exceptions. Neither the shell nor vi will run. ZPC will help with vi but the shell is too intimately tied to the machine's hardware to have any hope of running on the Z-100. Unless you have special need for the other programs, I don't think the Toolkit will be worth your time or money.

Tools For Everyone?

No. The Toolkit is fairly expensive and the learning curve is steep. If you're new to DOS, this is not your cup of tea. Learn DOS well before you attempt to go beyond it.

However, if you're an old hand at DOS who is beginning to feel restricted by COMMAND.COM's limitations and want more control over your machine then, by all means, give the Toolkit a try.

The MKS Toolkit
Mortice Kern Systems Inc.
35 King Street North
Waterloo, Ontario N2J 2W9
Canada

MS-DOS	$249
OS/2	$349

Requirements

PC Compatible, 512k RAM, MS-DOS 2.x or greater, or OS/2, 1.2 M bytes of disk space (although useful subsets of the kit may be installed on 360k floppies). Can use extended RAM and runs under networks.

[1] Aho, Alfred V., Brian W. Kernighan and Peter J. Weinberger, The Awk Programming Language, Addison-Wesley 1988

[2] Bolsky, Morris I. and David Korn, The Kornshell Command and Programming Language, Prentice Hall 1989

[3] Hewlett Packard, The Ultimate Guide to the vi and ex Text Editors, Benjamin/Cummings, Redwood City, CA 1990

[4] Kochan, Stephen G. and Patrick H. Wood. Unix Shell Programming, Hayden Books, Indianapolis, 1985

Rolodex is a registered trademark of Rolodex Corp.

Unix is a registered trademark of AT&T

WordStar is a registered trademark of WordStar USA

XTree is a registered trademark of XTree Co.

How Components Fail

Robert C. Brenner
Brenner Information Group
9340 Black Mountain Road, Suite E
San Diego, CA 92129

Each time we fondly look at our computer system, we envision a fine-tuned machine purring quietly–barely containing the power within. But when our system begins to malfunction–producing bizarre displays, inconvenient system crashes, and weird responses, sharp pain pierces our consciousness, worry lines form and our thoughts turn to disaster.

It's at this point that we desperately need information. Not only do we need to understand how our system works, but also how it degrades, and the things that we do to expedite the process. Plenty is written about system operation. Not enough is written about how they fail. This article describes how and why the components in your Heath/Zenith personal computer fail. As in life, everything is destined to wear out. But also like in life, we can have a direct impact on when and how wearout occurs.

If it moves, it wears. Therefore, everything in your system that contains mechanical moving parts will wear down and eventually fail. This includes the keyboard and the disk drives.

If it's electronic, current paths short or open. Even an intermittent problem is temporarily a short or an open conductive line. This includes the ICs, capacitors, resistors, inductors, delay lines, and anything else in your computer that operates on electrical current and voltage potential.

This article explores four primary areas of failure: the ICs, the keyboard, the display, and the disk drives.

Integrated circuits, transistors, diodes, capacitors, resistors and other electronic components (collectively called "the electronics") are fabricated using materials (metals and compounds) that react in a known way when placed in an electric potential or current path. The thousands of wires and solder joints of earlier systems are now contained in millions of micron-size traces and connections. Unused ICs and most other components have a shelf life of thousands of years. However, in a circuit, there is a finite operational lifetime. Each time we turn our system on, then off, then on again, we produce thermal stress. The electronics heat up, operate, cool down, and then heat up again. This stress eventually wears the component's atomic structure down to a point where operation becomes intermittent or stops altogether.

The failure mechanisms are shorting or opening of tiny current and signal paths. This also applies to cables and connectors. Every time we bend, twist, or otherwise move wires, pins, and other parts, we degrade the life of that component. Power cycles–brownouts, blackouts, heat, and humidity all affect the stress on those components.

Each time you touch your keyboard on a hot, dry day and discharge an electrical spike into your computer, you degrade the operational life of your system. Electrostatic discharge is one of the most serious, most damaging threats to electronics, yet, it's unseen so we usually forget to protect our systems properly.

Keyboards typically fail with one or more keys acting erratic or not working at all. Failure occurs in key contacts that remain open preventing key action from being recognized by the system; or a wire in a coiled cable breaks (opens) when you stretch the cable beyond its limits. Keyboards can also fail when someone spills liquid (usually coffee or soda pop) over the keys shorting out internal circuitry.

Keyboard failures are one of the two most common troubleshooting problems in computer systems. Usually the problem is bad key action caused by a key masher whose large, sausage-like fingers move like those on an 800-pound gorilla. Males are usually the culprits, especially when we have operated a mechanical typewriter where key action meant rapidly forcing a key down. The harder we pushed the key, the darker the resulting text on the paper. This behavior continued with electronic keyboards. However, electronic keyboards are not designed for rough handling. A slight pressure (mere ounces) is all that is required to cause the system to register key action. Pushing too hard on a key causes a membrane below the keys to bend. The farther it bends away from the keys, the more pressure needed to get the screen to echo our action. After a while, we get intermittent operation (usually a carriage return/enter key goes first) and then total failure of the key to respond.

Another problem involves those who feel a six foot coiled keyboard cable should extend to just that–six feet. They don't realize that six feet and 0.00004 inches is enough to break one of the wires inside causing intermittent operation or total keyboard interface failure.

Disk drive failures are the second most common failure items in a computer system. Your first indication is typically a read/write error. Or you could get a file, with missing or damaged data.

On older disk drives, a rubber belt, driven by a rotating motor turned a spindle shaft holding the disk. Just like in your automobile, belts can wear, stretch, or break. Usually, the belt stretched causing the speed of spindle rotation to gradually drift off. Degradation in the mechanical side of the drive system didn't affect the electronic side, which continued trying to write and read data at its clock speed. Unfortunately, when the speed drifted out of tolerance, the electronics could no longer have data ready when the disk was ready. Thus the "UNABLE TO READ" errors.

Today, plastic gears have replaced disk drive belts. And electronic circuitry is included to help keep the disk rotation speed within specs. Now it takes longer for the mechanical parts to wear out of tolerance.

Another cause for inability to read occurs when a read/write head get misaligned with the tracks on the disk. The head twists such that it can no longer be positioned directly over a track. This causes a serious reduction in the amount of magnetic signal that can be sensed from the track. Eventually, data that was written on the disk originally can no longer be read.

When you format a disk (hard or floppy), the controller causes identification marks to be written on the disk surface. These marks tell the system where tracks are and where sectors start and end. During use, these marks remain on the disk surface. The data in the sectors can be changed over and over again, but the ID marks remain. Therefore, as the head drifts out of alignment, it continues to write on the disk, but the information that gets written, is slightly off track. It's like taking the track area between a sector's ID marks and shifting it in or out slightly. Eventually, the head can no longer detect the ID marks and you start getting errors. This is when we do a low level format on a hard disk to re-establish the ID markings in line with the sector data.

We can cause mis-alignment by rough handling and improper use of the disk media. When you jam a floppy disk into a drive and slam the door shut before the disk is fully seated over the spindle, you press marks in the center area of the disk. Or you misalign the door mechanism and the disk seating alignment. Door misalignment can make the system unable to recognize disk change actions.

