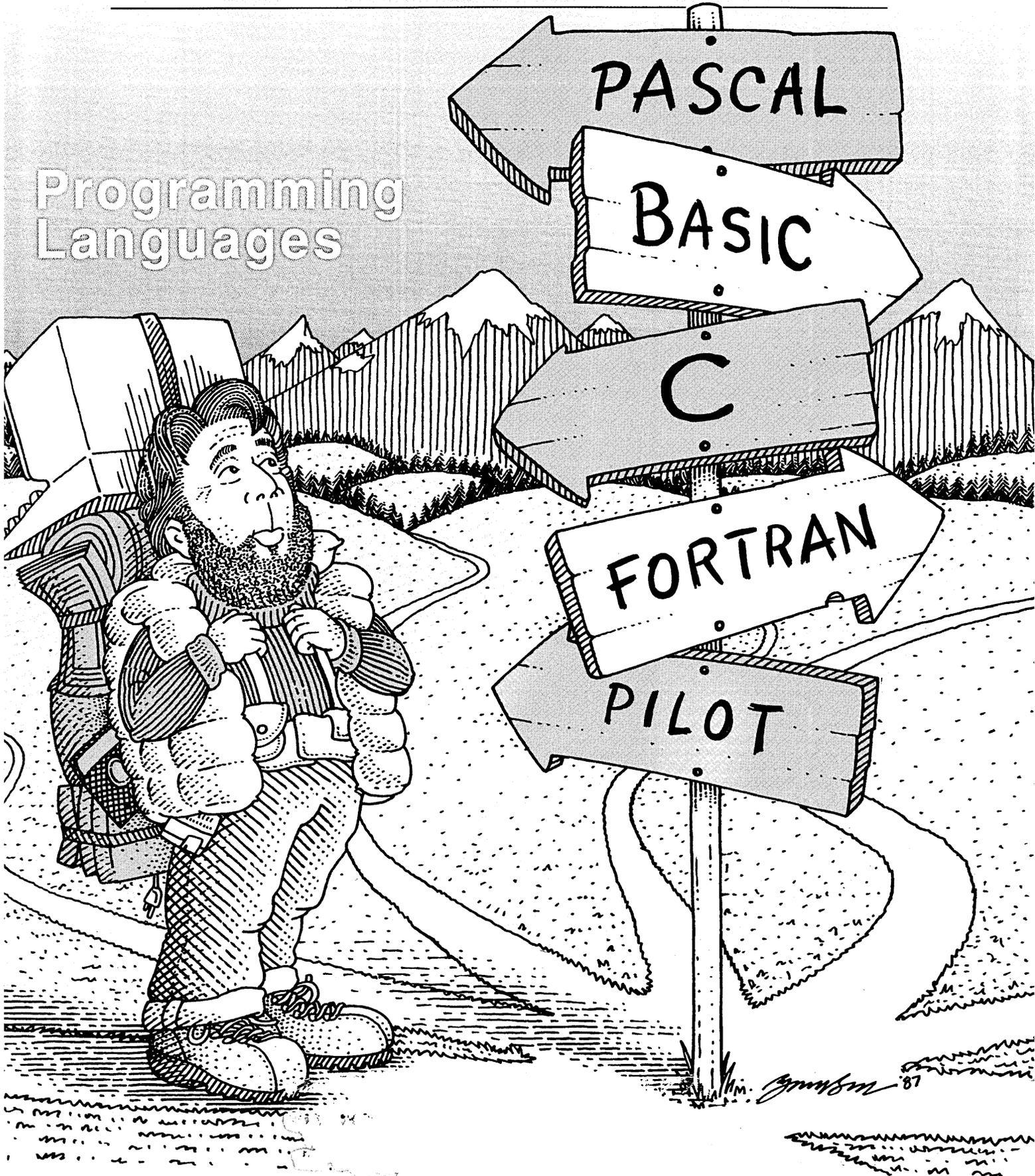


MORROW OWNERS' REVIEW

The International Magazine for Users of Morrow and Other CP/M Computers

Programming Languages



New Low Priced Products for both CP/M & MS/DOS computers!

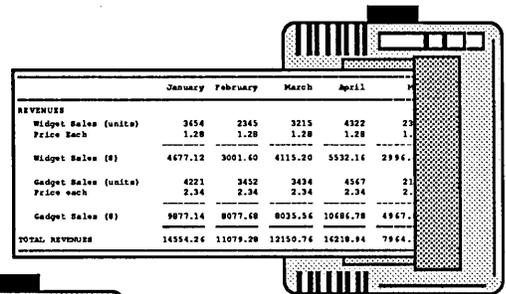
TWIST & SHOUT!

Sideways & Banner Printing Utility for Dot-Matrix Printers

TWIST & SHOUT! is two great pieces of software packaged together at one low affordable price!

TWIST lets you print out huge spreadsheets (or any text file) *SIDEWAYS* on your dot-matrix printer. It literally *twists* your printer's output 90 degrees — no more cutting and pasting to put together a fragmented printout. Unlimited width printouts!

SHOUT prints giant banners in letters from two to eight inches high using three special type styles. Make banners, posters and signs with ease. Anything you can type on the keyboard can be *SHOUTed* in gigantic letters!



	January	February	March	April	
REVENUES					
Widget Sales (units)	3654	2345	3215	4322	2
Price Each	1.28	1.28	1.28	1.28	1
Widget Sales (\$)	4677.12	3001.40	4115.20	5532.16	2396.6
Gadget Sales (units)	4221	2432	3434	4547	2
Price each	2.34	2.34	2.34	2.34	2
Gadget Sales (\$)	9877.14	8077.68	8035.56	10686.78	4567.6
TOTAL REVENUES	14554.26	11079.29	12150.76	16218.94	7964.2



SHOUT YOUR MESSAGE IN A BANNER!

For any CP/M or MS/DOS computer, just

\$34.95

Special Offer: one MS/DOS and one CP/M version for only

\$49.95

Hardware requirements: TWIST & SHOUT runs on any computer supporting CP/M, MS/DOS or PC/DOS. 'IBM Compatibility' is *not required*. Specify machine when ordering. **Printers supported:** Anadex, Apple, C. Itoh, Centronics, Data Products, DEC, Epson and compatibles, Hewlett Packard, IBM and compatibles, IDS, MPI, NEC, Mannesman Tally, Okidata, Smith Corona & Toshiba printers! More are being added daily.

Solve Computer Incompatibility!

MEDIA MASTER

Lets your IBM compatible (PC, XT or AT) read, write and format up to 140 disk formats. Exchange data and programs with other computers for just **\$39.95**.

MEDIA MASTER PLUS

All of the file transfer capabilities of MEDIA MASTER *plus* it gives your IBM compatible the ability to actually run CP/M software. All for just **\$59.95**.

ACCELERATE 8/16

Includes MEDIA MASTER, an upgraded emulator *and* a special easy-to-install NEC V20 chip which lets your IBM compatible run CP/M software at maximum speed. V20 chip and software is just **\$99.95**.

Call or write for full compatibility chart

MILESTONE

Automatic Business Project Planner

Originally \$295 — Now only \$99.95!

Never blow another deadline! Whether you're opening a branch office, hiring a sales force or launching a new product, nothing boosts the bottom line like being on time and within budget. Yet that kind of careful planning can take up a lot of your valuable time. Milestone brings all of the sure-fire planning methods of Critical Path Management together in one easy-to-use package for your personal computer. Here's how it works: You divide your major project into a number of smaller tasks. At Milestone's prompts you enter the manpower, cost and time estimates for each activity and indicate which tasks must be completed before others can start. Then Milestone does the rest, working out your entire schedule and budget (with holidays, sick days and even lunch hours taken into account). Milestone's clear and concise Time Schedule Report gives you a week by week chart showing detailed start and completion dates for each task, critical deadlines, manpower requirements and costs. Milestone is available for *any* CP/M, MS/DOS or PC/DOS computer and any printer. Please specify computer.

