



BUFFALO BILLS



NEW YORK GIANTS



NEW YORK JETS



PHILADELPHIA EAGLES



CINCINNATI BENGALS



NEW ORLEANS SAINTS



LOS ANGELES RAIDERS



LOS ANGELES RAMS



PITTSBURGH STEELERS



CHICAGO BEARS



HOUSTON OILERS



DENVER BRONCOS



GREEN BAY PACKERS



SAN DIEGO CHARGERS



INDIANAPOLIS COLTS



TAMPA BAY BUCCANEERS

ZENITH
data systems



DALLAS COWBOYS

Groupe Bull



MIAMI DOLPHINS



The Official Personal Computer
of the National Football League.



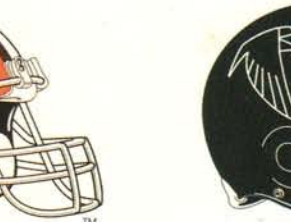
NEW ENGLAND PATRIOTS



MINNESOTA VIKINGS



CLEVELAND BROWNS



ATLANTA FALCONS



PHOENIX CARDINALS



DETROIT LIONS



SEATTLE SEAHAWKS



WASHINGTON REDSKINS



KANSAS CITY CHIEFS



SAN FRANCISCO 49ers



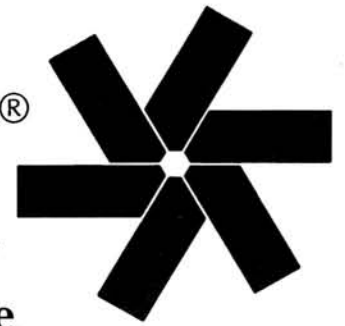
Share the Knowledge!

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 product, be it software or hardware, for your system? By submitting this information in the form of a major article, you can share with others the knowledge of a particular subject. REMark magazine is currently looking for authors (novice or professional) to write articles. Even if you have never written before, give it a try! As a REMark author, you will receive up to \$400 for each article accepted and published. (For more information on current policies, call Lori Lerch at 616-982-3794.) So, Let's get to it and ...

Share the Knowledge!

REMark[®]

November 1991



The Official Zenith Data Systems Users Magazine

Harvard GeoGraphics

George P. Elwood 7

Getting the Most From Your Computer Part 3 (Correction)

John P. Lewis 12

Getting the Most From Your Computer Part 6

John P. Lewis 13

I Can Do It! — Part 1

David Warnick 19

Using a Full Size Keyboard on a Laptop Computer

Pat Swayne 21

Harvard Graphics 3.0

George P. Elwood 25

Adding Expanded Memory to Your PC

Dan Jerome 31

First Aid for Writing

Mark Haverstock 33

Polishing Your Writing Style With RightWriter 4.0

Tom Bing 37

Utility

Sanford Shapiro 41

Zenith Data Systems and the NFL

..... 44

Bonehead Plays in C

Tom Bing 45

Advertising

	Page No.
<i>FBE Research Co., Inc.</i>	36
<i>QuikData, Inc.</i>	18
<i>WS Electronics</i>	6

Resources

Software Price List	2
Renewal Form	4
Buggin' ZUG	5

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)

ZUC
Software Orders
(616) 982-3463

Contributing Editor
William M. Adney

Printer
Imperial Printing
St. Joseph, MI

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

Contributing Editor
Robert C. Brenner

To Locate your Nearest:

Dealer 1-800-523-9393
Service Center 1-800-777-4630

	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. Check ZUC Product List for availability of bound volumes of past issues. Requests for magazines mailed to foreign countries should specify mailing method and include appropriate, additional cost.

Send Payment to: Zenith Users' Group
P.O. Box 217
Benton Harbor, MI 49023-0217
(616) 982-3463

Although it is a policy to check material placed in REMark for accuracy, ZUC 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 Zenith Data Systems Computer Centers.

ZUC is provided as a service to its members for the purpose of fostering the exchange of ideas to enhance their usage of Zenith Data Systems equipment. As such, little or no evaluation of the programs or products advertised in REMark, the Software Catalog, or other ZUC publications is performed by Zenith Data Systems, in general, and Zenith Users' Group, in particular. The prospective user is hereby put on notice that the programs may contain faults, the consequence of which Zenith Data Systems, in general, and ZUC, 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.

REMark is a registered trademark of the Zenith Users' Group, St. Joseph, Michigan.

Copyright (c) 1991, Zenith Users' Group

Software

PRODUCT NAME	PART NUMBER	OPERATING SYSTEM	DESCRIPTION	PRICE
H8 - H/Z-89/90				
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
AUTOFIL (Z80 ONLY)	885-1110	HDOS	DBMS	30.00
BHBASIC SUPPORT PKG	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	CHKBK 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 PKG	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 SECT SUPPORT PKG	885-1121	HDOS	UTILITY	30.00
HDOS PROG. HELPER	885-8017	HDOS	UTILITY	16.00
HOME FINANCE	885-1070	HDOS	BUSINESS	18.00
HUG DISK DUP UTILITY	885-1217-[37]	CPM	UTILITY	20.00
HUG SOFTWARE CATALOG	885-4500	VARIOUS	PROD TO 1982	9.75
HUGMAN & MOVIE ANIM	885-1124	HDOS	ENTERTAINMENT	20.00
INFO SYS AND TEL. & MAIL SYS	885-1108-[37]	HDOS	DBMS	30.00
LOGBOOK	885-1107-[37]	HDOS	AMATEUR RADIO	30.00
MAGBASE	885-1249-[37]	CPM	MAGAZINE DB	25.00
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 DEC '80	20.00
RUNOFF	885-1025	HDOS	TEXT PROCR	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 PKG	885-1127-[37]	HDOS	UTILITY	20.00
STUDENT'S STATISTICS PKG	885-8021	HDOS	EDUCATION	20.00
SUBMIT (Z80 ONLY)	885-8006	HDOS	UTILITY	20.00
TERM & HTOC	885-1207-[37]	CPM	COMMUN & UTIL	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)				
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	CHKBK 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 CAT UPDT #1	885-4501	VARIOUS	PROD 1983 TO 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
X-REFERENCE UTIL FOR MBASIC	885-1231-[37]	CPM	UTILITY	20.00
ZTERM	885-3003-[37]	CPM	COMMUNICATIONS	20.00

Price List

This Software Price List contains all products available for sale. For a detailed abstract of these products, refer to the Software Catalog, Software Catalog Update #1, or previous issues of REMark.

PRODUCT NAME	PART NUMBER	OPERATING SYSTEM	DESCRIPTION	PRICE
H/Z-100 (Not PC) Only				
CARDCAT	885-3021-37	MSDOS	BUSINESS	20.00
CHEAPCALC	885-3006-37	MSDOS	UTILITY	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
DUNGN & DRAGONS (ZBASIC)	885-3009-37	MSDOS	GAME	20.00
ETCHDUMP	885-3005-37	MSDOS	UTILITY	20.00
EZPLOT II	885-3049-37	MSDOS	PRINTER PLOT UTIL	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	UTILITY	20.00
HELPSCREEN	885-3039-37	MSDOS	UTILITY	20.00
HUG BKGRD 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
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	ATRONOMY UTILITY	20.00
SMALL-C COMPILER	885-3026-37	MSDOS	LANGUAGE	30.00
SPELL5	885-3035-37	MSDOS	SPELLING CHECKER	20.00
SPREADSHEET CONTEST PKG	885-3018-37	MSDOS	VARIOUS SPRDST	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				
ADVENTURE	885-3016	MSDOS	GAME	10.00
BACKGRD 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
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 CAT UPD #1	885-4501	MSDOS	PROD 1983 - 1985	9.75
HUGMCP	885-3033	MSDOS	COMMUNICATION	40.00
ICT 8080 - 8088 TRANSLATOR	885-3024	MSDOS	UTILITY	20.00
MAGBASE	885-3050	VARIOUS	MAG DATABASE	25.00
MATT	885-8045	MSDOS	MATRIX UTILITY	20.00
MISCELLANEOUS UTILITIES	885-3025	MSDOS	UTILITIES	20.00
PS' PC & Z100 UTILITIES	885-3052	MSDOS	UTILITIES	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
REMARK VOL 11 ISSUES 120-131	885-4011	N/A	1990	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				
CARDCAT	885-6006	MSDOS	CAT SYSTEM	20.00
CHEAPCALC	885-6004	MSDOS	SPREADSHEET	20.00
CLAVIER	885-6016	MSDOS	ENTERTAINMENT	20.00
CP/EMULATOR II & ZEMULATOR	885-6002	MSDOS	CPM & Z100 EMUL	20.00
DUNGEONS & DRAGONS	885-6007	MSDOS	GAME	20.00
EZPLOT II	885-6013	MSDOS	PRINTER PLOT UTIL	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' 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 UTIL	20.00
TCSPELL	885-8044	MSDOS	SPELLING CHECKER	20.00
ULTRA RTTY	885-6012	MSDOS	AMATEUR RADIO	20.00
YAUD (YET ANOTHER UTIL DSK)	885-6015	MSDOS	UTILITIES	20.00

Attention!!

Zenith Data Systems Owners

When ordering ZUG software, be sure to specify what disk format you would like us to put it on. If you have an H-8, H/Z-89, or H/Z-90, you have the choice of using hard- or soft-sectored disks depending on your drive type. Order soft-sectored by adding a -37 to the end of the part number (i.e., 885-8009-37). Leaving off the -37 specifies a hard-sectored disk (i.e., 885-8009). If you own an H/Z-100 (not PC) series computer, you will always use the -37 at the end of the part number. For PC users, you have the choice of 5-1/4" (-37), 3.5" (-80), or 2" (-90) disks. Just add this number to the end of the ZUG part number (i.e., 885-3009-37, 885-3007-80, 885-3007-90).

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, MasterCard, and American Express phone orders, telephone the Zenith Users' Group directly at (616) 982-3463. Have the part number(s), description(s), and quantity ready for quick processing. By mail, send your order to: Zenith Users' Group, P.O. Box 217, Benton Harbor, MI 49023-0217. VISA, MasterCard and American Express require minimum \$10.00 order. No C.O.D.s accepted.

Questions regarding your subscription? Call Lisa Cobb at (616) 982-3463.

REMark Magazine Subscription & ZLink/COM1 Bulletin Board Information

Your subscription entitles you to receive REMark, our monthly magazine containing articles specific to Zenith Data Systems computer and generally to other PC Compatible computers. All articles in REMark are submitted by readers like you. We welcome YOUR articles, and will pay you for any we accept!

A REMark subscription also allows you full access to the ZLink-COM1 bulletin board system (COM1, for short). There are many, many megabytes of free and shareware software available for downloading to registered COM1 users. Full access also lets you order products from the "Bargain Centre" section of COM1. The money you can save in the Keyboard Shopping Club will pay for decades of REMark subscriptions.

Last, but definitely not least, your subscription puts you in touch with thousands of other Zenith Data System computer users, from whom invaluable information can be exchanged.

REMark subscriptions, currently \$22.95, can be obtained in one of three ways. First, by ordering one on the COM1 bulletin board (see the Keyboard Shopping Club section); second, by phone with VISA, MasterCard, or American Express; and third, through the US Mail using a credit card, money order or check made payable to: Zenith Data Systems. Our address is:

Zenith Data Systems Users' Group
P.O. Box 217
Benton Harbor, MI 49023-0217
(616) 982-3463

Once you receive your ID number, registration on the COM1 BBS is NOT automatic. It requires that you log on, enter your first name and last name EXACTLY as they appear on your REMark mailing label, and then enter your ID number as your password. The FIRST time you access the board, you must elect to start a NEW ACCOUNT and answer the various questions. Once you've done this, our automated scanner will compare the system's database against the subscription database. If you made no mistakes, you will be verified and given full access within 24 hours.

Once you've been authorized as a full member, several important things happen. First, you're given full downloading privileges of up to one megabyte per day. Secondly, you'll have full access to the message boards. And finally, you'll be able to take full advantage of the Bargain Centre product savings.



Detach this form, enclose your check, money order or credit card information (no cash please).

REMark Subscription / Renewal Form

New Member: Yes No Credit Card # _____

ID Number: _____ Exp. Date _____

Address Change?

		Renew	New
Name: _____	U.S. Bulk Mail	<input type="checkbox"/> 19.95	<input type="checkbox"/> 22.95
Address: _____	U.S. First Class	<input type="checkbox"/> 32.95	<input type="checkbox"/> 37.95
City, State, Zip: _____	APO/FPO Surface Overseas	<input type="checkbox"/> 32.95	<input type="checkbox"/> 37.95
Daytime Phone #: () _____	Air Printed Overseas	<input type="checkbox"/> 52.95	<input type="checkbox"/> 57.95

Buggin' ZUG

Hard Drive For Your Portable

Dear ZUG:

Several past articles have addressed the inability of some Zenith Data Systems portables to accommodate an external drive. It has been suggested that this can be accomplished by use of the ZA-3034-EB expansion box with an appropriate card ("The Complete Travel-Office System, RE-Mark December 1989, p. 12) "Adding External Floppy Drives," RE-Mark January 1990, p. 7). This solution is not particularly cheap (\$400+ for the expansion box, \$110 for a CompatiCard, and the cost of the disk drive) nor compact. Fortunately, there is now a much cheaper and more compact solution.

Micro Solutions, 132 W. Lincoln Highway, DeKalb, IL 60115 (815-756-3411) makes an add-on external disk drive called the "Backpack" which comes in several configurations: 5-1/4" 360K (DSD), 5-1/4" 1.2 MB (DSDH), 3-1/2" 1.44 MB (DSDH), and 3-1/2" 2.8 MB (DSED). The list price for most versions is \$349, while the list price for the 2.8 MB disk is \$425. I ordered the 5-1/4" 1.2 MB version by telephone from Micro Warehouse and paid only \$266.

The Backpack 5-1/4" 1.2 MB disk drive comes in a metal case 2-1/4" x 6" x 10-1/2" deep. A cable connects the Backpack to the parallel printer port at the rear of the computer. Another parallel printer port is located at the rear of the Backpack disk drive case, allowing "daisy chaining" of additional disk drives or a printer. A small AC transformer powers the Backpack, and an off/on switch is located at the back of the case.

Software is provided which allows the computer to recognize the additional disk drive. It takes only a matter of minutes to do the necessary installation, following clear instructions. It is compatible with DOS 2.0 and higher. The same Backpack external disk drive can be connected to any number of computers by different manufacturers, so you are not locked into an investment which will be obsolete if you change com-

puters. I have attached my Backpack to my TurbosPort 386, giving me the best of all possible worlds: both the built-in 3-1/2" 1.44 MB disk drive (drive A) and the external 5-1/4" 1.2 MB (drive D). Since my office computer has only the 5-1/4" drive, the addition of the 5-1/4" Backpack drive to the TurbosPort has eliminated the tedious necessity of carrying a 5-1/4" disk to a computer with a 3-1/2" disk drive and copying the data to a disk I can use with the TurbosPort. Due to its compact size, the Backpack can be a travelling companion to the TurbosPort itself.

Sincerely,
Charles R. B. Kirk
10 Bolinas Avenue
San Anselmo, CA 94960

Problem With AutoCAD Rel. 11

Dear ZUG:

I have encountered a problem with AutoCAD release 11 on Zenith Data Systems 386 SX computers at school. We have some 16 MHz computers with DOS version 3.3+ and some 20 MHz computers with DOS 4.01. Release 10 seemed to work OK with both; Release 11 has a problem with the 20 MHz with 4.01.

The problem: When returning to the graphic screen from text, there is an uneraseable line on the graphic screen?

Question: Has Zenith Data Systems run release 11 of AutoCAD on the 20 MHz with 4.01? If so, have they had the same problem?

Would appreciate an answer to the question as soon as possible.

Gil Hoellerich
2617 Country Way
Fayetteville, AR 72703

Faint or Missing Characters

Dear ZUG:

I have a Zenith Data Systems 286 SupersPort laptop computer, and I really enjoy it. Unfortunately, I am limited in some of the programs that I run on it

because of the liquid crystal display. Some programs do not show up well (or at all) on it. I would like to be able to use my Zenith Data Systems ZVM-123 monochrome monitor that I used with my old Zenith Data System 161 "luggable". The monitor has only a conventional TV cable type input port.

How can I make use of that monitor? If I can use that terminal, will it "solve" my problem?

How can I change the display on the 286 to bring out the faint or "missing" characters?

Thanks for your help in this problem.

Sincerely yours,
William Carter
1441 MacArthur Boulevard
Munster, IN 46321

Care and Feeding of MinisPorts

Dear ZUG:

I just finished reading Mary Ellen Schutz' article in the July 1991 issue, "The Care and Feeding of MinisPorts." Thanks for publishing it: I found it very helpful.

I have a SupersPort SX, and have wondered about buying a second battery pack to use as a backup. Thus, if I want to spend \$149 for the luxury, I will be able to save everything, power down, slap on the new battery pack, and quickly get back to where I was. Then I can recharge the first battery, and have it ready when the second one blinks at me.

Or almost. There is one problem. From the time the battery blinks red until it shuts down is about half an hour. I could easily build a cord to plug into the tired battery, with a built-in load to run the battery down safely before beginning to recharge it. But I don't know what resistance would be safe.

Can anyone out there offer suggestions, or have any experience with such an arrangement?

Larry Monk
16690 S. Pam Drive
Oregon City, OR 97045

Die Hard Z-100 User

Dear ZUG:

I am a die hard owner of a Z-100 (ZFG-121-32) All-in-One desktop Zenith Data Systems computer manufactured in 1984 or 5.

Recently, I have run into a problem. I used to go to Zenith Data Systems store in Jericho, LI, NY and a genius mechanic, Bert, would help me. As you know, they are

no longer there and NJ Zenith Data Systems dealers never heard of my model. Bert upgraded my RAM to 768 and installed RGB chips and I wish the hell I knew where to find him or anyone else on Long Island, who could help me.

Anyway, this is my problem. The computer works like an old war horse, but:

Dots in irregular formations of half the brightness of the print on screen constantly appear. My screen, you might say, has a case of the measles. They appear in vertical and horizontal lines, some 3 dots down and others many dots across. Of course, it causes some lettering to be unclear.

If I run the cursor over them, they disappear, but rear their ugly heads again.

Any ideas as to the cause and remedy or must my horse be dying and I must live with it. I cannot afford to buy a new machine yet.

I have been told Zenith Data Systems no longer supports these old models. I do not know any User members on the Island,

who might be helpful unless you happen to know some leads for me to take in that direction.

I would appreciate any input you might have.

Thank you in advance.

Sincerely,
Arthur R. Siegel
1 Ash Place #5B
Great Neck, NY 11021

Prodigy Will Not Operate in Color

Dear ZUG:

A simple question gets me ambiguous answers. Perhaps you can give me a definitive response.

My equipment:
Zenith Data Systems 158-42
Monitor ZVM-135 RGB Composite
My problem:

Prodigy will not operate in color with this system. Because my eyes are going bad, the color would be most helpful. Prodigy bulletin board says "EGA card with the CGA monitor will produce Prodigy in color." Local Heath/Zenith dealer says it won't work unless I buy a new monitor to go with the EGA card.

Who's right?

Any suggestions as to EGA card to use with this monitor?

How do I install? Does present video-floppy card need to be adjusted or reset?

Thanks loads for any help you can give.

Incidentally, I have found REMark to be most helpful and this goes double for Bill Adney's always clear, always pertinent, always sensible columns.

Sincerely,
Charles K. Warriner
2760 Chipperfield Road
Lawrence, KS 66047



W S Electronics

(513) 376-4348

**** Since 1975 ****

(513) 427-0287

1106 State Route 380, Xenia, Ohio 45385

W S ELECTRONICS INTRODUCES KRIS TECHNOLOGIES

System 48C-3
80486-33MHz CPU
and
System 48C-2
80486-25MHz CPU

- Internal co-processor and 8KB cache.
- 128K secondary cache onboard.
- Socket for optional Weitek co-processor.
- 2MB 70ns RAM, expandable to 16MB onboard.
- Shadow RAM for System & Video BIOS.
- ROM-based setup utility.
- Six 5.25" drive bays available.
- Eight 16-bit expansion slots.
- Dimensions: 8.5"Wx18.5"Hx18"D.

\$3550⁰⁰
and
\$2899⁰⁰

System 38SX-2
386SX-20MHz CPU
and
System 38SX-1
386SX-16MHz CPU

- Socket for 80387SX co-processor.
- 1MB 80ns RAM, expandable to 8MB onboard.
- Page mode interleaved memory design.
- Shadow RAM for System & Video BIOS.
- Two 5.25" & two 3.5" drive bays.
- Six 16-bit & two 8-bit expansion slots.
- Dimensions: 7"Wx13"Hx16.5"D.

\$1225⁰⁰
and
\$1025⁰⁰

System 38M
80386-33MHz CPU
64K cache

- Socket for 80387 or Weitek co-processor.
- 2MB 70ns RAM, expandable to 16MB.
- Page mode interleaved memory design.
- Shadow RAM for System & Video BIOS.
- ROM-based setup utility.
- Two 5.25" & two 3.5" drive bays.
- Six 16-bit & two 8-bit expansion slots.
- Dimensions: 7"Wx13"Hx16.5"D.

\$1999⁰⁰

System 28
80286 60-bit CPU

- Socket for optional 80287 co-processor.
- Standard 1MB RAM onboard, configurable to 640K base & 384K extended memory.
- Three 5.25" & one 3.5" drive bays.
- Six 16-bit & two 8-bit expansion slots.
- User selectable 8/4/2 DMA clock speed.
- Dimensions: 17"Wx6.5"Hx16.5"D.

\$750⁰⁰

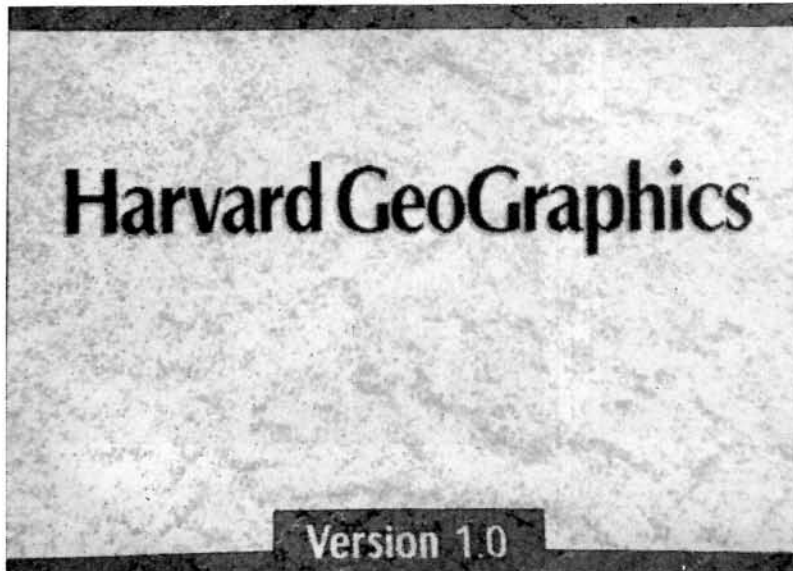
System 38
80386-35MHz CPU

- Socket for 80387 or 80287 co-processor.
- 2MB 80ns RAM, expandable to 16MB.
- Page mode interleaved memory design.
- Shadow RAM for System & Video BIOS.
- Four 5.25" & two 3.5" drive bays.
- Six 16-bit & two 8-bit expansion slots.
- Dimensions: 19"Wx6.5"Hx16.5"D.

\$1499⁰⁰

*ALL COMPUTERS HAVE A 2 YEAR PARTS, 18 MONTH LABOR WARRANTY!
*Monitors, Video Cards, Harddrives are optional.





George Elwood
1670 N. Laddie Court
Beavercreek, OH 45432

The Software Publishing Corporation has added a new product to its line of easy to use business graphics programs. The new product, Harvard GeoGraphics, provides a quick and easy means to create maps of both the U.S. and many foreign countries. Using user provided data, color coded maps can be easily generated. The product is a replacement for SPC's USMAP.

The large box contains an easy to understand manual and both 3.5 and 5.25 inch floppy media. This makes it easy to install on any computer without having to mail back disks or cards. The files on the disk are compressed and take up more than 3.5Meg on your hard disk. You will need at least 512K of RAM for the program, which supports video cards up through VGA. It is even possible to load the software on a network and run it from there, one user at a time.

GeoGraphics comes with a map library that include all 50 U.S states and over 63,000 major U.S. cities. SPC has also added 193 countries and 3000 major international cities to the package. As part of the U.S maps, two additional libraries are included and can be added to the State maps. The first is all 3140 counties and the second is a break out of the three digit postal ZIP codes.

The resulting files can be sent directly to a printer, plotter or film recorder or converted into several different file formats. The maps can be converted to Harvard Graphics charts or symbols, Computer Graphics Metafiles (CGM), Encapsulated Postscript (EPS), or PCX formats. All of these choices are available from a menu option.

The installation is very easy. Simply insert the first disk in a floppy drive and

make it the default drive. Type INSTALL and follow the on-screen instructions. Because of the compression of files, this can take a long time. The program does tell you the approximate time based on the installed CPU. You can install either the entire program or parts that you will need, if space is a problem. If you select the default installation procedure, a sub-directory named HGG will be created for the files. Within this sub-directory, two additional directories will be created, one for the MAPS and the second for the SYMBOLS.

After installation, simply type "HGG" to start the program. The program will display an opening screen similar to Harvard Graphics 2.3 and 3.0 and then the main menu will be displayed. To access any of the pull-down sub menus you can move the cursor with a mouse to select, move the cursor with the cursor keys, or press the desired function key. The F1 key is the Help key and can be used anyplace in the program if you run into problems.

SPC has added speed keys to GeoGraphics. This permits you to select a function without having to go to the pull-down menu. The speed keys use a combination of CTRL or ALT and one other key. The keys even make sense. For example, Ctrl-Q will quit the program, Ctrl-S will save the current map, Alt-M will move a selected object on the map.

Creating maps is very easy. You can use some predefined groups for a map or define the areas you wish to show. You can select from the group library either all 50 or just the 48 States. There are groups with the major cities in each state, or groups of regional states. On the foreign side, there are groups by continents, special areas like the Middle East, Arab League, the Euro-