Data can be scraped off a disk surface by a read/write head that gets contaminated with debris. In my books on PC repair, I include a drawing that shows how big a strand of hair and a dust particle are compared to the tiny write/read head. In a floppy disk drive, the heads ride on the physical surface of the disk media. If you operate your computer in a workplace that is dirty, or if you smoke or eat crumb-producing food near your disk drives, you set your system up for debris- catastrophe. Sticky tars and nicotine can settle on the heads collecting dust particles and crumbs to clog the pathway. If hard particles stick to the heads, something must give during the fast rotation of the disk media. Either the heads will gouge a path on the disk surface, or the heads will chip, crack and split open. Both options are bad.

You can cause a similar problem by using poor quality disks. It's very difficult to determine the quality of the disks that you buy. The 49-cent disk may be terrific and work reliably for a long time. It may also have an oxide coating that is barely held to the disk by its binder material. Then when you use it in your drive, the oxide easily rubs off, decreasing the signal strength of the information that you stored on the disk and packing debris against the read/write heads as the disk rotates.

Yes, preventive maintenance and cleaning are worth the effort. And sometimes, paying a little more for your disk media yields a large return in reliable operation and data storage.

Displays work hard and long without complaining, but occasionally they fail—usually after you've filled the screen with critical information and are pressed for time to complete a project. Symptoms of display failure include an internal short or open, the center of the screen looks worn, a deposit forms on the inside of the screen, or the performance simply degrades to a point it is no longer useful.

Symptoms of an internal short include a "hum" noise, a bright bar on the screen, a single point of bright light, diagonal lines on the screen, or very poor contrast. An internal open looks like a black screen or lack of response to changes in the brightness or intensity control.

If the inside of the screen gets worn, the letters and numbers begin to "bloom" with a surrounding halo. Intensity control is poor. You can turn the intensity down and brightness is O.K., but black is very black and there are no or very poor shades of gray.

Deposits on the inside of the screen caused by electrons continuously bombarding the phosphor on the inside surface may cause the picture to look fuzzy and have poor brightness control. It can get so bad (especially at the edges) that no display image can be seen.

Marginal performance is indicated by lowered brightness control, screen magnetization, or lower sharpness caused by dirt and dust building up on the inside and outside of the display unit chassis. The electrons that strike the phosphor are negatively charged ions. Dust is positively charged. This makes your display a terrific air cleaner—sucking dust particles from all over the room onto the screen surface. Daily or weekly cleaning may become necessary to keep your display image bright and sharp.

As described in an earlier article, many things can cause system failure—blocking vents causing internal overheating, rough handling, spilling liquids into the circuitry, installing the wrong part in the wrong way, improperly connecting external devices, and operating our systems in electromagnetic or radio frequency interference fields that cause data transfer errors.

Our systems do fail. But there are things that we can do to extend their operational life. In another article I'll share preventive maintenance techniques. For now, be forewarned, and be careful. The system you save may be your own. ✱

"SEVEN TIMES EIGHT? NO PROBLEM. I'LL JUST SWITCH TO THE C DRIVE, CHANGE DIRECTORIES, LOAD THE SPREADSHEET, LET A1=7 AND A2=8, THEN A3 EQUAL A1 PLUS A2 ... AND THERE IT IS... 15! UH, NO... WAIT, LET'S SEE..."

Help for eaZy PC Owners, ZDS MS-DOS 4.0

William M. Adney
P. O. Box 531655
Grand Prairie, TX 75053-1655

To all of you who answered my July request for assistance to eaZy PC owners, thanks from me and from all of the eaZy PC owners who read this column. There are a few sources for the eaZy PC modules, but supplies seem to be dwindling rapidly. And these sources seem to be few and far between, with one exception. And that exception comes from a long-time supporter of Heath/Zenith systems: FBE Research Company, Inc.

eaZy PC Enhancements

In response to my request for information for eaZy PC modules, Dave Brockman of FBE Research advised me that some new products will be available by the time you read this. This includes the EZM-128 memory expansion module that adds 128 K of base memory so that the eaZy PC has the maximum of 640 K. The EZCLOCK calendar/clock attachment for the EZM-128 is also available, and it provides automatic date and time settings when the eaZy PC is booted. And Dave also sent me some information on connecting a modem to the mouse port, which will be included in this article.

Like the IBM AT, the eaZy PC was originally equipped with 512 K of base memory. This leads to the obvious question: "Why was the memory limited to 512 KB instead of the maximum 640 KB?" Basically, the answer is that memory chips are available in various sizes, such as 64 Kb, 128 Kb, 256 Kb, 512 Kb, 1 Mb, and so on. Memory chips are simply not manufactured in the 640 Kb "size", and you will note that each size listed is exactly twice the previous value. That is simply because computers work in binary, which is a "power of 2." The whole point is that it would take an extra "bank" of eight or nine memory chips (depending on whether or not parity is used) to complete the 640 KB configuration, and that takes a certain amount of space which was not available inside the eaZy PC (or the IBM AT). For the eaZy PC, the additional expansion was originally provided by a memory module that plugged into an option connector on the rear of the computer. For the IBM AT, the additional memory expansion required the purchase and installation of an additional memory card.

FBE Research's EZM-128 module consists of a printed circuit board with a protective cover and installation documentation. The installation is easy because the unit plugs into the option connector and is secured with two screws, so it takes just a couple of minutes. To give you an idea of what the EZM-128 contains, Dave sent me Photo 1 which shows the prototype of the board without the cover and without the EZCLOCK connector.

This is a working prototype, but the production board will look slightly different. If you are interested in adding a clock to your eaZy PC system, FBE has the EZCLOCK that plugs into the EZM-128 module, but the EZCLOCK connector is not shown in Photo 1.

Dave also mentioned that he specifically did not include the serial port and the modem in the design of the expansion module in an effort to keep costs down in low-volume production. He also suggested that he would consider developing a standard serial port as a plug-in unit to the EZM-128 in lieu of the EZCLOCK card if there is sufficient demand. Be sure to write to Dave if you are interested in a standard serial port for the eaZy PC.

As previously mentioned, the EZCLOCK provides automatic date and time settings when the system is booted. Why do you need something like this? Well, I believe it is very important to have the correct date and time settings established when the system is booted so that when you create or update a file, DOS will have the correct date and time. That may not seem too important, but it can be critical if you are looking for the most current version of a file you created. Unfortunately, I have noticed that most users seem to press RETURN in response to the date and time prompts on most computers which do not have a clock feature. That means virtually all user-created files show up with a January 1, 1980 date (the default), so it is impossible to tell when a file was actually created or updated. In short, I recommend a clock accessory for any computer that does not have this built-in feature, and that includes virtually all 8088-based CPU and compatible systems, such as the eaZy PC. All computers that have an 80286 CPU and later already have this built-in feature.

The EZCLOCK includes an MS-DOS assembly language program (with source code) that sets the date and time when the system is booted. The same program is also used to set the EZCLOCK's internal date and time. The unit includes a non-replaceable lithium battery with an expected life of ten years or more. The battery is used to maintain the date and time settings when the computer is powered off.