We accept VISA, MASTERCARD and AMERICAN EXPRESS

Order by mail or call our 24 hour toll free order line from anywhere in the US or Canada:

800-628-2828 (Extension 918)

Technical questions, orders: 818-716-1655 (9-5 PST)

Add \$4 per order postage/handling. Overseas, add \$12. US funds only. CA residents add 6% tax (LA County 6.5%)

TECHNOLOGIES, INC.
22458 Ventura Blvd., Suite E
Woodland Hills, CA 91364

MORROW OWNERS' REVIEW

P.O. Box 5487
Berkeley, CA 94705
415/644-2638

MCI Mail: MREVIEW (ID 217-1566)

Editor & Publisher
Sypko Andreae

Managing Editor
Serge Timacheff

Operations & Advertising
Greg Smith

Subscriber Services
Margo DeMello

Layout/Art
Steve Albel

MORROW OWNERS' REVIEW (USPS 765-130, ISSN 0888-5087) is published bimonthly, starting with February, by the MORROW OWNERS' REVIEW, P.O. Box 5487, 2920 Domingo, Ste. 201B, Berkeley, CA 94705.

POSTMASTER: Send address changes to the MORROW OWNERS' REVIEW, P.O. Box 5487, Berkeley, CA 94705. Second class postage paid at Berkeley, CA, with additional mailing offices at Fulton, MO.

Copyright © 1987, MORROW OWNERS' REVIEW. All rights reserved. Reproduction in whole or in part without express written consent is strictly prohibited.

The opinions expressed by writers are not necessarily condoned by MOR, nor does MOR vouch for the technical accuracy of the articles. While MOR makes an effort to screen advertisers, MOR cannot be responsible for quality or delivery of merchandise advertised; nor does MOR verify the accuracy of claims made in ads. Investigate before buying. If problems arise, write to the advertisers explaining your complaint, and send MOR a copy of your letter.

We assume all rights in letters or other communications sent to MOR as unconditionally assigned for publication, and copyright purposes are subject to our unrestricted right to edit and to comment editorially.

SUBSCRIPTIONS: Subscriptions cost \$18 per year, \$32 for two years. Add \$8 per year for Canada and Mexico. For all other countries, add \$7/year for surface mail and \$16/year for air mail. MOR is delivered by second class mail in the U.S.; for first class mail, add \$8 per year.

MOR is published every other month, starting with February. We must receive your subscription application before the 15th of the preceding month in order to begin your subscription with the next published issue. To ensure continuity of your subscription, renew at least one month prior to the expiration month noted in the upper right corner of your mailing label. The number preceding that date is your Subscriber Number. Use it in all correspondence. Please allow two months for a change of address to take effect.

Table of Contents

Features

C Notes	
<i>By Lynsee Cain</i>	6
Given a Choice, I'll Take Pascal!	
<i>By Ken Grymala</i>	14
The Simple Beauty of Turbo Pascal	
<i>By Lee McKusick</i>	16
Cheap Disks	
Low cost may not mean low performance	
<i>By Ted Jensen</i>	18
Toolwork's C/80	
<i>Some questions answered</i>	
<i>By David Allen Giunti</i>	21
Small-C	
A public domain C compiler with notable limitations	
<i>By Erik Westgard</i>	23
The Reluctant Programmer	
Mom doesn't want you to spend your time that way, does she?	
<i>By Bruce Gowens</i>	29
Good Timing	
A nuts and bolts approach to building your own real-time clock	
<i>By Mike Allen</i>	35
A Customized Mailing List	
Simple basic programming eases your work	
<i>By Erik Westgard</i>	38
Computers and the IRS	
<i>By Serge Timacheff</i>	43
Make Those Function Keys Work!	
<i>By Gordon Northrup, MD</i>	44
The Basic Approach	
Programming with Basic	
<i>By Jesse Adams</i>	45

Reviews

Nevada Pilot	
<i>By Jesse Adams</i>	48
Mix C Compiler	
<i>By Lee McKusick</i>	49

Columns

CP/M Alive!	
<i>By Ken Grymala</i>	5
From the Mailbox	
<i>By Bill Steele</i>	7
Tools for Tyros	
<i>By Mike Allen</i>	10
Forever Z	
<i>By Rick Charnes</i>	12

Departments

Editorial.....	2
Letters to the Editor.....	3
Important Announcements.....	4
MOR Products.....	25
Table of Latest Morrow Revisions.....	42
Special Credits.....	43
Reference List.....	47
Classified Ads.....	52
List of Advertisers.....	52

Editorial

Enhance your Morrow or switch to Blue

A few people wrote and asked me to answer the question posed by Tim Evans in his letter in the previous issue. Tim wrote that the real question in a world of falling prices for computers (read: MS-DOS computers) is: "Given the fact that complete PC-clone systems can be put together for a thousand dollars, should I spend anything near that amount to *enhance* my Morrow?"

If you need to upgrade your system the major question is whether to either *enhance*, meaning that you improve your existing Morrow, or *switch*, meaning you trade your Morrow in for an MS-DOS machine. There are other alternatives (Macintosh, Atari, etc.) but we won't go into those here.

Before you tackle the *enhance* or *switch* question, you ought to ask yourself whether you need to upgrade at all. If your requirements are outgrowing your current system, figure out your specific needs. Most Morrow owners upgrade for one or more of the following reasons: to attain greater convenience, speed, space or support; access to new software not available under CP/M; or compatibility with another MS-DOS system. Let's look at how enhancing your current machine can, or cannot, meet each of these requirements.

Convenience can be enormously enhanced for MD3s by installing the ZCPR3 system. Soon this will also be possible for MD2s.

Speed of computation can't be increased much. Unless you need to engage in serious number crunching, computation speed is usually not an issue. Our Morrow machines do fine with WordStar or NewWord word processing programs. Disk speeds, however, do make a difference in word processing. It doesn't matter if you have a CP/M or an MS-DOS machine so much as whether you have the faster hard disk or the slower floppy drive. The IBM PC is notorious for poor disk speed, today enabling George Morrow to make a living developing a product to improve it.

Memory space (RAM) in CP/M could be a problem if your work requires large spreadsheets. The need for more disk space can be taken care of by the new add-on hard disks available that, combined with ZCPR3, give you an MD3 system that rivals and in some cases surpasses the quality of the Morrow hard disk machines. Prices are dropping, and MOR subscribers have access to a group-discount plan.

Support for a machine that is no longer made is always a potential problem. That's why MOR has worked long and hard to locate spare part sources and repair shops. We feel

confident that parts will be available for years to come and that repair shops across the country will provide service to Morrow users. If you have problems locating parts or service, we at MOR will help out where we can. The new MOR BBS -- with over a hundred subscribers registered at the end of 1986 -- has been a great help to many already. Many who use a Morrow at home and an MS-DOS machine at work assume that two systems are hopelessly incompatible. While this may be true for programs such as Lotus 1-2-3, it is hardly so for simple text files, which can be transferred between the machines with the excellent program UNIFORM (MOR Vol. 2, #2, p. 23).

Access to new software could be a real problem. Mounting an MS-DOS run coprocessor board will not enable you to run most IBM PC programs, just the few *generic* MS-DOS programs. The cost of an upgrade, in expense and time, is something you should consider carefully. You may get a great deal on MS-DOS hardware only to find yourself paying through the nose for software.

As a Morrow owner, you may have been spoiled into thinking that software is something you rarely pay for. CP/M programs either come bundled with the machine, are available in the public domain (read: free) or can be purchased commercially at moderate or downright low prices.

In the MS-DOS world software generally doesn't come bundled with the machine. While a growing body of public domain software is available, it is often *shareware* and you are expected to pay for it somewhere along the line, or support for it will disappear. Commercially available MS-DOS software tends to be much more expensive than CP/M programs. Before deciding to enhance or switch, ask yourself at least the following questions: How much do I really need an upgrade? If my work requires added functions or compatibility, how much of this is available through hardware or software additions to my Morrow? How will the time it takes to learn the MS-DOS system compare to the time required to modify my Morrow? How does support for my Morrow compare to that for an MS-DOS system regarding both availability and cost? How about user group support? What will my expenses be for each option, both initially and in the long term? Which option will be more exciting and fun to do?