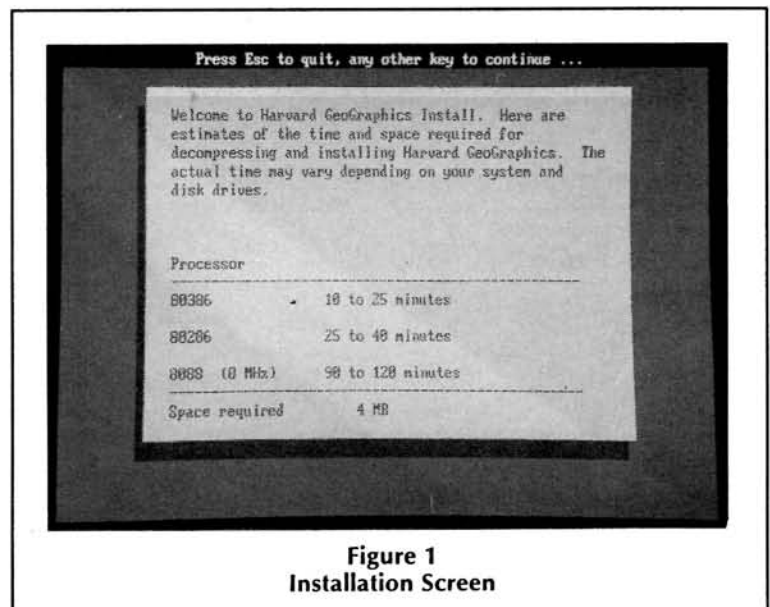


Figure 1
Installation Screen

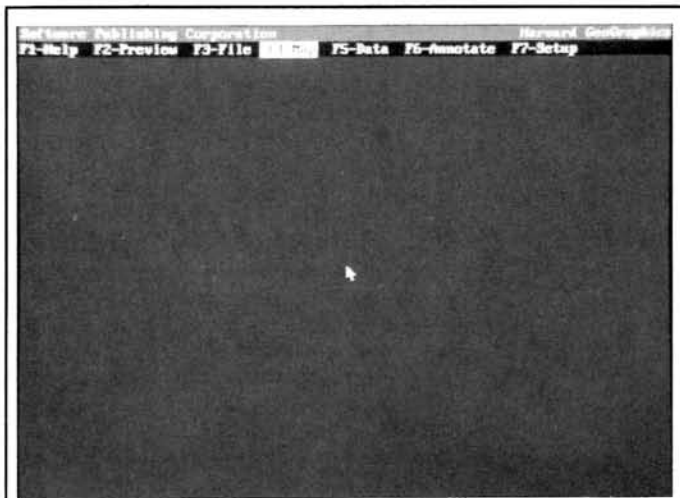


Figure 2
Main Menu

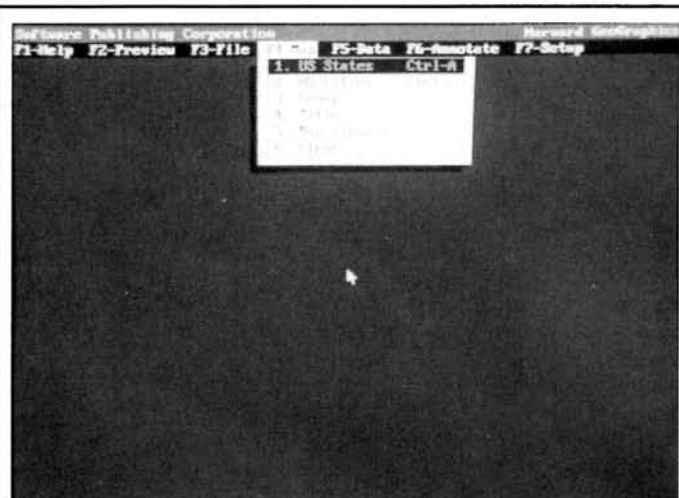


Figure 3
MAP Selection Menu

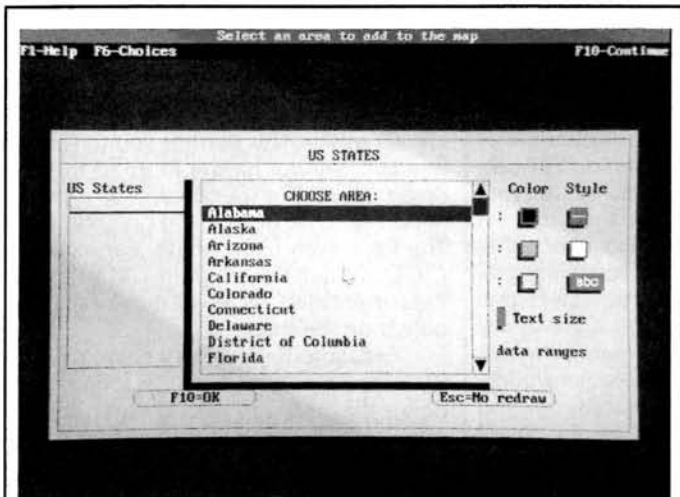


Figure 4
State Selection Window

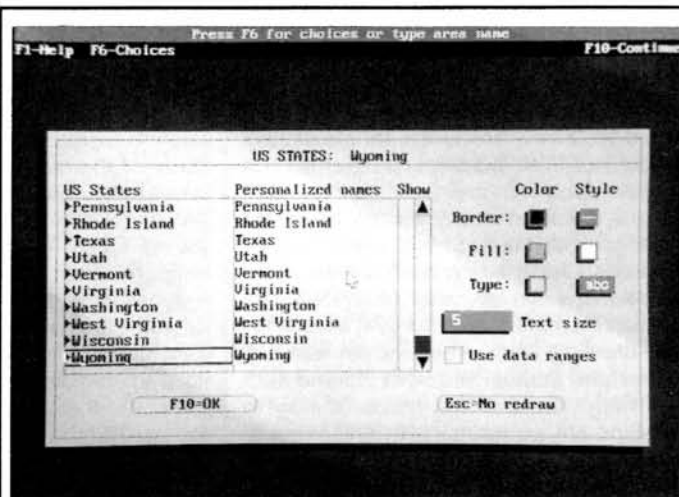


Figure 5
Completed Selection Window

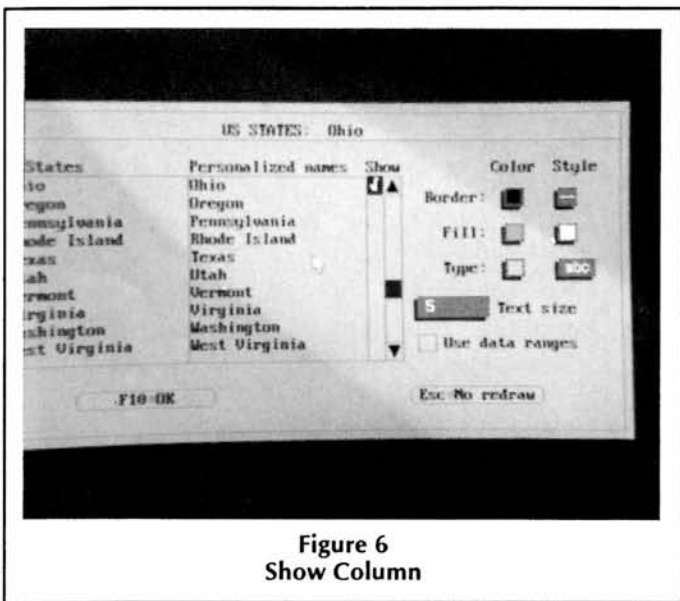


Figure 6
Show Column

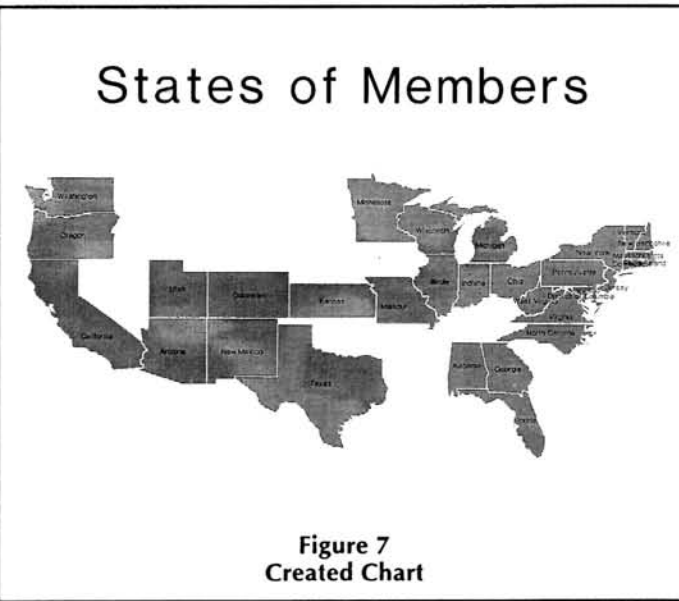


Figure 7
Created Chart

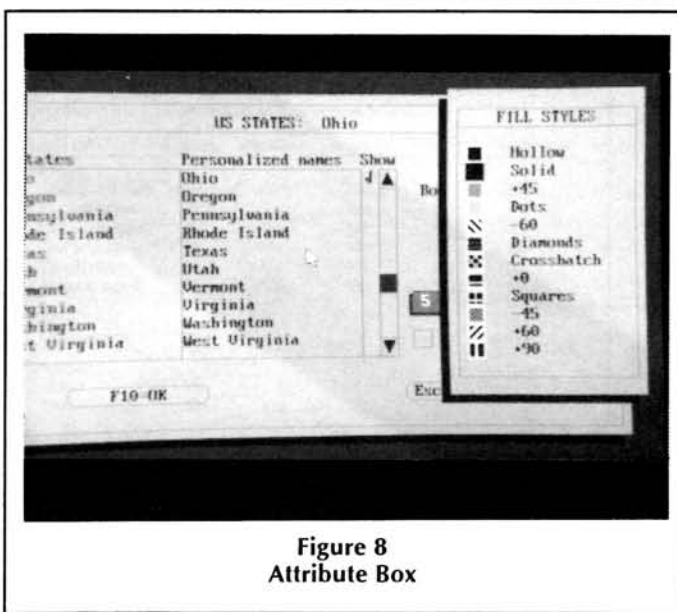


Figure 8
Attribute Box

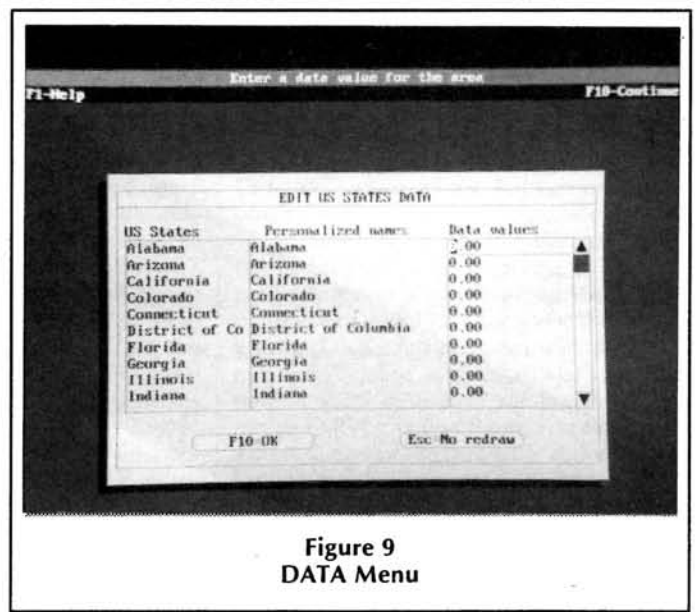


Figure 9
DATA Menu

pean Economic Community or NATO. These groups permit you to rapidly construct a map to meet your needs. Each group is made up of the entities from the selected libraries. If you need to make a map of the NATO countries, you would have to select the Countries map library first.

If you wish to make a map using your own selected areas, it is just as easy as selecting a group. As an example, to select a group of States you must first have the State Library set selected. In the maps menu you can select the state by highlighting and pressing the Enter key or by typing in it's name. As you type, GeoGraphics will start to search based on the letters keyed in.

ized name block. This block will reflect the state names shown on the left. You can change this name to whatever meets your needs by moving into the block and type over the name in the block.

To display the selected states press F10. GeoGraphics will then display all selected states in the correct position. The overall size will be determined by the number of states selected and their location. In the default mode, the state names will not be displayed inside the state. This can be turned on and the size changed by moving to the Show column and pressing the space bar. This will turn on the check mark and the state name will be displayed on the map.

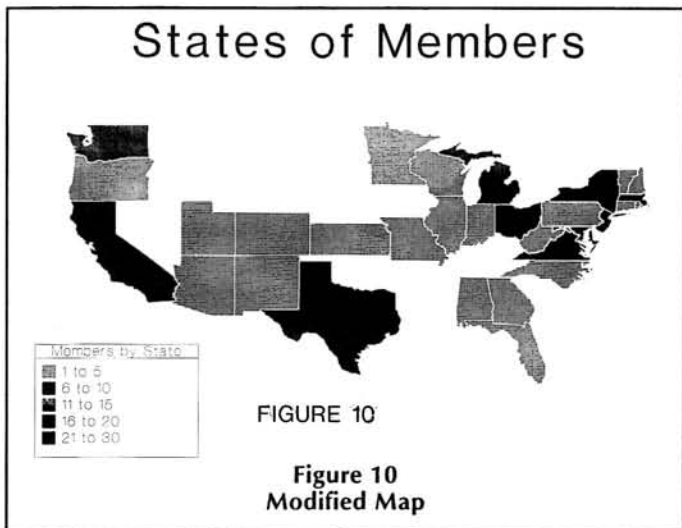


FIGURE 10

Figure 10
Modified Map

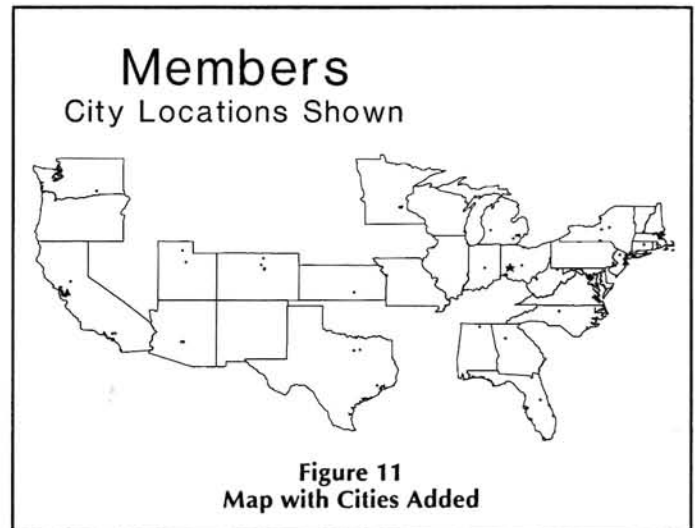


Figure 11
Map with Cities Added

Most states can be selected with one or two letters.

As an example of how to create this type of map, I will create a map of states showing where members of a test network live. With the list in hand, press F4 to display the pull-down map menu. If the state library is selected, it will be shown on this menu. Select the state option and the display will change to show the states. Move the selection bar into the area where the States are shown. Press Shift/Enter to select. You can then type the first few letters of the next desired state. The selection box will move to that state. When you press Shift/Enter, the name will move slightly indicating that it has been selected. When you are on the last state press <Enter> and all the selected states will be displayed in the left selected box. On the right side of the select box is the Personal-

If you do not like the colors, border, or text size, GeoGraphics permits you to modified these characteristics. Press F4 to redisplay the MAP menu and press the TAB key to move into the options area. You can change the border color and line style, the fill color and fill style, and text type and size. Like Harvard Graphics, you can move into the selection box and press the space bar to move through the options or press F6 to display all the options. You should set these options, at the start of the map. If you do not, you may change all state options by moving down the list of states and pressing Shift/Enter. This will insert a small diamond in front of the state name. Once all of the states are selected, you may then change the options as a group.

One of the features of GeoGraphics is its ability to use and

County Map Using Cities

Erie Lackawanna

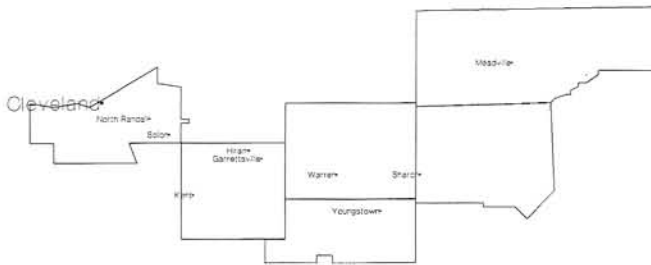


Figure 12
EL Cities with Counties

OHIO Three Digit ZIP Codes

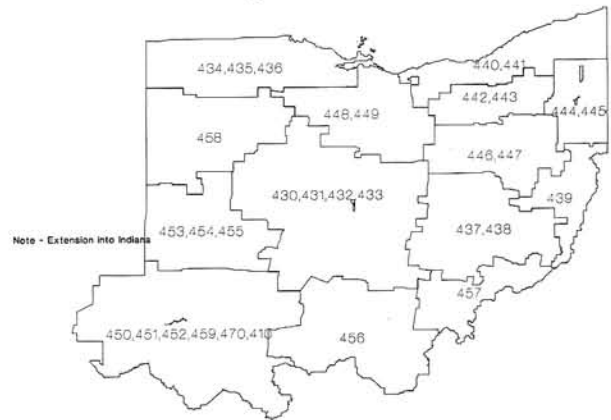


Figure 13
Ohio ZIP Codes

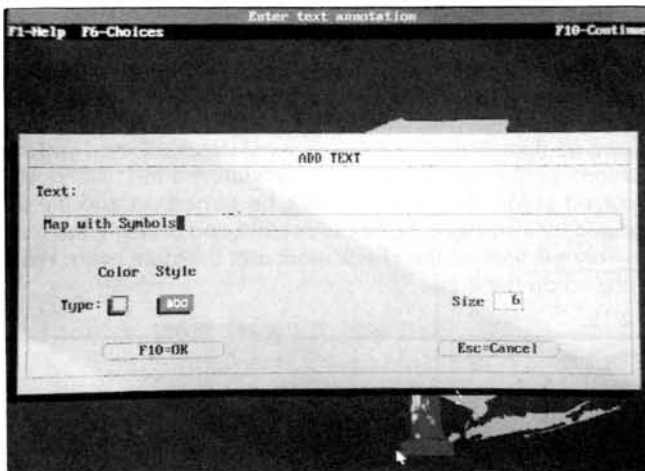


Figure 14
Title Option Menu

Map with Symbols

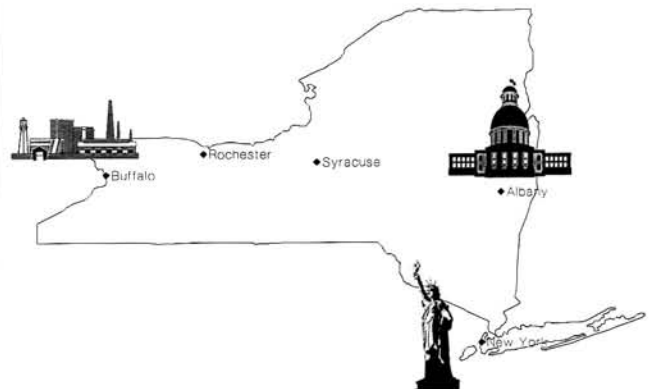


Figure 15
Map with Symbols

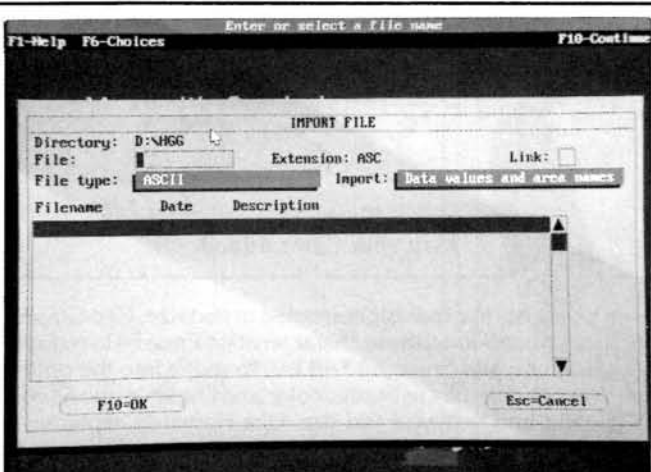


Figure 16
Import Data Menu

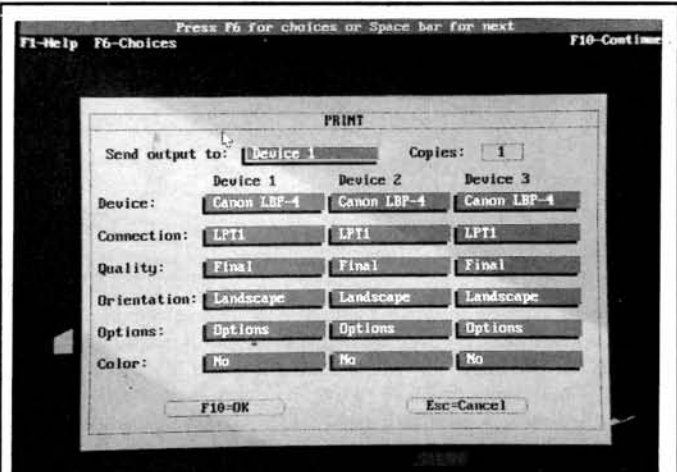


Figure 17
Output Menu

display data on the map. In Figure 10, I wanted to show the number of members in each state. I elected to use the data range function in the program. At the Main Menu, select Data (F5). On

this menu select the Ranges option. You can now set the maximum value for chart if you have manual calculation set. If you have automatic set, the program will calculate the maximum

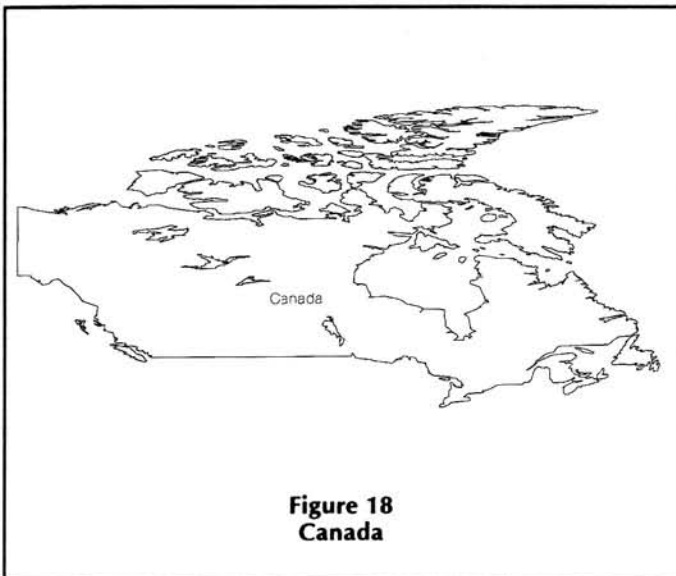


Figure 18
Canada

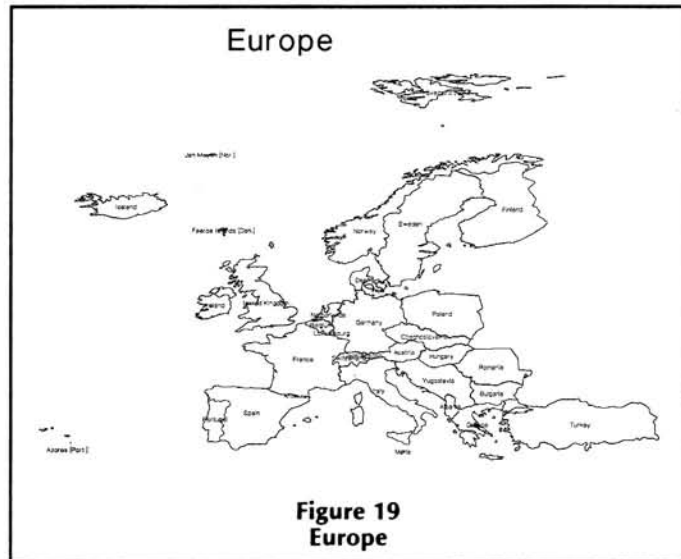


Figure 19
Europe

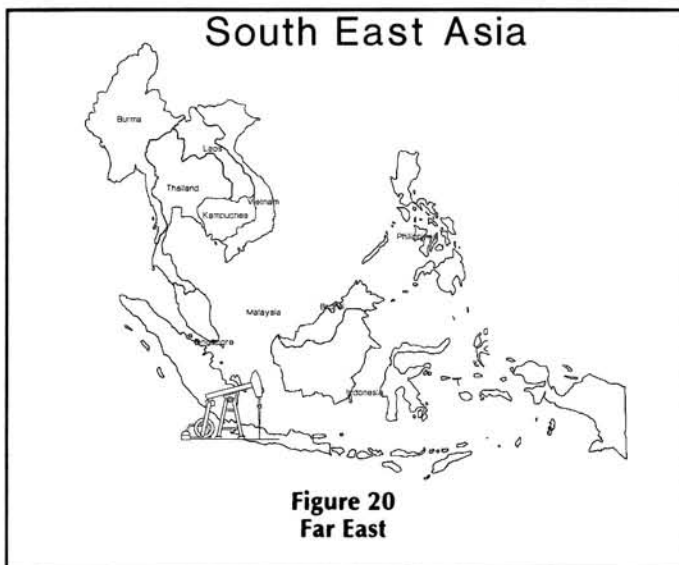


Figure 20
Far East

number to give an even 16 units. The program permits up to 16 ranges (the maximum number of colors). You may change the color or fill style in this menu if desired. When you press F10 to redraw the map, each state will now be displayed in colors or fill based on the ranges established above. Using other available options, you can display a legend on the map.

On another map, I wanted to show the cities where the members were located. Using the base line map, I selected CITIES from the Map Menu. The program has over 63,000 U.S. cities in its database so the cities I need would surely be included. To enter the cities I simply started to type the name. Before you select your first city, make sure you set the attributes you want for selected cities. You can set Icon style, font, size and location of each city. With each letter typed, GeoGraphics will move closer to the desired city. When you have a city that exists in several states, simply move the cursor over the desired city and press Shift/Enter. Remember, Shift/Enter permits you to enter many cities. When you have entered the last city press Enter and all of the selected cities will be saved. Press F10 to display the map with the cities. If you have many cities, you may wish to turn off the names. (Figure 11)

It is also possible to create a map by entering just the cities.. Harvard GeoGraphics will select the state required for each city. With over 63,000 cities, many small towns are included in the database. As an example, Alpha, OH is included. This is only a

post office with about 20 houses.

GeoGraphics also has a U.S. counties group. If selected, you can add the counties of any state or group of states. Selecting the Counties group from the MAP menu, you can select the counties or use the cities selection method used above. Many years ago I worked on the Erie Lackawanna Railroad out of Cleveland Ohio. With the counties option on, I named the cities and towns the line went through and created the following MAP. Note, that the map extended into PA because the last city selected was Meadville, PA.(Figure 12)

The program also has 679 three digit ZIP codes. Again, these can be selected from the MAP group option. See Figure 13.

Note that in the examples, each MAP has added text. Text can be added to each map by using the Annotate option from the top line menu. The color, location, size, and style can be set using the text options.

Harvard Graphics comes with a large set of symbols. Harvard GeoGraphics comes with a few symbols but you can use the symbols that come with HG to further enhance your map. You should move the symbols that come with GeoGraphics to the HG directory and set the path in the program. All of the functions available with HG symbols are available in GeoGraphics along with some that are available in Draw Partner, like rotation.

If you have a lot of data, GeoGraphics will import data from Lotus 1-2-3 and ASCII files. Rather than typing in the names of the state and then adding the numbers, you can import it directly. I generated the membership list out of an Enable database which I saved in an ASCII file. Using the import capability, the entire map creation was automated. You can even link the map to the external data so that when it changes, the map will be updated.

Now that you have created the map you would like to output it. Harvard GeoGraphics can output directly to many printers and plotters. Three different output devices can be selected and installed in the program. The printed output supports shades of gray so all selected items will not be black, as was the case with Harvard Graphics, before version 2.3.

You can save the map in several formats. There is a default map format that can be used only in GeoGraphics. You can also save the map as a symbol or chart for use in HG. Note, it is possible to create a map that cannot be used in HG version 2.3 and below. The program will warn you if this condition exists. You can also save the map in CGM or PCX formats which are used in some desktop publishing programs. The last two formats are Encapsulated Postscript or HP Graphics Language (HPGL). I have created a map and saved it in CGM format and then used it on a SUN workstation.

So far I have talked about GeoGraphics in the U.S. The

program does support 193 countries. These countries are selected from a library like the U.S. States, counties, and ZIP codes. The same functions are available with the countries as with the U.S. States.

This program is a welcome addition for those of you who need to develop map-based business graphics. The user interface is not windows, but the easy to learn and use Harvard Graphics

format with the new pull down menus.

Software Publishing Corp.
1901 Landing Drive
P.O. Box 7210
Mountain View, CA 94039-7210

\$395.00



Getting the Most From Your Computer

Part 3

John P. Lewis
6 Sexton Cove Road
Key Largo, FL 33037

Inadvertently, we left out part of Listing 2 from Getting the Most From Your Computer — Part 3 that appeared in the August 1991

issue page 43 of REMark.

We're sorry for any inconvenience this may have caused.

```
Unit DcHxunit;
Interface
Procedure Execute_DcHx;
Implementation
uses Crt, Dos;
Type
String20 = String[20];
String1 = string[1];
const
HexString : String20 = '0123456789ABCDEF';
Var
OutString, InString, Dum : String20;
St : String1;
Value, Dec : Longint;
Dumy : Integer;
Ch : Char;
Procedure Binary(Val : Longint);
Var len, Byte_Count, Format : Integer;
Begin
Len:=0;Format:=8;
Outstring:='';If val < 256 then Byte_Count:=1[no "; when
using else ]
Else if (val > 255) and (val < 65536) then
Begin
Byte_Count:=2;
Format:=17;
end;
While Val > 0 do
Begin
str(val mod 2,st);
{ get val modulo 2, convert to string }
outstring:=st+outstring;{ append outstring to st }
Val:=Val Div 2;{ integer division by 2 }
len:=len+1;{ add to length of outstring }
if len = 8 then{ check for end of byte }
outstring:=' '+outstring;{ tack on a space }
end;
If Length(Outstring) < Format then { check for length
}
repeat
begin
insert('0',outstring,1);
{ Pad the output string with zeros }
len:=length(outstring);
end;
until len = Format;{ are we done ? }
end;
Procedure Hex_ToDec(HexStr : String20);
Var Len, I : Integer;Count : Longint;
Found : Char;
begin
Len:=Length(HexStr);I:=0;Dec:=0;Count:=1;
while Len > 0 do begin
```

```
while ch <> Found do
begin
Found:=HexString[I+1];{ I = index into Hexstring }
ch:=HexStr[Len];I:=I+1;
end;
Dec:=-Dec+(I-1)*(count);count:=count*16;
Len:=Len-1;Found:=' ';I:=0;
end;
end;
Procedure Dec_ToHex(Index : Longint);
begin
ch:=HexString[Index+1];
end;
Procedure Construct(Val : Longint);
Var Index, Len, I : Longint;
begin
outstring:='';
while Val > 15 do begin
Index:=Val Mod 16;
Dec_ToHex(Index);Outstring:=ch+Outstring;
Val:=Val Div 16;
end;
Dec_ToHex(Val);Outstring:=Ch+Outstring;
Dum:=Outstring;
end;
Procedure Execute_DcHx;
begin
clrscr;gotoxy(4,4);
write('Decimal to Hex, Hex to Decimal conversion
calculator ');
gotoxy(12,6);write('1. Convert Decimal to Hex ');
gotoxy(12,8);write('2. Convert Hex to Decimal ');
gotoxy(8,10);write('Enter your
choice ');readln(Value);dumy:=round(value);
case dumy of
1:begin clrscr;gotoxy(4,4);
write('Enter a decimal value for
conversion ');readln(Value);
Construct(Value);Binary(Value);
gotoxy(6,6);Write('Hex Value = ',Dum);gotoxy(6,8);
write('Binary = ',Outstring,', Press Enter to
continue ');readln;end;
2:begin clrscr;gotoxy(4,4);write('Enter the Hex value
for conversion ');
readln(InString);Hex_ToDec(InString);Binary(Dec);gotoxy(6,6);
write('Decimal value = ',Dec);gotoxy(6,8);
write('Binary = ',Outstring,', Press Enter to
continue ');readln;end;
end; [end case of]
end;
end.
```



Getting the Most From Your Computer

Part 6

John P. Lewis
6 Sexton Cove Road
Key Largo, FL 33037



Our programming project for this month is quite interesting. I think you will agree with my assessment when we get into the description. It is not terribly complicated but does use some Turbo features that we haven't investigated yet. It also provides a vehicle for another lesson in dynamic memory allocation. A rather complex subject and one that causes quite a bit of confusion. First we'll look at an area of data management that needs some tools for proper handling.

Most of the readers of this series no doubt utilize a word processor in their day to day computer usage. Any word processor worth its title will include "word-wrap" in its bag of tricks. What's so strange about that? Nothing really, but the text produced with this feature enabled, uses a "soft" carriage return when it automatically wraps a line at the specified right margin. This idiosyncrasy can cause some rather strange problems when trying to view the file produced, using many of the tools available. Even Borland's "Readme.com" fails to display the text as it should be. Worse, if you are building a data base for use by others, how do you present it?

I have been quite busy lately, building a rather large data base. The data base program uses a combination of "Records" which give access to a much larger text file. These files include many instances of "soft" carriage returns. I intended to use a version of this month's listing to provide the user with a viewing utility for the included text files. It has been part of my source code inventory for some time, but I suddenly found that it needed some improvement if it was going to look professional in its text presentation. As a consequence, you get the benefit of my latest source code modification.

This month's listing solves that prob-

lem while providing us with a look at some other facets of programming logic. One area that we will explore is the use of compiler directives which will enable us to write one version of source code and then use it in several different ways by changing one or two compiler directives. This is a very powerful feature, one that I haven't used myself until recently. I hadn't realized how much trouble could be saved by employing this option (I quite often have two or more different versions of many of my programs).

I learned quite a bit about file I/O while putting this month's listing together. Actually, I thought I knew quite a bit about how Turbo Pascal works with files but I found that there was (and still is) much to be learned. My education is a continuing process.

In this instance, file I/O is tied closely to the way we are going to deal with the "soft" carriage return. When reading a file a line (`readln(fileVar, Var)`) at a time, Turbo Pascal senses the end of a line by the presence of a carriage return, line feed pair. If none is present then Turbo will just continue inputting characters and assign them to the same line. Talk about confusion! This causes some rather disgusting things to happen to the screen display of our file. The first solution for this problem that I tried, is the one I eventually used. Even though it increases file I/O duration, the subsequent file memory image is right on target.

A "soft" carriage return is created by turning on the msb (most significant bit) of the byte containing our carriage return. Since our byte originally has an ascii value of 13, turning on the "msb" results in a value of 141 (decimal). The msb has a value of 128, this added to 13 gives us 141. What to do? One solution to this problem is to

turn the msb back off (change from a 1 to 0). The most logical way to accomplish this is to "AND" the byte containing the ascii 141 with a value that will turn off the msb and leave the rest intact. Decimal 127 is equal to 01111111 in binary and decimal 141 is equal to 10001101 in binary. Use your dechex conversion utility (from part 3 of this series) to verify this if you want, my binary arithmetic may be a bit rusty. As you can see, "anding" the two bytes together will change 10001101 to 00001101 or 13 decimal (anding zero with one will leave zero).

The methodology for achieving this is shown in the listing (see procedure "assn" within the `{ $Ifdef Soft }` compiler directives). You may think of a better method to filter the soft carriage returns but I tried several and the included code works very well, if a bit slow. I have configured this program so that it will use a line at a time file I/O or a byte at a time, by incorporating `{ $Ifdef Soft }` and `{ $Ifndef Soft }` statements around the two "assn" (assign) procedures. That way, you will be able to use either configuration quite readily and test both versions to see how they work with your files.

The `{ $Ifdef }` (Note: must be accompanied by a matching `{ $Endif }`) statements are merely compiler directives. They add no code to the "exe" file but provide a great deal of versatility to this program. By inserting a space in front of the "\$" within the `{ $Define Soft }` statement, the "define" is nullified, thus the actual configuration can be changed by inserting (or deleting) a single space. As you no doubt surmised, the `{ $Ifndef }` statement corresponds to "if not defined".

I added a few more `{ $Ifdef }` statements to the code that allow changing the program from a "stand alone" program to

a "unit", or conversely, from a unit to a "stand alone" program. All that is needed is the insertion (or deletion) of a single space in the { \$Define Standalone } statement at the very top of the code (first line). That should also provide a little illustration of the power of Turbo Pascal. Be sure that the first line of the listing looks like this: { \$Define Standalone } (defeating the define statement) for code generation to be used with our REMark project program (Project_1.pas). I hope you will want to compile the program both ways, incorporating the unit version (Showunit.tpu) in our project and compiling it again as a stand-alone version (probably RENaming SHOWUNIT.EXE to SHOW.EXE). As alluded to previously, this program started life as an included module in a much larger (130 K exe version) database program for use in the Photo industry. The original version did not require soft carriage return handling due to the nature of its data.