If you already have a memory module for your eaZy PC, or if you are more adventurous, the FBE SmartWatch pack-

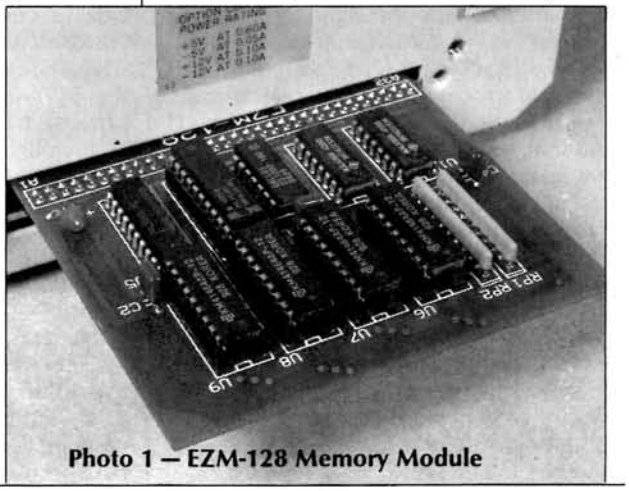


Photo 1 — EZM-128 Memory Module

age is also available. Installation of the SmartWatch requires partial disassembly of the eaZy PC, and if you are not familiar with disassembling your computer, you should get an experienced friend to help you. The SmartWatch consists of a 28-pin IC "socket" with the clock circuitry and battery embedded in it. After disassembling the computer, you carefully unplug the ROM, insert the SmartWatch into the ROM socket, and plug the ROM into the SmartWatch. Of course additional instructions are included with this package, but that is the basic installation procedure. And like the EZCLOCK, this package also includes an MS-DOS assembly language program (with source code) that sets the date and time when the system is booted. The same program is also used to set the SmartWatch's internal date and time. The unit includes a non-replaceable lithium battery with an expected life of ten years or more. The battery is used to maintain the date and time settings when the computer is powered off. As you may have guessed, the same technology is used by both the EZCLOCK and the SmartWatch. And the SmartWatch is also available for other 8088-based Heath and ZDS computers, such as the Z-151 and Z-100.

For those of you not familiar with FBE Research, they are highly reputable and have provided support to Heath/Zenith computer users for a long time. Even though I have not personally tested these units (because I do not have an eaZy PC), I have no hesitation in recommending them because I have a lot of confidence in their products. Now let's take a look at another problem-solving trick that Dave let me know about.

Adding a Modem to the eaZy PC

The eaZy PC included two ports: one was a standard parallel port that could be used for a printer, and the other was designated as a "mouse" port. The mouse port is actually a modified serial port that was intended to be only used for a mouse because not all of the "standard" serial port signals were provided for one reason or another. For additional information on the standard serial port signals, you should refer to the *POWERING UP* columns published in the July and September 1990 issues. In any case, one of the original eaZy PC modules included an "internal" modem, but there is a way to connect an external modem to the eaZy PC. This information is listed on an FBE Research product specification sheet called "EaZy PC Enhancements" dated July 90 and contains the following:

"The mouse port appears to be a standard 8250-type serial port with an AT-type 9-pin serial port minus a few input signals. In reality, there is no 8250 serial chip in the eaZy PC. It is simulated using the serial port built into the V40 microprocessor used by the eaZy PC.

The mouse port is physically addressed as COM1 but, because of the way things work, it will be recognized by DOS as COM1 if it is the only serial port present. This means that if you want to use the DOS MODE command to setup the port, use COM1. However, if you have a modem program which directly accesses the serial port (most do), use COM2. The mouse port supports only the standard baud rates (110, 150, 300, 600, 1200, 2400, 4800, 9600, 19200). Non-standard rates will not work.

The missing signals are Carrier Detect (pin-1), Data Set Ready (pin-6) and Clear to Send (pin-8). Because of the way the eaZy PC emulates the serial port, these signals will always appear to be true when read by software. Whether this will have any effect depends on the modem program being used. Some programs won't mind; some will.

To operating the modem, you will need a cable wired as follows:

eaZy PC	**MODEM**
DB9S	**DB25P**
2	3
3	2
4	20
5	7

If you do not wish to make your own cable, the Tandy Computer Products (Radio Shack) part number 26-269 cable also works.

The mouse port has been successfully used with the popular shareware program BOYAN v4.0 which is available on many bulletin boards. BOYAN is an excellent program, easy to use and powerful. Please register your copy! If you do not have access to a copy of BOYAN, you can order one from FBE for $9 to cover shipping and handling."

This solution may provide some help, but Dave told me that there is apparently a problem in using the mouse port with the Prodigy software. As previously mentioned, I have not tested any of this, but if Dave Brockman says it will work, you can rest assured that it WILL work. If you decide to connect an external modem to your eaZy PC using the cable that Dave suggests, I recommend that you also use the BOYAN program that he mentions because I have no idea what other programs might work (the Prodigy software apparently does not). Now that you know of one reliable source for the eaZy PC memory expansion module, let's move on to some other possible sources for eaZy PC accessories.

Other eaZy PC Sources

There are a couple of versions of the expansion modules that were mentioned in the letters I received. The EZA-11 is a combination of the Memory/Modem/Serial Port module, and the EZA-15 consists of only the Memory/Serial Port (without the modem).

Don Moore (St. Louis, MO) says that the EZA-11 (Memory/Modem/Serial Port) is available from Famous-Barr (a department store) in St. Louis for $149.99, and they have a toll-free number: (800) 528-2345. As I recall from my trips to St. Louis, Famous-Barr accepts both MasterCard and VISA, but I cannot remember for sure.

Jim Garrett (Gillette, WY) tells me that he was able to find the EZA-15 (Memory/Serial Port) for $78 at May D&F, 16th Street at Tremont Place, Denver, CO 80202; phone (303) 620-7500. Jim also mentioned that May D&F will ONLY accept a May D&F Charge Card, but he was able to obtain one in less than an hour by using a FAX. I presume that May D&F would also accept a check, but you might want to call them to make sure.

Richard Fitzgerald (S. Venice, FL) sent me a page from the July 1990 *Computer Shopper* that lists the EZA-15 (Memory/Serial Port) for $79.99 that is available from Tredex, 1875 Century Park East, Suite 2633, Los Angeles, CA 90067. Tredex can be reached at (800) 338-0039 (outside California) or (213) 277-4168 (in California). The ad shows that MasterCard and VISA are accepted, and there is no extra charge for using a credit card. Al Jones (Michigan City, IN) mentioned the same ad, but said that the $79.99 price is only when the module is purchased with an eaZy PC (shown in the same ad); the module ordered separately is $99.99. Al also mentioned that he order some EZA-15s (Memory/Serial Port) from Heath back in March and April of this year, but I understand that they are completely sold out now.

Richard Boneta (Richmond, KY) mentioned that an expansion unit for the eaZy PC was mentioned in *REMark*, but did not mention which issue. He says that the unit has its own power supply and will hold two PC expansion cards. The list price is $495, and the unit can be ordered from Connect Computer Company, 9855 West 78th Street, Eden Prairie, MN; phone (612) 944-0181. That is another option for expanding the eaZy PC, but I think that price is a little steep for a unit that only holds two cards.

All of this information was provided by a number of thoughtful HUG members, and I have not checked out any of this, so I will leave that up to you. Thanks again to all of you who answered my request for some help in locating sources for the eaZy PC accessories.