For more questions and arguments, pro and con, check out the file BAM-PBBS.MSG on one of this month's FLOBs. It contains a lively discussion among BAMDUA-BBS users on the very topic of whether to enhance or to switch. Read and make your own choice.



Letters to the Editor

GEORGE'S SECRET MANUALS AND SCHEMATICS

Dear Editor:

Let me commend you on the marvelous job you did on publishing the Morrow maintenance manual, and at a price nobody can refuse. I suspect it is much more complete than the manual none of us USERS were even qualified to purchase (available only to qualified dealers), at a price quoted at nearly today's hardware cost of an equivalent computer.

I must admit that I am impressed by the Morrow circuitry, now that I have had the opportunity to even see the schematics, but I can only say George had his gall! You could get complete schematics of most of his competitors's equipment from SAM's photofacts, or from most technical bookstores for the nominal cost of the booklet. What was Morrow trying to hide? A USER wasn't going to go to Taiwan and clone his damned machines and a serious manufacturer had only to purchase a couple of his MD3's and take them apart to find out all he needed to know, including a readout of all his precious ROMs.

Herb Hartman, Sacramento, CA

[Years ago Morrow, Inc. was afraid of a thousand phone calls a day by beginners asking for help in explaining the technical manuals. I guess that fear was a bit unfounded, but it sure angered a lot of users.--Ed.]

LIMITATIONS OF THE CP- 88 COPROCESSOR

Dear Editor:

I read with interest those letters of CP-88 coprocessor owners who are

concerned whether "IBM-PC compatible" software will run on their machines. It most assuredly will not. Only software which is described as "generic MS-DOS" will run on the CP-88, and the list of manufacturers of such software is minute, indeed.

Todd Merriman, Decatur, GA

NO FINAL WORD ON QUIET DISK DRIVES

Dear Editor:

I was sorely disappointed at the June Issue charting the final word on quieting those MD3 disk drives of mine. Yes, I know that you had published the way to quiet these drives in your first issue. But, the next issue was full of debate about the advisability of altering the drives at all. Learned friends of mine tell me to just leave it the way it was designed. However, I still would like to ask you, my final resolve, if there is a way for a non-tech to follow a "recipe" published in your magazine and feed in the code to permanently quiet the drives and live in heaven evermore...quietly. If there is a safe procedure that users could provide please let us know.

Ken Nielsen, Tigard, OR

[It looks as though there will never be a final word to this. Since MOR published the articles to which you refer, we have discovered that there are more differing BIOS versions for Morrows in the world than you would care to believe. About six months ago I bought a group of Shugart SA455 DSDD drives from Priority One that were almost totally quiet. If the same would have worked for you is hard to tell.--Ed]

TWEEZERS AND DENTAL FLOSS FOR YOUR KEYS

Dear Editor:

A strong hug for MOR and especially Lee McKusick; his article in MOR Vol. 3, #4 about keyboard key repair saved me. After one of my keys didn't register, or registered too many times, I followed Lee's clear instructions and the keycap and keyswitch came apart as he said they would. I dry swabbed the contacts with a Q-tip and voila! The key worked perfectly. May I suggest that to the list of needed supplies and tools (knife, screwdriver, dental floss) that a pair of tweezers be added? Essential for those whose well sized fingers get in the way.

Milton Wolff, El Cerrito, CA

RETIRED NEWSPAPER EDITORS KNOW

Dear Editor:

You are doing a terrific job with MOR. Please keep it up! While other industry periodicals have experimented and failed, or continually have changed formats until they have worn out the subscribers, good old MOR has steadily plunged ahead with the greatest degree of service.

MOR gets better all the time, it continually looks better and more professional. I know, I am a retired newspaper editor. But the most important thing is what it delivers: timely, informative, useful bits of information that have continuity.

Thanks to your staff for making MOR the valuable publication that it is.

Ray Nish, Modesto, CA

Advanced Concepts "Mini Winnie" HARD DISK SYSTEMS For Your Z-80 Morrow!

Our EXTERNAL systems install in seconds! Includes an attractive vertical cabinet with integral P/S and fan, 3 foot cables. Everything necessary to plug in and go!

Our INTERNAL systems do not displace a floppy drive, do not require a power supply upgrade, use the highest technology Winchester available. No compromises!

COMPLETE SUBSYSTEMS

Plug in and run! Includes software, documents, cables, brackets as req'd.

	\$465. 5mb External	\$665. 10mb External	\$790. 20mb External	\$740. 10mb Internal	\$845. 20mb Internal
Kaypro (1,2,4,10)	✓	✓	✓	✓	✓
Big Board I,II	✓	✓	✓		
Xerox 820-I,II	✓	✓	✓		
Micro Mini 5B-160	✓	✓	✓		
Gemini Zorba	✓	✓	✓		
Osborne	✓	✓	✓		
Sanyo MBC-1150/ 1160/1200	✓	✓	✓		
AVL Eagle (with mods)	✓	✓	✓		
Alspa II	✓	✓	✓		(You may delete \$90 for external systems less P/S enclosure)
Epson QX-10	✓	✓	✓		Add \$6.00 for extra long cables.)
Morrow MD-2,3	✓	✓	✓		
Z-80 S-100 Computers	✓	✓	✓		

The key to the Mini-Winnie system is the INSTALL program, which links your drive and your computer, the way YOU want it. Extensive menu options allow selection of any ST-506 compatible drive, as 1 to 4 logical drives within CP/M at any drive letter. Diagnostic and format capability provided to match the installation. Distributed in SOURCE form. Compatible with most other add-ons, including SWP CoPower, ZCPR3 and microsphere.

We have club, user group, OEM/Dealer programs. Call for quantity schedule. If for any reason you are not satisfied, return for full refund within 30 days. All systems warranted for 90 days, info and manual \$8.00 credited with system purchase. Specials subject to availability.

ORDER INFO: Call (305) 482-7302. Terms: UPS cash COD, check or M.O. Please allow 4 weeks for delivery. Fla. add 5% tax. Shipping extra. Please specify desired floppy format.



Advanced Concepts Engineering & Consulting
8926 S.W. 17th Street
Boca Raton, Florida 33433

Get what you've always wanted— Get the Proportional Printing Program which REALLY WORKS!

PropStar gives type-set quality true proportional spaced printing from ordinary NewWord and WordStar document files. Prints on daisy-wheel printers with ps type-wheels (and certain matrix & laser printers), including Diablo, Qume, NEC, Brother, C-Itch, Juki, Silver-Reed, Canon, Fujitsu & others.

PropStar is a stand-alone program, not a patch, gives higher quality print than modified WordStar. **PropStar** follows the rules for typesetting, maintains spacing between letters, never crowds text, offers continuous underline, increases letter spacing in "bold" text to avoid run-together letters. **PropStar** doesn't move text, so you get what you see on your w.p. screen, supports most W.S. print features.

Available on more than 40 CP/M formats including MD-2/3/11. MS-DOS version also available. Only \$ 49.95, Visa & M/C accepted, or send check with order and we pay postage. Comes fully pre-installed so please specify computer and printer models.



CIVIL
COMPUTING
CORPORATION

2111 RESEARCH DRIVE, SUITE 1
LIVERMORE, CALIFORNIA 94550

(415) 455-8086

Important Announcements

THE NEW MANAGING EDITOR

MOR has a new Managing Editor, Serge Timacheff, who joined us in early December. He brings to MOR a strong background in editing, writing and production and has a varied array of skills including marketing and PR that will come to MOR in good stead. Glad to have you, Serge.

Outgoing Managing Editor Marilyn has been with us for almost a year during which she almost single-handedly redesigned MOR and gave it the typeset look. We wish her much success with her new book that she yearns to spend more time on.

SUBSCRIBER DISCOUNT ON ADD-ON HARD DISK

After completing new arrangements now all owners of either the MD2 or MD3 can enjoy the group discount given MOR subscribers by Advanced Concepts for their add-on hard disks. Previously this discount was only available to those who purchased the ZCPR3 kit provided by MOR. These hard disks have to be purchased from Advanced Concepts, a MOR advertiser. You'll get an as yet undetermined discount after you contact MOR to get on the group discount list. For more information on the disks call Tony Nicotra of Advanced Concepts in Florida at 305/482-7302.