You will notice that I added a bit of code to the "asn" procedure (within the {\$Ifdef Soft} compiler directives) that might look a bit mysterious. Near the bottom of the procedure is a routine that prints "Please wait..." a character at a time with a computed interval between characters. This code is purely a panacea for the user, it gives him/her something to do besides waiting for the computer to read and convert the file being processed. It is, in my opinion, quite effective in accomplishing its objective.

Since the object of this program is to exert absolute control over the text file being processed, we need a method of traversing the file, a line at a time in either the forward direction or backwards. Furthermore, it should update the screen almost instantaneously. Additionally, we need the ability to update the file image a page (screen) at a time-instantaneously. A rather tall order if you think about it but you will soon have such a tool in your software arsenal. This program accomplishes that objective quite well, in fact its speed (after reading the file into memory) is beyond reproach.

We just touched on the reason why this program is so fast-the entire file resides in memory, on the "heap". Each line is accessed through a pointer to a Turbo Pascal "record" (similar to a structure in "C"). Each "record" has the address of its partners, the record preceding it and the record succeeding it.

Part five of this series dealt with records and dynamic memory allocation, also part of the program's charisma was the utilization of a linked list, but only in one direction -forward. Properly called a singly linked list. Each individual record held a pointer to the next element (if any). This month's listing employs some of the same methodology but adds a link to the previous record,

allowing a retreat (no burned bridges syndrome) to the preceding line.

Let's take a look at the listing to clarify

how this is accomplished. Look just below the "Type" definitions to the statement: Ptr = ^line; line = record.

Listing 1

```
{ $Define StandAlone }
{ $Ifdef Standalone }
Program Show;
{ $Endif }
{ $Ifndef Standalone }
Unit showunit;
Interface
Procedure Execute_Show;
Implementation
{ $Endif }
uses crt, dos;
Label escape, oneline;

Type
String30 = String[30];
String80 = String[80];
Ptr      = ^Line;
      Line = Record
      Next : Ptr; { provide pointers to next and previous records }
      Prev : Ptr;
      Words : String80; { Holds text for each line }
End;

Const
LnPd : Integer = 10;
Mask : Integer = 127; { And bit 7 with a zero to remove }
Message : array[1..30] of char = 'Please wait.....';
EndMess : string = '***->> End '+chr(24)+' '+chr(24)+' of File <<-***';
LastChar : Integer = 30;

Var
Filehandle : Text; { create a file of type text }
BytFile : File of Byte; { create a file of type byte }
Filenam : String30;
Buf : String80;
Start, P, RunP, T_ptr, B_ptr : Ptr;
Col, Row, Len, I, J, Divsor, Period, numchar, LastRow : Integer;
Bt : Byte;
Ch : Char;
EndF, ok, one : Boolean;
Sz : word; { Change to longint for BIG files }
{ $Define Soft } { Insert a space in front of '$' to change config }

Procedure Clear_Down(Lcol, UpRow, Rcol, Brow : Integer);
begin
window(Lcol, UpRow, Rcol, Brow); { make window with coordinates }
clrscr;window(1,1,80,25); { clear it and then expand to Pgm screen }
end;

Procedure Explain;
begin
clrscr;gotoxy(2,7);
write('Sorry, I can't find your file. Press enter to Exit ');
readln;halt;
end;

Procedure GetFileName;
begin
clrscr;gotoxy(2,7);
write('Please enter the file name to be displayed ');
readln(filenam);
end;

Procedure OpenFile(Fname : String30);
Begin
{ $Ifdef Soft }
assign(BytFile,Fname);{$I-}
reset(BytFile);{$I+}; ok:= IOresult = 0;
{ $Endif }
{ $Ifndef Soft }
assign(Filehandle,Fname);{$I-}
reset(Filehandle);{$I+};ok:=IoResult = 0;
{ $EndIf }
if not ok then explain;
End;

Procedure Init;
Begin
new(p);Start:=P; { Initialize link }
```



```

P'.Prev:=-Nil;P'.Next:=-Nil;
RunP:=P;
end;

```

```

Procedure Rest;

```

```

Begin
  New(p);P'.Prev:=RunP { Form chain, linked backwards and forwards }
  RunP'.Next:=P;P'.Next:=-Nil { enable identification of last link }
  RunP:=P;
end;

```

```

{$ifdef Soft }

```

```

Procedure Assn;

```

```

var I : Integer;
begin
  I:=1;
  while(bt <> ord(LnFd)) do
    begin { Use line feed character to indicate }
      read(ByteFile, bt); { end of line }
      bt:=((Bt) and (mask)); { strip bit 7 from byte }
      P'.Words[i]:=chr(bt);
      dec(Sz); { count the byte read }
      inc(i); { increment subscript }
      if(Sz < 1) then begin
        EndF:=True;numchar:=i;
        exit;end;
      end;
      P'.Words[0]:=chr(i-1); { assign length byte to string }
      bt:=0; { reset bt }
      If((Sz mod Divsor = 0) and
        ( J <= LastChar)) then begin { Spell out "Please wait..." }
        write(message[J]);inc(J);end;
      end;
    {$endif }
  end;
  P'.Words[0]:=chr(i-1); { assign length byte to string }
  bt:=0; { reset bt }
  If((Sz mod Divsor = 0) and
    ( J <= LastChar)) then begin { Spell out "Please wait..." }
    write(message[J]);inc(J);end;
  end;
  {$endif }
end;

```

```

{$ifndef Soft }

```

```

Procedure Assn;

```

```

begin
  while not EoLn (Filehandle) do begin
    readLn(filehandle,P'.Words);{ read file into string storage, }
  end;
  end; { a line at a time }
  {$endif }

```

```

Procedure Scroll_Window(Up_Lin, Low_Lin, L_Col, R_Col, Num, Dir : Integer);

```

```

Var { Very handy screen scroll }

```

```

REGS: REGISTERS; { procedure }

```

```

begin

```

```

  with Regs do

```

```

    begin

```

```

      if Dir = 1 then AH:=6 else AH:=7; { load ah to scroll in right }

```

```

      AL:=Num; { direction, al with number }

```

```

      BH:=7; { bh with attribute for blank lines }

```

```

      CH:=UP_Lin; { load ch with top line }

```

```

      CL:=L_Col; { cl with left column }

```

```

      DH:=Low_Lin; { dh with bottom line }

```

```

      DL:=R_Col; { dl with right column }

```

```

      Intr($10, REGS); { now call interrupt routine }

```

```

    end;

```

```

  end;

```

```

{$ifndef Standalone }

```

```

Procedure Execute_Show;

```

```

Label escape, _oneline;

```

```

{$endif }

```

```

Begin
  clrscr;If paramcount < 1 then { Allow command line file spec }
  GetFileName else FileName:=Paramstr(1);
  Openfile(Filenam);clrscr;
  {$ifdef Soft }

```

```

  Sz:=FileSize(ByteFile); { get size of file }
  {$endif }

```

```

  one:=False;

```

```

  { done:=False;Endf:=False;J:=1;

```

```

  If(Sz > 20) then Period:=Sz div 20 { variables used by "assn" }

```

```

  else Period:=1;Divsor:=Sz div PERIOD { to compute interval }

```

```

  if Divsor = 0 then divsor:=1; { Can't allow division by zero! }

```

```

  gotoxy(15,25);

```

```

  write('Commands: PgUp, PgDn, Arrow Up, Arrow Dn, End ');

```

```

  gotoxy(30,12);{ place cursor near center of screen }

```

```

  init;assn;

```

```

  {$ifdef Soft } { see text }

```

```

  While ((Sz > 0) and (Endf <> True)) do begin

```

```

    {$endif }

```

```

    {$ifndef Soft }

```

```

    While NOT EOF(Filehandle) do begin

```

```

      {$endif }

```

```

      rest;assn;

```

```

    end;

```

```

    if(Start'.Next = Nil) then { only one line in file }

```

```

      begin one:=True;goto oneline;end;

```

```

      P:=start;col:=1;Row:=1;

```

```

      clear_down(1,1,80,24);

```

```

      while P'.Next <> Nil do

```

```

        begin

```

```

          repeat { Print first screen full of file }

```

```

            gotoxy(col,row); { 24 lines, or until end of file }

```

```

            write(P'.words);

```

```

            P:=P'.Next;inc(row);

```

```

          until (row = 25) or (P'.Next = nil);

```

```

          T_ptr:=Start;B_ptr:=P'.Prev;

```

```

          if(P'.Next = Nil) and

```

```

            (row < 23) then

```

```

              begin { If end of file, inform user }

```

```

                gotoxy(1,row+1);write(EndMess);

```

```

              end;

```

```

            gotoxy(1,row);

```

```

          end;

```

```

          row:=row-1;LastRow:=Row;P:=P'.Prev; { point to top of next screen }

```

```

          gotoxy(1,row);{ initialize variables }

```

```

        repeat

```

```

          ch:=readkey;

```

```

          if (ch = #0) and keypressed

```

```

            then begin { Look for key pad key, ignore rest }

```

```

              ch:=readkey;

```

```

              case ch of

```

```

                'Q':begin { *** PgDn *** }

```

```

                  if (P'.Next = Nil) then

```

```

                    goto escape;I:=1; { if P'.Next = Nil exit loop }

```

```

                    RunP:=B_ptr;P:=B_ptr;

```

```

                    clear_down(1,1,80,24);

```

```

                    while (P'.Next <> Nil) and (I <= 24) do

```

```

                      begin

```

```

                        gotoxy(1,I);write(P'.Words);

```

```

                        P:=P'.Next;inc(I);

```

```

                      end;

```

```

                    If(I >= 24) then begin

```

```

                      T_ptr:=B_ptr;

```

```

                      B_ptr:=P'.Prev { Note the lack of semicolon here }

```

```

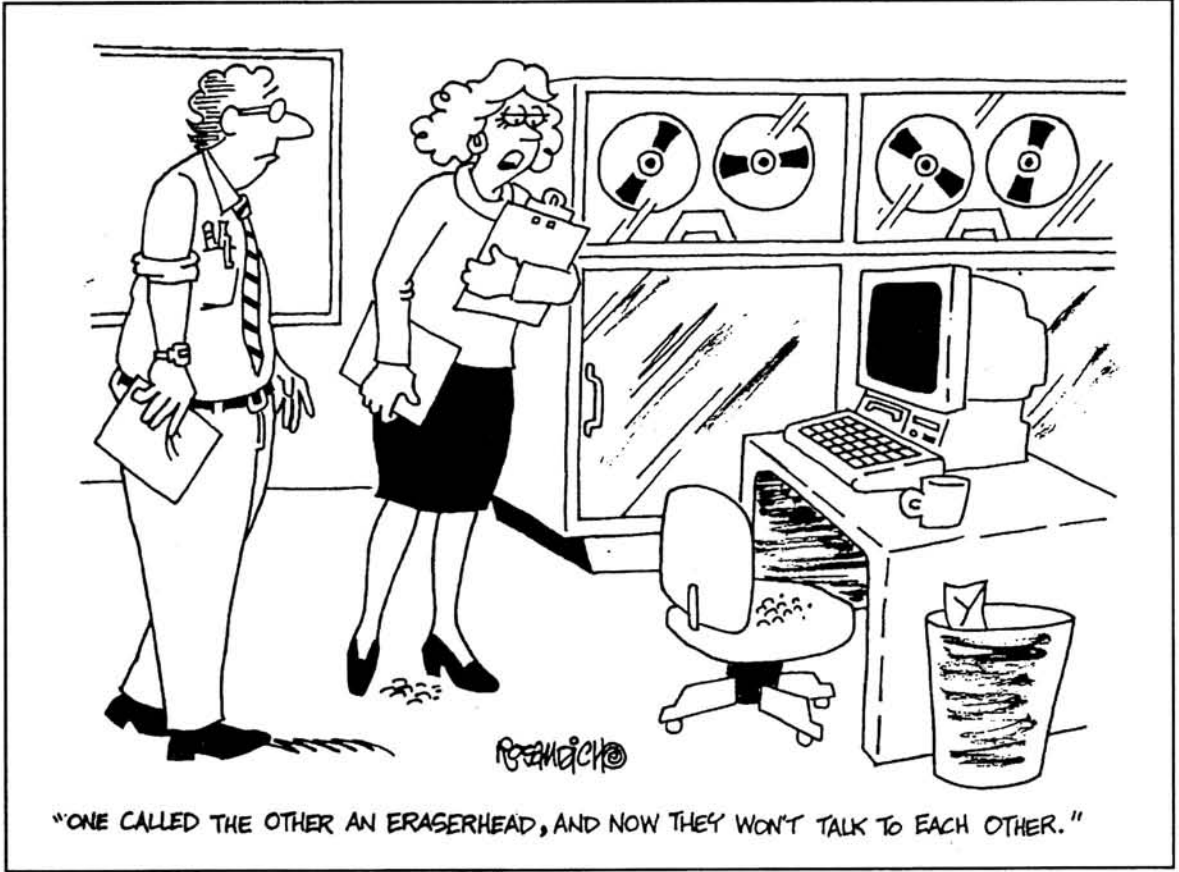
                    end else begin

```

```

writeln(p^.Words); { and print it, quit }
writeln(EndMess);ch:=readkey;end;
{$ifdef Standalone }
end;
{$endif }
end.

```



"ONE CALLED THE OTHER AN ERASERHEAD, AND NOW THEY WON'T TALK TO EACH OTHER."

```

T_Ptr:=RunP; { VERY IMPORTANT! with "else" }
P:=B_Ptr;end;
LastRow:=I;
if(P^.Next = Nil) or (I < 24) then
begin
if I = 25 then begin
scroll_window(0,23,0,79,1,1); { don't print on bottom }
I:=I-1;T_Ptr:=T_Ptr^.Next;
B_Ptr:=B_Ptr^.Next;end; { of screen (row 25) }
gotoxy(1,I);writeln(EndMess);
end;
end;
'I':begin clear_Down(1,1,80,24); { *** PgUp *** }
I:=I;P:=T_Ptr; { Store pointer }
while(P^.Prev <> Nil) and
(I < 24) do { Slide up a screen full }
begin { of pointers, checking }
P:=P^.Prev;I:=I+1; { for top of file as we go }
end;T_Ptr:=P;I:=1;
while_(P^.Next <> Nil) and
(I < 25) do begin
LastRow:=I;Row:=I; { We have new pointer, }
gotoxy(1,I);I:=I+1; { now display new screen }
write(p^.Words);P:=P^.Next;
end;B_Ptr:=P^.Prev;
gotoxy(1,Row);
end;
'H':begin Row:=I;P:=T_Ptr { *** UpArrow *** }
if(T_Ptr^.Prev <> Nil) then
P:=T_Ptr^.Prev
else goto escape;T_Ptr:=P; { get pointer for line }
scroll_window(0,23,0,79,1,0); { above, scroll screen }
gotoxy(1,1);clreol;
write(p^.Words); { & print top line }
gotoxy(1,Row);Inc(LastRow);
B_Ptr:=B_Ptr^.Prev;
end;
'P':begin If(P^.Next = Nil) { *** DnArrow *** }
or (LastRow < 24) then goto escape;
row:=I;P:=T_Ptr;I:=1 { get pointer for }
while (P^.Next <> Nil) and (I <= 24) do { top line }
begin
P:=P^.Next; { descend 24 lines }
Inc(I); { see if we reached }
end; { end of file }
if(P^.Next = Nil) and (I < 22) then
begin gotoxy(1,1);writeln(EndMess);
goto escape;end;
If(P^.Next <> Nil) then begin { If not, establish }
B_Ptr:=P;T_Ptr:=T_Ptr^.Next; { new pointers }
scroll_window(0,23,0,79,1,1); { scroll screen }
gotoxy(1,I-1);clreol;
write(p^.Words); { write new bottom line }
gotoxy(1,Row);
end;
end;
'O':begin window(1,1,80,25);
clrscr;exit;end; { *** End Key, exit *** }
end;
end;
escape:
until ((keypressed = True) and (ch= 'O'));
oneline;if(one = true) then begin { if only one line in }
end;
gotoxy(1,1);P^.Words[0]:=chr(numchar-3) { file, compute its length }

```

First we are defining a pointer to "line" which is a Turbo Pascal "record". The caret "^" symbol tells the compiler that it is dealing with a pointer type - NOT the NAME (ptr), we tacked on to it. We could just as well have named it "bird_dog" and the compiler wouldn't care. Each record contains a variable (Words) to hold the text for the line, a (memory) pointer to its predecessor (Prev) and a pointer to its successor (Next). There is one very important point to be stressed here, we can't allow the program to lose track of its current pointer to a memory location. If this is allowed to happen, you'll be able to hear the (program) crash for miles. Well, maybe not for miles - feet perhaps?

Since the position of our memory pointer is so precarious, we have to take steps to ensure that it doesn't step out into oblivion. Take a look at the procedure "init". After allocating memory for the record "new(p)", we make "Start" equal to "P" as a means of always being able to find the first record. Then the memory location reserved for a pointer to the PREVIOUS record is made equal to NIL, thus giving us an easy way of identifying the (beginning) end of our file. Assuming that the current pointer is somewhere near the middle of the file, we can traverse in a backward direction by using the code fragment: if P^.Prev <> Nil then P:=P^.Prev; Thus allowing the program to take a step to the rear (safely). Conversely, the succeeding record may be addressed using: If P^.Next <> Nil then P:=P^.Next;

As each ensuing record is processed, the link in the chain is forged using: "New(p);P^.Prev:=RunP;RunP:=P;". Then "RunP" is made current "RunP:=P", and "P^.Next:=Nil;" provides a means of identifying the last link in the dynamic chain.

Examine how the individual program routines, dealing with "ArrowUp", "PgUp", "ArrowDn" and "PgDn" (found in "case ch of"), process the cursor movement commands. Each routine has a unique problem to deal with, but uses much the same logic (alluded to above) in solving it.

After you have copied the source code for this month's listing into your computer and eliminated the "typos"; I would strongly suggest that you make some changes in those parts of the program which may not be clear to you and then "run" the modified version. There is no substitute for a "hands on" approach to the acquisition of programming skills.

Although Turbo Pascal has a rather extensive library of screen control procedures, there is one feature that is missing - that of scrolling an area (window) of the screen. Of course there are ways of giving the appearance of a screen scroll without actually invoking such a routine, or, a window may be created and either "InsLine"

or "DelLine" may be called to scroll the window a line at a time. In either case the coding is more complex (and larger). Included in this month's listing is a routine for scrolling any part of the screen, one line or several, in either direction. This procedure should certainly be a candidate for inclusion in your library of useful routines.

Let's look at "Procedure Scroll_Window(Up_Lin, Low_Lin, L_Col, R_Col, Num, Dir);" for a minute. As you can see by the parameters passed, it will scroll any portion of the screen you wish - as many lines as you wish (up to a screen full (25)) and as many columns as you wish - from one to eighty. Here is a procedure that could come in very handy in the creation of a "help" utility to run within a "window". I can think of numerous other uses as well, your imagination can take over from here.

Scroll_Window makes use of "Bios" services to perform the window manipulation for us. The "IBM PC Bios" can be used to provide a number of screen utilities, some of which are part of the Turbo Pascal standard library - such as gotoxy(col, row) and clrscr.

Notice the statement: "REGS : REGISTERS;" within our "Scroll_Window" procedure. Pseudo registers are being created to emulate the real thing (used to pass values to the CPU in assembly language, "REGISTERS" is a Pascal data TYPE defined within the DOS unit). The "REGISTERS" are loaded with the parameters to be passed to the BIOS interrupt and then the interrupt is called (by number) using the statement: Intr(\$10, Regs);

In the interests of eliminating redund-

ant code listings, the modified code for "Projct_1.pas" will not be included in this article, Instead I will describe the changes to be made. They are few in number and quite easy.

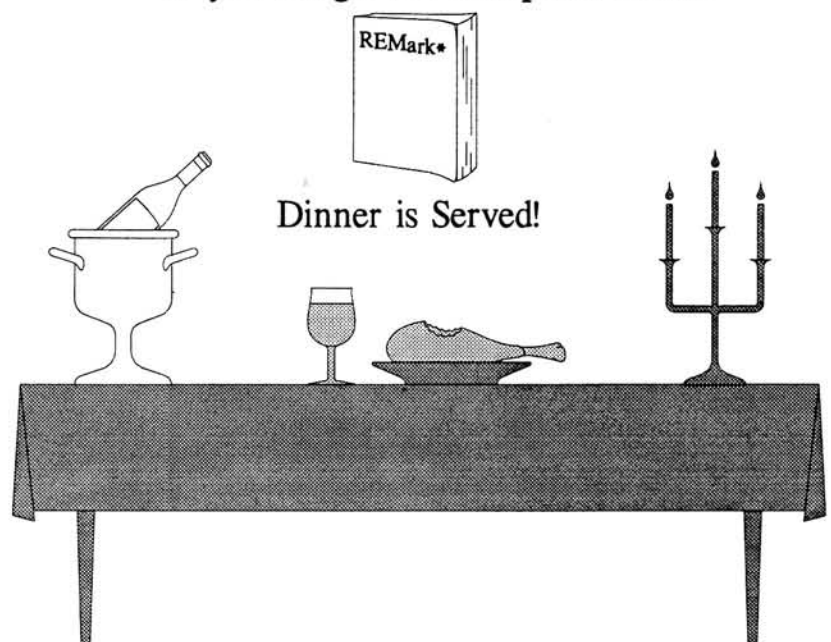
First, on the second line, add to the uses statement: "Showunit". The modified line should read: "uses boxunit, crt, calcunit, DchxUnit, calnunit, ShowUnit;". Next, on line four of the "menu" statements change: "4. Function not yet implemented." to: "4. View ascii text file.". Now move down to the "case option of" statement and add: "4:begin window(1,1,80,25); Execute_show; end;". Make sure that you have configured the program listing to produce a "TPU" version of the file when compiled (by inserting a space in front of the "\$" in the { \$Define Standalone } statement. Now compile the modified version of "Projct_1.pas". Before I leave this subject, let me add that Turbo Pascal unit FILE NAMES MUST MATCH the unit name within the source code.

Success! Our project program is taking on the appearance of a very handy utility while you are building a library of quite useful routines.

One feature of the program that I have not yet mentioned is its ability to accept filenames from the command line (stand alone version). Assuming you have named our new program "Show", and you wish to examine an ascii text file named: "Myfile", the command: "show myfile" <Return> will invoke "show" and display the contents of "myfile".

To code well, bodes well. See you soon. ✨

If you hunger for Computer news...



QUIKDATA - 16 YEARS OF H/Z SUPPORT!

YOUR H/Z ENHANCEMENT EXPERTS!

ACCELERATE YOUR PC/XT/AT!

From Sota Technologies, Inc., the fastest and most proven way to give new life to your H/Z PC/XT computer, giving it -AT compatible speeds! Turn your turtle into a -286 rabbit with a 12.5 Mhz 80286 or 80386 16Mhz SX accelerator board. Complete with 16K on-board CACHE.

The 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! **You won't believe your stop watch!**

EXP-12 - \$275

EXP386 - \$449 Much faster 16Mhz 80386 SX version

For your H/Z241 or 248 we have the new direct replacement **WESTERN IMAGING Z-33 ZUNI MOTHERBOARD** that's just loaded with features. 33Mhz 80386 main board with ten expansion slots and built in are two serial ports, one parallel port, floppy disk controller and an IDE disk controller! That gives you 10 slots, add video (and your old MFM controller if you desire) and you still have plenty of expansion left! Support to 32 meg on-board RAM using SIMMs, and 32 additional megs with memory expansion.

WIZ-33 - \$849 with 0K RAM

80387-33 - \$195 Coprocessor

SIM1X9-7 - \$59 1 meg 70ns SIMM RAM

MEMORY UPGRADES

Note: All memory upgrades come without memory chips. 150ns 256K DRAMs are \$1.79 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.

EAZYRAM - \$89 Upgrades EaZy PC from 512 to 640K

ZMF100 - \$55 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 - \$55 Similar to ZMF100 above, but for new motherboards with p/n 181-4918 or greater.

3MB RAM BOARD for Z241/248 computers is an excellent memory card. Will backfill your 512 to 640K, and provide both extended and expanded RAM; all can coexist. Uses 100ns M256-10 RAM chips, 36 per megabyte desired. Minimum of 18 DRAM chips required (\$1.95 each).

EVATRD - \$119

Z248/12, Z286LP RAM UPGRADE Z605-1 consists of 2MB SIMM 80ns RAM kits to upgrade your H/Z systems.

Z605-1 - \$149

Z386/20, Z386/25, Z386/33, Z386 EISA 2MB SIMM 80ns UPGRADE to add increments of 2MB to these systems. Two required.

ZA3600ME - \$89

ZA3800MK - \$349 4 megabyte SIMM upgrade for Z386/20, /25, /33, /33E. Must have 4-1 meg SIMMs installed first.

WINCHESTER UPGRADE KITS

PCW20 - \$239 Complete MFM winchester setup for a H/Z150, 148, 158, 159, 160, PC etc. Includes **21 meg** formatted half-height Segate ST-124 37ms drive, controller, cable set, doc.

ST-124 - \$195 Bare drive only

PCW30 - \$295 32 meg with 28ms Segate ST-138-1.

ST-138 - \$249 Bare drive only

PCW40 - \$319 42 meg with 28ms Segate ST-251-1.

ST-251-1 - \$269 Bare drive only

PCW80 - \$595 80 meg with Segate ST-4096 full size drive.

ST-4096 - \$549 Bare drive only

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

WDCON - \$59 PC/XT hard drive controller board

WDATCONF - \$95 1:1 interleave HD/floppy controller for AT's

FLOPPY DRIVE SAMPLE

MF501 - \$71 5" 360K DS/DD drive

MF504 - \$79 96 TPI 1.2 meg AT/Z100 drive

MF353 - \$79 720K 3.5" drive in 5" frame

MF355 - \$85 1.4 meg 3.5" AT drive in 5" frame

TM100-2R - \$69 40tk DS reburp (H8/89/Z100 PC type)

Also other drives and full line for older systems

ADD AN EXTERNAL HARD DRIVE TO ANY LAPTOP

OR PC DESKTOP with our EXPORT hard drive. Plugs into a parallel port (do not lose the port) to give you an affordable way to easily add a hard drive to your computer. Fast!

EWIN20 - \$495 20Mb unit

EWIN40 - \$625 40Mb unit

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

ADD ANY FLOPPY TO ANY LAPTOP OR PC WITH A PARALLEL PORT

easily and inexpensively. With Backpack, you simply connect the external unit to your parallel printer port (do not lose printer function), install software, and away you go! No expansion boxes needed for laptops, and no slots required. Too easy to be true, but it is. Want a 2.8mb/1.4mb/720K floppy on your MinisPort? Want to add a 1.2 meg to your laptop? Want to add an additional drive to your desktop? Plug it in and go. 2.8MB 3.5" version will read and write 2.8 meg, 1.4 meg, and 720K format. 1.2 meg version will read 1.2 meg and 360K and write 1.2 meg format.

BPACK2.8 - \$319

BPACK1.4 - \$269

BPACK1.2 - \$269

BPACK360 - \$269

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 BIOS ROM upgrades and batteries for most H/Z PC/XT/AT computers, spike protection filters, backup power supplies, tape backup units, modems, printers, cables and ribbons, disk drives and diskettes of all types, external hard drive and floppy 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! **Need a PC/XT/AT computer?** Tell us what you want and we will quote you a price on one of our custom assembled QD computers made up to your exact specifications!

Call or write in to place your order, inquire about any products, or request our 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

I Can Do It!

Part 1

**David E. Warnick
RD#2 Box 2484
Spring Grove, PA 17362-2484**

A long time ago, in a place far away (I think it was 1981 in Spring Grove, PA), an already middle-aged man, then employed as a Microwave Systems Engineer decided to get his feet wet in the new hobby of computing. After studying all available literature, reading every possible catalog, and talking to several (there weren't many in 1981) owners of computers, the choice was clear. The industry leader was a company called Heath.

This came as no surprise, as this middle-aged fellow had built stereo equipment, ham radio gear, test equipment, and several TV sets from kits made by Heath Company, and he was very familiar with their excellent quality, and their exceptional value. So, it was off to the Credit Union for a loan, then to the Baltimore, MD Heathkit Store, and finally home with boxes and boxes of parts which would eventually become a computer and a printer.

And what a system it was. The processor was an 8-bit Z-80, the hottest thing around back then. There was 48K of RAM, more than enough for most jobs a computer could do, and there was a floppy disk drive capable of storing 100K bytes on-line. This was quite an improvement over the more common cassette tape units of the day. Yes, the old Heathkit H-89 was the start of something wonderful for this fellow.

But, computers were so new, and information about them was hard to find. Magazines of that day did not help very much at all. A home study course from The Heath Company on programming in the 'BASIC' language was a big step toward better understanding and control of the system. But this fellow's next step, an inexpensive and seemingly insignificant one, opened the doors to many answers he sought. It provided a forum for the sharing

of information among thousands of folks who, like him, had an interest in this leading-edge technology. Yes, he joined a group called "The Heath Users' Group", or, more simply, "HUG". This is the organization which grew to become the "Zenith Users' Group", or more simply "ZUG". What followed were many months of studying, learning, and trying new things.

Then it happened. This fellow found that he had discovered how to do something he had not seen in the books and magazines he devoured with every chance he got. Now was the time to try to share this information with his new-found friends in HUG. He sat down and wrote what he hoped would become an article in HUG's magazine, "REMark". He carefully typed it into his word processor, printed a copy on paper, and packed that, along with a disk, and sent it to Michigan.

Then came the waiting. Every day seemed like a week. Was his work wrong? Had he really not discovered something? Would the folks in the Heath Users' Group care about what he had done? Finally, the letter arrived. His article was accepted. He was about to become a real honest-to-goodness author!!! Let this be a lesson to all of you ZUG members. You can do it, too. There's a need for your submissions to REMark, and a lot of folks would appreciate the information you can share.

OK, for anybody who doesn't know by now, I'm the person in the story above. My first article was published in May 1982. What has followed that rather unsensational beginning has been several upgrades to the old Heath H-89, a big move up to a Heath HS-151 IBM PC-Compatible with a color monitor, a hard disk drive, and 640K of RAM, and finally, to my present 12 MHz 80286-based system.

More importantly to me, were a total

of about thirty (30) articles published in REMark. We dealt with things like Sequential Files, Random Files, Games, and Serial Communications. Then came a change in my career direction, made possible partially by this great hobby of ours. I give the Heath Company and Zenith Data Systems much of the credit for my ability to move to the field of Data Processing, specifically to Data Communications. It has been a great career move, but has demanded very much of me, time-wise. Something had to give, and my experimenting and writing stopped rather abruptly.

That was about six years ago. I changed jobs in December 1983, and I have missed the writing terribly. I miss the contact a REMark author has with readers worldwide, even though attempts to keep up with correspondence can be quite frustrating. So it is that I've decided to get back to writing. That is, if what I have to offer is what the members of ZUG want and need.

In the past, all my work has dealt with programming and with understanding what really goes on inside the computer. The language I used in those past articles was always 'BASIC'. However, over the years I have migrated to the 'C' programming language. For my purposes, I find it excellent. It's not as hard to learn as many would have you believe. Those of you who have read my work in the past know that I also believe in the liberal use of flow charts and visual aids to make my work more easily understood.

If you're reading this article, that means that my writing has been accepted for publication. If so, you have become one of the directors of the way I should write. After all, this magazine is by and for members of ZUG. You must let me know what you want, and I will do my best to follow an appropriate course.

As a matter of information, I don't have a spreadsheet on my system. My choice in Database Management Systems is *Informix*, my CAD system is *Drafix CAD Ultra* and my C Compilers include Computer Innovations *C86* and *C86 PLUS* and the *Microsoft C*, release 5.1 by Microsoft. Finally, these articles are written using *WordStar Professional*, release 5.5. I don't claim any of these to be the best, and I refuse to debate their worth relative to products which do the same or similar things. I only present this information as readers sometimes want to know these things. The best thing for each computer user to do is to get a product and stick with it. Learn that product and use it well.