ZDS MS-DOS 4.0

I just received my ZDS MS-DOS update to version 4.0, and ZDS did a GREAT job on it. I am extremely impressed, and it is perhaps the nicest MS-DOS release so far. Although I really have not had a lot of time to check it out, it has some very interesting differences compared to the previous version 3.3 Plus. Perhaps the most interesting addition is the MS-DOS Shell, which is a Windows- like program that provides many of the point-and-shoot features for starting programs. Most notable by their absence in this release are the ZDS General Disk Utilities programs, GDU and GDUTSR, and the COMPACT disk "compression" program is also missing. I suspect one reason the GDU programs are absent in this release is because I would imagine, based on the letters I received, that these program caused all kinds of support problems, especially GDUTSR. Even though I have mentioned many times that one should NOT load a memory-resident program like this as a standard practice (i.e., in AUTOEXEC.BAT), especially without thorough testing with application software, it seems that quite a few people installed GDUTSR anyway. It turns out that GDUTSR was not "compatible" with some versions of Microsoft Word and Word Perfect, but that was really more of a problem with the way those word processors worked with files as opposed to a problem with GDUTSR. And I still think it is a waste of time to load GDUTSR for any application that provides for a backup of a previous version of a file like Word and Word Perfect. As most of you know, I am extremely cautious about backing up files on my hard drive, but I thought that GDUTSR was overkill, unless I was doing maintenance (i.e., deleting old files from the drive). It never occurred to me to load GDUTSR before starting an application, and I suspect that it did not occur to anyone at ZDS either. With 20/20 hindsight, it is easy to suggest that ZDS should have placed a caution about the use of GDUTSR in the Command Reference, but it is not always easy to predict how software will be used. Oh well, I guess that is just a matter of experience.

I suppose that some compromise was necessary because I received my update on three, 3.5-inch floppy disks which I ordered, but you can also order the update on 5.25-inch floppy disks. To give you an idea of the contents of the 3.5-inch floppy disks, I have included a DIR listing of their contents as Figures 1, 2, and 3.

Disk #1 contains two new commands in version 4.0: DSKSCAN.EXE and MEM.EXE. As far as I can tell DSKSCAN works essentially like the Mace Utilities REMEDY program that I discussed in last month's column. DSKSCAN attempts to read each sector on a disk. If any sector cannot be read successfully on the FIRST try, then DSKSCAN saves all available data from the cluster containing the bad sector, copies the data to a new cluster, modifies the FAT to reflect the new cluster number far that file, and updates the FAT to show that the old cluster is "bad." For the long-time ZDS MS-DOS users, you will note that the DETECT command has apparently been replaced by DSKSCAN, and the Bad Sector Table no longer exists. I believe that this is a much better approach because it allows a dynamic update to the FATs as the hard drive ages, and it does not require a FORMAT to the hard drive to "install" any additional bad cluster marks. DSKSCAN does not have any optional switches because it automatically performs its function.

The MEM command is one of the most interesting and helpful commands I have seen in a long time. Most everyone knows by now (I hope) that the CHKDSK command can be used to display used and available memory within the usual 640 KB address space, but there was no way for DOS to display the availability of expanded (EMS) and extended (XMS) memory until the MEM command. You can use the MEM command three ways: without any optional switch (see Figure 4), with the /PROGRAM switch (see Figure 5) or with the /DEBUG switch (see Figure 6).

The MEM command by itself displays the base memory available and used below 640 K (like CHKDSK), as well as the total expanded (EMS) and extended (XMS) memory available as shown in Figure 4. My Z-386/16 system has a total of 5 MB of memory installed (Z-515 and Z-505 cards) with 1 MB of base memory, 1 MB of extended memory, and 3 MB of expanded memory as shown. The MEM command with the /PROGRAM switch provides the display shown in Figure 5.

The MEM command with the /PROGRAM switch displays information about the programs loaded into memory, as well as information about the sizes and types of available memory. Figure 5 shows that I have the memory-resident SETVID program (provided with the ZDS FTM monitor for screen time-out) loaded, as well as the DOS SHARE program (because my C partition is 48 MB). The MEM command with the /DEBUG switch provides the display shown in Figure 6.

The MEM command with the /DEBUG switch displays program information similar to the /PROGRAM switch, but also includes additional information about internal DOS device drivers (e.g., COM1, LPT1, etc.) and the expanded memory file handle.

There are also two new commands for the CONFIG.SYS file: INSTALL= and REM.

```
 Volume in drive B is 1322-01A
 Directory of  B:\

COMMAND  COM    37744  11-20-89   2:48p
CONFIG   SYS      104  11-13-89   2:36p
AUTOEXEC BAT       32  12-19-89  12:49p
SELECT   COM    55203   2-08-90  10:01a
SYS      COM    11600  12-29-89  10:30a
COUNTRY  SYS    12806  10-06-88  12:00a
DISKCOPY COM    10396  10-06-88  12:00a
DISKCOMP COM     9857  10-06-88  12:00a
DISPLAY  SYS    15692  10-06-88  12:00a
FORMAT   COM    23563   1-02-90   8:23a
KEYB     COM    15176  12-01-89   4:12p
KEYBOARD SYS    23328  10-06-88  12:00a
MODE     COM    22960   7-10-89   1:26p
SHARE    EXE    13424  10-14-88  12:00a
ANSI     SYS     9105  10-06-88  12:00a
APPEND   EXE    11154  10-06-88  12:00a
APPLY    COM     2353  12-01-89   4:06p
ASSIGN   COM     5753  10-06-88  12:00a
ATTRIB   EXE    18263  10-06-88  12:00a
BACKUP   COM    36880  10-14-88  12:00a
CHKDSK   COM    17787  10-14-88  12:00a
COMP     COM     9459  10-06-88  12:00a
DEBUG    COM    21574  10-06-88  12:00a
DRIVER   SYS     5241  10-06-88  12:00a
DSKSCAN  EXE    16401  12-21-89  10:41a
EDLIN    COM    14069  10-06-88  12:00a
EMM      SYS     6544   1-12-90  10:20a
EMM386   SYS    87776   3-22-90   2:34p
EXE2BIN  EXE     7963  10-14-88  12:00a
FASTOPEN EXE    16718   4-07-89  12:00a
FC       EXE    15807  10-06-88  12:00a
FILESYS  EXE    11129  10-06-88  12:00a
FIND     EXE     6037  12-01-89   4:11p
GRAFTABL COM    10239  10-06-88  12:00a
GRAPHICS COM    16693  10-06-88  12:00a
       35 File(s)     37888 bytes free
```

Figure 1 — MS-DOS Disk #1 (3.5-inch) Contents

```
Volume in drive B is 1323-00C
 Directory of  B:\

CONFIGUR COM   115193   1-02-90  10:17a
CONFIGUR HLP     5813  12-21-89  10:21a
DSKSETUP COM    76157   1-02-90   9:14a
DSKSETUP HLP     3690  12-21-89   2:05p
{CW}MSG  DAT    12459  12-18-89   4:34p
{CW}ERR  DAT     3118   6-19-89  12:00p
{CW}     OVL     2915  12-01-89   4:36p
HIMEM    SYS     6261  10-06-88  12:00a
IFSFUNC  EXE    21739  10-14-88  12:00a
JOIN     EXE    17813  10-06-88  12:00a
LABEL    COM     4458  10-06-88  12:00a
LINK     EXE    43988  10-06-88  12:00a
MACHINE  COM    15345  10-13-89   9:55a
MEM      EXE    20005  10-06-88  12:00a
MORE     COM     2134  10-06-88  12:00a
NLSFUNC  EXE     6878  10-06-88  12:00a
PART     EXE    74619   1-02-90   4:48p
PART     HLP     9782  12-29-89   3:26p
PREP     EXE    24911  12-29-89   5:22p
PRINT    COM    14739  12-01-89   4:15p
PRINTER  SYS    18914  10-06-88  12:00a
RECOVER  COM    10588  10-06-88  12:00a
GRAPHICS PRO     9397  10-06-88  12:00a
       23 File(s)    196608 bytes free
```