DIAGNOSTIC PRICE GOING UP

After it became clear that MOR was loosing money on the distribution of Diagnostic II software by SuperSoft it was decided to raise the price from \$37.50 to \$45 as of January 1, 1987.

NEWSTAR SOFTWARE SOLD TO MICROPRO FOR THREE MILLION

NewStar Software, the makers of NewWord, has been sold to MicroPro International and NewStar has closed shop by the time you read this. The latest CP/M version of NewWord is 2.17. MOR has not been able to reach Stan Reynolds, NewStar's past president, to comment on NewWord's future support. Rumor has it that MicroPro plans to incorporate NewWord's neat enhancements in yet another (CP/M?) WordStar. If MOR learns anything it will be posted on the MOR BBS (415) 654-3798.



CP/M is alive on my clone!

For the past year or so, I've been wanting to get a computer that would run MS-DOS/PC-DOS programs. When we moved into our new office a couple of months ago, we decided that I'd get a new computer and my wife would retire the MD-2 and replace it with my MD-11.

However, before we could really put the MS-DOS machine into full use, we absolutely had to have nearly full compatibility between the XT clone and our MD-11. We not only needed to easily transfer text files between the machines, but we also needed the capability to run some of our CP/M software on the new machine.

I did a search to see what would provide the best CP/M capability. Many people replace the microprocessor in their MS-DOS machines with the NEC V-20 microprocessor. This is a very low cost method of obtaining CP/M capability on an MS-DOS/PC-DOS machine. The V-20 costs about \$20 and the programs to run CP/M are in the public domain. Complete V-20 kits, including software, are available for around \$100. These kits usually

Send your questions and comments to Ken Grymala at P.O. Box 2540, Manassas, VA 22110.

CP/M Alive!

by Ken Grymala

include programs that will read and write to nearly all CP/M disk formats.

Unfortunately, the V-20 microprocessor will not run all CP/M programs. And, I knew I had a lot of CP/M programs that would not run on the V-20.

I began to look at a number of coprocessor boards. The coprocessor that I chose to provide the MS-DOS to CP/M capability was the RUN/CPM board built by Micro Interfaces Corporation. I purchased the board from MOR advertiser Westwind Computer for \$199. Westwind promptly sent me my RUN/CPM board. They also sent me a complimentary "CP/M Lives" t-shirt.

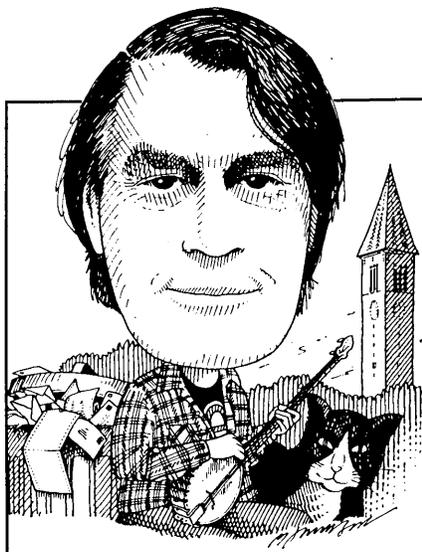
The RUN/CPM board and its accompanying software work well. CP/M capability can easily be invoked when you boot up the computer or at any time. Once it is invoked, it

remains dormant in the PC's memory until you decide to run a CP/M program. CP/M programs are run exactly as if they were MS-DOS programs. I haven't done any timing analysis, but the CP/M programs appear to run at approximately the same speed as they do on the Morrows.

The nice thing about RUN/CPM is that CP/M and MS-DOS programs can co-exist on the hard disk, the only difference being the file name extension. MS-DOS programs remain as ".COM" programs, while the CP/M programs must be renamed as ".CPM" programs. If the programs are run from a floppy disk rather than the hard disk, they don't even need to be renamed.

An extremely nice feature is the capability to set up the floppy disk drives as MS-DOS drives or to nearly any 5-1/4" CP/M format ever conceived. Since I have two floppy disk drives and a hard disk, I can set up one drive as a Morrow drive and one as a standard MS-DOS drive. I can then easily copy directly from one format to another much easier than I could ever do with Uniform or Multiform on my CP/M machines.

Most of my CP/M programs work well with the RUN/CPM board. NewWord is one of the few CP/M programs that I have found that won't run properly. However, CP/M



More from John William Johnson of Bloomington, IN about using DICTEDIT to add the words in AUXDICT.TXT to the Correct-It master dictionary. He sent along a printout that suggests he's doing everything right, but says the new dictionary still doesn't contain the words he wanted to add. The printout lists a lot of words from which the final letter has been dropped (e.g., "collaboratin," "ethiopi.") If words are being added in truncated form, of course Correct-It wouldn't recognize the complete form. Has anyone else ever encountered an error of this sort? With this as with a lot of other problems, one approach is to find someone else with a Morrow and try running their software on your files, or vice-versa, to find out if the problem is with the program, the file or the computer. If all else fails, I'll try anything once, if you send a disk and return postage.

Mr. Johnson also can't get the public domain sideways program PDSW on Aug. 86 FLOB No. 1 to run with his MD3 (ROM Rev. 3.1,

We're more on our own than ever, so if you know a solution to a problem, don't assume someone else will write in; there are not so many of us that you can count on that. Whether you have questions or answers, keep them coming: Bill Steele, P.O. Box 782, Ithaca, NY 14851, 607/273-2132; MCI: WSTEELE, 254-5833.

From the Mailbox

by Bill Steele

CP/M 2.2, Rev. 3.1, Liberty 50 terminal and Epson FX80 parallel printer). "PDSW.COM freezes. I ran ASM on PDSW.ASM, resulting in a PSDW.HEX and a PDSW.PRN file; PDSW.COM converts from a 2K file to a 0K file. I ran LOAD on the PDSW.HEX file and got an error message--"ERROR: INVERTED LOAD ADDRESS, LOAD ADDRESS 0000.'" According to the CP/M manual this may be due to a weakness in LOAD.COM. Try loading the program with DDT. Call DDT with the line "A>DDT PDSW.HEX." When it loads you'll see the following:

```
NEXT PC
XXXX 0100
```

This means the program starts at 0100 hex and ends just before XXXX hex, whatever number that actually is.

Subtract 100 from XXXX (using hex arithmetic) and you get the length of the program, divide by 100 hex and that's the number of 256K blocks the program needs (because 100 hex = 256 decimal). Exit from DDT with ^C, then type "A>SAVE XX PDSW.COM" where XX is the number of blocks you calculated.

However, if the original PDSW.COM didn't run, there's no guarantee the one you get this way will do any better; it may be freezing because it's not set up correctly for your printer or terminal. Anyone else have experience with this program?

Another way to find the number in the SAVE command (for this or any other work with DDT) is to run STAT.COM. Put the CP/M system disk in A: and NW disk in B: and type "A>STAT B:NW.COM". The number under "Recs" will be the number of 128K blocks the program occupies; use half of that number in the SAVE command. I just did that on NW Ver. 2.16 and got 54, so the command to use when patching NW would be "SAVE 26 NW.COM".

Today's admonition: with the present state of the art in computers, you really can't manage without learning to use hex numbers.

Jim Rosack in Englewood, Colorado can't get MagicPrint to run on his MD2 and MP100. He installed for CP/M List Output Driver and assumed the MP100 emulates the Diablo 1610/1620. "According to the instructions I was supposed to pick the

ETX/ACK communications protocol, however my WordStar runs with no communications protocol, so I tried both. Neither worked. MagicPrint prints a single character at the top of the page followed by an underscore, then spits out 70 lines of nothing, then prints about 10 characters on top of one another." That sounds a lot like what my Teletext (Olympia) printer did once when I installed NW for a Diablo printer I had been told it emulated; I suspect the MP100 doesn't emulate one either.