I cannot stress too much how important it is to learn your system or application software (word processor, compiler, spreadsheet) and to use it well. Too often, I see computer users jumping from one product to another based on a magazine article. Believe me, when I tell you that a good product used to its full capacity will outperform most excellent products which have not been learned in depth.

So much for the history lesson, and for the idle chatter about me. The purpose of any article which I write is to teach the reader, as painlessly as possible, either how something inside the computer works, or how to make these wonderfully powerful machines do your bidding. I know that you can do it. There is no doubt that if you can read this far, you can learn everything I have learned, and that you can apply it to do even more than I did. The most complex work I have done to date on a PC is the Ultra-RTTY program sold by ZUG as part number 885-6012 and found in each month's Software Price List. It performs many functions like converting Teletype codes to Computer talk, recognizing the special function keys, communication through HAM Radio, reading and writing disk files, controlling a printer, and much, much more.

How does anybody know how to do all those things? How does one write a program so complex? And how would you ever do it in a language like 'C'? The answer to each of these questions is the same. You can do all of these things if you just do them one step at a time. Each step is easy by itself. So, all you have to do is a lot of easy things, one after another. That's how your computer does it. When you type a character and it appears on the screen, that isn't one complex job for your computer. Instead, it does a lot of easy things, one at a time. It looks at the keyboard. If there is a character waiting (that is, if a key has been pressed), the second step, get that character, is performed. Then, it looks at the display. If the display is ready to accept another character, the computer completes step four, give the character to the display. Then, as I'm sure you've already guessed, the computer

looks at the keyboard again.

It's all pretty simple when you look at it that way, isn't it. You see, I've already sneaked in a lesson. While many of you already think this way about how the computer operates, I hope I've helped some newcomers. Other lessons will be just as easy, or even easier, and before we're done, you'll be able to do things far more complex than the 'Ultra-RTTY' program. Just stick with me, and you'll see.

When I started writing this series, I couldn't come up with a good name. As I worked on the individual articles, the title, *YOU CAN DO IT!*, came to mind. That seemed OK, but didn't show the self-confidence I know each of you can feel as we go along. So it was that I came up with *I Can Do It!* That's what you should tell yourself every time you read one of these articles. I can do it!! And well you can. Yes, there will be failures. But are they really failures? Not if you learn from them, they're not. You didn't fail in trying to do something. You learned how NOT to do it. That may sound trivial at first, but as you go along, you will begin to appreciate how important those lessons are, too.

We will look at both 'BASIC' and 'C' language here. If you use 'Pascal', 'FORTRAN', 'COBOL' or any other language, stick with us. It's amazing how similar many of the words or instructions are among the many available programming languages. It's definite that the things your computer can do will not change, just because the instructions were written a different way. Printing a sentence on the printer is the same no matter what language you use to make it happen. It's important for you to recognize that the principals are the same. So, hopefully, everybody can learn something from this series.

Before I stop writing this month, I'd like to discuss the things your computer can do. There aren't that many, and they form the basis for all the work to come. Yes, the following paragraphs form a lesson, but it's an easy one, so don't fret. It's also a lesson which you may want to refer to often at first, so now would be a good time to stick a paper clip on this page to mark your spot.

Input Data

Your computer can input data. This function can take many forms. The data could come from the keyboard as you type, it could come from a floppy disk, or from a hard disk drive. There are other sources of data for your computer, also. These include a modem connected to a telephone line, a mouse used in graphics work, or any of a number of other devices. But, no matter where the data is coming from, or no matter what the computer does with this data, the computer is doing just one simple thing. It is inputting data from somewhere.

Output Data

This is just the opposite of the function above. Data which the computer holds is sent elsewhere. It could go to the display screen for you to read, to the printer or a plotter to produce a paper copy, to a modem for transmission to some far-away place, or to one of your disk drives to be saved for another time. Again, it is easier to think of this as a simple function. Just output the data. We can worry about where it should go and how to do it later.

Make Comparisons

Sometimes it matters whether two things are the same or different. If they are different, it may be important to know whether the first item is larger or smaller than the second item. Not to worry! This is perfect work for the computer. It can tell you whether an item is smaller than, equal to, or greater than another item. This is how a computer sorts mailing lists by name, by zip code, or any other way you want. It may have to make thousands of comparisons to do such a job, but, the basic step is the same, and it is a simple one.

Make Decisions

This one can really take the stress out of your life. That little guy sitting on your desk, humming away quite happily will make a lot of decisions for you. Perhaps a teacher has stored all the numeric grades for a test in a file on a disk, and now has to assign letter grades for each student. It would be easy to tell the computer that a grade of 91 to 100 is an 'A', a grade of 81 to 90 is a 'B', etc. and let the computer make all the decisions. Or you could tell it that if the balance in your checkbook program is less than zero, it is to turn red.

Each of these things is a decision, and the computer will make them consistently, faultlessly, and as often as you like.

Do Math

If you can write a mathematical formula, the chances are that your computer can solve it for you. That's why and how these little fellows can print thirty years of mortgage payments by knowing only the amount of the loan, the interest rate, and the repayment period. Again, the computer may have to make many, many calculations to give you the results that you want, but it's only doing it one step at a time, and it's only doing one thing, and that's math.

Loop

WAIT A MINUTE!!! This guy is trying to see if we're still awake. Airplanes loop. Computers just sit on a desk. Well, not quite. Loop is what we call doing the same thing several times. A classic assignment given to new programming students is to print the numbers 1 to 10, each on a separate line. Behind each of those num-

Continued on Page 48



Using a Full Size Keyboard on a Laptop Computer

Pat Swayne
ZUG Software Engineer

A Handy
Utility to
Make It Work
Better

Some models of Zenith Data Systems laptop computers, including the TurbosPort 386, TurbosPort 386e, and SupersPort SX, can use a full size desktop keyboard in place of the built-in keyboard. Unfortunately, there are some special functions accessible on the built-in keyboard that you cannot access from a standard keyboard. You can switch the computer's speed between fast and slow, change the gray-scale palette on the LCD, and switch the video between the LCD screen and an external CRT.

I have written a program called FSKBD.COM to solve this problem. It is a TSR (Terminate and Stay Resident) utility that you run once each time you boot your computer. While it is active, your standard keyboard can access the special functions listed above using the same function keys as on the laptop keyboard, except that you use the Shift and Ctrl keys together instead of the Fn key. For example, on the laptop keyboard you can set the slow speed by typing Fn-F3 (hold down Fn and press F3). On a standard keyboard, with FSKBD.COM active, you would type Ctrl-Shift-F3 (hold down Ctrl and Shift and press F3).

I have provided several ways for you to get this program. The easiest way is to get it is to call Zlink-COM1 at 616-982-3956 with your modem and download FSKBD.ZIP. If you don't have a modem, you can type in the BASIC program in Listing 1. When you run it, it will create FSKBD.COM in your default directory. Or, if you really like to type, you can type in the assembly source code in Listing 2, and assemble it to get FSKBD.COM.

How to Use FSKBD.COM

Before you use FSKBD.COM, you should copy it to your default directory, or a directory pointed to by your PATH statement. To load FSKBD.COM into memory and activate it, just type

Listing 1. A BASIC program that creates FSKBD.COM.

```
10 PRINT "CREATING FSKBD.COM
20 OPEN "O",1,"FSKBD.COM":L=100
30 FOR I=1 TO 79 :C=0:FOR J=1 TO 12
40 READ B:C=C+B:PRINT #1,CHR$(B);:NEXT J:READ S
50 IF S<>C THEN PRINT "TYPING ERROR IN LINE";L:STOP
60 L=L+10:NEXT I:CLOSE #1:SYSTEM
100 DATA 233,209,0,144,70,83,75,68,66,32,32,49,1061
110 DATA 46,48,1,0,0,0,0,0,0,0,0,0,0,95
120 DATA 0,128,252,79,116,5,46,255,46,79,0,46,1052
130 DATA 128,62,78,0,1,117,83,60,29,116,80,60,814
140 DATA 157,116,84,60,42,116,88,60,54,116,84,60,1037
150 DATA 170,116,88,60,182,116,84,46,128,62,87,0,1139
160 DATA 3,117,51,80,36,128,46,162,88,0,88,36,835
170 DATA 127,180,90,60,61,116,26,180,91,60,62,116,1169
180 DATA 20,180,92,60,66,116,14,180,93,60,67,116,1064
190 DATA 8,180,94,60,68,116,2,235,2,138,196,46,1145
200 DATA 10,6,88,0,180,79,207,46,128,14,87,0,845
210 DATA 1,235,247,46,128,38,87,0,254,235,239,46,1556
220 DATA 128,14,87,0,2,235,231,46,128,38,87,0,996
230 DATA 253,235,223,46,128,62,78,0,2,116,5,46,1194
240 DATA 255,46,83,0,14,31,51,192,142,192,190,79,1275
250 DATA 0,191,84,0,252,165,165,190,83,0,191,160,1481
260 DATA 0,165,165,14,7,180,73,205,33,142,6,44,1034
270 DATA 0,180,73,205,33,251,205,32,160,93,0,162,1394
280 DATA 218,2,252,186,219,2,180,9,205,33,180,82,1568
290 DATA 205,33,38,139,71,254,142,216,161,1,0,140,1400
300 DATA 202,59,194,115,49,142,192,190,4,1,139,254,1541
310 DATA 129,239,192,0,30,14,31,185,5,0,243,167,1235
320 DATA 116,17,140,6,213,2,31,161,3,0,64,140,893
330 DATA 219,3,195,142,216,235,209,140,6,215,2,198,1780
340 DATA 6,217,2,1,235,232,14,31,128,62,217,2,1147
350 DATA 1,117,64,142,6,215,2,50,192,186,17,4,996
360 DATA 128,62,218,2,68,116,36,176,2,186,66,4,1064
370 DATA 128,62,218,2,85,117,19,139,30,215,2,59,1076
380 DATA 30,213,2,115,14,186,94,4,180,9,205,33,1085
390 DATA 205,32,254,200,186,42,4,80,180,9,205,33,1430
400 DATA 88,38,162,78,0,205,32,128,62,218,2,85,1098
410 DATA 116,86,14,7,190,0,1,191,64,0,185,212,1066
420 DATA 0,243,164,184,21,53,205,33,137,30,79,0,1149
430 DATA 140,6,81,0,184,40,53,205,33,137,30,83,992
440 DATA 0,140,6,85,0,184,21,37,186,89,0,205,953
450 DATA 33,184,40,37,186,223,0,205,33,186,24,3,1154
460 DATA 180,9,205,33,128,62,93,0,68,117,12,198,1105
470 DATA 6,78,0,0,186,17,4,180,9,205,33,186,904
480 DATA 20,1,205,39,186,141,4,180,9,205,33,205,1228
490 DATA 32,0,0,0,0,0,0,13,10,70,83,75,283
500 DATA 68,66,32,45,45,32,70,117,108,108,32,83,806
510 DATA 105,122,101,32,75,101,121,98,111,97,114,100,1177
520 DATA 32,85,116,105,108,105,116,121,32,102,111,114,1147
```

FSKBD

at the DOS prompt, and press Enter. If you need to run a program that uses the combination of Shift, Ctrl and the function keys, you can temporarily disable FSKBD by entering

FSKBD D

To enable it again just enter FSKBD as you did to load it in the first place. If you need to remove FSKBD from memory (it only uses a few hundred bytes), enter

FSKBD U

You can only remove FSKBD if it is the last TSR program loaded.

Note: A on a TurbosPort 386, you can only swap the display between the LCD and an external CRT at the monitor prompt. You cannot do it after you boot MS-DOS, so FSKBD will not work for that function.

How FSKBD Works

This section of this article is for those of you who have been following my series on assembly language. It provides you with an example of an interrupt-driven TSR program.

Every TSR program is divided into two sections – the resident section, and the setup/control section. At the beginning of the FSKBD listing (after some memory location definitions) is a jump around the resident part to the setup/control part. The resident part contains two interrupt processing routines which handle interrupts 15h and 28h. Interrupt 15h is a special software interrupt used by the BIOS for many different functions. It is sometimes called the Device Control interrupt. It can be issued either by the BIOS or by a user program depending on the function, which is designated by the number in the AH register. In an AT-compatible or newer computer, interrupt 15h is issued by the BIOS with 4Fh in the AH register to allow external processes to translate keyboard scan codes. The AL register contains the scan code, and all the processing routine has to do is replace it with another code and then return from the interrupt.

FSKBD watches for the codes of the Shift and Ctrl keys, and records in a flag when they are down or up. It also looks for the codes for the F3, F4, F8, F9, and F10 keys, which are used for the special Laptop functions that FSKBD controls. If one of these codes is detected, and the Shift and Ctrl keys are both down, then the routine replaces the code in the AL register with the code generated in the laptop when the key is pressed with Fn down.

FSKBD also handles interrupt 28h, which is issued repeatedly by MS-DOS while it is in a loop waiting for keyboard

```
530 DATA 32,76,97,112,116,111,112,115,44,32,118,46,1011
540 DATA 32,49,46,48,13,10,10,36,70,83,75,68,540
550 DATA 66,32,105,115,32,110,111,119,32,105,110,115,1052
560 DATA 116,97,108,108,101,100,32,105,110,32,109,101,1119
570 DATA 109,111,114,121,46,32,32,89,111,117,32,109,1023
580 DATA 97,121,32,99,111,110,116,114,111,108,32,105,1156
590 DATA 116,13,10,98,121,32,101,110,116,101,114,105,1037
600 DATA 110,103,13,10,10,32,32,70,83,75,68,66,672
610 DATA 32,68,9,9,84,111,32,100,105,115,97,98,860
620 DATA 108,101,32,115,117,112,112,108,105,109,101,110,1230
630 DATA 116,97,108,32,107,101,121,115,46,13,10,32,898
640 DATA 32,70,83,75,68,66,9,9,84,111,32,648
650 DATA 101,110,97,98,108,101,32,115,117,112,112,108,1211
660 DATA 105,109,101,110,116,97,108,32,107,101,121,115,1222
670 DATA 46,13,10,32,32,70,83,75,68,66,32,85,612
680 DATA 9,9,84,111,32,117,110,105,110,115,116,97,1015
690 DATA 108,108,32,70,83,75,68,66,46,32,32,89,809
700 DATA 111,117,32,99,97,110,32,111,110,108,121,13,1061
710 DATA 10,9,9,9,117,110,105,110,115,116,97,108,915
720 DATA 108,32,105,102,32,70,83,75,68,66,32,105,878
730 DATA 115,32,116,104,101,32,108,97,115,116,13,10,959
740 DATA 9,9,9,84,83,82,32,108,111,97,100,101,825
750 DATA 100,46,13,10,36,70,83,75,68,66,32,105,704
760 DATA 115,32,110,111,119,32,100,105,115,97,98,108,1142
770 DATA 101,100,46,13,10,36,70,83,75,68,66,32,700
780 DATA 105,115,32,110,111,119,32,101,110,97,98,108,1138
790 DATA 101,100,46,13,10,36,70,83,75,68,66,32,700
800 DATA 105,115,32,110,111,119,32,117,110,105,110,115,1181
810 DATA 116,97,108,108,101,100,46,13,10,36,70,83,888
820 DATA 75,68,66,32,105,115,32,110,111,116,32,116,978
830 DATA 104,101,32,108,97,115,116,32,84,83,82,46,1000
840 DATA 32,32,67,97,110,39,116,32,117,110,105,110,967
850 DATA 115,116,97,108,108,46,13,10,36,70,83,75,877
860 DATA 68,66,32,105,115,32,97,108,114,101,97,100,1035
870 DATA 121,32,110,111,116,32,105,110,115,116,97,108,1173
880 DATA 108,101,100,46,13,10,36,0,0,0,0,0,414
```

Listing 2. Assembly source code for FSKBD.COM.

```
        PAGE          ,132
;       FSKBD -- A UTILITY TO AID IN USING FULL SIZE
;       KEYBOARDS ON CERTAIN LAPTOP COMPUTERS.
;       THIS UTILITY REPLACES THE SLOW, FAST, PALETTE,
;       AND CRT/LCD KEYS.  TO USE FSKBD, ENTER
;
;       FSKBD          TO INSTALL OR ENABLE IT
;       FSKBD D        TO DISABLE IT
;       FSKBD U        TO UNINSTALL IT (MUST BE LAST TSR)
;
;       BY P. SWAYNE, ZUG SOFTWARE ENGINEER 12-SEP-91
;       (C) 1991 BY ZENITH DATA SYSTEMS USERS' GROUP
;       ALL RIGHTS RESERVED.

CODE    SEGMENT
        ASSUME  CS:CODE,DS:CODE,ES:CODE,SS:CODE

;       DEFINE SOME MEMORY LOCATIONS

        ORG      1
PSPSEG LABEL WORD          ;PROGRAM SEGMENT PREFIX
SEGMENT ADDR.
        ORG      3
MCBSIZE LABEL WORD        ;MEMORY CONTROL BLOCK SIZE
        ORG      2CH
ENVSEG LABEL WORD         ;ENVIRONMENT SEGMENT
        ORG      5DH
FCBARG LABEL BYTE        ;FCB ARGUMENT
        ORG      100H

START:  JMP      SETUP    ;GO SET UP PROGRAM
        EVEN

SIG     DB       'FSKBD 1.0' ;PROGRAM SIGNATURE

ENFLG  DB       1          ;ENABLE FLAG
INT15V DW       0,0       ;INT 15H VECTOR
INT28V DW       0,0       ;INT 28H VECTOR
```


input. During this interrupt, it is safe for a program to use MS-DOS service routines, so FSKBD uses this interrupt time to issue the commands to remove its resident part from memory when it is necessary. This interrupt was originally provided to allow PRINT.COM to print in the background, but it can be used by a TSR to accomplish any background work. However, it is not issued while an application program is running that does not use an MS-DOS routine for keyboard input, so it should not be the only interrupt you use to drive a background task. FSKBD works in the background (sort of), but since it only deals with processing keyboard codes, it only needs to be driven by a keyboard interrupt. A program that does a background job unrelated to keyboard input could use the timer interrupt to drive itself. Perhaps I will include such a program in a future assembly language article.

The setup/control section of FSKBD prints a sign-on message and then uses an internal MS-DOS function (52h) to get a list of pointers. From this list, it gets the pointer to the first Memory Control Block, which can be used to locate programs that are resident in memory. By following the "chain" of Memory Control Blocks, FSKBD can determine if its resident part is in memory or not. I will not explain this in detail here, but you can study the code, and use it virtually as is if you are planning to write your own TSR program.

If FSKBD does not find itself in memory, it first copies the resident part down so that it starts at 40h instead of 100h. This is done to save memory, and is the reason why references to memory in the resident part have "-0C0H" after them (100h - C0h = 40h). Then it installs the vectors for the interrupt processing routines, and checks for a command line argument. The only valid command line argument (other than none) when you are installing the program in memory is D, and if this is entered, FSKBD will clear the enable flag. Then it prints some instructions and exits with the resident part left in memory.

If the resident part is found in memory, FSKBD will check the command line. If the argument is U, FSKBD will check to see if its resident part is the last TSR in memory, and will flag that part to unload itself if it is the last TSR. If the argument is D, it will clear the enable flag, and if there is no argument, it will set the enable flag.

When you use the U command to unload FSKBD from memory, you will see MS-DOS issue two prompts instead of one. That is because there are actually two programs exiting to MS-DOS - the non-resident copy of FSKBD that you are running to issue the command, and the resident part that is unloading itself. ✖

```

CSFLG  DB      0                ;CTRL-SHIFT FLAG
KEYHB  DB      0                ;KEY HIGH BIT

;      INT 15H PROCESSOR. KEYS ARE PROCESSED HERE

INT15:  CMP     AH,4FH           ;IS THIS A KEY PRESS?
        JZ     GOTKEY          ;YES
        JMP     CS:DWORD PTR INT15V-0C0H ;ELSE, GO ON
GOTKEY:  CMP     CS:ENFLG-0C0H,1 ;FSKDB ENABLED?
        JNZ     INT15X        ;IF NOT, EXIT
        CMP     AL,1DH         ;CTRL DOWN?
        JZ     CTRLDN         ;YES
        CMP     AL,9DH         ;CTRL UP?
        JZ     CTRLUP
        CMP     AL,2AH         ;LEFT SHIFT DOWN?
        JZ     SHFDN         ;YES
        CMP     AL,36H         ;RT SHIFT DOWN?
        JZ     SHFDN         ;YES
        CMP     AL,0AAH        ;LEFT SHIFT UP?
        JZ     SHFUP         ;YES
        CMP     AL,0B6H        ;RT SHIFT UP?
        JZ     SHFUP         ;YES
        CMP     CS:CSFLG-0C0H,3 ;CTRL AND SHIFT DOWN?
        JNZ     INT15X        ;NO, EXIT
        PUSH    AX            ;SAVE KEY CODE
        AND     AL,80H         ;ISOLATE HIGH BIT
        MOV     CS:KEYHB-0C0H,AL ;SAVE IT
        POP     AX            ;GET KEY CODE
        AND     AL,7FH         ;STRIP HIGH BIT
        MOV     AH,5AH         ;ASSUME F3 (SLOW)
        CMP     AL,3DH         ;F3?
        JZ     FIXKEY         ;YES, FIX IT
        MOV     AH,5BH         ;ASSUME F4 (FAST)
        CMP     AL,3EH         ;F4?
        JZ     FIXKEY         ;YES
        MOV     AH,5CH         ;ASSUME F8 (PALETTE )
        CMP     AL,42H         ;F8?
        JZ     FIXKEY         ;YES
        MOV     AH,5DH         ;ASSUME F9 (PALETTE )
        CMP     AL,43H         ;F9?
        JZ     FIXKEY         ;YES
        MOV     AH,5EH         ;ASSUME F10 (CRT/LCD)
        CMP     AL,44H         ;F10?
        JZ     FIXKEY         ;YES
        JMP     SHORT FIXHB    ;NOT A SPECIAL KEY
FIXKEY:  MOV     AL,AH         ;USE VALUE IN AH
FIXHB:   OR      AL,CS:KEYHB-0C0H ;RESTORE HIGH BIT
        MOV     AH,4FH         ;RESTORE AH VALUE
INT15X:  IRET
CTRLDN:  OR      CS:CSFLG-0C0H,1 ;MARK CTRL DOWN
        JMP     INT15X
CTRLUP:  AND     CS:CSFLG-0C0H,11111110B ;MARK CTRL UP
        JMP     INT15X
SHFDN:   OR      CS:CSFLG-0C0H,2 ;MARK SHIFT DOWN
        JMP     INT15X
SHFUP:   AND     CS:CSFLG-0C0H,11111101B ;MARK SHIFT UP
        JMP     INT15X

;      INT 28H PROCESSOR. USED FOR UNINSTALLING FSKDB.

INT28:  CMP     CS:ENFLG-0C0H,2 ;UNINSTALL FSKDB?
        JZ     UNINS          ;YES
        JMP     CS:DWORD PTR INT28V-0C0H ;ELSE, EXIT
UNINS:  PUSH    CS            ;POINT TO THIS SEGMENT
        POP     DS
        XOR     AX,AX
        MOV     ES,AX         ;AND TO INT. SEGMENT
        MOV     SI,OFFSET INT15V-0C0H ;POINT TO INT 15 VECTOR
        MOV     DI,15H*4      ;WHERE IT GOES
        CLD
        MOVSW                ;RESTORE VECTOR
        MOVSW
        MOV     SI,OFFSET INT28V-0C0H ;POINT TO INT 28 VECTOR
        MOV     DI,28H*4      ;WHERE IT GOES
        MOVSW                ;RESTORE VECTOR
        MOVSW
        PUSH    CS

```

```

POP      ES
MOV      AH,49H
INT      21H
MOV      ES,ENVSEG
MOV      AH,49H
INT      21H
STI
INT      20H

ENDRES:

;
; INSTALL FSKDB IN MEMORY
SETUP:   MOV      AL,FCBARG
MOV      FCBARG1,AL
CLD
MOV      DX,OFFSET SIGNON
MOV      AH,9
INT      21H
MOV      AH,52H
INT      21H
MOV      AX,ES:[BX-2]
MOV      DS,AX
MOV      AX,PSPSEG
FNDLPL: MOV      DX,CS
MOV      AX,DX
CMP      CHKFNDD
JAE      CHKFNDD
MOV      ES,AX
MOV      SI,OFFSET SIG
MOV      DI,SI
SUB      DI,0COH
PUSH    DS
PUSH    CS
POP      DS
MOV      CX,5
REPZ    CMPSW
JZ      GOTCC
MOV      LSTSEG,ES
FNDNXT: POP      DS
MOV      AX,MCBSIZE
INC      AX
MOV      BX,DS
ADD      AX,BX
MOV      DS,AX
JMP     FNDLPL
GOTCC:  MOV      FNDSEG,ES
MOV      BYTE PTR FNDNXT,1
JMP     FNDNXT
CHKFNDD: PUSH    CS
POP      DS
CMP      BYTE PTR FNDNXT,1
JNZ     NOTFND
MOV      ES,FNDSEG
XOR     AL,AL
MOV      DX,OFFSET DISMSG
CMP      FCBARG1,'D'
JZ      SETFLG
MOV      AL,2
MOV      DX,OFFSET UNMSG

; PUT ES HERE
; RELEASE THIS PROGRAM'S MEMORY
; POINT TO ENVIRONMENT SEGMENT
; RELEASE ENVIRONMENT SEGMENT
; ENABLE INTERRUPTS
; EXIT
; END OF RESIDENT CODE
; SAVE FCB ARGUMENT
; PRINT SIGN-ON
; GET MEMORY CNTRL BLOCK ADDR.
; GET FIRST MCB SEGMENT
; POINT TO IT
; GET PSP SEGMENT
; IN THIS SEGMENT?
; IF SO, CHECK IF FSKDB FOUND
; ELSE, PSP SEGMENT TO ES
; PROGRAM IS MOVED DOWN
; SAVE MCB SEGMENT
; PUT DS HERE
; 5 WORDS IN SIGNATURE
; IS FSKDB HERE?
; WE FOUND FSKDB
; SAVE PSP SEGMENT
; RESTORE MCB SEGMENT
; GET MCB SIZE
; CORRECT IT
; CALCULATE NEXT MCB SEGMENT
; POINT TO IT
; TRY AGAIN
; SAVE SEGMENT WHERE FOUND
; MARK PROGRAM FOUND
; FIX DS
; WAS FSKDB FOUND?
; NO
; GET SEGMENT WHERE FOUND
; ASSUME DISABLE WANTED
; DISABLE FSKDB?
; YES
; ASSUME UNINSTALL

; UNINSTALL?
; NO, ENABLE FSKDB
; GET SEGMENT WHERE FOUND
; COMPARE WITH LAST PSP FOUND
; OK TO UNLOAD
; ELSE, SAY "CAN'T UNLOAD"
; ELSE, ENABLE FSKDB
; SAVE FLAG
; PRINT APPROPRIATE MSG
; GET FLAG
; SET CONDITION
; AND EXIT
; UNINSTALL?
; CAN'T, IT'S NOT INSTALLED!
; FIX ES
; POINT TO START OF PROGRAM
; PUT IT HERE
; SIZE OF PROGRAM
; MOVE PROGRAM DOWN
; GET INT 15H VECTOR
; SAVE IT
; GET INT 28H VECTOR
; SAVE IT
; INSTALL NEW INT 15H VECTOR
; INSTALL NEW INT 28H VECTOR
; PRINT INSTALLATION MSG
; DISABLE?
; NO
; ELSE, KILL ENABLE FLAG
; SAY "FSKDB DISABLED"
; EXIT WITH FSKDB RESIDENT
; SAY "NOT INSTALLED"
; LAST PSP SEGMENT
; SEGMENT WHERE FOUND
; PROGRAM FOUND FLAG

FCBARG1,'U'
ENABLE
BX,FNDSEG
BX,LSTSEG
SETFLG
DX,OFFSET NOUNMSG
AH,9
INT 21H
20H
AL
DX,OFFSET ENMSG
AX
AH,9
INT 21H
AX
ES:ENFLG-0COH,AL
20H
FCBARG1,'U'
CANTUN
CS
ES
SI,OFFSET START
DI,40H
CX,ENDRES-START
MOVSB
AX,3515H
21H
INT15V-0COH,BX
INT15V-0BEH,ES
AX,3528H
21H
INT28V-0COH,BX
INT28V-0BEH,ES
AX,2515H
DX,OFFSET INT15-0COH
21H
AX,2528H
DX,OFFSET INT28-0COH
21H
DX,OFFSET INMSG
AH,9
INT 21H
FCBARG,'D'
NOTDIS
BYTE PTR ENFLG-0COH,0
DX,OFFSET DISMSG
AH,9
INT 21H
20H
NOTDIS: MOV      DX,OFFSET ENDRES-0COH
INT      27H
CANTUN: MOV      DX,OFFSET NINMSG
MOV      AH,9
INT      21H
INT      20H
LSTSEG DW 0
FNDSEG DW 0
FNDNXT DB 0

```

Continued on page 36

Harvard Graphics

3.0

George P. Elwood
1670 N. Laddie Court
Beavercreek, OH 45432

As a long time user of Harvard Graphics, I learned to accept the limitations of the program. Some of the options left something to be desired. I did find Harvard Graphics easy to learn and use. As part of my job, I wrote two instructional manuals on Harvard Graphics and taught over 1500 people how to use the product. The drawing capability was acceptable, although not real good. I was able to create over 500 symbols using the drawing capability, which I provided as part of the class.

Version 2.3 of Harvard Graphics (HG) fixed some of the problems with earlier versions of the program, but this resulted in a repaired version of the current products, V2.13. Software Publishing Corporation has completely revised Harvard Graphics in the new release V3.0 (HG3). The product, while retaining the basic look and feel, has been improved a great deal. They have included Draw Partner in the Draw Menu so you do not have to leave the program. Use of colors has been improved. You may now use 64 colors on screen at one time out of a total of 16 million possible colors/hues. Real VGA support is now built in to support the color options. It is not Windows compatible, but that is coming later this year. The present version will now run in Windows as a DOS application program.

Like the other Software Publishing Corporation products, the program comes with both 3.5 and 5.25 inch disks. The installation is easy. Load disk one of the software into a floppy drive and type "INSTALL". The files are all compressed on the disks and installation time will vary depending on the CPU. The first screen in the installation procedure will give the estimated time and the amount of disk space required. The installation procedure does permit you to select the parts of the program you wish to install if space is a problem.

Installation will place the files in a sub-directory called HG3 as a default. Under this sub-directory, others are created for the sub-functions of the program. These ten sub-directories are DATA, DRIVER, EXPORT, FONT, GALLERY, IMPORT, OUTPUT, PALETTE, SYMBOL, and TUTORIAL.

The contents are evident from the names.

After installation, all you have to do is type HG3 and the program will load. The program is bigger, so it will take a little longer than the earlier versions of HG to load. During the load, HG3 will check to see if a mouse is installed. If not, a prompt

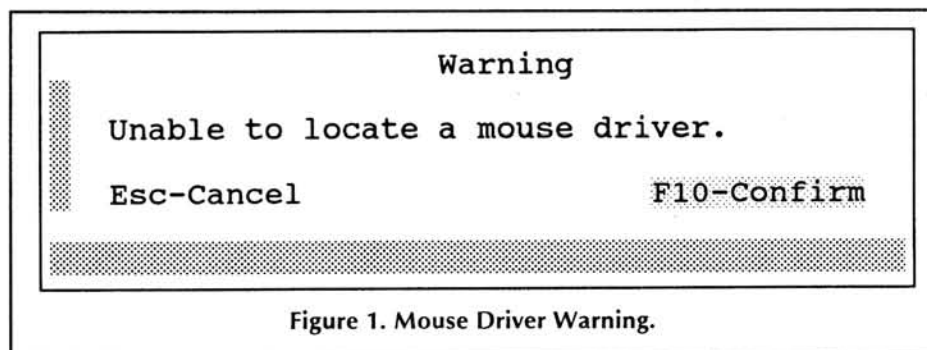


Figure 1. Mouse Driver Warning.