Figure 2 — MS-DOS Disk #2 (3.5-inch) Contents

The INSTALL= command is similar to the DEVICE= command. As most of you know, the DEVICE= command is used to load various device drivers that have a SYS

```
  Volume in drive B is 1324-00C
  Directory of  B:\

4201      CPI      6404  10-06-88  12:00a
4208      CPI       720  10-06-88  12:00a
5202      CPI       370  10-06-88  12:00a
EGA       CPI     49068  10-06-88  12:00a
LCD       CPI     10703  10-06-88  12:00a
REPLACE   EXE     19415  10-06-88  12:00a
RESTORE   COM     36946  10-06-88  12:00a
SEARCH    COM      7636  10-13-89   9:57a
SHIP      COM       917  10-13-89  10:00a
SORT      EXE      5882  10-06-88  12:00a
SUBST     EXE     18467  10-06-88  12:00a
TREE      COM      6302  10-06-88  12:00a
VDISK     SYS      3121  10-13-89  10:01a
XCOPY     EXE     17055  10-06-88  12:00a
ZCACHE    SYS      5208  12-20-89   9:00a
ZCOM      EXE     24117  10-13-89   9:59a
ZSPOOL    COM      3007  10-13-89  10:01a
SHELL     CLR      4406  10-06-88  12:00a
SHELL     HLP     66527  10-06-88  12:00a
SHELL     MEU      4588  10-06-88  12:00a
SHELLB    COM      3894  10-06-88  12:00a
SHELLC    EXE    153855  10-06-88  12:00a
DOSUTIL   MEU      6660  10-06-88  12:00a
PCIBMDRV MOS        263  10-06-88  12:00a
       24 File(s)    262144 bytes free
```

Figure 3 — MS-DOS Disk #3 (3.5-inch) Contents

```
655360 bytes total memory
    655360 bytes available
    556480 largest executable program size

   3145728 bytes total EMS memory
   3145728 bytes free EMS memory

   1048576 bytes total extended memory
   1048576 bytes available extended memory
```

Figure 4 — MEM Command Display

```
Address      Name        Size       Type
-------      --------    ------     ------
000000                   000400     Interrupt Vector
000400                   000100     ROM Communication Area
000500                   000200     DOS Communication Area
000700       IO          002D00     System Program
003400       MSDOS       008E40     System Program
00C240       IO          008C20     System Data
                EMM      002B30      DEVICE=
                         000820      FILES=
                         000100      FCBS=
                         0048E0      BUFFERS=
                         0001C0      LASTDRIVE=
                         000CD0      STACKS=
014E70       MSDOS       000030     - Free -
014EB0       SETVID      000080     Environment
014F40       SHARE       0018A0     Program
0167F0       COMMAND     001640     Program
017E40       COMMAND     000100     Environment
017F50       SETVID      000250     Program
0181B0       MEM         000070     Environment
018230       MEM         012F00     Program
02B140       MSDOS       074EB0     - Free -

  655360 bytes total memory
  655360 bytes available
  556480 largest executable program size

 3145728 bytes total EMS memory
 3145728 bytes free EMS memory

 1048576 bytes total extended memory
 1048576 bytes available extended memory
```

Figure 5 — MEM/PROGRAM Command Display

file type (or extension, if you prefer), such as ANSI.SYS, EMM.SYS or VDISK.SYS. In contrast, the INSTALL= command is used to load specific EXE support files: FASTOPEN, KEYB, NLSFUNC, and SHARE. These four commands can also be entered from the keyboard (because they are EXE files) or added to the AUTOEXEC.BAT, but the INSTALL= command gives you another option. If you use any kind of hard disk optimization, unfrag or compression program however, I recommend that you NOT use the FASTOPEN command as a standard practice because it may have a conflict. But if you do use FASTOPEN, be absolutely SURE that you remove it from the CONFIG.SYS or AUTOEXEC.BAT file (and rebooting the system) before you run an optimization program. Otherwise, you may find the files on your hard drive have been "scrambled" by the optimization program. If you want to check this out on your own system, I recommend that you do a complete hard drive backup before you run your optimization program with FASTOPEN, but I suggest that it is best to avoid problems by rebooting the system without the FASTOPEN command.

The REM command is also new to the CONFIG.SYS file, and it allows you to add comments just like its counterpart in the AUTOEXEC.BAT file.

Two new device drivers are included with ZDS MS-DOS 4.0: EMM386.SYS and HIMEM.SYS. As you might expect from its name, EMM386.SYS can only be used on 80386-based systems (including the SX). If your computer has extended memory, this device driver essentially "converts" the extended memory to expanded memory that can be used by programs that support EMS. The ZDS User's Reference (page D.14) contains all kinds of cautions about the use of this device driver, so be sure to read it before you attempt to add EMM386.SYS to the CONFIG.SYS file.

HIMEM.SYS supports the XMS (eXtended Memory Specification) version 2.0 which provides a standard, hardware-independent (i.e., not ZDS-dependent) method for accessing extended memory above 1 MB. Therefore, it cannot be used on an 8088- based system, such as a Z-151, and it can only be used on computers with at least any 80286 or later CPU.

How useful these two new device drivers are depends on what kind of computer you have and what software you are using. I still maintain that it is a waste of money to buy extended or expanded memory unless you have software that either requires it (e.g., OS/2 requires extended memory) or can use it. For example, Quattro Pro can use expanded memory, but it really does not help much unless you have a VERY large spreadsheet.

Other MS-DOS 4.0 Enhancements

You may remember that I criticized the "new" PART command in version 3.3 Plus because it was difficult to use and did not always work as you might expect. ZDS has completely redesigned the PART command in this version, and it is quite clear and easy to use — my compliments to the programmer who did the job. CONFIGUR has also been redesigned, and I found it easy to use also, although I did not have any problems with it in 3.3 Plus.

Although there seem to be a number of minor enhancements to existing commands throughout this version, FORMAT has the most visible changes. Instead of displaying the cylinder and head number during a hard drive FORMAT, this new version displays a "percent complete," which is more meaningful to most people. FORMAT also has a new switch that will make it easier to figure out how to format "low-density" floppy disks in high-density drives. The /F switch is used to specify "Format capacity," which can be entered according to the capacity of the disk. Since I continue to receive letters about this problem, let me digress a moment to discuss how this works.