As I've said before, questions on the same topics seem to bunch up. Yet another installation problem from **Dennis Cogswell of Radford, VA**, with the game "Wordwiggler," which produces bizarre patterns on his Televideo 950. This program comes with "configuration files" for the Osborne and Kaypro. For any other computer/terminal arrangement you have to write your own file, using the non-document mode of your word processor. What you put in this file are the graphics codes your terminal must receive to print the characters for drawing a crossword grid on the screen, i.e., vertical and horizontal lines, corners, intersections, etc. MDT60 and MD70 terminals use single-character codes numbered above the regular ASCII code for graphics characters. On the MDT50 you have to enter an escape code that turns on graphics mode, a code for the letter of the alphabet corresponding to the graphics character you want, then another escape code that turns off the graphics mode (because there may be a regular letter in the next space). I don't know which approach the Televideo uses; check your terminal manual.

Mr. Cogswell also writes that he's thinking of buying a Commodore 128 for his son. The manual says it reads and writes a number of CP/M formats, including Osborne SSDD, and wonders if it will read Morrow format. Probably not, but your Morrow will read and write Osborne format, with

OSB.COM. However, it won't format an Osborne disk, and the Commodore probably won't either, so you'll have to get someone else to format some disks for you before you can exchange anything. Some computer supply houses sell pre-formatted disks, and the programs Uniform and Multiform, discussed previously in MOR, will format disks in Osborne and other formats.

With Morrow out of business, **Bradley H. Kline of Franklin Park, NJ** is wondering if he should buy an MS-DOS machine, and asks if he will be able to use his old files and programs. Files can be transferred to MS-DOS disks with programs like Uniform and Multiform. To use WS/NW files directly you would have to be running an MS-DOS version of WS or NW. (The MS-DOS NW was highly rated by *PC Magazine*, incidentally.) For other word processors you need conversion programs. Logicalc and Pearl files can't be used by other spreadsheet and database programs, but you can usually write reports from them that other programs can use; for instance, you can write a report from Pearl that can be loaded into dBase II. Of course none of your CP/M software will run on the new machine, but WestWind, a regular MOR advertiser, offers IBM clones with an optional CP/M board.

James E. Bottom of Salem, OR asks if adding double-sided drives to an MD2 and patching CP/M really gives you an MD3, or "does the earlier version ROM (Rev. 1.3 in my case) give you limitations?" He can get a Rev. 2.0 PC board, ROM and CBIOS from Silicon Valley Surplus for about \$65 and wonders if it's worth it. What makes an MD3 is double-sided drives and a Centronics port, so I guess it depends on whether or not you need the extra port. Also note Denis Kuwahara's comment in the last issue about a software fix for head stiction in newer ROM's.

Having earlier asked which

terminals were made by whom, Mr. Bottom now also asks about the secret identities of Morrow printers, noting that they are now on sale at good prices. The MOR staff came up with the following (and said it was surprising how hard it was to get this): The MP100 seems to be a Silver Reed EXP 500, the MP200 is the EXP 550 and the MP300 is the 770. Actually it seems these were made and shipped directly to Morrow by Silver Reed's parent company in Japan, Silver Seiko, so there could be minor differences between the Silver Reed and Morrow versions. The MP100 and MP200 are essentially similar, except that the first has an 11 inch carriage and the second a 15 inch carriage. We can't find anyone who has seen an MP300; the "equivalent" EXP 770 was apparently never sold in the U.S. In a quick scan I could find only one New York City dealer advertising any of the Silver Reed printers, and that was the EXP 550 at \$279.95.

Meanwhile, **Laurence A. Moore, Jr. of Gualala, CA** disagrees with my earlier answer to Mr. Bottom about terminals: "I have an MDT60 terminal and an MD3 computer. They came together from Morrow in San Leandro. Both Morrow and my dealer said it was an ADM-31. Both Morrow and my dealer have been known to speak with crossed wires. I've always entered ADM-31 when asked what my terminal is, and it has always worked." Well, I'll stick to what I've been told many times: the MDT-60 is basically a Zenith Z-29. Maybe Morrow released a few ADM-31's with that label on them; anybody know for sure? One way to check: I don't think an ADM-31 would have the same function key codes as the MDT-60/Zenith. Other than that, most popular terminals use the same control codes for functions like moving the cursor, clearing the screen, etc., so you can often install for one or another and get the same results (although there are subtle differences, as pointed out in

relation to WS losing characters).

John A. Birmingham of Broken Arrow, OK writes that when he tried programming function keys to work with NewWord 2.17 (as described in my article in Vol. 1 #4, Oct. 1984), functions involving ^T wouldn't work. For instance, ^T^A^U (to exchange words) would delete the second word and just stop. I had the same problem when I tried to put my function key table into the upgrade. As the 2.xx versions come from the factory, they are set so that the type-ahead buffer stops accepting characters after all the delete functions and the up and down scrolling functions. The reason is that if you hold down, say, ^T or ^Y on an auto-repeat key you may delete more words, lines or whatever than you meant to. Holding down a scrolling key may cause you to overshoot, and a friend of mine recently reported that holding down ^X too long caused NW to crash. The

buffer action is controlled by the "ahead flag," which is at address 0786 in 2.xx versions and is on Special Patch Menu #2. Changing the byte at this address from 00 to FF will allow your function keys to work with the delete and scrolling functions.

Douglas Baker of Wilton, CA, who wrote previously that KEY.COM wouldn't work with his new ROM chip (Vol. 3 #4) found his own solution "The newer KEY.COM has a 'Fast' vs. 'Slow' mode; all I had to do was add a 'SWITCH FAST' line to my definition files." He found the answer in the upgraded Hard Disk Micro Decision manual. See his letter on page three, last issue.

Timothy Slater writes all the way from West Germany "Like everybody who writes in languages other than English I was delighted to hear that this seven bit ASCII code is finally going to be replaced by an eight bit code that provides accents and

other signs. But since I hear that CP/M 2.2 resets the 'high-order' bit, the question of compatibility arises. Would it be enough to switch to CP/M 3.0 to use the eight bit code, and is this possible for an MD2? Which of our favorite 2.2 programs would not run under 3.0?" As previously noted in this column (Vol. 2, #6), CP/M 3.0 comes with a 2.2 "emulator." I'll leave the compatibility questions to experts like Ken Grymala. Seems to me the big advantage of the eight bit code would be in controlling the printer, and you probably could use an eight bit printer with any of our software by writing a new printer driver for it.

Someone called my answering machine a while back offering an answer to Gary Andersen's request for software to convert an MD11 to a larger capacity disk, but I must have copied the number down incorrectly. Please call or write again. ■

THE 10-MINUTE 1040

TAXWORKS™ Your Fastest, Easiest Tax Preparation Program

GOODBYE TO SERVICE BUREAUS

You'll save money if you're now using a service bureau. And instead of waiting for pickups and deliveries, you can run a return in 10 minutes at any time of day... or night. If you now prepare returns manually, TAXWORKS will turn out more returns in less time, with greater accuracy... and a lot less aggravation!

EQUIPMENT REQUIRED

IBM-PC, MS-DOS or CP/M-80 Computers. 80 Column Printer

PROFESSIONAL FEDERAL & CALIF. _____	\$500.
PROFESSIONAL FEDERAL ONLY _____	\$400.
SELF-EMPLOYED _____	\$175.
INDIVIDUAL _____	\$100.

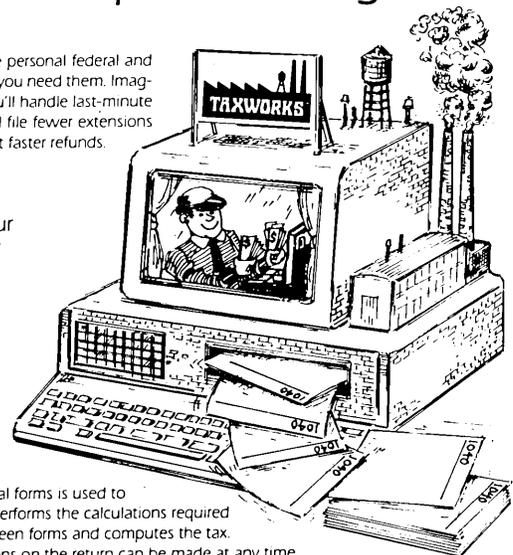
NO TURNAROUND TIME

TAXWORKS gives you fast, accurate personal federal and California returns when and where you need them. Imagine what this will mean in April: you'll handle last-minute returns and changes on-time. You'll file fewer extensions and corrections. Your clients will get faster refunds.