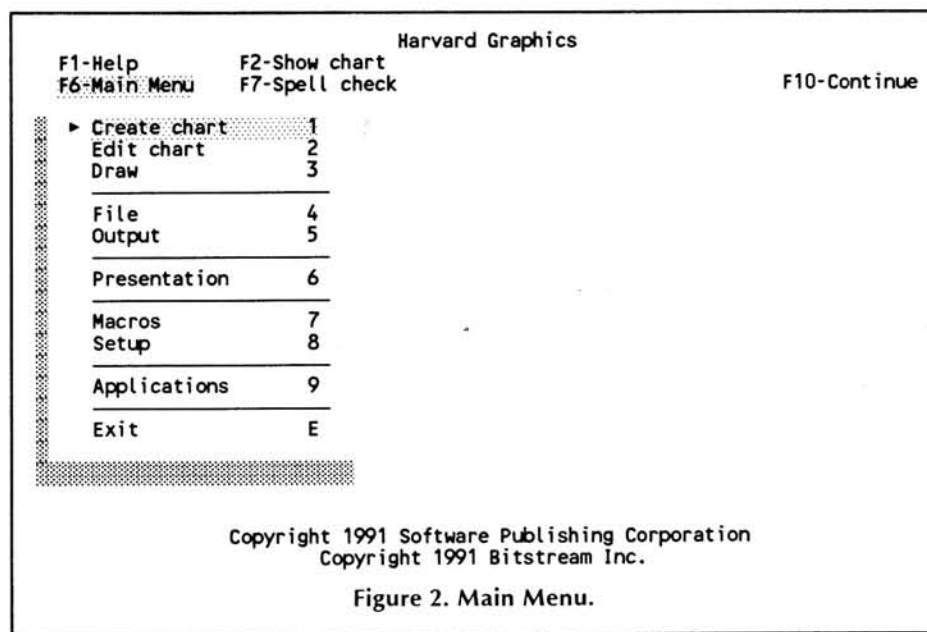


Figure 2. Main Menu.

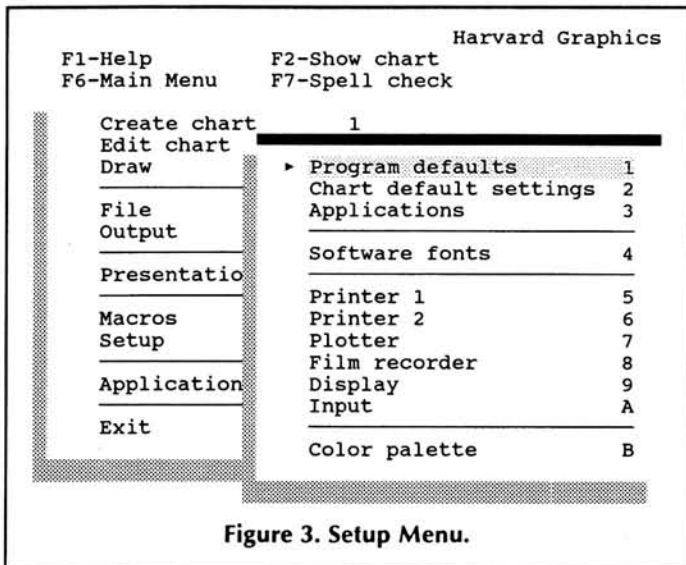


Figure 3. Setup Menu.

will be displayed (Figure 1). Pressing F10 will continue the load. This is a convenient stop if you do not load your mouse driver from the AUTOEXEC.BAT file. You can operate the program without a mouse, as long as you are creating text or data charts without graphics. During the loading, HG3 will check to see what type of graphics card you have. It will then configure itself to use the correct video driver. When the program loads, the first menu looks very similar to those of earlier versions.

Like earlier versions of HG, options from the menu can be selected by moving the cursor over the option with the mouse and pressing the left mouse button, typing the first letter of the option or its number, or moving the highlight bar with the cursor keys over the option and pressing <ENTER>. The available options have changed slightly. Two new options have been added, Applications and Macros. One of the options in Applications permits you to leave HG3 and return to the DOS shell. Typing EXIT from the DOS shell will return you to HG3.

Before we create our first chart, let's explore the Setup Menu (Figure 3). The Setup Menu permits you to set up the options that are used throughout the program. The first option, *Program defaults*, permits you to define directories for the various files used by HG3. You can also define the palette and the use of expanded and/or extended memory if available. The next option, *Chart default settings*, permits you to set the font, orientation, and other formatting options, such as date and time formats. The Applications option permits you to define the external programs you wish to run.

HG3 now supports software fonts. Included with the package are several fonts provided by BitStream. The Software Font option (Figure 4) permits you to view the available fonts. Additional BitStream fonts may be added.

The next two options are used for printer selections (Figure 5). HG3 has added drivers for several more printers. This option permits you to select two printers and define how they are connected to the system. Another new capability of HG3 is a built-in print queue. In earlier versions, you could not work on a chart or start another while printing. HG3's queue now allows the printing to be done in the

This is where you can define the 64 colors you wish to use out of the 16 million colors available.

Now that we have gone over the basics, let's jump right in and create a few charts (Figure 7) and see what improvements will appear.

Selecting option one from the Main Menu displays a second menu. This menu lists all the types of charts that can be created using HG3. The first option is the text charts. (Figure 8) Pressing <Enter> or clicking on this option with the mouse will open another menu displaying the type of text charts that are available. Three types of text charts can be created with this option: Title, Bullet, and Table.

While the total number of text chart types has been reduced from earlier versions, the functions are still there along with some new capabilities. The Title Chart (Figure 9) can be used for many charts. Unlike the earlier versions, the number of lines is now 11.

Each of the three defined areas can be assigned attributes (Figures 10-12). To create a simple list, you would add more lines of text in the middle and bottom area and make the attributes the same. A user can access three pages of mode functions available by pressing the F8 key for the Options Menus.

background. A very welcome addition.

The next option permits you to define the plotter. The plotter support has also been improved in HG3. The next option is the Film Record option. If you plan on sending your files to a film recorder, make those option choices here.

The last option is Palette (Figure 6). With this option, you can define the palette you wish to use or make your own palette.

One new feature that will please HG3 users is the text overflow procedure. In earlier versions of HG, you were limited to 60 characters per line. In HG3, the text will shrink-to-fit to the available chart size.

The next type text chart is bullet. This chart is the basic one used to create larger Text Charts. You can turn off the bullets to create a basic list chart. Six different bullet types can be selected. The attribute selection is similar to the Title Chart discussed above.

HG3 will automatically generate charts that build to show progression. One example is a bullet chart. Using the auto-build capability, HG3 will create three separate charts if you have three points. When displayed as part of

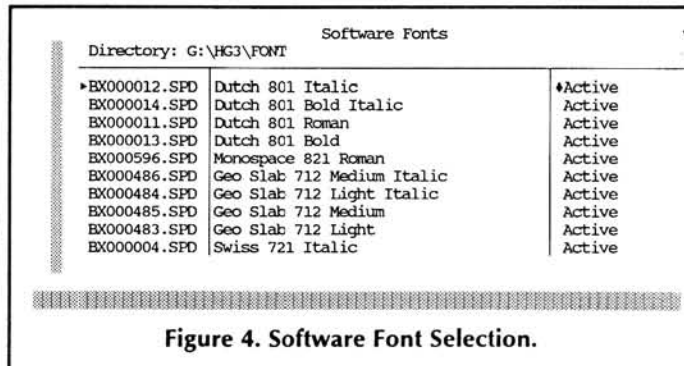


Figure 4. Software Font Selection.



Figure 4A. Sample of Fonts.

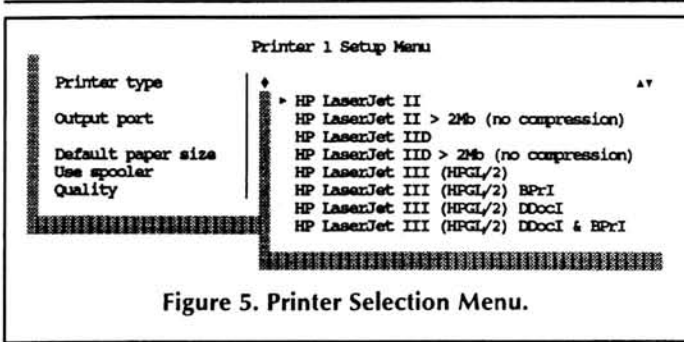


Figure 5. Printer Selection Menu.

a screen show, the transition is smooth.

The last text type is Table (Figure 13). One of the questions I received while instructing HG was "What if I need more than three columns in a chart?" The older versions of HG limited you to three columns. In HG3, this limitation has gone away. You can now create Table Charts (Figure 14) with up to 24 columns with 24 rows each. Each row can contain up to 50 characters. The size of the characters will be smaller with each added column, so this must

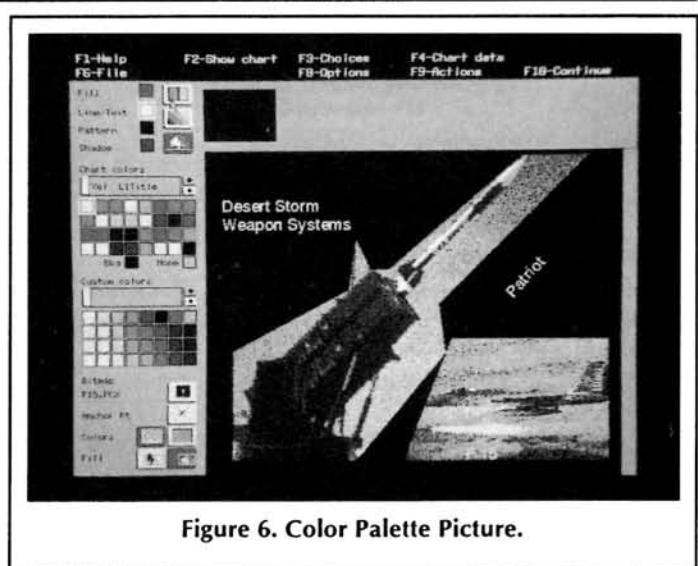


Figure 6. Color Palette Picture.

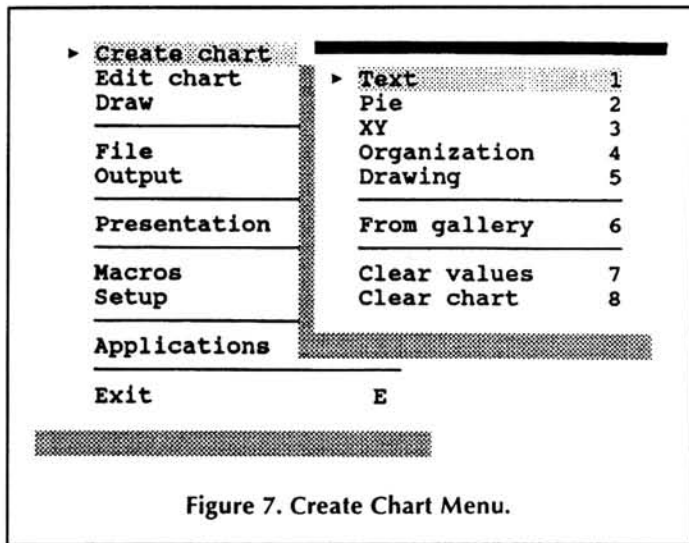


Figure 7. Create Chart Menu.

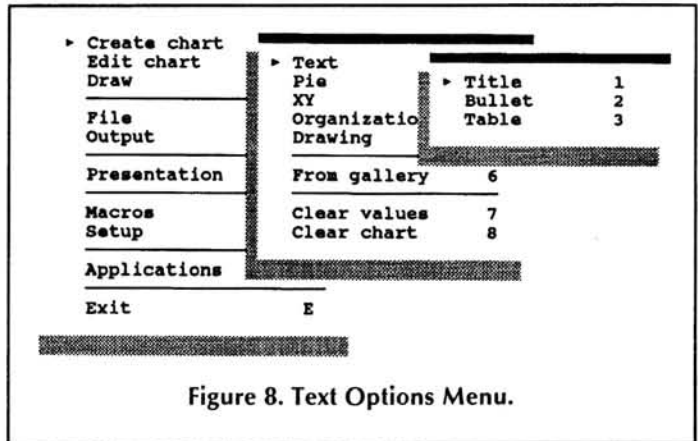


Figure 8. Text Options Menu.

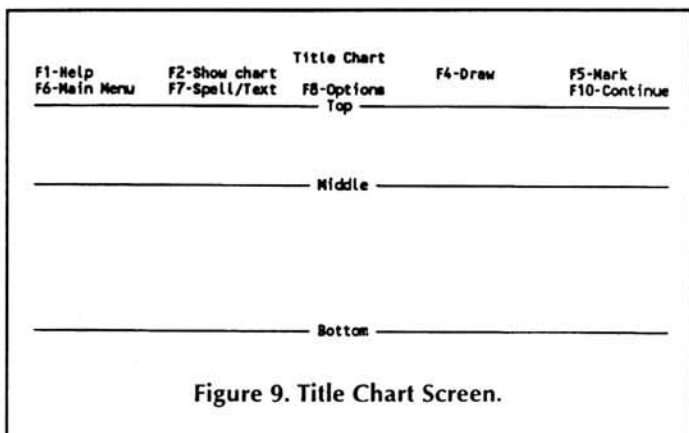


Figure 9. Title Chart Screen.

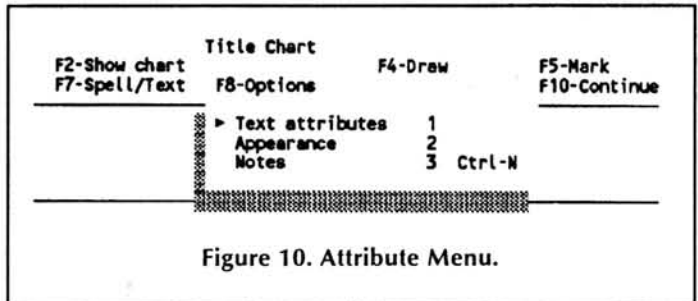


Figure 10. Attribute Menu.

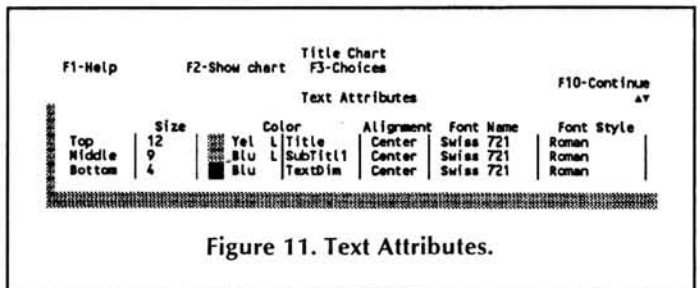


Figure 11. Text Attributes.

be considered when making this chart. The text will be centered on the available chart. All of the normal attributes are available.

The next type of chart available is the Pie Chart (Figure 15). One of the questions that always came up in class is "Does HG allow you to cut slice 3-D pie charts?" The older version of HG did not do 3-D cut slice (Figure 16), although it was possible to create one using the drawing capability. HG3 now supports 3-D cut slice. The Pie Chart function has also been improved in that you can now create and display six Pie Charts (Figure 17) on one chart. It

provides some unique capabilities. The pie chart data can be displayed as a pie or column chart. It is possible to link the data between the parts to use the second pie/column to expand data in the main pie.

The last major group of data charts is the XY group (Figure 18). Within this group are eight types of charts (Figures 19 and 20).

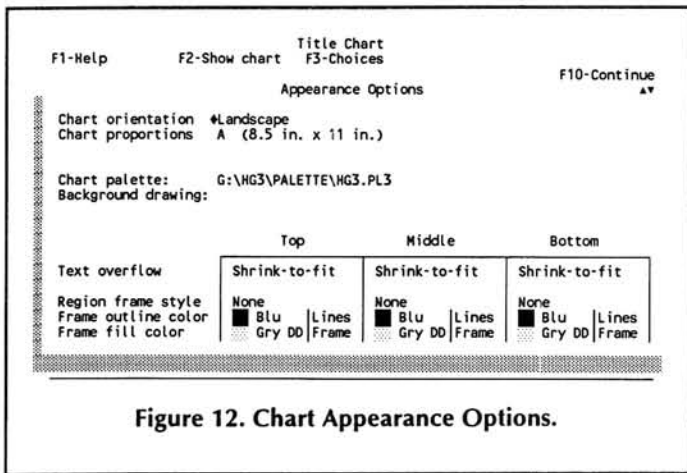


Figure 12. Chart Appearance Options.

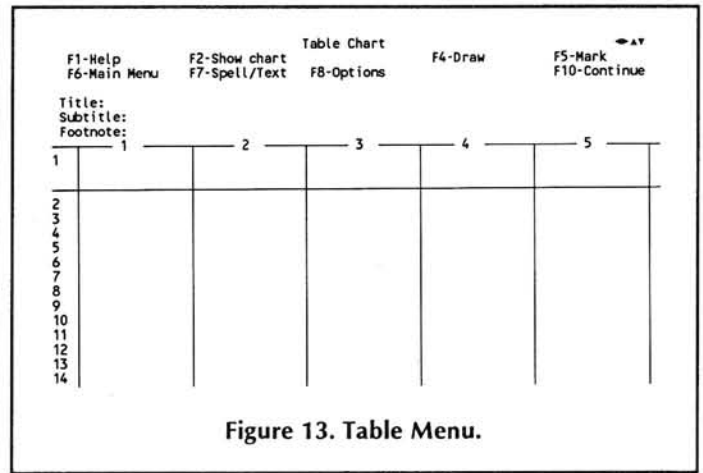


Figure 13. Table Menu.

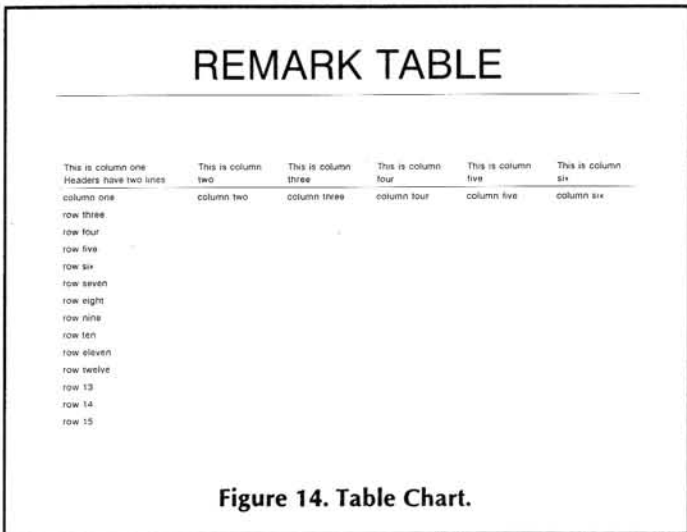


Figure 14. Table Chart.

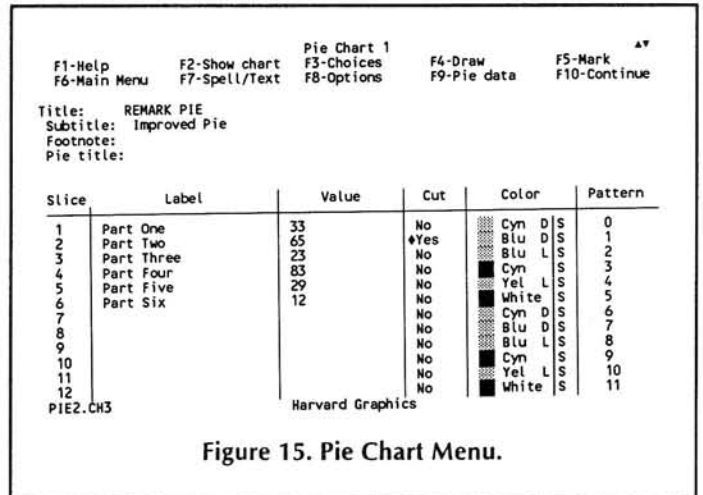


Figure 15. Pie Chart Menu.

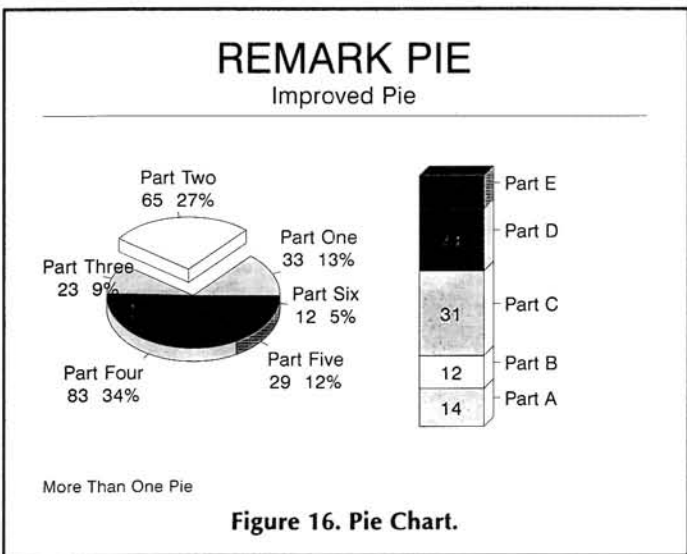


Figure 16. Pie Chart.

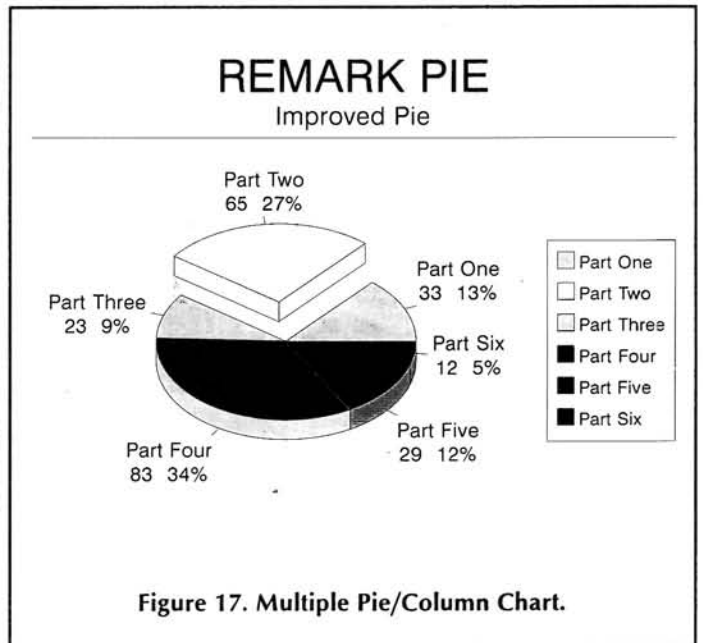


Figure 17. Multiple Pie/Column Chart.

These charts are basically the same as HG with the new menus and color options. You can use the data from one chart and create different types of charts. Using this capability, you select a chart that will get the point over in your presentation. These charts can contain up to 16 columns of data and several hundred rows, if necessary. Remember, if this data has to be displayed, too many points may overwhelm the audience.

Another complaint during class was the lack of a good Organizational Chart (Figure 21) capability. During class, many people would exceed the number of boxes permitted and nothing would be displayed. HG3 has improved the Organizational Chart to a real ORG Chart (Figure 22) capability. You are now limited to

F1-Help	F2-Show chart
F6-Main Menu	F7-Spell check
<ul style="list-style-type: none"> ▶ Create chart Edit chart Draw File Output Presentation Macros Setup Applications Exit 	<ul style="list-style-type: none"> Text ▶ XY Organization Drawing From galler Clear value Clear chart
	<ul style="list-style-type: none"> Bar 1 Stacked bar 2 Overlapped bar 3 Bar/Line 4 Line 5 Area 6 High/Low/Close 7 Point 8
	E

Figure 18. XY Choice Options.

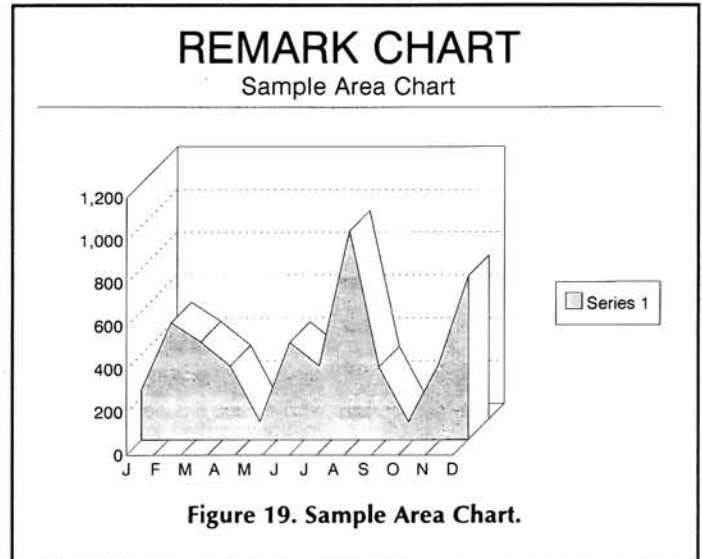


Figure 19. Sample Area Chart.

Organization Chart

F1-Help	F2-Show chart	F4-Draw	F5-Mark
F6-Main Menu	F7-Spell/Text	F8-Options	F9-Organization F10-Continue

Title: REMARK ORG CHART
 Subtitle:
 Footnote:

Name: Lower Boss Title: Boss 2b Comment:

Figure 20. Organizational Chart Building Screen.

Organization Chart

F1-Help	F2-Show chart	F4-Draw	F5-Mark
F6-Main Menu	F7-Spell/Text	F8-Options	F9-Organization F10-Continue

Title: REMARK ORG CHART
 Subtitle:
 Footnote:

▶ Add/Edit box text	1	Ctrl-Z
Add subordinate	2	Ctrl-Ins
Add staff position	3	
Move box/group	4	
Switch left	5	Ctrl-L
Switch right	6	Ctrl-R
Delete box/group	7	Ctrl-Del
Undelete	8	
Set top of chart	9	
Hide subordinates	A	Ctrl-H
Show subordinates	B	Ctrl-U

Name: Title: Comment:

Figure 21. Organizational Chart Options.

210 boxes. Adding, deleting or modifying the chart is very easy. You can see what the chart will look like during the construction. To add information to a box, simply move the cursor to the box and press <Enter>. HG3 will display the contents of the selected box in the upper right corner of the screen.

You still have all of the enhancement capabilities. These are available from the Top Line Menu by either pressing the function key or moving over it with a mouse controlled cursor and selecting.

Included with HG3 is a gallery of charts. These are pre-built baseline charts of all types. If you want to create a chart based on one of the baseline charts, select *Gallery*, and then select the type chart you would like to create. This chart is brought up and then you must replace the generic data with your own data. This is a quick way to create charts without having to go through all of the menu selections.

The most improved area of HG3 is the Drawing function. As a temporary fix, Software Publisher provided Draw Partner to assist in making complex drawings. This was a separate program that you had to run outside HG and then import the results. In HG3, this capability and much more has been incorporated into the basic program.

While the basic drawing capability was acceptable (if you learned how to use it), a lot was left to be desired. I learned to get around in this area and created over 500 symbols that were provided to the classes I taught. Some of these were complex, such as a three color tiger head. With the new capabilities, these symbols could have been created with less effort. The additional colors would make the results even better. As with HG, HG3 drawings can be added to any chart.

HG3 will permit you to import and use .PCX files. In HG, I created a presentation with PCX files that were used as underlays

Harvard Graphics

F1-Help	F2-Show chart
F6-Main Menu	F7-Spell check

Create chart 1

Edit chart

Draw

File

Output

Presentation

Macros

Setup

Applications

Exit

E

▶ Get chart	1	Ctrl G
Get template	2	
Apply template	3	
Save chart	4	Ctrl S
Save as template	5	
Save as symbol	6	
Import	7	
Export	8	

Figure 22. File Menu.

for charts. This worked OK, but the 16 color limitation made these images less than fully desirable. With HG3, you can even print the PCX files used in the charts. As an example, I scanned a page into the system as a PCX file. This image was then printed as part of a chart with excellent results.

Software Publishing Corporation has included over 500 symbols with HG3. Included with these are several animated



Photo 1. Patriot Missile.

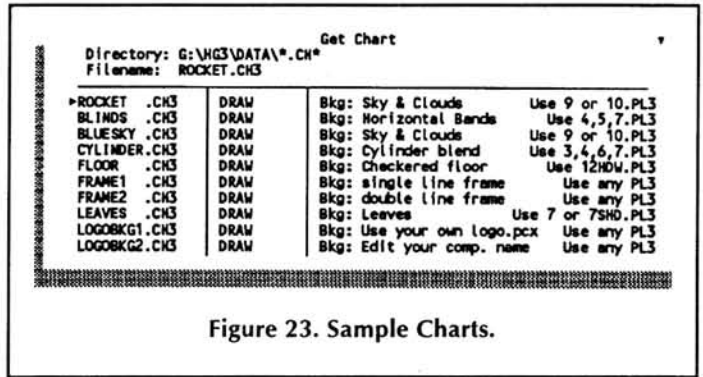


Figure 23. Sample Charts.

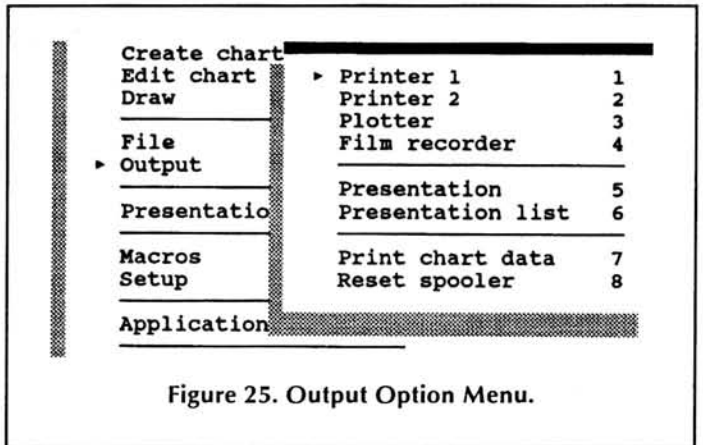


Figure 25. Output Option Menu.

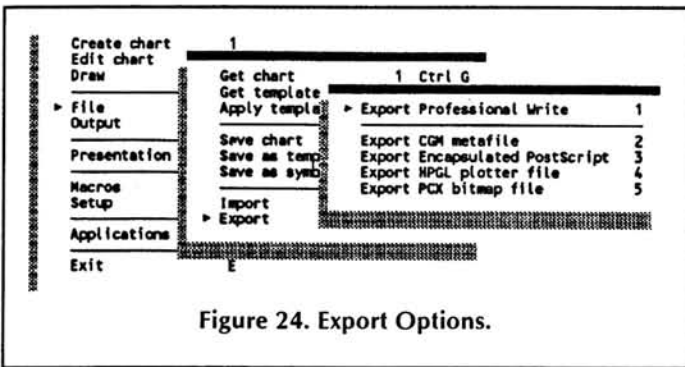


Figure 24. Export Options.