The most important point is that a floppy disk drive will always attempt to format a disk at the maximum capacity of which it is capable. For example, if you have a high-density (1.4 MB) 3.5-inch floppy disk drive, FORMAT will always

```
Address      Name         Size        Type
-------      --------     ------      ------
000000                    000400      Interrupt Vector
000400                    000100      ROM Communication Area
000500                    000200      DOS Communication Area
000700       IO           002D00      System Program
                 CON                    System Device Driver
                 AUX                    System Device Driver
                 PRN                    System Device Driver
                 CLOCK$                 System Device Driver
                 A: - D:                System Device Driver
                 COM1                   System Device Driver
                 LPT1                   System Device Driver
                 LPT2                   System Device Driver
                 LPT3                   System Device Driver
                 COM2                   System Device Driver
                 COM3                   System Device Driver
                 COM4                   System Device Driver
003400       MSDOS        008E40      System Program
00C240       IO           008C20      System Data
               EMM        002B30       DEVICE=
                          000820       FILES=
                          000100       FCBS=
                          0048E0       BUFFERS=
                          0001C0       LASTDRIVE=
                          000CD0       STACKS=
014E70       MSDOS        000030      - Free -
014EB0       SETVID       000080      Environment
014F40       SHARE        0018A0      Program
0167F0       COMMAND      001640      Program
017E40       COMMAND      000100      Environment
017F50       SETVID       000250      Program
0181B0       MEM          000070      Environment
018230       MEM          012F00      Program
02B140       MSDOS        074EB0      - Free -

   655360 bytes total memory
   655360 bytes available
   556480 largest executable program size

Handle       EMS Name     Size
------       --------     ------
     0                    000000

  3145728 bytes total EMS memory
  3145728 bytes free EMS memory

  1048576 bytes total extended memory
  1048576 bytes available extended memory
```

Figure 6 — MEM/DEBUG Command Display

attempt to format it to the 1.4 MB capacity, even if you use a 720 KB (double density) disk. That will usually fail because a 720 KB disk is not manufactured to "accept" the recording density required by the 1.4 MB format. In MS-DOS 3.3 Plus, the only way to successfully format a 720 KB disk in a 1.4 MB drive is to use a command like: FORMAT B:/N:9. The /N:9 switch tells FORMAT to initialize 9 sectors per track for the 720 K format, instead of the maximum drive capability of 18 sectors per track used in the 1.4 MB format. Similarly, most people used a command like "FORMAT A:/4" to format a 360 K floppy disk in a high-density (1.2 MB), 5.25-inch drive.

The /F switch in this new FORMAT command makes it much easier because all you have to know is the disk capacity. To initialize a 720 K disk, all you need to enter is a command like: FORMAT B:/F:720 (or 720K or 720KB). To initialize a 360 K disk, you can enter a command like: FORMAT A:/F:360 (or 360K or 360KB). Much, much easier if you don't want to have to remember technical stuff like how many sectors per track there are in these formats.

Windows 3.0

All reports indicate that there is a considerable interest in Windows 3.0, and many people think this latest version is really neat. However, I have already seen some reports of general problems, and there is one that is apparently unique to some Heath/Zenith computers.

First, a general problem that is NOT unique to Heath/Zenith computers. The problem is that the Windows' 3.0 SMARTDRIVE program is not compatible with some third-party hard drive partitioning schemes. For example, Ontrack's Disk Manager program (furnished with many Seagate hard drives) is known to cause problems with high-capacity hard drives that have more than 1024 cylinders. If you have already spent the money to upgrade to that kind of hard drive, it is false economy to not spend the additional $49 to buy an updated version of ZDS MS-DOS (using the update card) that directly supports the drive. Frankly, I would not trust any third-party partitioning software anyway because I have also heard that they cause problems for other programs, such as Mace and Norton Utilities. You are far better off using an operating system that directly supports these large drives, and I highly recommend ZDS MS-DOS 4.0 for that purpose.

There are also some reports that the "generic" (i.e., Microsoft) version of Windows 3.0 has problems running on some Heath/Zenith laptop computers, notably the SupersPort series and the discontinued TurbosPort 386. I suspect that most of those reports are probably true because the generic Windows 3.0 does not provide specific video drivers for any laptop LCD displays, which may cause problems on some systems. If you are considering buying Windows, I recommend that you buy the ZDS version that contains the video drivers for all Heath/Zenith computers, including the SupersPort laptops. Which leads to a question that I get all the time...

What is the REAL difference between a ZDS version of Windows and a "generic" version? As I just mentioned, perhaps the most important difference is that the ZDS version includes video drivers specifically for Heath/Zenith computer hardware, including laptops. And because these video drivers are specifically customized for the hardware, they will probably have an improved display speed over a generic driver. Which brings up the second difference... The ZDS version of Windows is "optimized" to take advantage of Heath/Zenith hardware features which will give you better system performance (i.e., speed). I think that the custom video drivers and the better system performance are the key reasons to use the ZDS version of Windows.

Speaking of video drivers, I keep hearing reports that Windows 3.0 seems to have problems with several different "brands" of computers, including some Heath/Zenith computers. After checking this out, I found that the problem seemed to be with the video display card (not the computer itself), but I was not able to identify any specific manufacturers. For the two Heath/Zenith computers that I know of that had the problem, neither user knew what brand of video card was installed except that it was NOT a Heath/Zenith card, such as the Z-449, Z-549 or HVB-550. Since I was on a consulting assignment, it was not appropriate for me to begin disassembling these computers, and the technician responsible for maintaining the computers was not available in the limited time I had. The important point is that not all video cards may be compatible with all software, including Windows 3.0. By the way, the major symptom of this problem is that Windows will simply cause the system to go into a "hard freeze", which requires a power-off reset to recover. Apparently, Windows does something at the hardware level that causes problems with some video boards. In general, I prefer to use either a Heath/Zenith or Video 7 video board in my computers, and I have not had any problems with them.

For what it is worth, I also have been seeing a number of reports that Windows 3.0 is not necessarily compatible with applications that have been designed to run on an earlier Windows version. Even early versions of Microsoft's own Word for Windows and Excel apparently have problems with Windows version 3.0, even though both programs ran fine under earlier versions. If you are having a problem running some Windows-based software under version 3.0, then I recommend that you contact that software vendor to find out when an update will be available that runs under Windows 3.0.

And finally, there is one reported and confirmed problem that is unique to Heath/Zenith computers running Windows 3.0. The problem is specific to Z-241 and Z-248 computers which still use the 84-key keyboard (NOT the 101-key keyboard). The symptom is that typed keys are "missed" when running Windows 3.0 on these systems, and that is due to a timing problem between the keyboard and the system. Based on the information I have, it seems that Microsoft got into the low-level hardware again, and that is what causes the problem. Although I am told that ZDS is working on the problem, the only quick cure is to use a 101-keyboard with these systems. Again, the only problems reported are with the Z-241 and Z-248 which use the 84-key keyboard. If you are using the 101-key keyboard with any Heath/Zenith computer (including laptops which use a 101-key "emulation"), you will not have this problem. As I was researching this, I wondered if Microsoft kind of ignored the 84-key keyboard...

Powering Down

Regardless of whatever the problems are, reports are that Windows 3.0 is selling fast. Indeed, there is so much interest in Windows 3.0 that I will be writing a column about it. Stay tuned for more information.

For help in solving specific computer problems, be sure to include the exact model number of your system (from the back of the unit or the model series from the Owner's Manual), the ROM version you are using (use CTRL-ALT-INS to find it), the DOS version you are using (including both version and BIOS numbers from the VER command), and a list of ALL hardware add-ons (including brand and model number) installed in your computer. The list of hardware add- ons should specifically include memory capacity (either added to an existing board or on any add-on boards), all other internal add-on boards (e.g., modems, bus mouse or video cards), the brand and model of the CRT monitor you have, and the brand and model of the printer with the type of interface (i.e., serial or parallel) you are using. Also be sure to include a listing of the contents of the AUTOEXEC.BAT and CONFIG.SYS files unless you have thoroughly checked them out for potential problems (e.g. TSR conflicts). If the problem involves any application software, be sure to include the name and version number of the program you are running when the problem appears.

If you have questions about anything in this column, or about Zenith or Heath systems in general, be sure to include a self-addressed, stamped envelope (business size preferred) if you would like a personal reply to your question, suggestion, comment or request.