- Complete Returns on Your Own Personal Computer — No Waiting Time!
- 10-Minute Entry Calculation, Print Out

SIMPLE TO OPERATE

A screen format similar to the official forms is used to enter tax data. The program then performs the calculations required on each form, transfers totals between forms and computes the tax. Corrections and last minute additions on the return can be made at any time. The completed tax return can be printed on official tax forms or on blank paper for xeroxing with plastic overlays included with the program.



FOR MORE INFORMATION, IN CALIF. CALL:
1-800-232-1040



**HOLLINGSWORTH
BUSINESS SERVICES**
881 Alma Real Dr. Suite T-2 Pacific Palisades, CA 90272
(213) 459-2746

Well, folks, as I am writing this it is coming up on the Christmas holidays and I am, as usual, behind the power curve. I'm still learning about my MD5 and really enjoying this fine machine. I've been having some spirited discussions on the BAMDUA BBS about the pluses and minuses of CP/M vs. MS-DOS and am gaining quite an education. I'd like to suggest that the various BBSs are great learning forums. It seems that all it takes is for someone to make some controversial statement and many opinions come out of the woodwork. There are some very knowledgeable people "lurking" around the BBSs and all that is needed is for someone to attract their interest. A lot of these discussions are not technical. There was one "thread" that had to do with the driving habits of Bostonians. Fascinating stuff. Reminds me of the way Ham Radio used to be when I was first licensed back in 1954. The thrust of this paragraph is that computers can be wonderful communications devices for people. If you don't have a modem and communications software you are missing out on an exciting experience.

Let's talk a little about the FORMAT program. FORMAT is a transient command. That means that it resides on your disk as the file FORMAT.COM. Now what does FORMAT do for you? When you get a new disk, it has nothing at all on it. It is like a blank audio tape. From my

Send your questions and comments to Mike Allen at 400J Summer Wind Way, Glen Burnie, MD 21061.



Tools for Tyros

by Mike Allen

last column we found out that for our computer to use a disk it must be organized in tracks and sectors. What the FORMAT program does is write a lot of information on your disk identifying the various tracks and sectors. Then CP/M can find where it is supposed to write and/or read information. It is kind of like taking a blank notebook and putting page numbers in it. This is not all done without cost. As I mentioned in the last column, a double sided MD disk can store 400k of which 16k is tied up in the system tracks and the directory. Actually there is 500k on the disk. 100k is used in formatting.

To use FORMAT, just type

```
FORMAT<cr>
```

and then answer the questions. It will ask you which drive to use and if you have an MD3 or a MD-HD

whether to use a single or double sided format. The single sided format is used for MD1s and MD2s. BE CAREFUL and make sure you select the proper drive. If not you may format a disk that already has data that you want. Formatting a disk destroys what ever data was on the disk. This can be a real disaster in the case of a hard drive.

There is a short-hand method of using FORMAT. Just type

```
FORMAT B S<cr>
```

to format a single sided disk on drive B.

SYSGEN is the program for the floppy MDs that places a copy of CP/M on the system tracks of the disk. This copy of CP/M must be there if you are ever going to use that disk in drive A. You don't gain any disk space by not doing it so you might as well do it on all your disks.

When you invoke SYSGEN it will ask you where to get CP/M from. Answer "A". Then it asks you where to put it. Type the appropriate drive. You can SYSGEN a bunch of disks from one calling of SYSGEN and it doesn't take much time.

COPYSYS is the CP/M 3 version of SYSGEN. If you are using a MD-HD this is the program that does the SYSGEN function. Hopefully you will never need to do this to your Hard Disk, but it is possible. Since the MD-HD can boot from a floppy, you need to use COPYSYS on any floppy that you may wish to boot from (like your back-up floppy.). COPYSYS is slightly more complex. To make a

floppy with CP/M on it (a "bootable" disk) you type

```
COPYSYS SYSLDRF.COM<cr>
```

The program will ask you for the destination drive. Enter "B:<cr>" and away it goes. Like SYSGEN you can repeat for a bunch of disks. Unlike CP/M 2.2, CP/M 3 requires another file called CPM3F.SYS for the floppy. Use PIP to copy this file to your new bootable disk. If, Lord forbid, you ever have to reformat your hard disk you will have to accomplish the same operation on it. There are two differences. With the COPYSYS command use SYSLDR.COM and not SYSLDRF.COM. Also copy CPM3.SYS not CPM3F.SYS. The "F" in those two file denotes "Floppy" and are for booting from a floppy. Without the "F" they are for the Hard Drive.

AUTO is a neat little program that allows your floppy MD to execute a program directly when you first turn it on or when you punch the reset button. To use it is very easy. For instance, type

```
AUTO *B: NW<cr>
```

to make the disk in drive B go directly to NewWord if that disk is in drive A when you turn the MD on. There is some discussion of AUTO in the MD owners manual, but they forgot to mention that you can tell the program which drive the disk you want to set up is on by including the "*D:" in the command line where "D" is the destination disk.

CP/M 3.0 for the MD-HDs is a little more versatile. There is a file on your boot disk called PROFILE.SUB. You can edit this file in your word processor in the non-document mode.

This file contains a bunch of commands that you want executed whenever you turn your system on. You can put most anything you want in there. Use your imagination!

I said that I was going to delve into the differences between some of the CP/M 2.2 and CP/M 3.0 commands but, to be honest, I'm running out of time. One quickie. The STAT command doesn't exist in CP/M 3.0. Use the SHOW command to see the type of things you could display with STAT and use the SET command to set the things you could set with STAT. For further information, go

```
HELP SHOW<cr>
and
HELP SET<cr>
```

I hope the New Year sees all of you happy and well. My best of the season to everyone and keep on computing. ■

Out - Think

By Kamasoft

Logically Outlines Your Thoughts and Work

Out-Think is a powerful *idea processor* that takes full advantage of CP/M. Capture your ideas and organize them ... Out-Think supports your thinking process by naturally working the same way your mind does.

The Outline processor is simple, easy to use, and will appeal to all users.
-- Mark Renne,
InfoWorld Magazine

Brainstorm. Track your thoughts with full screen editing.

Develop your thoughts into outlines that follow a natural, logical flow.

Collapse the details and take an overall look.

Expand the outline to see detail.



Outline Processing
Word Processing
Information Retrieval
File Management
Documentation

Special Price - \$45
Out-Think retails elsewhere for almost \$70, but through MOR you get almost \$25 off!
Order MOR Product #805 on page 28 of this issue.

WordStar is a registered trademark of MicroPro, Inc. CP/M is a registered trademark of Digital Research, Inc. Kamasoft and Out-Think are trademarks of Kamasoft, Inc.

I'd like to dedicate this first column to the many Morrow owners who have ordered the special bootable Z-System disk. The response has been very gratifying, and I hope to offer tips, hints and general "Z" miscellany to all intrepid explorers embarking on cosmic adventures through the Z-System skies. I can't guarantee that you'll be the same person when you return, but perhaps I can help with the navigation a bit.

I will cover two topics for this first column, as they will likely be the most useful and enjoyable programs you have: shells and aliases.

SHELLS

There are three shell utilities on the Morrow disk - VFILER, VMENU and MENU. All Z users use one or more of these daily and find them indispensable and quite extraordinary.

What is a shell? In a way, you could say it's a secondary operating system. Once you run a program that's a shell, you can then run other programs from it. These other programs, rather than returning to your operating system prompt upon exit, return to your shell utility.