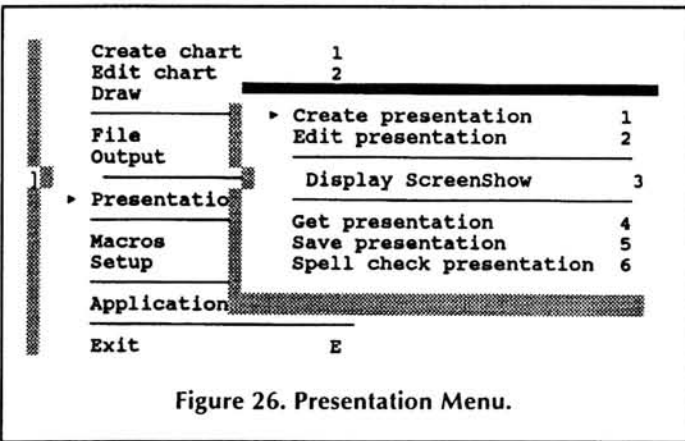


Figure 26. Presentation Menu.

symbols which will move across the screen when the chart is displayed. They are real eye openers when included as part of a screen presentation.

I highly recommend a mouse when you work with the Draw option. I am not a great fan of a mouse unless I have to draw. Moving from the keyboard to the mouse and back again seems like a waste of motion. When I draw, I find the mouse very important. HG3 uses the mouse much better than HG. When you move into the draw option, you will immediately notice the new menu structure and capabilities.

Rather than the one line of menu options, HG3 now displays two lines with

now open a window below the font/size option where you will type in the text. Pressing [ESC] will display the text in the selected location. To move the text, position the cursor/arrow over the text and press the select key. The text block will be defined with six small blocks. Holding the left mouse button down, you can move the block to the new location.

With the options on the menu, you can mark and delete, duplicate, copy, align, space, rotate, skew, and hide objects. You can flip objects either horizontally or vertically. With the second column of options, you can create objects. Boxes, polygons, and circles/ovals are just some of the ob-

jects that can be created. PCX files are selected from this options column. Any of the created or selected objects, including PCX files, can be fit into created shapes. Note the Patriot missile in the polygon shape shown in Photo 1.

On the second page of options are some unique capabilities. One of these is Evolve Chart. This options permits you to evolve one shape into another using user-defined increments. I used this option to evolve a chicken from an egg (oval). The Draw options are fun but they do take a little time to master.

The next option on the Main Menu is File Menu. This option is used to save and retrieve charts. Note that speed key options are now available. Ctrl-G will open an existing chart without having to move through the menu structure.

One of the new functions that is available with the Get Chart option is Sort. As you create charts, they are inserted into the directory in random order. If you have many charts, searching through this list can be time consuming at the very least. When you select Sort, the charts are displayed alphabetically.

Note in the figure that Sample Charts (Figure 23) are included with the program. You can select one of these charts as a background for your work. One example is the sky background. This is a sky blue background with added clouds. You can create similar backgrounds for use in your presentations as required.

Continued on Page 36

Adding Expanded Memory to Your PC

The Everex 3000 Delux AT Memory Board

Dan Jerome
801 E. 132nd Street
Burnsville, MN 55337-3870



With the progressive waves of increasingly large programs for our PC computers, it is necessary for the user to increase his computer's storage capacity just to keep current. Those of you who have purchased PC Tools V.7 or other contemporary programs will know exactly what I mean.

This article was written for computer users who have computers with the motherboard filled "to the gills" with DRAM chips for a total of 1MB, and who are looking for a means of adding EXPANDED memory to handle these extremely large programs, such as a desktop publisher, word processor, and related programs.

For the sake of the uninitiated, memory may be categorized in three classifications: 1) Base memory — that memory from zero to 640 k, 2) Extended memory — memory above 1 megabyte that can only be accessed by a '286 or higher processor in the protected mode, and 3) Expanded memory — memory above 1 megabyte accessible on any computer using an EMS device driver. This article deals with expanded memory, but on a board designed for use in AT ('286) computers.

I have a CompuAdd AT with a 286 processor, EMS, three floppy drives, and an enhanced EGA monitor. It has a large case, since when I purchased it I envisioned having more than one hard drive and certainly more than two floppy drives. I have been happy using this computer, but there came a time when it was evident that I needed to purchase additional EXPANDED memory. My computer is two years old so it is fairly primitive. It can only handle discrete DRAM chips instead of SIMS, etc. ("Discrete" means individual parts.)

When I contacted Henry Fale of QuikData, Inc. on the phone, he recommended a RAM 3000 Deluxe, AT Memory Expansion Board for this purpose.

The unit arrived in 4 working days. For reasons of economy, I decided to order only 1MB of 80 ms 256k DRAM chips. Although the unit can hold 3 MB I felt that, if necessary, I could always add more DRAM

later. Having received my "computer technician's certificate" as a Heath H-89A graduate in the school of hard knocks from 1981 to the present, I felt no compunction about outfitting my Everex RAM 3000 board and installing it, having done several such installations in the past.

Prior to opening the package, I reached out and touched my grounding wire so that I couldn't transmit any static electricity to the unit and thereby "kill" the chips. I feel that this is always a very necessary caution.

The Everex 3000 Deluxe AT Memory Expansion Board measures 13-7/8 inches long by 5 inches high, measuring from the tip of the card edge connector. It requires all of the space from the front of my computer to the back. To guard against possible shorts between the board and the computer case, I decided to cover the top edge with electrician's tape so Murphy's Law couldn't take effect. The object was to prevent a short from the top of the board to the computer case. As it turned out there was about 1/8 inch clearance between the top of the card and the computer case.

The first thing I did was to bend the RAM chip pins to match the spacing of their sockets. This consists of first touching your grounding strap to relieve any static charge, removing a chip from the insulated foam, and laying it on its side on the work table, preferably on top of a soft cloth. To bend the chip leads, you simply push the leads down all at once on the hard surface of the table. As you complete one chip, inspect it to insure that the pins are all bent the same and that no pin is cockeyed. In this manner, you can install each of the chips. This operation will take a while, but you will be more than adequately compensated for your time.

Next, examine the circuit area where the chips will be installed for obstacles that you will have to work around. If any obstacles are present, I recommend you begin installing the first set of chips in the sockets right next to the obstacle. If you don't do this, you may be forced to remove

some of the chips near the obstacle later to provide room to install chips next to it. Chances are the last chips to be installed will be crowded out.

If you need to remove a chip, it is always best to use the proper tool. If you don't have a chip puller, you can resort to the "field expedient," as my old US Army sergeant used to say. In this case, you can use a tiny flat-bladed screwdriver with a blade that will just clear the pins without shorting them. Then, CAREFULLY pry the chip straight upward until it comes loose. You shouldn't have any chip just lying by itself on the work table. Best technician practices call for inserting all chips into insulating foam — even for temporary storage.

Make sure the computer is turned off and the power cord unplugged. Then remove the computer cover. Assuming the EMS board has been populated with the DRAM, simply position the two jumpers and three dip switches. Then carefully install the EMS card into a full-sized slot on your motherboard. Push it into place, but do not permanently screw it down with the holding clamps.

Now insert the Everex software disk into a 5-1/4" disk drive, turn on the AC power, and boot the computer. Create a new subdirectory named "EVEREX." Now run the INST159.EXE program to copy all of the Everex files to the directory you just created on your hard drive. Shortly, you will see a menu on the screen covering the EV-159 Utility Program. This menu will provide three options:

- Memory configuration and testing
- Install EMS driver
- Exit to DOS

Select the "Memory configuration and testing" option. Although you can't test memory at this time, you will see a data screen on your monitor. It will look something like Figure 1.

The next task is to check the responses and insure that they are correct for your specific situation. Note the default data al-

```

Configure Memory EV159 1.02 F1=HELP F10=Exit
+-----+-----+
| Amount of Existing Base Memory (Multiple of 128k) | 640k |
+-----+-----+
| Amount of Existing Extended Memory                | 1024k |
+-----+-----+
| Amount of Base Memory to Be Added                 | 0k    |
+-----+-----+
| Amount of Extended Memory to Be Added             | 0k    |
+-----+-----+
| Amount of EMS Memory to Be Added                  | 3072k |
+-----+-----+
| EMS I/O Port Address                              | 258H  |
+-----+-----+

+-----+-----+
| EMS Fast Mode                                     (Y/N) | N    |
+-----+-----+
| Speed of DRAMS is 120 Nanoseconds or Faster (Y/N) | Y    |
+-----+-----+
| Enable Parity Checking                           (Y/N) | Y    |
+-----+-----+

+-----+-----+
| / \
| | | Move Cursor:F2 Config/Test Menu:F4 Load Config:F8 Save |
| | |
| \ /
+-----+-----+

```

Figure 1

ready residing in the "response" areas. For example, reading from the top you will see: 640k, 1024k, 0k, 0k, 3072k, and 258H. For your convenience, I'll repeat the definitions of the terms used:

Base Memory is the memory from zero to 640k in your computer.

Extended Memory is the memory above 1 megabyte.

Expanded Memory is any memory above 1MB accessed via an EMS driver.

Expanded memory pertains to the amount of memory that you will now be adding. For example, if you are adding 1 megabyte of memory, type 1024k. If you are adding 2 megabytes, type 2048k. Finally, if you are adding 3 megabytes of memory, type 3072k—or select the default value.

The instructions for using this menu are at the bottom of the data box. For example, the up and down arrow keys are used to move the cursor up and down in the data table. F1 is HELP, F2 is Configure/Test Menu. F4 is Load Configuration. F8 is Save Configuration.

The object of this part of the install program is for you to change the data in the data table so that it applies to your specific system. In my case I was only adding 1MB of EMS memory, so I changed the appropriate figure from 3072k to 1024k. That was the only entry that required changing.

The second part to the data table consists of three options. The first is: EMS Fast Mode (Y/N). The second is: The Speed of DRAMS is 120 Nanoseconds or Faster: (Y/N). The third is: Enable Parity Checking (Y/N). I went with the defaults and these were, in succession, N, Y, Y. This is a good choice, but if your DRAM is slower, indi-

cate "N" in the middle block.

If you have any problems, press F1, but if you are done, DO NOT press F2, or you will be directed to pass through the data table cycle again. Instead, press the F8 key to save the configuration. Name the resulting file EV159.001. Press <ENTER> to save this configuration.

Now the fun part, press F4 and a graphics picture of the card will appear on your screen. It will tell you exactly where to set the two jumpers and three dip switches to let your computer know it now has EMS memory. This is where EVEREX shines! This display technique is not only an example of beautiful graphics, but it is completely fool-proof and easy to use! No other supplier of EMS cards has this neat and unique feature! Using pencil and paper, make a note of the settings as shown in the picture. Then press the F2 key and you will be returned to the initial menu. Select "Install EMS Driver." You CONFIG.SYS file will be modified. Now exit the Utility Program and turn off your computer (be sure to park your hard drive, if your hardware requires this). Then once more unplug the AC line cord.

To set the jumpers, first carefully remove the EMS card from your computer and lay the card down on a soft cloth on your work table. Referring to your notes, use small needle nose pliers to move the jumper blocks. To set the dip switches, use a tiny flat-bladed screwdriver—or in a pinch, try a heavy duty paperclip. The dip switches have two possible positions: 0 for "off" and 1 for "on." Be certain to set the switches right the first time. Then check your work to insure that you have done it correctly. We don't want any errors at this

late stage of the game. Then install the EMS card into your full-size slot PERMANENTLY! Do not yet put on the computer cover. You can do that later after you are done playing with your new accessory. When you do, make sure you first have your hard drive parked (if you hard drive unit requires it). If you move your computer or reinstall the cover without parking the hard drive, you can cause bad spots to appear on the hard drive disk surface.

Plug in the AC power and turn on your computer once more. Take a few moments to run your Memory Configuration/Test in INST159.EXE and check the validity of your DRAM. This test should come out just fine. At least you will have the peace of mind that all is well and no mistakes have been made. When you are satisfied that all is up to snuff, go to MS-DOS and call your CONFIG.SYS into an ASCII editor. Then add a "DEVICE=filename.sys" statement to the file to tell your system which memory driver to use. This driver will handle your Expanded Memory, when you run a DRAM-hungry program, if you observe it carefully. Most of the operations to EMS Memory are user-transparent, meaning that the typical user will not notice them.

Now that you have successfully completed the hardware installation of your EMS card, the final step is to consider two additional utilities that Everex provides. These are a RAM drive and a hardware spooler for your printer.

I think these are real neat options! Since I began with MS-DOS computers I always wanted a hardware spooler for my MS-DOS clone to be analogous with HDOS' "SP.DVD" written by Jim Tiexeria. The purpose of this is so that I could save precious time by continuing to write while my printer is dutifully chugging out my last document.

To install a printer spooler, put the following phrase in your AUTOEXEC.BAT file toward the end of the file:

```
ESPOOL /D:256 /X /P:1
```

This creates a 256k spooler in EMS memory, with the files it handles spooled out to parallel port LPT1. If you skip the 256, a default value of 64k is used.

To install a RAM disk, put the following phrase in your CONFIG.SYS file:

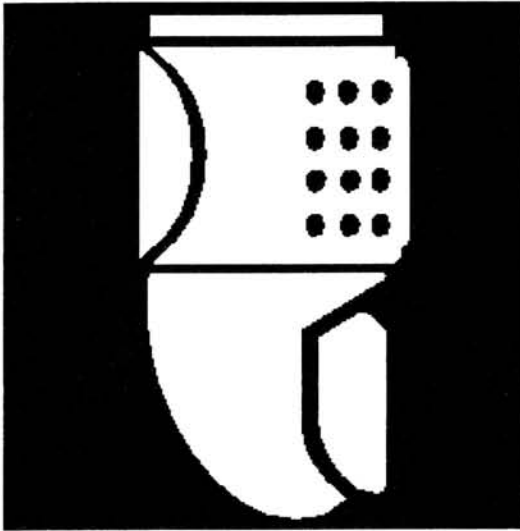
```
DEVICE=C:\EDISK.SYS 256 512 /X
```

This creates a 256k RAM disk with a sector size of 512 bytes, and tells your computer to set it up in EMS memory.

To activate the RAM disk and the spooler the first time, all you have to do is warm boot your computer. Subsequently, each time you boot your computer the RAM disk and the spooler are recreated and enabled.

In case you want to deactivate your RAM disk, just put your CONFIG.SYS file in a text editor and type "REM" (ignore the quotes) between the expression we dem-

Continued on Page 48



First Aid for Writing

Mark Haverstock
6835 Colleen Drive
Youngstown, OH 44512

Several Slick Programs Can Take Your Ailing Documents Off the Critical List

You don't have a second chance to make a first impression. What kind of first impression does your writing give?

Eventually, you're going to write something, whether it's a memo to the boss, a resume or a letter to your Aunt Gladys. Simple errors can make the difference between someone reading your writing seriously or pitching it in the trash. An uncertain grasp of English can become a handicap for even the brightest person.

If you use your Zenith Data Systems or other PC-compatible for word processing, there are several useful programs that can help you with your written expression. Spelling and grammar checkers can help make your documents more readable, and weed out those pesky errors you might overlook.

Remember that these tools act as advisors, not referees. You must decide whether to make the changes they suggest. Generally, these programs go strictly by the book. Some of their suggestions may not be appropriate for the context or style of your writing. We'll discuss some specific examples later in the article.

Spelling Checkers

Most word processing programs contain some kind of spell check function. After you finish your document, you press a key or combination of keys and the checker goes to work, pointing out suspected errors. The key word here is *suspected errors*. To understand why, it's best to know how the checking function works.

A spelling checker does the tedious job of checking each word in your document against its own dictionary (Figure 1). It goes to the first word and compares that word to its dictionary list. If the word appears in the dictionary, it moves on to the next word. The process repeats itself until it finds a word not contained in its dictionary. Anything that isn't in the dictionary is flagged as an error.

Once it detects an error, the checker will offer several alternatives. These are usually based on sound-spelling patterns—ones similar to the word that's in question. You usually have three options: 1) choose one of the alternatives as a replacement, 2) edit the word (if you know the spelling), or 3) ignore the word and move on.

Check Again

All words the spelling checker flags aren't necessarily errors—they're just words that it doesn't recognize for one of several reasons.

Proper names. If a name isn't listed in the dictionary, it probably will show up as an error, though the spelling may be correct. For example, in the sentence "Dan Quayle is our Vice-President," **Dan Quayle** would be considered an error. Despite your own political beliefs, at least the spelling is okay.

Abbreviations. Most spelling checkers will recognize common abbreviations, but not all. The spelling checker in Word-Perfect recognized FBI (Federal Bureau of Investigation), but not FFA (Future Farmers of America).

Foreign words. Phrases like "deja vous" may be in your speller's dictionary, but "glasnost" may not.

A spelling check won't reveal all problems in a document. Look at the following sentence:

I ate too donuts for breakfast.

Remember to think like the spelling checker. Would the word **too** be contained in the dictionary? Yes, so it won't be flagged as an error. But common sense tells us we've used the wrong word. This brings us to another class of words the spelling checker programs fail to find.

Homophones (homonyms). These are the words that sound alike, but have different spellings and meanings. Word combinations like *too/to/two* and *there/their/they're* are often confused and used in the wrong context.

Of course, the word **two** would be the correct word in the situation presented earlier. You'd need to edit this manually using the word processor.

Here are some other mistakes spelling checkers will overlook. Read the sentences below:

Lenny are the best runner on the squad.

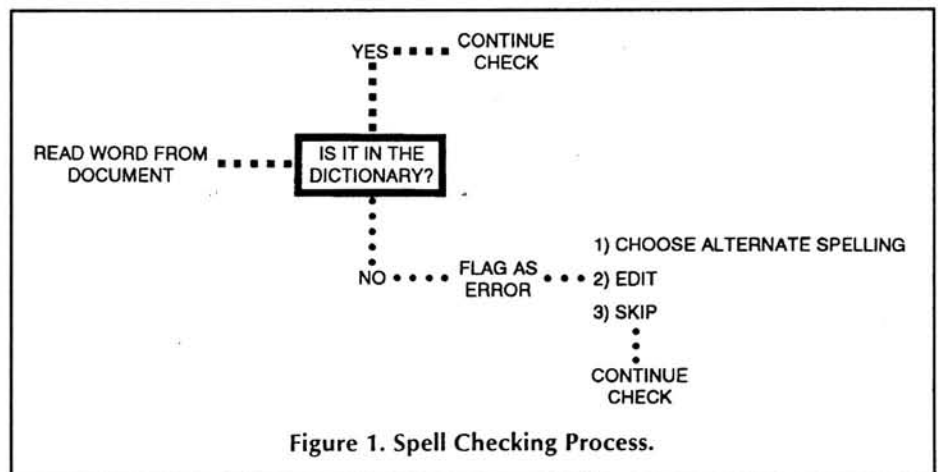


Figure 1. Spell Checking Process.

Ellie and her figured out the answer.

Again, all the spellings are correct, but some words don't seem to fit. They're not the right choices in this context.

Agreement. Words may not fit because they don't match in person, case, number or gender. Since Larry is a singular subject, the correct wording would be "Lenny is." In the second example, "Ellie and she" would be the best choice since both are subjects (nominative case).

There are two other considerations when using spelling checkers. They won't detect these errors:

Words in wrong order/missing. A nonsense sentence like "His name Fred is" will simply be checked for spelling - nothing more. An incomplete sentence will get the same treatment.

Punctuation. Most spelling checkers have no provisions for finding errors in punctuation or even verifying if punctuation is present in your sentences.

Spelling checkers do a respectable job in most circumstances, but don't be lulled into a false sense of security. They have their own limitations and shortcomings. Think of them as a silent partner to give your documents the once-over. So be prepared to do a second reading yourself, or enlist the help of another class of programs - grammar checkers.

Grammar Checkers

Do the grammar gremlins get you each time you sit at the word processor? Is your writing torn apart by split infinitives and misplaced commas? A grammar checker could be a cure for your grammar and style problems. These programs won't turn you into an instant Shakespeare or Hemingway, but they will help turn your thoughts into good, organized prose.

What does a grammar checking program do? First, it creates a copy of your original document and saves it to disk. Next, it analyzes the original for grammatical and stylistic problems, such as punctuation, passive voice, incomplete sentences, cliches, unpaired parenthesis and quotations, split infinitives, and excessively long sentences (like this one!) A short comment explains the problem and usually suggests alternatives. You can then make correc-

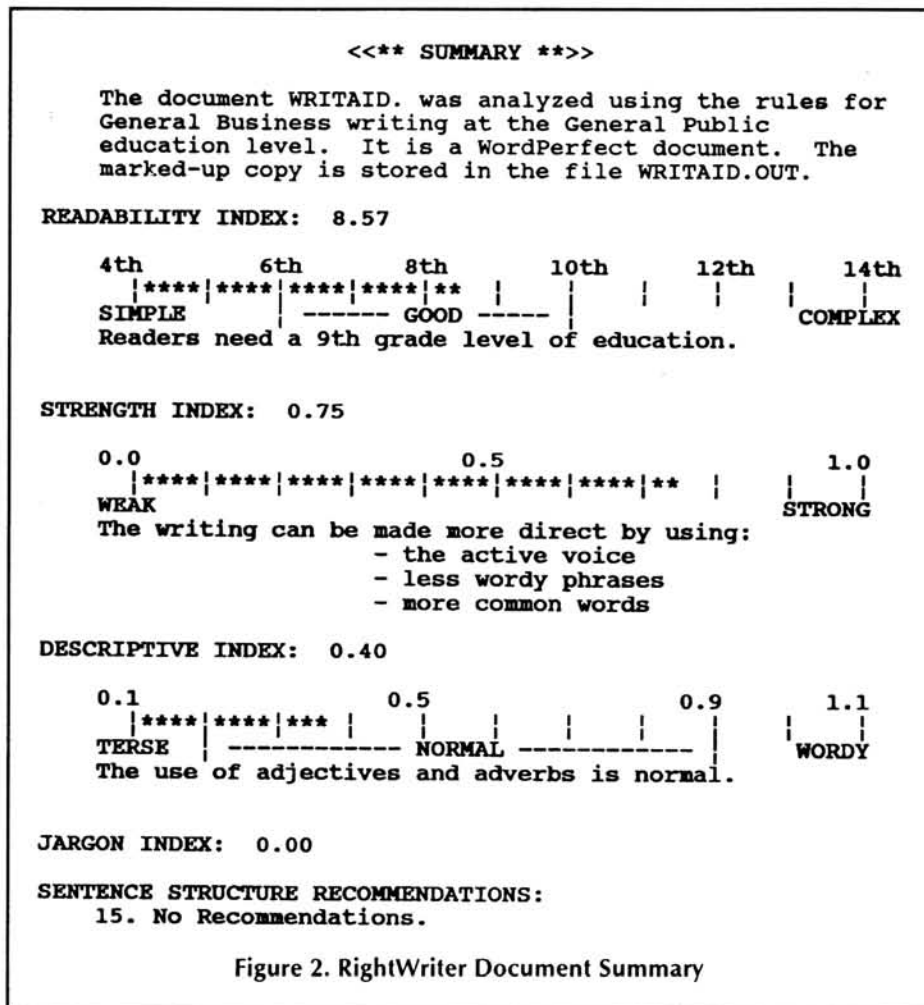


Figure 2. RightWriter Document Summary

tions, if necessary.

In addition, these programs will determine readability with indexes that give you a general idea of how easy your document is to understand. These indexes are based on sentence length and the number of multi-syllabic words per sentence. Most grammar checkers use three readability indexes: Flesch-Kincaid, Flesch Readability and Gunning's Fog Index. These measure by using grade levels or ease of reading on a scale of 0 to 100.

By watching the readability levels, you can better tailor writing for your audience. Most adults, for instance, prefer reading material written at the 7th-9th grade level.

If your readability levels come out higher, it would be wise to simplify your writing by using shorter words and sentences.

Caution!

English is a complicated (and messy) language, so a few words of caution are in order. These programs are not the same as a human editor. They don't understand the actual meanings of words in context, nor can they judge how the writing sounds.

Consider these two sentences: "I'll record tonight's concert," and "I hope to set a new record." You shouldn't have any problem figuring out the meaning of **record** in each sentence. However, the computer could have a rough time making sense of them.

Since grammar checking programs can't deal with exceptions or ambiguous rules, they may flag correct phrases as errors or miss incorrect ones. This is the reason they won't automatically make the changes for you. Instead, the program will mark the phrase in question and give a recommendation. You'll decide if changes need to be made.

Of course, grammar checkers may not be as intuitive (or picky) as your high-school English teacher. But they can be an indispensable writing tool to help you com-

Table 1
Hot-Key Support

RightWriter

Microsoft Word 4.0, 5.0
Multimate Advantage II
pfs: Professional Write 2.0
thru 2.2
WordPerfect 5.0, 5.1
WordStar (except Wordstar 2000)
Q&A 3.0
Q&A Write 1.01, 1.02
LetterPerfect

Grammatik

Microsoft Word 4.0, 5.0
Professional Write 2.1, 2.2
WordStar 5.0
WordPerfect 5.0, 5.1
XyWrite III

municate more effectively. Use them as a guide, not the final word, in your writing.

One final word of advice – run your spell check **before** you do a grammar analysis of your document. The words need to be spelled correctly so the grammar check will be as accurate as possible.

Using Grammatik and RightWriter

Grammatik and RightWriter are the two most popular and well-established grammar checkers on the market. Both are compatible with files from almost all major word processing programs in their native format. They'll also accept files in ASCII format. To begin, you simply load the grammar checker and specify the file you want to analyze.

You also can start your check within your word processor. One keystroke is all it takes to access these programs from within several popular word processors. This means you can move from typing your document to checking it and back again – without loading a separate program. Table 1 includes a list of word processors supported by a "hot-key" combination. Since the list frequently changes with product upgrades, it's a good idea to check this feature before you buy.

RightWriter begins by analyzing your entire file, inserting comments when it finds a problem. A comment explains the problem, and often suggests an alternate word or phrase. Next, RightWriter creates an output file with the comments, called a marked-up copy.

At the end of the marked-up copy, you'll receive a summary that gives an overall critique (Figure 2). It contains indexes for the documents readability, use of descriptive language, strength of delivery, and use of jargon. The summary also includes suggestions for improving your sentence structure. You can now go back into your word processor and edit the marked-up copy as necessary. When you're done, RightWriter removes its comments and leaves you with your corrected copy.

Grammatik offers another option for checking your document. As in RightWriter, it will generate a marked-up copy with advice. You also can choose an interactive mode.

When you're in the interactive mode, Grammatik stops at each suspected problem. It offers advice so you can make corrections on the spot. There are several choices: You can replace the word(s) in question, edit, ignore, mark the problem for later reference or skip to the next problem.

Once Grammatik completes its analysis, you'll see the document summary. This summary interprets the following statistics about your document: grade level (Flesch-Kincaid score), reading ease (Flesch Readability Score), percentage of passive voice

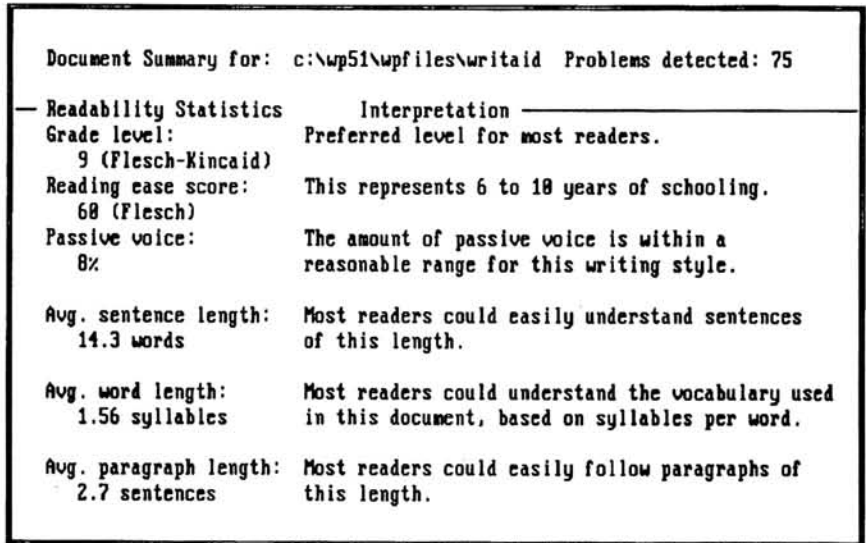


Figure 3. Grammatik Document Summary.

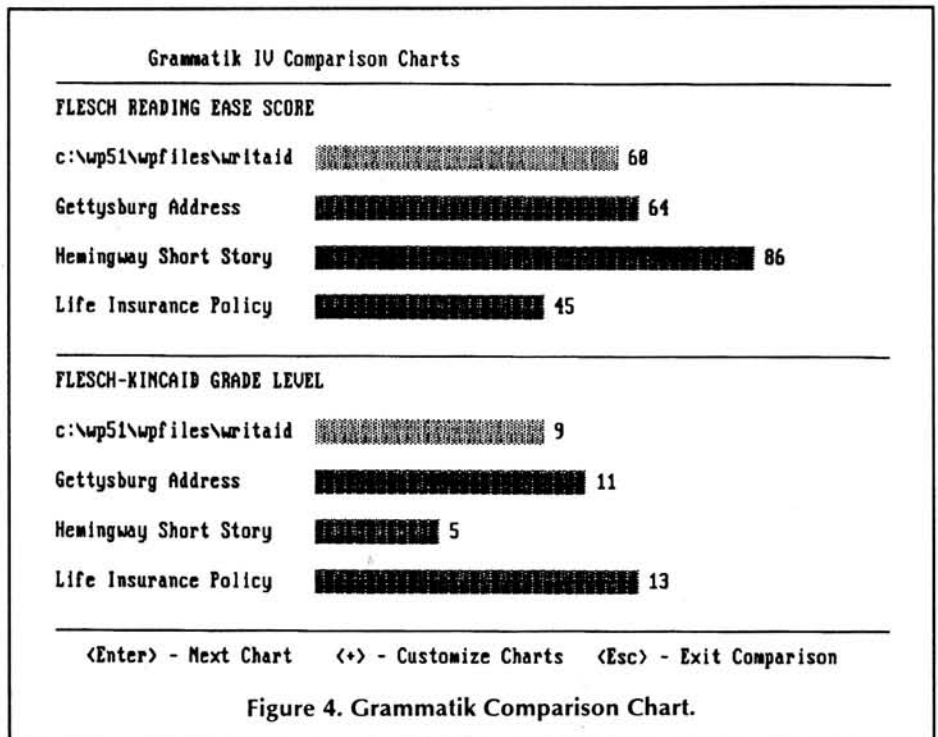


Figure 4. Grammatik Comparison Chart.

and average word sentence and paragraph lengths (Figure 3). More detailed information is available in the document statistics screen. If you prefer statistics in chart form, Grammatik will compare your document with three standard works: The Gettysburg Address, a Hemingway short story and a typical insurance policy (Figure 4).

Writing Styles

Both Grammatik and RightWriter can

tailor their proofreading for a specific writing style. These features tell the grammar checkers to apply different proofreading criteria, depending on the writing style you choose (Table 2). For example, business writing requires stricter rules than fiction writing. Long Faulkner-style sentences may be okay for fiction, but wouldn't be appropriate for a business letter.

If you don't choose a writing style, these programs default to a "general" writ-

ing style. This would be a suitable choice for most writing.

Changing the Rules

When you become more familiar with grammar checking programs, you may have the urge to customize them to your own writing style. RightWriter and Grammatik allow you to turn rule classes on and off, customize the summaries, change rule dictionaries and choose proofreading options. This makes the program smarter and fine tunes it for your needs.

For example, I turned off the mixed capitalization rule in Grammatik so it would not flag software names. Before doing this,

Right Writer	Grammatik
Technical reports	Technical
General business	Business
Articles	Informal
Proposals	General
Manuals	Fiction
Fiction	Custom
Custom	

names like WordPerfect, WordStar (and even REMark) came up as errors. You can choose as many rules as you wish to turn on, off or even modify.

Summary

The current crop of writing aids are more sophisticated and easier to use than ever. Along with your own careful proofreading, they'll act as an electronic "second opinion" for your writing. Use them regularly to take your ailing documents off the critical list.