Products Discussed
HUG SOFTWARE
Powering Up (885-4604) $12.00
Heath/Zenith Users' Group
P. O. Box 217
Benton Harbor, MI 49022-0217
(616) 982-3463 (HUG Software only)

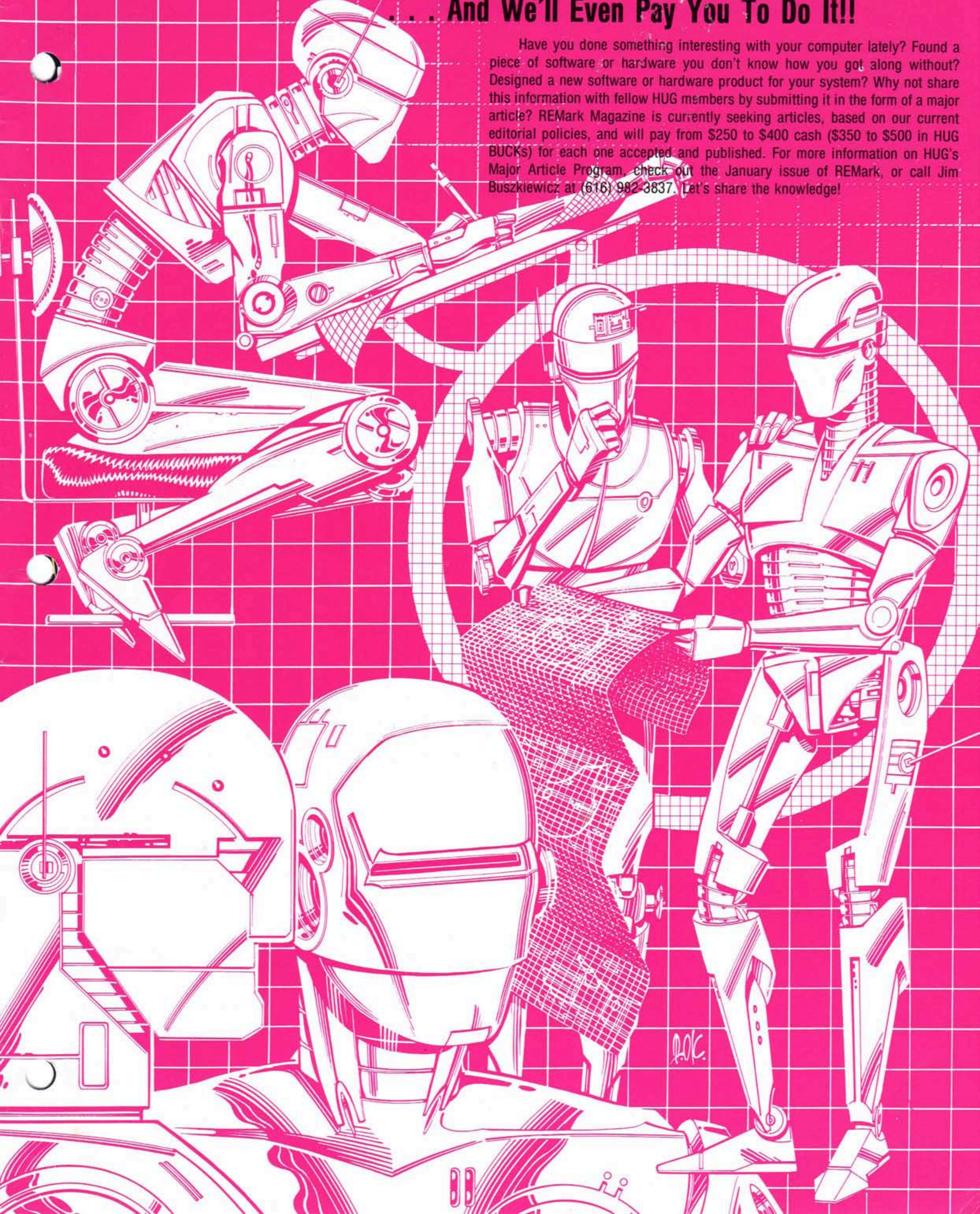
HARDWARE
For eaZy PC only
EZM-128 128 K Memory Module $125.00
EZCLOCK 35.00
FBE Research Company, Inc.
P.O. Box 68234
Seattle, WA 98168
(206) 246-9815 (MasterCard and VISA orders)

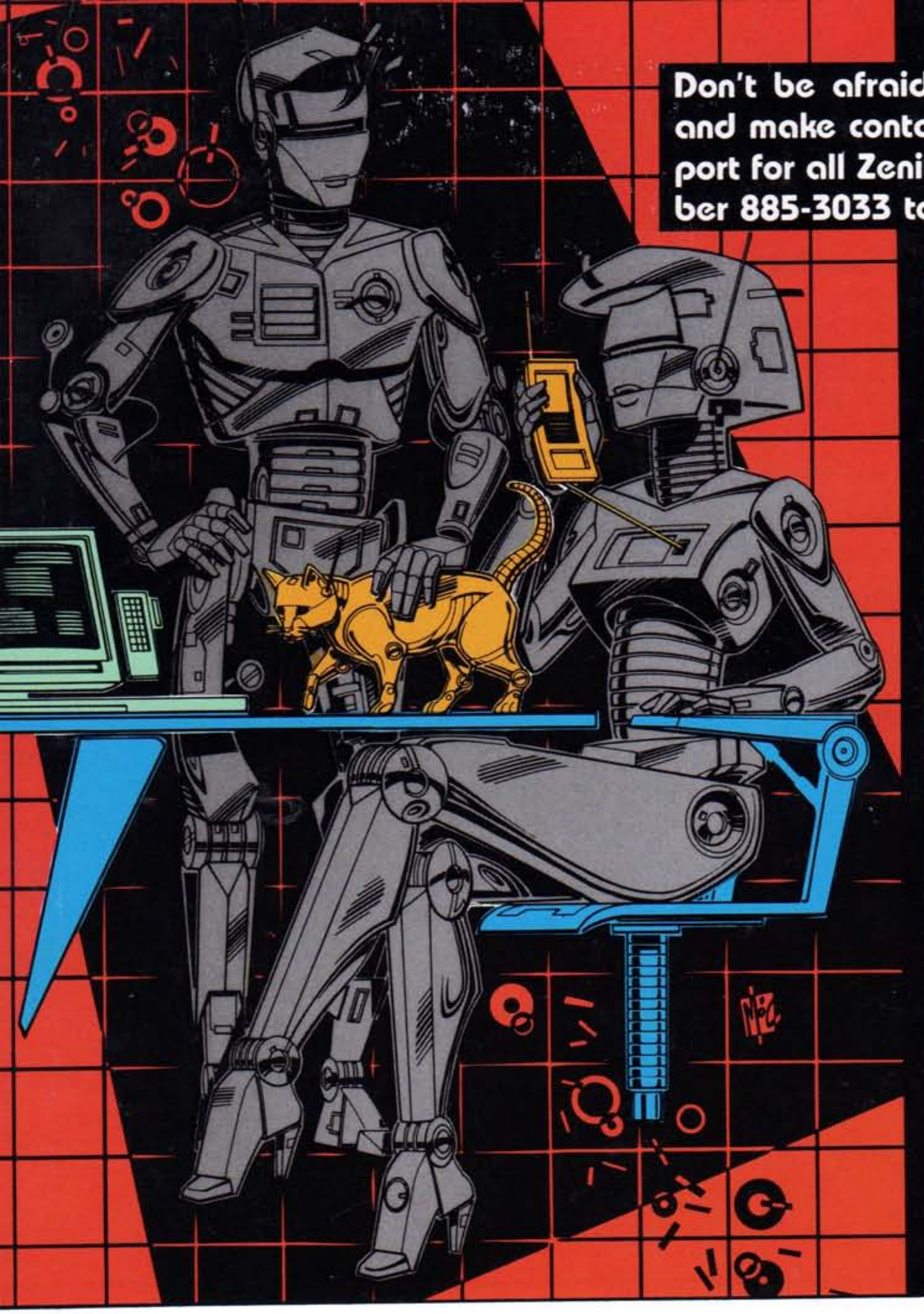
SOFTWARE
MS-DOS Version 4.0 $149.00 (List price)
(Mail order with update card only) 49.00
3.5-inch version (OS-103-MS)
5.25-inch version (OS-105-MS)
Heath/Zenith Computer Centers
Heath Company Parts Department
Hilltop Road
St. Joseph, MI 49085
(800) 253-0570 (Heath Catalog orders only)