Shells exhibit one of the most intriguing aspects of Z-System ... the way its disparate parts are related to each other in a thousand subtle, and generally user-unnoticed, ways. Z is a

Rick Charnes bought his MD-2 many years ago when he thought of himself primarily as a writer. Now a self-admitted fledgling technical type, his computer has turned from tool to toy. His immense love affair with ZCPR3 leaves some occasional time for mundane things like work. He is president of BAMDUA.



Forever Z

By Rick Charnes

unique operating system because it has such a large number of utilities that feature this dynamic and two-way relationship.

You'll often hear the word "utility" bandied about in "Z" literature, and here it has a specific meaning. You see, these programs are sort of lonely by themselves. In order for them to run at all, they first have to be given the "imprint" of the operating system. Dozens of tiny little messages are constantly being sent back and forth between the utilities and the operating system. General information is shared about the "system environment," as well as information specific to each program/operating system interaction. It's somewhat like school children passing notes to each other in class while no one else sees it going on ... that's the kind of information that makes the world go round, and Z-System run.

The most useful shell for everyday use and the one which you will use from the first day you boot up, is VFILER. If you can imagine

NSWEEP, itself an extraordinary program, taken to the 100th power, you will begin to have some idea of what it is. Perhaps the thing you might want to do with that magical moment when you've got Z-System installed for the very first time is to just type V-F-I-L-E-R, carriage return, and stand back.

The first thing you'll notice is that your files are displayed in five columns, unlike NSWEEP which gives you one. This way you can generally see the contents of an entire disk in one screen. You'll notice many of the familiar regular and "mass" operations of NSWEEP - copy, rename, delete, etc., but you have much more. You have the ability to "move" files from one disk/directory to another. Files are copied to the new area, then the old ones deleted. The features, however, that makes VFILER and the other shells shine are their menu functions, as I mentioned in my article last issue. Any kind of operation can be done from these menus, and in fact for a challenge I sometimes see if I can run my entire computer session from VFILER. It can be done, and the process is most educational. You can write these menus yourself or start off with the ready-made ones available from Z-Nodes in the file DEMO.LBR.

I have to warn you, though ... if you start enjoying yourself, you'll find yourself spending half your time "putting the finishing touches on" your menus instead of doing what you set out to do.

With VFILER you can write 36 different command lines that can be run either by pressing any of the 10 number keys alone, or by first pressing your escape key and then any of the letters. Remember the concept of a "pointer file." This means that

your command line will act upon whichever file VFILER's friendly "pointer" (see it there?) is pointing to. You reference this file in your command line with "%f" for the entire file, "%n" for the filename and "%t" for the file type.

Here's a sample command line I use for opening up and inspecting the contents of an .LBR file. Remember, you're generally allowed 200 characters in your command line, which is about two and one half screen-widths. Think of what you can do with that! For the purposes of this article, of course, I can only show it on two separate lines, but when you type it in you enter it all on a single line. There's a Z-System tradition to display long lines in print form by separating them with ">>" and that's what I've done here. I invoke this "macro" (command line) with "N", since it uses NULU. Individual commands are always separated with a semicolon. (Fig. 1)

To analyze: The initial "N" indicates the character that will invoke it from VFILER. You actually will be typing <ESC>, which is the lead-in character, then "N". Then VFILER checks the file type with the "IF" command to make sure it is indeed a library. If so, it goes to the next command, which "echoes" the sequence "<ESC>G4" to my terminal.

```
N if %t=libr;echo ^[G4 OPENING UP %F AS A LIBRARY...^[G0 >>
sak /p4;nulu -o %n -f;else;echo %f is not a library;fi
```

Figure 1

If you read my article last issue you know how much I like reverse video, which this enables. The text message tells me that it's opening the file up, appropriate substituting the library name for "%F" in the echoed message. Then "<ESC>G0" which appears here and on your screen as "^[G0" turns off the reverse video.

Because it looks so nice in reverse I like this attractive message to remain on the screen for a few seconds, so the

"SAK /P4" tells the program SAK to pause operations for four seconds. Then NULU awakens and opens up the file, substituting its name for "%N". If the file is not a library (sometimes you'll hit "N" by mistake, I assure you!), the "ELSE" command takes over and we're echoed a message to that effect. The "FI" command is the equivalent of "ENDIF" and ends the "IF" interaction.

VFILER is an extraordinary program and I'm quite sure you'll find it to be one of the most treasured friends in your arsenal of Z tools.

The other two shell utilities are MENU and VMENU. The difference among this trio is interesting. VFILER shows you a screen's worth of files, or by hitting the "#" key the menu, but not both at the same time. MENU is at the other extreme. You never see your files, but you have much more room for descriptive material of your menu. Such beautiful displays are possible! In my word processing menu for MENU.COM I am able to edit the pointer file with NewWord, edit it with Wordstar, correct it with Correct-It, print it with my print spooler, soften it with HRDSFT, filter it with FILTW.COM or count the number of words in it --- each with a single keystroke.

VMENU is right in between the

two and is lovely in a different way - the top half of your screen fills with files and the bottom half with your menus. Yes, that's menus -- plural; 255 menus are possible in both VMENU and MENU. You move from menu to menu, and list of files to the next, with a single keystroke. Ah, what infinite variety of forms in this Z world!

I am sure that you'll find uses for all three utilities and will enjoy

learning about and taking advantage of the special features of each.

ALIASES

An alias is essentially a submit file wrapped up into a single .COM file package. You'll probably spend hours dreaming up favorite combinations of commands you want to stuff into one, so I'll only tell you one of my simplest and most favorite. I've always disliked CP/M's reverse "statement formation logic," as in the following:

```
PIP NEWFILE=OLDFILE
```

... so I've taken Z-System's resident copying command, CP, and made myself a nice MS-DOS style copy program called appropriately enough, COPY.COM. To wit:

```
CP $2=$1
```

When I want to copy LOVELTR.TXT to drive B0:LETTERS> (remember, we can give names to our directories in Z-System) I can type ...

```
COPY LOVELTR.TXT LETTERS:
```

Ge -- it makes me feel like I have a IBM PC...

Again, your only thing to watch for is the 200 character buffer, so you'll enjoy learning how to put the most in without hitting that magic upper limit.

I hope you are having a excellent time with Z-System. Don't forget to log on to your nearest Z-Node for the latest in the large number of excellent public domain utilities available. This grand adventure is just beginning.

Next time I'll cover, among other things, an amazing program called HSH that will actually let you recall and edit previously issued command lines as well as the one you're currently writing.

Z you next time... ■

Given a Choice, I'll Take Pascal!

by Ken Grymala

I guess I'm fortunate. I get to work in a variety of computer languages, and since I do people frequently ask me which is my favorite language. Quite often I also get asked what the best language is if someone decides that they want to learn a computer language.

My answer to these two questions is almost always "PASCAL!"

As you may have noticed, I said "almost always." There certainly is no "best" language. There are many languages available, and each has its good and bad points depending on the application. However, I feel that most people would agree that Pascal is an excellent language to begin with if you are just learning programming. As it has matured, it has also become an excellent all-purpose language that is available for nearly every type of computer from the Commodore to the largest, fastest mainframe.

The original purpose of Pascal was for teaching good programming habits and practices. It is known as a "structured" language, meaning that it forces you to conform to a very specific format.

Most Pascal versions require that the first line of your program contain its name. Next, you must define all of the constants and variables that you will use. If you have several subroutines needed during program execution, they come next. Each of these routines must also be given a proper name. Last of all you have your main program. The main program and each subroutine must also have a "begin" and an "end" statement.

Pascal is also known as a "self documenting" language. This is kind of a misnomer, but if a little care is taken when a Pascal program is written, it can be easily read and understood, even by someone who doesn't know Pascal or programming.

Ken Grymala authors the MOR monthly column CP/M Alive! He loves cats, CP/M and hot new technical theories.

The listing accompanying this article is the "source code" of a Pascal program named "EXAMPLE." This is a rather useless program that only adds two numbers together and then multiplies the result by 1000. But, it demonstrates the layout and "self documentation" of a Pascal program.

If you are a BASIC programmer, first you will notice that there are no line numbers. Secondly, notice that the constant, the variables, and the procedures all have understandable plain-English names.