Despite their limitations, grammar and spelling checkers work. If you lower the amount of grammatical errors and increase the readability of your documents, you're bound to become a better writer. ✨

Continued From Page 24

```
FCBARG1 DB 0 ;FCB ARGUMENT COPY

SIGNON DB 13,10,'FSKDB -- Full Size Keyboard Utility for Laptops, v. 1.0'
DB 13,10,10,'$'
INMSG 'FSKDB is now installed in memory. You may control it',13,10
DB 'by entering',13,10,10
DB ' FSKDB D To disable supplemental keys.',13,10
DB ' FSKDB To enable supplemental keys.',13,10
DB ' FSKDB U To uninstall FSKDB. You can only',13,10
DB ' uninstall if FSKDB is the last',13,10
DB ' TSR loaded.',13,10,'$'
DISMSG DB 'FSKDB is now disabled.',13,10,'$'
ENMSG DB 'FSKDB is now enabled.',13,10,'$'
UNMSG DB 'FSKDB is now uninstalled.',13,10
NOUNMSG DB "FSKDB is not the last TSR. Can't uninstall.",13,10,'$'
NINMSG DB "FSKDB is already not installed.",13,10,'$'

CODE ENDS
END START
```



EaZy PC: EZM128 128K Memory Expansion, \$95; EZCOM Serial Port \$85; EZCOMBO Memory Expansion and Serial Port, \$145

SmartWatch: No-slot calendar/clock module. Software included. For all H/Z PC's, \$32

H/Z-148: ZEX-148 1-1/2 Card Expansion Bus, \$79.95; ZP-148 704K Memory PAL, \$19.95

H/Z-151: VCE-150 removes existing video card, allows use of EGA/VGA card, \$49.95; ZP640+ PAL modifies existing memory card to 640/704K using 256K RAM chips, \$19.95; ULTRA-PAL modifies existing RAM card to 640/704K plus 512K EMS/RAM disk, \$39.95; COM3 kit changes existing COM2 to COM3, allows internal COM2 modem, \$29.95

H/Z-100: ZMF100a modifies old motherboard for 768K memory, \$75; ZRAM-205 converts Z-205 card into 768K RAM disk, \$39

H89: H89PIP two port parallel printer interface card, \$50; Printer cable \$24; SLOT4 adds extra expansion slot to right-side bus, \$39.95

Call Or Write For Additional Information

Order by mail, phone or FAX. VISA/MasterCard/AMEX accepted. No charges for UPS Ground or USPS shipping. WA residents add 8.2% tax. Hours: M-F 1-5 PM Pacific. We return all calls left on our answering machine!

FBE Research Company, Inc.

P.O. Box 68234, Seattle, WA 98168

206-246-9815 Voice/FAX TouchTone Selectable

Continued From Page 30

Edit Presentation			
F1-Help	F2-Preview	F4-Effects	F5-Mark block
F6-Main Menu	F7-Add/Edit	F9-HyperShow	F10-Continue
#	Filename	Type	Description/Directory
1		DRAM	Background drawing for presentation
2			
3			
4			
5			

Figure 27. Create/Edit Presentation Menu.

One point I make in class is the use of the description block. This is a short block where you can add information about the file. This makes it easy to find the chart you are looking for. The few seconds you spend typing this information into the system will save you many minutes later.

Also available in the File Menu is the Export (Figure 24) capability. HG3 will save the chart in various formats, including PCX, bit-mapped and CGM.

The Output option (Figure 25) on the Main Menu will permit you to select the output device. Note the spooler option. This will permit you to delete files in the HG3 spooler.

The next option on the Main Menu is Presentation (Figure 26). This function per-

mits you to build a slide show which can be run on a PC. HG3 will automatically build your slide show, if you desire. When your charts have been completed, you can then move to the *Presentation* option and add the transition effects and timing.

The Presentation capability also permits you to create interactive presentations where the viewer can select different paths to view.

Note in the Create/Edit Menu (Figure 27) that you can now select a background drawing for the show. To add charts to the presentation, press Ctrl-Ins and all available charts will be displayed.

This completes a brief overview of Harvard Graphics 3.0. This product is a vast improvement over earlier versions. If you use business graphics, you should look this product over. Although it does not have the Windows interface yet, it is easy to learn and use. ✨



Polishing Your Writing Style with RightWriter 4.0

Tom Bing
2755 Carolyn Drive
Smyrna, GA 30080

Word processing on the personal computer is a power trip. You move paragraphs with a few keystrokes. You tighten a sentence with a few more. Global changes take only seconds. Ignore that typo; the spelling checker will catch it. Your finished work prints crisply, with straight right margins and numbered pages. A laser printer adds authority to your writing; its fonts and graphics defy anyone to challenge your words.

Then, along comes RightWriter (RW), letting all the air out of your writer's ego. It's neither bossy nor demeaning. It comes on like a quiet colleague who asks, suggests, recommends. Is this clear? Would "try" be better than "attempt" here? Can you avoid the passive voice there? At the end of the document, RightWriter announces Judgment Day. Well, maybe it's not *that* bad; it simply summarizes the qualities of your writing. It tells how direct and forceful you are. It tells what level of education readers must have to understand your text. RW's overview of style, structure, and usage gives a global picture of your prose.

Do you need this? I surely do. My articles need editorial review and proof-reading. If I proof it myself, I'm too close to my work; some problems escape my notice. If I seek help, few of my friends have the time and editorial skills to straighten me out. If I find someone, they'll probably just tell me my faults and not write them down. I write for publication; I want editors to accept my work and readers to understand it. I have to make my words work, and that's where RightWriter comes in.

George Elwood wrote an article on Grammatik and RW for the November 1989 REMark ("Better Writing", page 19). That article compared earlier versions of these two widely available packages. Both programs have been upgraded since then, and I have chosen to focus on RW 4.0. I

hope that this article will complement George's very capable treatment of RW 3.0.

Why I Bought RW

I had been planning to buy either Grammatik IV or RightWriter for a long time. Either one was about \$60.00 at local discount stores, but I'm cheap. Then TigerSoftware made me an offer I couldn't refuse. For \$29.95, I could buy RightWriter with the software version of Strunk and White's *Elements Of Style* thrown in. To cap it all, they included a book: the *Word Perfect 5.1 Quick Reference* from Que. These three items had a total list price of \$168.85. Tiger offered this discount to registered WordPerfect owners. I took them up on it, and many others must have also. The RightWriter package was back-ordered and came after about two weeks.

Installation

Installation was easy. I copied my distribution disk and used the copy for installation. The software came on a 3-1/2" disk, so I used my B drive. The chapter on installation assumes that the disk is in drive A. However, if you substitute "B" for "A" when installing, it works fine. During the installation, the user identifies his word processor to the program. RightWriter is fully compatible with 23 different word processing programs. As the manual says, this means RightWriter will accept documents in the word processor's own format. Furthermore, the word processor will accept the marked-up documents produced by RightWriter. For 12 other word processing programs, RW works on an ASCII version of the document. For some of these, RightWriter recognizes the native word processing document format and translates it into ASCII. For others, it relies on the word processing program's own internal

ASCII translator. An editor or word processor that is not supported in either way may still be used with RightWriter. If you can convert its native format into ASCII, RW can scan the resulting file. For example, WordPerfect, WordStar, and MS-Word are three of the fully compatible word processors. WordStar 2000 is one of those 12 that RightWriter can translate to ASCII. Others, like Lotus Manuscript, must perform their own conversion to ASCII.

Eight of the fully compatible word processing programs can use the RightWriter Interface Program. When RW is installed, it creates a modified copy of your word processing program called WPRIGHT.EXE. This allows RW to run from inside the word processor. For instance, I type Alt-R inside WordPerfect and RW does its work. The other way, which works with all supported word processors, is to run RIGHT.EXE. Both methods create a marked-up copy from your original document.

The installation program changes your AUTOEXEC.BAT file (if you choose). In my case, it added the line **SET RIGHT=WP** to the file, since I use WordPerfect. It saved a copy of the original AUTOEXEC.BAT as AUTOEXEC.RW4, a reassuring safety feature. The manual recommends that you install RightWriter in the same directory as your word processing program. I did so, but only after I had scanned the RightWriter disk for duplicate file names. There were two: INSTALL.EXE itself and a READ.ME file. I copied these existing files to a new directory. This precaution proved necessary, since INSTALL replaced my old READ.ME without asking. This was a surprise, considering the protection that INSTALL had given to AUTOEXEC.BAT. Future versions of INSTALL may (I hope) check for such name collisions and automatically copy old files with matching names

<<*** SUMMARY ***>>

The document WPTEST.DOC was analyzed using the rules for General Business writing at the General Public education level. It is a WordPerfect document. The marked-up copy is stored in the file WPTEST.OUT.

READABILITY INDEX: 5.23

4th 6th 8th 10th 12th 14th
|****| | | | | | | |
SIMPLE |----- GOOD ----- | COMPLEX
Readers need a 5th grade level of education.

STRENGTH INDEX: 0.00

0.0 0.5 1.0
|* | | | | | | | | | |
WRAK |----- |----- |----- |----- |----- |
The writing can be made more direct by using:
- the active voice
- fewer weak phrases
- fewer cliches
- more positive wording

DESCRIPTIVE INDEX: 0.50

0.1 0.5 0.9 1.1
|*****|*****|*****|*****| | | | | | | | | |
TERSE |----- NORMAL ----- |----- |----- |
The use of adjectives and adverbs is normal.

JARGON INDEX: 0.00

SENTENCE STRUCTURE RECOMMENDATIONS:
15. No Recommendations.

<< WORDS TO REVIEW >>

Review this list for words that may confuse your message. These include words that are negative, frequently misused, colloquial, or jargon. As you review each word, think of its effect on the reader.

- 1 advantageous (Often misused)
- 1 doable (Possible jargon)
- 1 not (Negative)
- 1 severe (Negative)

<< END OF WORDS TO REVIEW LIST >>
<<*** END OF SUMMARY ***>>

Figure 1

to a safe place. The only other installation problem I had was the switch from Drive A to Drive B as described earlier. Apart from that, installation was simple.

Using RW's Output

RightWriter comes with an ASCII file, TEST.DOC, containing a single paragraph. Chapter 3 guides you through an analysis of TEST.DOC. The resulting file, TEST.OUT, contains each weakness noted by RightWriter and the summary and recommendations. However, the summary in the manual TEST.DOC differs from the one I got by running RIGHT.EXE. For instance, there are two sentence structure recommendations in the book. My trial run of RIGHT produced no such recommendations and flagged 10 errors; the book shows 9. I suspect the RightWriter folks are fine-tuning their error reporting. The software responds to tiny changes in the document.

By simply splitting "cannot" into "can not", with no other changes, the summary numbers changed slightly.

RightWriter analyzes the whole document at once, and I think I know the reason. If RW worked interactively, it would prompt the user for each suggested change. Each change would, in effect, create a new paragraph. RW might have to analyze the altered paragraph many more times. For spell checkers, interactive correction may work. However, RW's style checking is a "big picture" assessment, designed to scan a whole document at once.

The Summary boils down the strengths and weaknesses of your writing to a few major indexes. The Readability Index tells the education level (school grade) required to understand your document. The Strength Index is higher for forceful, direct delivery; lower for wordy, intricate writing. The Descriptive Index should be in the mid-range

(0.2 - 0.9) if adjectives and adverbs occur with normal frequency. If you have several sentences that are compound or have subordinate clauses, the Sentence Structure Recommendations will alert you to the problem. As the manual says, such complex sentences are more appropriate to longer documents. The Words To Review goes one step beyond a spelling checker. This section aims at "getting the right words", as well as "getting the words right". The words listed may be improper for the reasons given (Figure 1). The Summary provides an overall critique that shows the general tendency of your writing.

Customization

RW allows the user many choices. For instance, RW checks documents using the General Business style by default. You may choose among four other styles: Fiction, Manual (user guides, references), Proposal, or Technical. Each style has a slightly different customization of the rules used to check your work (see Figure 2). For example, RW will flag sentences longer than 25 words if the General Business style is chosen. For Fiction, the length is 30. You also may turn off individual rules in each of these groups: grammar, style, usage, and punctuation. You can also tailor the document summary to your needs. Three readability indexes are available. Ratios, such as words per syllable and syllables per word, are used to compute the indexes. The U.S. Government has adopted one of these as a standard (the Flesch-Kincaid Index). Documents produced under government contract for the armed services must use this index. The other indexes are the Flesch Index and the Fog Index (no, I don't think the last was devised by Professor Fog).

The /C command-line option for RW doesn't work. This option should allow you to substitute different configuration files at run-time. The configuration file tells RW what rules are turned on, which writing style to use, and so forth. If you need to keep different configuration files, store them in different directories. That way, you can overwrite the one which RW uses before each run. This work-around is messy. I called RightWriter's tech support 800 number to discuss this problem. They were very friendly and responsive. They answered the call (no "please hold") and routed me to the right technical person in less than a minute. They knew about the problem, and I got the impression that a fix is in the works. Changes made with the RW "CHANGE SETTINGS" menu need not be permanent. Temporary changes are the way to go if only a few one-time changes are involved.

RW includes a dictionary that the user can modify. However, it differs from those used by spell-check programs. The RW dictionary is a list of "problem words". When RW finds one of them, it flags the

Writing Types

The following table summarizes differences in writing types. For all five types, all rules not listed below remain at their current values.

Analysis	Fiction	Business	Manual	Proposal	Technical
Style					
Sentence length	30	25	22	28	28
Passive voice	Yes	Yes	Yes	No	No
Conjunct. start	No	No	No	No	No
End w/prep.	No	No	No	No	No
Contractions	No	No	No	No	No
Simpler words	Yes	Yes	Yes	Yes	No
Usage					
Computerese	Yes	Yes	Yes	Yes	No
Grammar					
Unusual					
Capitalization	Yes	Yes	Yes	Yes	No
Document Statistics					
Avg words/sent	Yes	No	Yes	Yes	No
Avg syll/word	No	No	Yes	Yes	No
All others	No	No	No	Yes	No

Changing writing types also changes (1) flags for complex sentences and (2) measures for strength and descriptive indexes. For example, if you choose Fiction, RightWriter allows more flowery, descriptive sentences.

5 - 7

Figure 2

word as a possible error (jargon, slang, and so on). By contrast, a spelling dictionary lists acceptable words. For instance, if much of your writing deals with PC's, you might want to add "MS-DOS" to your spelling dictionary. Otherwise, it will show up as a spelling error. You add words to RW's dictionary for the opposite reason - they may not be acceptable. For instance, you might add "MS-DOS" and other trade names to the RW dictionary. RW will then complain about seeing these words in a document. A purchasing manager might check for brand names in a "Request For Quotation" document, because it will go to several vendors. The brand names to avoid would go in the RW dictionary as "User Flagged" words. RW can also find a certain type of spelling error. For instance, someone copying the Gettysburg address types, "all mean are created equal." Since "mean" is a dictionary word, a spelling checker will let it pass. However, RW will flag it by asking, "DO SUBJECT AND VERB AGREE IN NUMBER?" Even though "mean" can be a noun, it is singular, and "are" is plural.

You may get tired of RW's gripes about passive voice or split infinitives. If so, you may want to turn off certain rules. Don't do it. A writer, like any other artist, will occasionally break a rule by design to create a desired effect. Winston Churchill's famous sentence, "This is the sort of English up with which I will not put," comes to mind. He made the point that you must either end some sentences with prepositions or use very awkward English. Even so, you should take RW's comments seriously; breaking the rules should be a deliberate decision. Turning off rules right and left is like turning up your car radio so you won't hear the engine coming apart. The main

goal of using RW is, after all, to improve your writing. An example is RW's "single word in quotes" rule. I often quote a single word to convey something other than its usual meaning. However, leaving the rule turned on guards against overuse of this device.

You can also change some of the file system defaults for RW. You can specify a default directory to search for document files and the extension to use for RW's marked-up copies. Experienced DOS users with specific needs may want to change the defaults for these settings.

What's in the Manual

RW now comes from Que Corporation, a well-known publisher of books on personal computers. The quality of the User's Manual is what you'd expect from a book publisher. The pictures of PC screens are clear and pertinent to the topics under discussion.

Chapters 1 through 5 include the material already covered above:

- Chapter 1. Introducing RW
2. Getting Started
3. RW Tutorial
4. Running RW
5. Customizing RW

After you've used RW a few times, you may want to study the rest of the manual, described below.

Chapter 6 gives a detailed description of each of RW's comments. It explains the principle of writing involved in each case, telling why you should make the suggested change. I suggest you skim this chapter. If you find that a specific comment occurs often in RW's analysis of your work, study its explanation in detail. You'll come away with specific ways to make your writing crisper, clearer, and more direct.

Chapter 7 tells how to interpret the

RW summary. RW tailors the summary to document length. If a document is 50 words or less, the readability index is all you'll see. For short documents, the other statistical measures are not significant. The summary was self-explanatory to me, but Chapter 7 gives all the background on it you'll need.

Chapter 8 covers the WP programs compatible with RW. It surveys each compatibility group (Fully, Print, or ASCII compatible) and highlights facts you'll need to know to use each program with RW. For example, Enable users must spell out the default WPF extension for Enable documents so RW can find them. You may have a new word processor with a document format that RW cannot use directly. This chapter ends with several suggestions for converting such documents so RW can scan them.

There are four appendices to the manual. The notes below should give an idea of their contents.

- A: *Using the Advanced Features.* Gives more background on the readability index and its meaning, with suggestions to improve your score in this area. This appendix also purports to cover all RW command line flags. However, it lists the /C flag without describing it; this is what prompted my call to RW technical support. Until some future version of RW implements the /C option, do not try to use it.
- B: *Troubleshooting RW.* This appendix lists error messages and gives possible causes of each type of error. This data probably came from the experiences of the RW technical support staff.
- C: *Using the WordPerfect Interface.* This appendix details the use of RW with this popular word processing program. For users who have WordPerfect Library, it also covers setup of the Library to recognize RW.
- D: *Using RW In Windows 3.0.* Several illustrations appear in this appendix. After all, what else would you expect in a discussion of a graphical interface? This appendix tells how to edit the Windows PIF file so RW and Windows can work

It's clear from the example why RightWriter analyzes the whole document at once, rather than attempting<<*_S13. REPLACE attempting BY FORM OF SIMPLER try? *>> to get the user's approval for each recommended change as errors are found<<*_S1. PASSIVE VOICE: are found *>>. <<*_G3. SPLIT INTO 2 SENTENCES? *>><<*_S3. LONG SENTENCE: 29 WORDS *>> Each change, in effect, creates a new paragraph which<<*_S22. SHOULD paragraph which BE paragraph that? *>> would have to be re-analyzed<<*_S1. PASSIVE VOICE: be re-analyzed *>>, with another prompt to the user to confirm or reject a suggestion.<<*_S4. IS SENTENCE TOO DIFFICULT? *>><<*_S3. LONG SENTENCE: 26 WORDS *>>

Figure 3

together.

The Glossary defines several terms of interest to the user. Some of them are general DOS terms, like "file extension". Others are more specific to RW, like "Fully compatible" and "Marked-up copy".

Are RW's Results Valid?

No grammar checker is perfect; they will all occasionally flag phrases that are correct, or let an error pass. According to RW ads, RW goes astray less often than the competition does. Once in a while, you will have to exercise judgement in accepting or ignoring RW's suggestions. Still, the general thrust of its analysis is right on target. For instance, a March 1990 *Unix Review* article covered the SCO Xenix version of RW. It cited passages from Hemingway and Steinbeck that scored 0.83 and 0.92, respectively on the strength index. That's evidence that RW recognizes good writing. I ran the Gettysburg Address through RW, using the "Fiction" style. RW surprised me by giving it a zero on the strength index, although the other indicators were in the mid-range. After that result, I ran Mr. Lincoln's speech through Grammatik III. It

didn't show a strength score, but made similar complaints about style. I doubt that any speech would win praise from a grammar checker. Speeches tend to use longer sentences with parallel phrasing to make them memorable. Figure 3 is a paragraph from an early draft of this article, with RW's comments. I used these suggestions, and RW found no fault with my final version of this paragraph.

Conclusion

Why bother with RightWriter or any other grammar checker? The answer is simple: To communicate more effectively. It's like transmitting a radio signal. If I'm sending and you're not receiving, communication has not taken place. It's the sender's job to overcome any obstacles. I must first gain your attention, by saying something you want to hear and starting in a way that "hooks" you. You may be a Ph.D. in English and an author of several books. If I'm smart, I won't make you use all that knowledge; a difficult style repels even those who can struggle through it. The best style is the one that the reader notices least. The best style focuses attention on what I am saying, not

how I say it.

RightWriter identifies these roadblocks to understanding and suggests ways to deal with them. The writer can then choose to adopt or ignore the suggestions. My experience suggests that you will take RW's advice most of the time. As a result, your writing will be more forceful, clearer, and accessible to a wider audience.

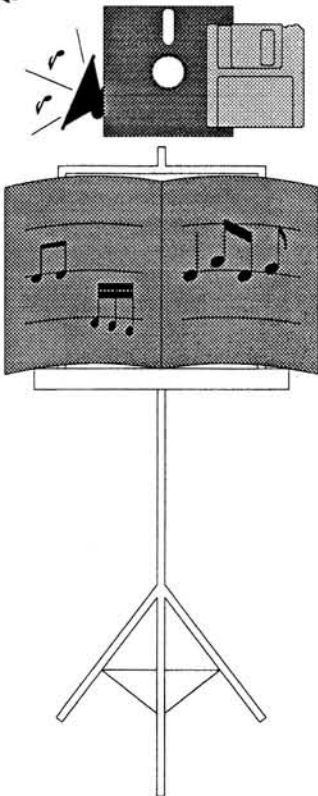
I bought my copy of RW from TigerSoftware at the address below. I paid \$29.95 plus shipping; their regular price is \$89.95. Other mail-order and discount suppliers also carry it. Addresses for TigerSoftware and RightWriter appear below.

RightWriter
Que Software
4545 Samuel Street
Sarasota, FL 34233
800-992-0244

TigerSoftware
800 Douglas Entrance
Executive Tower
Coral Gables, FL 33134
800-444-3363

*

New Software Product!



The Electronic Clavier
P/N 885-6016



u-til-i-ty (u-til'e-ti), n. [*pl.* UTILITIES (-tiz)], [Fr. *utilite*; L. *utilitas* < *utilis*, useful < *uti*, to use], 1. the quality or property of being useful; usefulness. 2. the greatest happiness of the greatest number. 3. something useful.

Sanford Shapiro
654 Gravilla Street
LaJolla, CA 92037

The dictionary defines utilities as useful things. Computers can be difficult to use — like shaving with an axe, or they can be easy — like making instant coffee. Good utility programs make the difference. For the beginning computer user, utility programs are like grains of sand — they can be found everywhere. But for the experienced user, finding the good ones can be tough — like trying to conquer the Himalayas on roller skates. Everyone has his favorites — and mine include disk and file organizers, typing shortcuts and memory managers.

Popularity does not guarantee satisfaction — some best selling programs, like PC Tools, are a disappointment. Nor can you judge by cost — many excellent utility programs, like 4DOS, are inexpensive. And PC Magazine provides some wonderful utility programs for free. From the PC Magazine collection, I particularly like the programs: RN, DR, DIRMATCH and SNIPPER.

PC Tools

What, you might ask, is wrong with PC Tools? Nothing, if you are just starting your collection of utility programs. Version 7 of PC Tools exploded on the computer scene like a Fourth of July firework. A Swiss Army Knife program — something for everybody — it offers file management utilities, disk management utilities, a shell program, a remote control modem program, a disk cache program and a comprehensive hard disk backup program. But, the serious user will already have other programs that do most of these tasks, and often do them better.

UNDELETE

For example, for years I relied on the PC Tools program, UNDELETE, to recover accidentally erased files. Then Zenith Data Systems released MS-DOS 3.3+ with the GDU (General Disk Utilities) program containing an even more powerful UNDELETE

program. [Figure 1]. The Zenith Data Systems UNDELETE program offers two options for recovering erased files: Cluster Analysis Priority (default method) and Cluster Sequence Priority (alternate method). [Figure 2].

Mr. Gale Horst, Zenith Data Systems Software Engineer, points out: "Analysis priority was chosen [as the default option] for the following reasons: 1) Most people don't compact their disks very often, 2) the files that are written and rewritten most often are your data files, such as those from your word processor, spreadsheet, etc. 3) The files that are most likely to be fragmented are the files that are written and rewritten the most often... At the time GDU was developed, all the other undelete utilities I tested used Sequence Priority and gave the same results as GDU. Then the

GDU analysis priority was developed which gives you a much better chance of recovery for certain types of files, depending on the conditions of the disk fragmentation. It was then determined that for the "average user" (whatever that means) the odds are that the GDU Analysis Priority would prove better undeletion results."

What this means is that "Cluster Analysis Priority" is better for text files, and "Cluster Sequence Priority" is better for .COM and .EXE files. The PC Tools UNDELETE program will sometimes tell you that a program has been successfully undeleted when it hasn't — and then you are out of luck. With GDU, if the undeleted file is corrupted, you can erase it again and successfully undelete it using the alternate method. With any UNDELETE program, however, whenever possible use the FC

(File Compare) utility to check your file's integrity. Although GDU is not included with MS-DOS 4.01, I have found the version with DOS 3.3+ to be fully compatible with both DOS 4.01 and DOS 5.0.

```
General Disk Utilities Ver 03.30.08
Copyright (C) 1988 Zenith Data Systems Corporation
```

```
[ General Disk Utilities Main Menu ]
DIRECTORY - Display Total Directory
           - Change Directory
           - Sort Directory
DISK      - Select Drive
           - Display System Parameters
FILES     - List/Edit in Hex and ASCII
           - Undelete a File
           - Rebuild File or Directory
           - Delete a File
MISC     - Display FAT Cluster Chain
           - Display/Edit Logical Sectors
           - Search/Display Data Clusters
EXIT     - Return to DOS
```

Figure 1

```
[ Undelete Options ]
Cluster Analysis Priority (general undeletions)
Cluster Sequence Priority (alternate method)
Restore From Backup Data (if GDUTSR was installed)
```

Figure 2

Backup

A backup program is one you hope never to use. And if the day comes that you need to restore your files — you don't want addi-

tional aggravation. The PC Tools backup program, CPBACKUP, got rave reviews from national computer magazines for speed, reliability and convenience of use, and I enjoyed using it until an emergency restore of my files became necessary. To my surprise, the restore process — unlike other backup programs I have used — could not be started from a floppy disk! I called Central Point technical support, and they confirmed that CPBackup is required to be on the hard drive for it to operate.

If your hard drive has been reformatted, a backup program I like — Fastback Plus — can be loaded from a floppy disk and the restore process immediately started. With CPBackup, however, you must first create a backup subdirectory on your hard drive, copy the backup files from floppies into this directory, restore your files — assuming there is still enough room — and then delete the duplicate backup files. I would prefer selling a ham to a kosher caterer than go through this rigmarole.

Disk Management

The larger your hard drive, the more easily it gets cluttered. Some commercial disk managers do a lot — they also take up gobs of disk space. The PC Magazine utilities are simple, efficient, take up little of your precious hard drive space and are available from any bulletin board system (BBS). I keep trying new disk management utilities, and I keep returning to DR, RN and DIRMATCH.

DR.COM

DR is a replacement for the DIR command (an updated version of DR from PC/Computing Magazine is DIRMAGIC — by the same author, Michael J. Mefford). It lists the files in a directory and provides a menu for the function keys. [Figure 3]

PC COMPUTING — DIRMAGIC	
(C) 1988 ZD ~ Michael J. Mefford	
F1 Copy	F2 Delete
F3 Rename	F4 Move
F5 Clear marks	F6 Mark all
F7 Sort Name	F8 Sort Ext.
F9 Sort Size	F10 Sort daTe
<i>When combined with Alt key</i>	
F1 Copy/V OFF	F2 Confirm OFF
F3 Protect ON	F4 Sort (A)
F5 Highbit ON	F6 Attributes
F7 File search	F8 New path
(Ret) View file, or CHDIR + exit	
String commands: F=Find L=Next	
Ctrl (Ret) Run file, or Load Dir	
Shift Ctrl (Ret) Run then pause	
+/- Mark/Unmark	Esc to Exit

Figure 3

The different function keys allow you to mark, copy, delete, rename, move and view a file. You can also sort the ways in which the files are listed. Combined with the "Alternate" key, the function keys may be used to change attributes, turn on "highbit" (useful for viewing WordStar files) and do searches. Pressing the "Return" key views the file.

RN.COM

After the directory listing program called "DR" was a success, PC Magazine said that if "DR" was the "doctor," then "RN" is the "registered nurse." Typing "RN" instantly puts a tree-like listing of your hard drive directories and subdirectories on your screen, plus a menu for the function keys. [Figure 4].

You can scroll up and down through the tree, and you can use the function keys to change, rename, make and remove directories and to change the attributes of files within the directories. The function key that I use the most is F10, which calls up

PC MAGAZINE — RN	
VERS 2 (C) 1988 ZD	
Michael J. Mefford	
F1	ChDir (or Ret)
F2	Rename
F3	MkDir
F4	RmDir
F5	Hide/Unhide
F6	Do/Undo R-O
F7	Set/Reset Arc
F8	Reread
F9	File count
F10	DR
Esc	to Exit

Figure 4

the DR program. From there, you can continue viewing, copying, moving and editing your files. Typing the Escape key puts you back into the Tree — hitting the Escape key again puts you back to the DOS prompt.

DIRMATCH

When you need to view and manipulate files in two different sub-directories, call up DIRMATCH. DIRMATCH requires that you specify a source directory and a target directory — two windows then open listing the files in each directory. Files with the same name in each directory will be on the

same line. If the files are not identical, the later file will be highlighted. A menu lists commands that allow you to mark, copy, delete, move, swap and print files. You can easily eliminate duplicate files and quickly replace earlier programs with their later versions. [Figure 5.]

These utility programs are not memory resident — they do not steal memory from other programs. They appear instantly when loaded, and they disappear when done. The utility program "Snipper," however, is a memory resident program — once loaded it hangs around waiting to be called. As much as I hate to have precious memory tied up with "terminate and stay resident" (TSR) programs, this utility has wormed its way into my heart. Fortunately, it only takes up 3.6K bytes of RAM.

Snipper

Sometimes you just want to print part of the text on your monitor screen. With Snipper loaded, typing a hot key, "Alt-W," puts a non blinking cursor in the upper left corner of the monitor screen. Using the arrow keys, you place the cursor at the upper left corner where you want the printing to start and type "Return." Then use the arrow keys to define the right and bottom margins of the area to be printed. Typing "P" will dump the defined area to your printer, adding carriage return/line feed characters at the end of each line. Typing "F" prompts you for a file name, which when given, saves the defined text to a disk file. This file can then be edited with a word processor. All of the illustrations in this article were captured from the monitor by the Snipper program.

4DOS

I like short cuts — especially in typing commands. Previously, I used the PC Magazine utility program: "DOSKEY," which provides shortcuts for common DOS commands and keeps track of previous commands. To repeat an earlier command, press the up arrow key until the command appears. MS-DOS 5.0 also provides a "DOSKEY" utility program. But DOSKEY is like a bicycle without tires compared to the shareware program "4DOS" (available from "COM 1", the official Zenith Data Systems Bulletin Board, and available commercially as NDOS, a Norton Utilities program).