✱

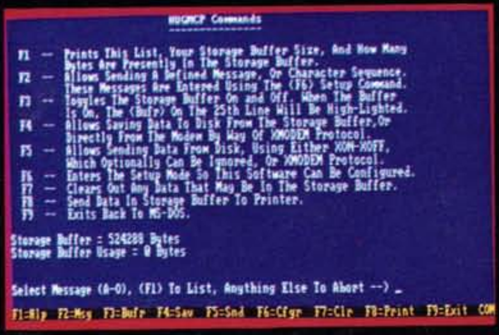
Authors, Share Your Knowledge With Other HUG Members, . . . And We'll Even Pay You To Do It!!

Have you done something interesting with your computer lately? Found a piece of software or hardware you don't know how you got along without? Designed a new software or hardware product for your system? Why not share this information with fellow HUG members by submitting it in the form of a major article? REMark Magazine is currently seeking articles, based on our current editorial policies, and will pay from $250 to $400 cash ($350 to $500 in HUG BUCKs) for each one accepted and published. For more information on HUG's Major Article Program, check out the January issue of REMark, or call Jim Buszkiewicz at (616) 982-3837. Let's share the knowledge!

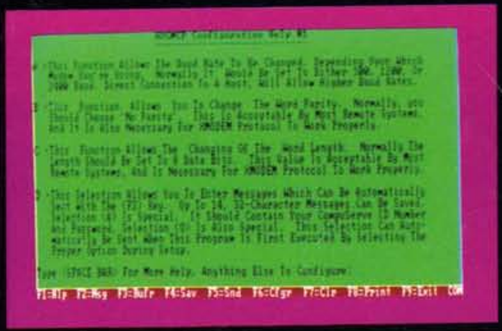




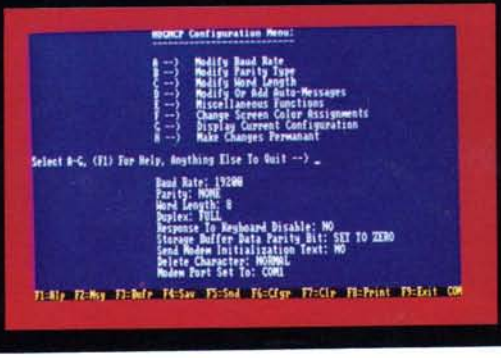
Don't be afraid to communicate! Get HUGMCP and make contact the easy way. Now with support for all Zenith Laptops, order HUG Part number 885-3033 today.



```
                         HUGMCP Commands
                         ---------------

F1 --  Prints This List, Your Storage Buffer Size, And How Many
       Bytes Are Presently In The Storage Buffer.
F2 --  Allows Sending A Defined Message, Or Character Sequence.
       These Messages Are Entered Using The (F6) Setup Command.
F3 --  Toggles The Storage Buffer On and Off. When The Buffer
       Is On, The (Bufr) On The 25th Line Will Be High-Lighted.
F4 --  Allows Saving Data To Disk From The Storage Buffer,Or
       Directly From The Modem By Way Of XMODEM Protocol.
F5 --  Allows Sending Data From Disk, Using Either XON-XOFF,
       Which Optionally Can Be Ignored, Or XMODEM Protocol.
F6 --  Enters The Setup Mode So This Software Can Be Configured.
F7 --  Clears Out Any Data That May Be In The Storage Buffer.
F8 --  Send Data In Storage Buffer To Printer.
F9 --  Exits Back To MS-DOS.

Storage Buffer : 524288 Bytes
Storage Buffer Usage : 0 Bytes

Select Message (A-O), (F1) To List, Anything Else To Abort --> _
F1=Hlp  F2=Msg  F3=Bufr  F4=Sav  F5=Snd  F6=Cfgr  F7=Clr  F8=Print  F9=Exit  COM
```



```
                  HUGMCP Configuration Help #1

A -This Function Allows The Baud Rate To Be Changed. Depending Upon Which
   Modem You're Using, Normally It Would Be Set To Either 300, 1200, Or
   2400 Baud. Direct Connection To A Host, Will Allow Higher Baud Rates.

B -This Function Allows You To Change The Word Parity. Normally, you
   Should Choose 'No Parity'. This Is Acceptable By Most Remote Systems.
   And It Is Also Necessary For XMODEM Protocol To Work Properly.

C -This Function Allows The Changing Of The Word Length. Normally The
   Length Should Be Set To 8 Data Bits. This Value Is Acceptable By Most
   Remote Systems, And Is Necessary For XMODEM Protocol To Work Properly.

D -This Selection Allows You To Enter Messages Which Can Be Automatically
   Sent With The (F2) Key. Up To 14, 32-Character Messages Can Be Saved.
   Selection (A) Is Special. It Should Contain Your CompuServe ID Number
   And Password. Selection (B) Is Also Special. This Selection Can Auto-
   matically Be Sent When This Program Is First Executed By Selecting The
   Proper Option During Setup.

Type (SPACE BAR) For More Help, Anything Else To Configure:
F1=Hlp  F2=Msg  F3=Bufr  F4=Sav  F5=Snd  F6=Cfgr  F7=Clr  F8=Print  F9=Exit  COM
```



```
                   HUGMCP Configuration Menu:
                   --------------------------

                   A --)   Modify Baud Rate
                   B --)   Modify Parity Type
                   C --)   Modify Word Length
                   D --)   Modify Or Add Auto-Messages
                   E --)   Miscellaneous Functions
                   F --)   Change Screen Color Assignments
                   G --)   Display Current Configuration
                   H --)   Make Changes Permanant

Select A-G, (F1) For Help, Anything Else To Quit --> _

                   Baud Rate: 19200
                   Parity: NONE
                   Word Length: 8
                   Duplex: FULL
                   Response To Keyboard Disable: NO
                   Storage Buffer Data Parity Bit: SET TO ZERO
                   Send Modem Initialization Text: NO
                   Delete Character: NORMAL
                   Modem Port Set To: COM1
F1=Hlp  F2=Msg  F3=Bufr  F4=Sav  F5=Snd  F6=Cfgr  F7=Clr  F8=Print  F9=Exit  COM
```

Heath
Users'
Group

P.O. Box 217
Benton Harbor, MI 49022-0217

$2.50
P/N 885-2129