4DOS was reviewed extensively in the October, 1990 issue of REMark ("A Replacement for COMMAND.COM" by Mike Wolfson), and I will only briefly summarize the program here. A winner of the PC Magazine 1989 "Award for Technical Excellence," PC Magazine stated: "4DOS is a shareware replacement for COMMAND.COM and we can all hope that Microsoft downloads a copy... and uses it as a guide for improving DOS. This little dynamo keeps the... DOS prompt, but

DIRMATCH 2.0 (C) 1989 Ziff Communications Co. ~ PC Magazine ~ Michael J. Mefford

F1 All Different	F2 Mark all	F3 Clear marks	F4 Copy	
F5 Copy & Delete	F6 Del	F7 Move	F8 Swap	F9 Print
F10 Disk free	+/- Mark/Unmark	Esc to Exit		

Figure 5

adds a world of power and flexibility to nearly every command."

4DOS enhances many of the DOS commands and adds new ones. The documentation is excellent—the manual on the shareware disk prints out to 108 pages. In addition, an "update" and a sample batch file for the "alias" command adds another thirty four pages of documentation.

The "alias" command lets you redefine a command, or a group of commands, with a shorthand definition. For example, I wrote a batch file, added to my autoexec.bat file, which had the command: "Alias L List." [Figure 6]

Now, typing "L" followed by a file name, invokes the "List" command, which is like "Type" but shows one screen of text

ALIASES.BAT

```
alias d dir
alias d2 dir/2
alias l list
alias up cd..
alias ov cd..%1
alias c copy
alias zap del *.bak
alias de del
alias d4 dir/4
```

Figure 6

at a time and allows forward and backward scrolling. Other "Aliases" in my batch file include: "D" for "Dir"; "D2" for "Dir/2" (which lists a two column directory); "UP" for "CD.."; and "OV" for "CD..%1" (changes to another subdirectory in the same directory).

I haven't begun to tap the power of this program. Other commands allow for complex batch file operations, and macro commands define sequences of keystrokes that automatically load and run programs. And best of all, you can use the arrow key to recall and rerun old commands — you don't waste time retyping them. You also can edit these commands before executing them.

Now, you may wonder, how much memory is required? 4DOS normally will take 3K bytes of conventional memory and swap the rest of itself (about 71K) to a disk file that it creates. If expanded (EMS) memory is available, 4DOS will use that instead of disk swapping. And if Upper Memory is available, 4DOS can be loaded into an upper memory block leaving only 0.8K bytes in conventional RAM.

An ideal utility program, 4DOS makes your computer easier to use and has no drawbacks. It is fast, invisible and takes up little conventional memory. For me to give up 4DOS and go back to COMMAND.COM would be like having to comb my hair with a broom.

QEMM-386

A discussion of utility programs is not complete without a mention of QEMM-386. If you have an 80386 computer, QEMM386.SYS — a memory manager — is magic. People, they say, are never rich enough or thin enough — and with computers, you never have enough memory. No

topics generate more confusion than different types of computer memory, the devices used to manage memory and the process of "multi-tasking" (running several programs simultaneously). Computer users are faced with

four types of memory: System memory (also known as conventional memory), Upper memory, Extended memory and Expanded memory.

System memory is the first 640 K bytes of memory and is used by MS-DOS pro-

Region	Area	Size	Status
1	C000 - C7FE	31K	Available
2	CA00 - DFFE	87K	Available
3	F400 - F7FE	15K	Available
4	FD00 - FDFF	4K	Available

Figure 7

grams. Upper memory refers to the addresses between 640K and 1024K. This memory is "reserved" memory — addresses used by video devices and other system

hardware. In 80386 computers, memory managers such as QEMM, can fill unused memory addresses in this area with programs and device drivers, thereby freeing up precious conventional memory.

Figure 7 shows a map of 137K bytes of RAM available in the "upper memory area" in my computer before loading any programs.

Figure 8 shows how this memory area is used in my system by TSR's and device drivers.

The first block, identified as "UMB," is the 2.7K bytes used by the program 4DOS. MIRROR is a DOS 5.0 utility that records information about the hard drive for use in restoring a reformatted drive. Cache-AT refers to VCache - a disk caching program. I use a 1024K byte cache that, although it runs in extended memory, still requires 30K of conventional memory.

EKDRIVER is a device driver for a Kodak 6.6 megabyte floppy disk drive in

Region	Area	Size	Status
1	C000 - C0AE	2.7K	Used (UMB)
1	C0AF - C0B6	0.1K	Available
1	C0B7 - C251	6.4K	Used (MIRROR)
1	C252 - C7FE	22K	Available
2	CA00 - D1B1	30K	Used (CACHE-AT)
2	D1B2 - DFFE	57K	Available
3	F400 - F52D	4.7K	Used (EKDRIVER)
3	F52E - F534	0.1K	Available
3	F535 - F5CE	2.4K	Used (FILES)
3	F5CF - F5E8	0.4K	Used (PDTIMPRK)
3	F5E9 - F608	0.5K	Used (HERCSAVE)
3	F609 - F7FE	7.8K	Available
4	FD00 - FD06	0.1K	Available
4	FD07 - FDF2	3.6K	Used (SNIPPER)
4	FDF3 - FDFE	0.2K	Available

Figure 8

my system. "FILES" is a utility that I use to load FILES=30 into upper memory. That allows me to load only 10 FILES in conventional memory and still have a total of 40 FILES loaded — thereby saving 2.4K of conventional memory. PDTIMPRK and HERCSAVE are hard drive parking and screen saving utilities, and SNIPPER is the TSR program already described. You can see that saves 50K of conventional RAM, and 87K of upper memory is still available for mouse drivers or other TSR programs. The memory maps in Figures 7 and 8 are

provided by the QEMM386 program, "LOADHI."

Extended memory is all memory above 1 megabyte, and it is not available for run-

ning programs under DOS. In 80386 computers and some 80286 computers, the first 64K bytes of extended memory, called "High Memory," also can be used by programs. DOS 5.0, for example, can load part of itself into High Memory. Otherwise, extended memory can only be used by certain types of programs such as RAM disks, print spoolers, disk cache programs and Windows.

Expanded memory is "shadow" memory — DOS does not know it exists. A memory manager maps an area of upper memory to serve as a window through which all expanded memory can be viewed. This clever memory manager then fools the DOS program into thinking that it is running in the conventional (0K-640K) memory area.

Where do these programs actually reside? Is there hardware on your video board, for example, that holds these programs? The answer is no. The programs reside in hardware on your extended or expanded memory board, and your extended memory manager (XMS) or expanded memory manager (EMS) makes DOS think the programs are running somewhere else.

A multi-tasking program, like DESQView, loads different programs into different physical memory areas but maps them — in alternating fashion — to the SAME DOS addresses. Each program thinks it is running in the same memory space — each program getting a portion of the CPU's time (called "time slicing"). Each

program runs a little slower and, in effect, each is running simultaneously. This is different from "task switching" where only one program runs at a time — the other programs are suspended.

Multitasking helps when your computer is tied up in one task, like downloading a large file over the modem, and you want to get back to your word processor or spread sheet — you don't have to sit around twiddling your thumbs waiting for the modem to finish doing its job. Multitasking can be done on an 80286 computer if special hardware is present. Motherboard memory must be disabled, and the addresses must be mapped from an expanded memory board (a process called "backfilling"). On an 80306 computer, no special hardware is needed. DOS 5.0 also provides an extended memory manager, "HIMEM.SYS" and an expanded memory manager, "EMM386.EXE." These memory managers compare to QEMM386 like an abacus to a computer.

QEMM386.SYS manages both EMS and XMS memory. It can load TSR's and device drivers into upper memory and convert extended memory into expanded memory and automatically back into extended memory as needed by programs. QEMM-386 also comes with "OPTIMIZE," a program that will analyze and automatically find the most efficient way of using upper memory for your particular needs. QEMM-386 with its associated utilities, like "OPTIMIZE," "LOADHI," and "FILES," makes memory management as easy as

scrambling an egg.

Products Mentioned

4DOS
J.P. Software
P.O. Box 1470
E. Arlington, MA 02174
(617) 646-3975
(800) 368-8777 (orders only)

Fastback Plus
Fifth Generation Systems
10049 N. Reiger Road
Baton Rouge, LA 70809
(504) 291-7283 (Tech. Support)
(504) 291-7221 (Customer Service)

PC Tools Version 7
Central Point Software
15220 N.W. Greenbrier Parkway
Beaverton, OR 97006
(503) 690-8080 (Tech. Support)
(800) 445-2110 (Customer Service)

QEMM-386
Quarterdeck Office Systems
150 Pico Boulevard
Santa Monica, CA 90405
(213) 392-9701

VCache
Golden Bow Systems
P.O. Box 3039
San Diego, CA 92103
(619) 483-0901

Zenith Data Systems and the NFL

I'll bet some of you football fans already know this, but for the people who don't, let me fill you in. Heath/Zenith has had a long standing association with the National Football League. When a request for proposal from the NFL to provide a four-user configuration for game day play-by-play statistics entry and reporting was issued, Zenith Data Systems responded. ZDS was awarded the contract because our portable product line offered power and flexibility in a small package and our support program through Heath/Zenith was the most comprehensive.

The ZDS NFL configuration consists of one SupersPort SX, two SupersPort 286e's and a MinisPort HD, plus various peripherals.

This equipment is networked into four workstations running a play-by-play statistics package called Superstat. The ZDS/Superstat configuration allows the home team stat crew to quickly enter each play of the game as it happens, automatically updating cumulative individual and team stats and providing virtually instant stat updates to the media and coaches.

Previously, this process was handled manually, with more potential for error and a lengthy delay in updating and reporting the stats.

There are a number of NFL-related advertising and promotional activities underway, including a series of ads in Sports Illustrated and the ZDS All-Pro Linebackers Challenge for our Medallion resellers. Our agreement with the NFL runs through 1995, and you will see more and more programs developed over the next several years. Until then, remember that the number one sport in North America is now associated with the number one PC manufacturer in North America.





Bonehead Plays in C:

Tom Bing
2755 Carolyn Drive
Smyrna, GA 30080

Pointers for Programmers

Programming in C is like riding a big Harley with no roll bar; it's exhilarating to control that much power, but you have to be alert all the time. C gets you closer to the machine than other languages do. For instance, you can access the same memory location in different ways, first as a character array, later as a floating-point number. You can mix data types in assignments. The C compiler will warn you about assigning a floating point value to an integer variable, but it won't keep you from doing it. As an instructor of mine once said, "The C compiler assumes that you are the smart one." That is, the compiler will go ahead and process your statements, no matter how meaningless they are, as long as they don't break the bare-bones rules of C syntax. The motorcycle analogy was deliberate. The motorcycle can go more places than an automobile, but it makes greater demands on the rider. It is far less forgiving of any failure to pay attention. My purpose here is to help the beginning C programmer, to instill an attitude of alertness that will keep the Harley out of the ditch, so to speak.

So much for generalities. Let's look at a few concrete examples of the trouble you can get into in C.

Saying '=' When You Mean '=='

Certain symbols in C, called operators, indicate logical and mathematical operations. In C the '=' operator by itself always means 'assignment'; take whatever is to the

left of equals and change its value. On the other hand, '==' is always an equality test. For instance, if 'a' is 12 and 'b' is 25, then the expression 'a == b' is logically false (zero). In contrast, the BASIC language uses '=' for both meanings and distinguishes them by context:

```
LET X = 1 (Assignment)  
IF X = 1 (Equality test)
```

Giving one operator multiple meanings is called operator overloading. There are some overloaded operators in C, such as '*' and '&'. However, '=' and '==' are not overloaded. Here's a typical mistaking of '=' for '==' in C:

```
if (char_count = 0) {  
    printf("Input file was empty.\n");  
    exit(1);  
}
```

This error will really clobber your program. First, no matter what char_count has before the 'if' statement, it's going to be set to zero when the 'if' is executed. Furthermore, the printf and exit statements will never be reached. Here's what will happen. In the 'if' statement, char_count will be replaced by zero, not compared with it. The char_count = 0 expression is the condition. The 'if' will execute the statements inside the '{}' only if the condition is true, that is, non-zero. The way C works, not only is char_count set to zero by the '=', but the entire char_count = 0 expression has a zero (false) value. Therefore, the condition is false and the statements inside the '{}' are never reached.

If the program prints out the value of char_count past this point, it will be zero. The programmer will wonder why it has a zero value and why the program continued past the 'if' statement. If char_count really is zero before the 'if' statement and the '=' goes unnoticed, it will appear that the program is doing something logically impossible.

We could write a scanner program that would read C source code files and change all occurrences of '=' to '==' in condition statements. This, however, is not desirable. Sometimes you want the '=' operator in a condition, particularly in a 'while' loop:

```
while (*cp++ = *xp++) {  
    ...  
}
```

This cryptic expression allows us to copy the contents of one character string (xp) to another (cp). Usually, in the body of the loop, you will use 'if' statements to pass over some characters, such as spaces and tabs. The point here is that an assignment ('=') sometimes has a place in a condition.

Knowing When to Stop

Remember the Harley? To continue our motoring metaphor, debugging C programs is often like finding brick walls in the dark. In a Jeep with no headlights. Sometimes it's not the abrupt halt that gets us, but the fact that we didn't stop — we went past the desired ending point. What's worse,



...the other cats get to sing along! And now, HEPCAT does Windows!



up over Windows™ 3, even when it is running in the standard (286) or 386 enhanced modes.

What Is HEPCAT?

HEPCAT (Handy Engineer's and Programmer's Calculation Tool) is a floating point calculator with several scientific/engineering features built in, and a binary (programmer's) calculator combined into one tiny, powerful program. HEPCAT is a memory resident program that "pops up" on your screen whenever you activate it by typing a special "hot key" sequence.

The Other Cats Can Sing Along

Unlike other pop-up calculators, HEPCAT is concurrent. That means that when

you pop it up over a running program, the program can continue to run. For example, if you pop it up while Lotus™ is busy loading a huge spread sheet, it will continue loading while you perform your calculations. And HEPCAT always pops up in the current video mode, rather than forcing the screen into a text mode like other pop-ups do. HEPCAT can pop up in any standard CGA, EGA, VGA, or Hercules™ graphic mode, as well as in any text mode. It can even pop up in some non-standard graphic modes (but it may not clear its window when you exit).

HEPCAT Works Harder

The floating point calculator in HEPCAT includes the following built-in functions: powers, pi, factorial, square root, sine, arc sine, cosine, arc cosine, tangent, arc tangent, log (natural and base 10), e^X and 10^X , and it does rectangular-to-polar and polar-to-rectangular coordinate conversion. It also includes several built-in US-metric and metric-US conversions. The binary calculator works in these number bases: binary, tetral (base 4), octal, split octal, decimal, and hexadecimal; and it supports these operations: MOD, AND, OR, XOR,

SHL, SHR.

The HEPCAT floating point calculator supports 8 significant digits and can display numbers four ways: floating point, fixed point, scientific notation, and engineering notation. Numbers are handled internally in BCD format to eliminate binary round off errors in addition and subtraction.

HEPCAT Eats Less

HEPCAT uses less than 18k of memory — less than any other pop-up calculator that we know of. It also uses less than 14k of disk space, so you don't have to worry about where to put it on a small system. HEPCAT is easier to learn, too, with commands that make sense.

If you are tired of pop-up calculators that can only sing solo, or calculators that can do DOS but not Windows (or Windows but not DOS) give HEPCAT a try. HEPCAT is available from ZUG as part no. 885-3045 for \$35.00 (plus S/H). It works on any Zenith Data Systems computer that runs MS-DOS or Z-DOS (including the Z-100 series), and on most PC-compatibles. If you have HEPCAT version 1, send in your original distribution disk and \$10 to upgrade to version 2. ✱


```

#include <string.h>
#define EOS ('\0')
#define MAX 255
int v;
char string1[MAX];
v = strlen(string1);
if (v > MAX - 1) {
    /* Set last array element to EOS */
    string1[MAX-1] = EOS;
    printf(
        "string1 has been truncated from %d to %d chars\n", v, MAX-1);
}

```

Figure 1. Placing an Ending EOS in a Character Array.

we tried to write to a memory location that wasn't in the program area, clobbered DOS, and maybe hung up the PC. In this case, we're not finding brick walls with our Jeep, but cliffs. Here's a simple example:

```

#define MAX 255
char input_line[MAX];

```

If MAX is our array size, then the last element of the array above is the MAX - 1 element, number 254 in this case. This is because C numbers array elements starting with zero. Thus, a 10-element array has 0 through 9 as valid subscripts. If we wanted to step through input_line array, we might set up a 'for' loop as follows:

```

for (k=0; k < MAX; k++)

```

The '<' operator above insures that the loop index k will never equal MAX. If we had used '<=' instead, the final value of k would have been one position beyond the end of the array.

A related problem is making sure that the last element of a character array is an 'end-of-string' (decimal zero) character. We'll refer to this special use of zero as EOS. Nearly all C string-handling functions rely on the EOS value as a string terminator. If you accidentally overwrite or leave out the EOS, most of the standard functions won't know where to stop. For instance, let's say that a last_name string had an EOS in position 30. Then a space replaces that EOS because of a program error. The next EOS in memory is, say, 4000 characters further on. A printf function to display this string would print the last name plus 4000 garbage characters. Figure 1 shows a way to avoid this problem.

Note the strlen(string1) in Figure 1. It is vital to distinguish between strlen(string1) and the sizeof(string1) value. The 'strlen' value indicates where the first EOS was found, while 'sizeof' computes the number of bytes required to store the array (MAX, in this case).

Sometimes, we let the computer figure out how big an array is at initialization. Consider the following array of character pointers:

```

static char *Expenses[] = {
    "Clothing", "Food", "Travel", "Utilities", "Other";
}

```

By leaving the array dimension out (see the '[]' above), the compiler will figure out the correct value (5). This is a small

array, but you can see how tedious it would be to compute the right value for bigger arrays. There is a clever way to specify the number of elements of any one-dimensional array. This method comes from Reference 1 (see list at end of article). It is:

```

#define DIM(array) (sizeof(array)/sizeof(*(array)))

```

Using this macro, we could walk through the 'Expenses' array by some statement such as this:

```

for (k = 0; k < DIM(Expenses); k++)

```

Notice the 'less_than' (<), since an array of dimension N has elements numbered 0 through N-1.

Another boundary value problem occurs when you step through a character string to break (parse) it into different parts. Let's say we're reading lines of the form "SMITH, JOHN" and trying to split them into last and first names. A code fragment to do the job appears in Figure 2.

In the Figure 2 'while' loop, after all tabs and spaces have been skipped, we are one character past where we need to be. Because of the increment (++) operator used in the looping statement, 'full_name' now points to the O in SMITH, JOHN. We want it to point to the J. We back up one position using the full_name - statement. The basic points to remember about handling character arrays and pointers are:

1. The last element of an array of N elements is number N-1.
2. Strings must end with an EOS character for functions to handle them correctly.
3. In parsing strings, pointers must sometimes be decremented after 'for' or 'while' loops to assure that they point to the correct character.

Perils Of Cut And Paste

If a programmer is in a hurry or very expert with an editor, he will often cut and paste existing sections of code to add features. It's a

quick and dirty method, and can get you into trouble if you aren't careful. Here is an example:

```

if ((fp = fopen(input_file, "r")) == NULL) {
    fprintf(stderr, "Can't open file %s\n",
        input_file);
    exit(1);
}

```

Later in the program, you may need to open an output file. If you copy this code section, you must remember to change "r" to "w" and change input_file to output_file. Otherwise, an output file problem may result in a very misleading error message. A section of code may recur throughout your program with only minor changes. If so, you should consider placing the section in a function and handle the minor differences through function arguments. Benefits of the function approach are:

1. Code that exists in only one source file needs to be corrected in only one place when it breaks.
2. In making the code re-usable, you make it more general-purpose, taking care of differences in usage through function arguments.

A Little Learning is Dangerous

Good C program design encourages portable code — that is, programs written so that the same source file will compile in many different environments (MS-DOS, Unix, VAX/VMS). Of course, the compiled code will be different because the machine instruction sets are different. I have sometimes assumed, however, that C somehow smooths over all machine differences, and I didn't have to think about it. Not so! Here's an example of opening a binary file for reading:

```

In Unix:
fp = fopen(in_file, "r");
In MS-DOS:

```

```

/* Full, last, and first names */
char *full_name, *lname, *fname;
char cc;

/*
Code here (not shown) would set lname and cause
full_name to point to the first space after the
comma in "SMITH, John"
*/
/* Get past any spaces and tabs after the comma */
while(cc = *full_name++) {
    if(cc == ' ' || cc == '\t') {
        ; /* Stay in this while loop */
    } else {
        full_name--; /* Now points to first name */
        strcpy(fname, full_name);
        break; /* Exit loop */
    }
}

```

Figure 2. Finding a First Name in a Full Name.

```

#include <string.h>
#include <stdio.h>

int asgn();

main()
{
    char retval[30];
    asgn(retval);
    printf("retval: %s\n", retval);
}

int asgn(sp)
char *sp;
{
    /* Simple assignment doesn't work.
    That's why this statement is commented
    out.
    sp = "ASSIGNMENT DOESN'T WORK";
    */

    strcpy(sp, "STRCPY WORKS");
    return;
}

```

Figure 3. Changing Main Program Values by a Function Call.

```
fp = fopen(in_file, "rb");
```

On a Unix system, the first 'open' statement would work on either a text or binary file; in MS-DOS, the "rb" mode must be used for binaries. There's a little history involved in the reason, dating back to the CP/M operating system. CP/M kept track of files on a disk-sector basis. Since a CP/M text file might not use all of its last sector, a ctrl-Z (decimal 26) was used to mark the end of a text file. MS-DOS adopted this convention to maintain compatibility with CP/M. Unix keeps track of files by their exact byte length. The ctrl-Z may occur anywhere in an MS-DOS binary file. Since we don't want to stop reading the file when the first ctrl-Z occurs, we have to tell the fopen which kind of file we're reading. The "rb" argument is the way to do this. If we had used "r", the file would have been read as a text file by default. Thus, the first ctrl-Z would have been interpreted as the end of the file.

By the way, this is not merely an MS-DOS quirk; not any more, at least. The ANSI C standard officially recognizes "rb"

as a legal argument to fopen. What's the point? Simply this: C allows you to write portable programs, but it doesn't force you to do so. You must avoid usage that is specific to a single machine or operating system — even if that usage is time-honored.

Pointers as Function Arguments

Calls to C functions do not change the value of the arguments in the calling program; this feature is known as "call by value". The function is working with a copy of the value used in the calling program. How then can a function have a lasting effect on main-program values? The solution is to pass the function an address of a value. The function can make a (lasting) change to the value at an address without changing the address itself. Changes to an address argument will do no good; they will last during the function call, but will be lost when we return to the calling program. Notice the listing in Figure 3. The asgn function places a new value in the retval array. The commented-out statements were a first attempt to do this; they didn't work. The strcpy statement was successful, because it changes the value to which sp points.

The term "call by reference" is used for function calls which make lasting changes in calling-program variables. The lesson of this example is that pointer or address arguments are needed for call-by-reference, but they're not enough.

Array notation, such as retval[k], and pointer notation, such as *(retval + k), refer to the same object in memory. Often, as in Figure 3, an array in a calling program "becomes" a pointer in a function. We could have defined retval as a character pointer in the calling program. The point to

remember is that char retval[30] sets aside 30 bytes of storage. In contrast, char *retval only allocates two or four bytes — the size of a character pointer on whatever compiler you're using. Using the pointer approach, we'd have to initialize retval with a call to malloc. This will insure that there is enough unused space for the value it will have after asgn call, like this:

```

char *retval;
retval = (char *) malloc(30);
asgn(retval);

```

Without the malloc call, the strcpy in the asgn function might clobber some variable next to retval in memory. The lesson here is: pointer and array notation are interchangeable. However, because of differences in the amount of storage set aside, pointer and array declarations have different results.

Conclusion

What resources are available to cut down on C coding errors? There are software tools to provide the C programmer a safety net, such as the PC-Lint utility (see Reference 3). There is also the /W3 flag used with the Microsoft compiler. Nonetheless, the errors described here were detected at run time. In other words, as far as the compiler could tell, the program was correct. A programmer should use whatever reliable tools come to hand to check code for errors. However, there will still be some errors for which the only diagnostic tools are the old calibrated eyeball and crafty mind scanning a printout.

The references below should help if you want to study further:

1. Book, "Portable C And Unix System Programming", by J. E. Lapin (Prentice-Hall, 1987).
2. Book, "A Programmer's Introduction To Debugging C", by Robert Ward (R & D Publications, 1989). This appears to be a reprint of "Debugging C" (Que, 1986) by the same author.
3. Article, "The Programmer's Craft: Tools Of The Trade", by Tom Bing (REMark, April 1991). ✨

Continued From Page 20

bers print columns with that number squared, and with the square root of that number.

Rather than writing the instructions necessary to print a number, calculate it squared and print that, and then calculate its square root and print that, ten times; it is only necessary to write the instructions one time, and to tell the computer to 'loop' through those instructions ten times.

Well, there you have it. As we progress through this series of articles, you'll see that you're only going to tell your computer to do six (6) things. They are:

- Input Data

- Output Data
- Make Comparisons
- Make Decisions
- Do Math
- Loop

So you see, you've only got to learn to do six things to become a programmer. Honest, it's that easy, as I'll begin to show you how next month. See you then. ✨

Continued From Page 32

onstrated here and the left margin. This will cause the computer to ignore the new feature(s). To re-energize one or both of them, just simply remove the applicable "REM" statement(s) and reboot.

There is a wealth of information available

on the EMS card and its utilities in the Everex manual. The above "real-world" examples provided for your convenience, briefly illustrate how to implement it. You should check the manual for further data. How do I like my "Everex RAM 3000 Deluxe AT Memory Expansion Board?" Boy, is this unit a real ZINGER! I wish I had installed it long ago!

Credits: I would like to extend public recognition to Henry Fale for his personal support, computer expertise, wisdom, and willingness to share. Also, I recognize John Toscano, a personal friend who assisted me in the installation of the EMS card, and whose advice is much appreciated. ✨

A decorative border of stylized orange and yellow flames surrounds the central text. The flames are jagged and have a red outline, set against a black background.

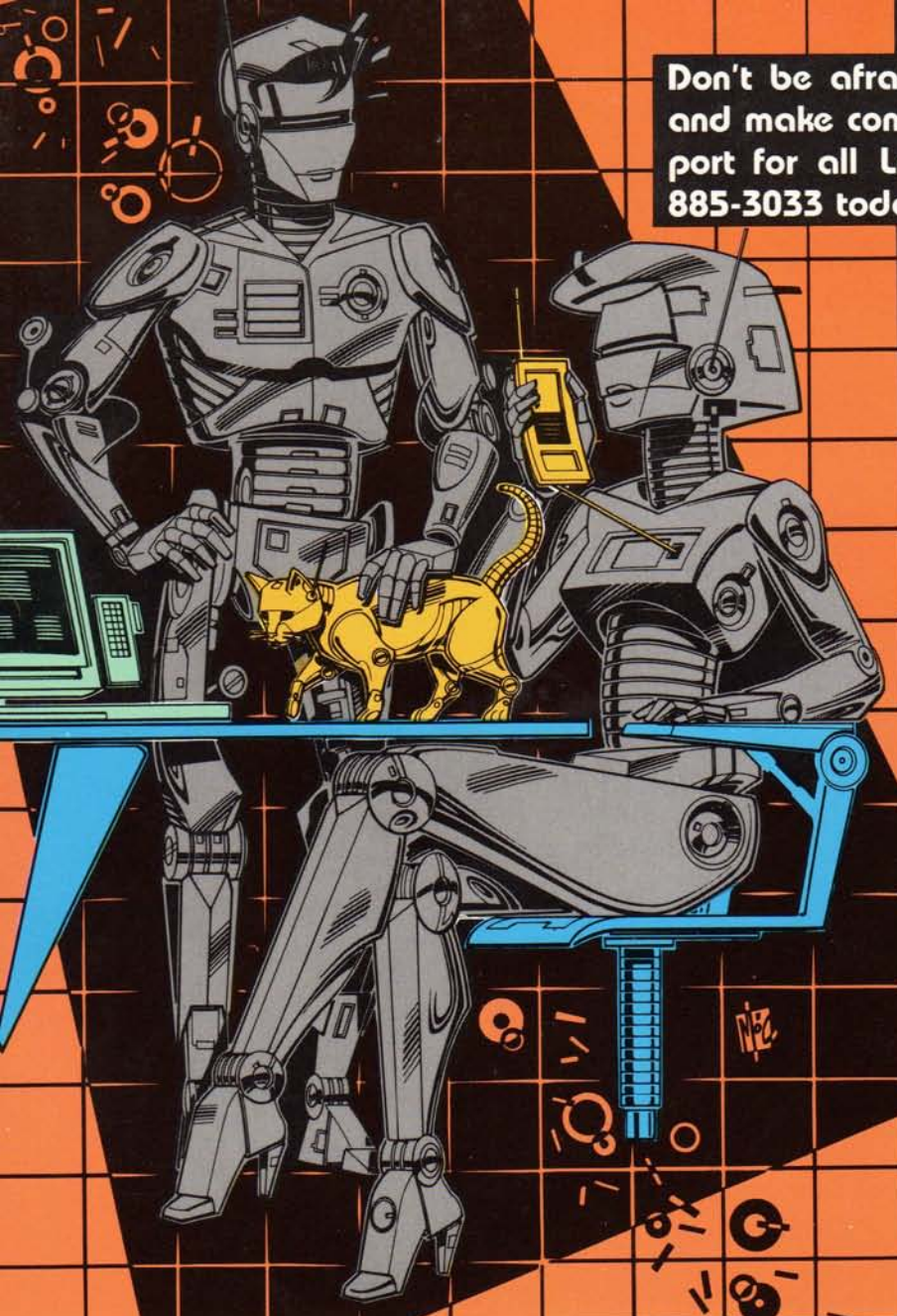
HADES II

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.

Don't be afraid to communicate! Get HUGMCP and make contact the easy way. Now with support for all 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 (F3) Setup Command.
F3 - Toggles The Storage Buffer On and Off. When The Buffer
      Is On, The (Buf) 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 XMODEM,
      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 - 324288 Bytes
Storage Buffer Usage - 0 Bytes

Select Message (0-0), (F1) To List, Anything Else To Abort --> _
F1=Hlp F2=Msg F3=Buf F4=Save F5=Send F6=Cfg F7=Clr F8=Print F9=Exit COM
```

```
HUGMCP Configuration Menu #1:
This Screen Allows The Best Rate To Be Chosen, Depending Upon Which
      Modem You Would Like To Use. It Offers The "Low" Or
      "High" And "Screen" Connections To A Modem. Will Allow Higher Baud Rates.

This Function Allows You To Change The Word Parity. Normally, You
      Would Change The Parity To "None" If Available To Your Modem.
      It Is Also Necessary For XMODEM Protocol To Work Properly.

This Screen Allows The Changing Of The Word Length. Normally The
      Length Should Be Set To 8 Bits. This Option Is Available To Most
      Modem Systems, And Is Necessary For XMODEM Protocol To Work Properly.

This Selection Allows You To Enter Messages Which Can Be Activated
      At Any Time. The (F2) Key Will Do It. Character Messages Can Be Long
      (Maximum 255 Characters). It Should Contain Your Computer's ID Number
      and Keyword. Selection 00 Is Also Special. This Selection Can Allow
      Access To The Setup Data Program. It First Invokes The
      "Screen" Option During Setup.

Type (F6) For More Help, Anything Else To Continue.
F1=Hlp F2=Msg F3=Buf F4=Save F5=Send F6=Cfg 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 Data-Messages
E --> Miscellaneous Functions
F --> Change Screen Color Assignments
G --> Display Current Configuration
H --> Make Changes Permanent

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 Level: NO
Delete Character: HUGMCP
Modem Port Set To: COM1

F1=Hlp F2=Msg F3=Buf F4=Save F5=Send F6=Cfg F7=Clr F8=Print F9=Exit COM
```

ZENITH
data systems 
Groupe Bull

BULK RATE
U.S. Postage
PAID
Zenith Users' Group

POSTMASTER: If undeliverable, please do not return.

\$2.50
P/N 885-2142