The main part of the program lies between the lines "begin {Main Program}" and "end. {Main Program}."

Again, notice that the program can be read like a plain-English outline. If you read each of these lines, you can refer to the purpose of that line by reading the procedure that corresponds to it.

The "source code" of a Pascal program generally must be "compiled" before it can be run. That is, the source code (written part, e.g. the listing below) is turned into computer code that is executable by the computer. The compiler, which is part of any Pascal package, strips out all the comments and spaces, and turns the written information into a very compact set of commands that the computer can understand. Most Pascal compilers are also very good at finding and reporting any typing or programming errors that you may have made.

There are a number of versions of Pascal available for CP/M. There are even several public domain versions available. But the all time favorite of mine, and of most everyone else, is "Turbo Pascal" by Borland International. As of this writing, CP/M Turbo Pascal lists for \$69.95. There have been so many excellent things said about Turbo Pascal in the dozens of reviews written about it, that I won't extol its features.

If you should decide to start with Pascal, or if you are already into Pascal, I have an excellent book to recommend. "COMPLETE TURBO PASCAL" written by Jeff

Duntemann and published by Scott, Foresman and Company is the best tutorial and reference book on Pascal I've run across. The book is available in most book stores that carry computer books and it lists for \$19.95.

I even recommend "COMPLETE TURBO PASCAL" over Borland's Turbo Tutor if you are just learning Pascal. While Turbo Tutor is an excellent way to learn Pascal, if you are a little short on cash, or unsure that you want to get into Pascal, buy Duntemann's book first. In my opinion it provides much clearer and better real-world programming examples than does Turbo Tutor which costs more than twice the price (list \$39.95) of the book. Turbo Pascal is available from MOR advertiser Central Computer Products, 330 Central Avenue, Fillmore, CA 93015 at discounted prices or directly from Borland International at the normal list price.

As mentioned, other Pascals are available. Some are less expensive, and some are more sophisticated. But there are so many users of Turbo Pascal, that it has really become the de-facto CP/M standard Pascal. There are also many hundreds of public domain programs available in Turbo Pascal source code.

```

program EXAMPLE;
{ Example PASCAL program
  Written by: K. Grymala}

CONST
  K = 1000; {Thousands multiplier}

VAR
  first_number,
  second_number,
  result,
  result_K : real {numbers};

{PROCEDURES}
procedure Get_First_Number;
begin
  Write ('Enter a number: ');
  Readln (first_number);
end;

procedure Get_Second_Number;
begin
  Write ('Enter a number: ');
  Readln (second_number);
end;

procedure Add_Numbers;
begin
  Result := First_Number + Second_Number;
end;

```

```

procedure Convert_to_thousands;
begin
  Result_K := Result * K;
end;

procedure Display_Results;
begin
  ClrScr {Clear Screen};
  Writeln ('Results');
  Writeln ('First Number + Second Number = ', result:9:3);
  Writeln ('Result expressed in thousands = ', result_K:9:3);
end;

begin {Main Program}
  ClrScr; {Clear_Screen}
  Get_First_Number;
  Get_Second_Number;
  Add_Numbers;
  Convert_to_Thousands;
  Display_Results;
end. {of program}

```



See page 47 for a complete reference list.

Diagnosics Software

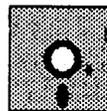
By SuperSoft

*You Need To Know
What's Going On.*

Keep track of every part of your system ... at a moderate price! Test each major part of your system:

- Disk Drives
- CPU
- Memory
- CRT
- Printer

Don't *guess* what your system is doing. Order MOR Product #802 on page 28 of this issue today for only \$45!



For only \$25, the complete **Diagnosics User's Guide** can also be yours. Order MOR Product #702. The complete manual to diagnosing your computer system.

The Simple Beauty of Turbo Pascal

By Lee McKusick

At one time I worked in a machine shop making machines. Similarly, as a programmer I now think of Turbo Pascal as my own personal machine shop for building software gadgets. On both my CP/M Morrow and my MS/DOS machine, Turbo Pascal is a major tool. I credit Turbo Pascal as my first choice programming language. I credit Pascal as my favorite environment for mastering the concepts of computer science.

For example, when a program in "Computer Recreations," a monthly column in *Scientific American*, catches my eye, I choose Turbo Pascal for playing around. Having worked as a database programmer, my work has required me to use existing database languages. The ideas I mastered with Turbo Pascal are present in the judgment and understanding I apply to

Lee McKusick uses a Morrow MD3-P to enhance his computer science skills. He is developing a career in technical writing, typesetting and programming in Pascal, dBase and C.

each programming task. As one learns Turbo Pascal, several stages of learning become apparent.

The first stage involves acquiring the software, installing it and then spending about 20 hours reading and running programs from Turbo Tutor or another tutorial book.

Next, beginning programmers go through the "It's beautiful! It's powerful! What can I do with it? How do I start?" experience. Passing through this stage requires copying and playing with existing programs, and it requires focusing on simple programs. It is hard to accept how simply Pascal accomplishes complex things, and the new user will endure a blizzard of error messages while learning that all variables must be declared before using them in Pascal.

In the next stage, the user designs and develops major programs focused on his or her own needs building a library of Pascal procedures and books suited for personal interests. At this level, some Pascal programmers write copyrighted and compiled commercial programs. I read an ad once stating that

the advertiser used Turbo Pascal for creating CP/M products. QMODEM, a shareware IBM communications program, is written and compiled in Turbo Pascal. Also at the third level are many programmers who create interesting programs used privately. Pascal is a wonderful language for teaching computer science and for doing all kinds of unusual things on a Morrow computer. Pascal programs have potential beauty that is satisfying in itself.

Acquiring Turbo Pascal

Buy Turbo Pascal from a vendor who sells software for your specific computer. Borland's Turbo Pascal Version 3 is available in a Morrow MD2-MD3 disk format for the CP/M operating system.

If you purchase the older Turbo Versions 1 or 2, I recommend that you promptly upgrade to the latest version, currently 3.X. This version's manual is better and includes useful features added to the program.

Turbo Pascal is sold for three

different operating systems. The operating systems are: CP/M, IBM-MS-DOS and CP/M-86. Even though Borland offers no low cost upgrade between operating systems, most CP/M Morrow Turbo Pascal source code compiles perfectly on MS/DOS. A file conversion utility such as Uniform is required to transfer the source code without retyping.

Turbo Pascal 3.0 comes with a difficult manual, frequently requiring you to hunt for information when learning Turbo. I add entries to the manual index with a pencil and I use yellow adhesive notes to mark important pages.

If you are a beginner, I recommend Borland's Turbo Tutor Version 2 book and disk. I learned Pascal with Turbo Tutor Version 1, but most reviewers say Version 2 is much better than Version 1.

The third part of the Turbo Pascal CP/M program set is the Borland Turbo Database Toolbox. I tried to save money by not buying the Toolbox, but I finally ordered it on a rush basis and paid an extra handling charge when I suddenly understood and needed the Toolbox's GINST. The Toolbox is often sold in a three program Turbo Pascal bundle.

If you have a limited budget, you can try Pascal by borrowing it. Turbo Pascal's software license allows the program to be loaned provided only one person runs Turbo Pascal at a time. You will want to borrow the essential reference manual, too.

Notes for Using Turbo Pascal

Turbo Pascal is used by Morrow owners with single sided, double density disk drives (MD-2 computers) with little trouble. The main Turbo Pascal CP/M programs total about 80k.

Turbo Pascal comes with the program "TINST" for terminal installation, which customizes Turbo

to use the display abilities of your computer's terminal. TINST has 24 terminal choices but, unfortunately, the Morrow MDT-60 is not among the choices. To install Turbo Pascal on my Morrow MD-3P (which has an MDT-60 terminal) I selected "Televideo 912" as the closest choice. A few further changes to the TVI terminal codes made a pleasant MDT-60 display. I have included the essential codes from my working MDT-60 installation file below.

NOTE: TINST.COM screen installation cursor codes for MDT-60 are developed from the TVI-912 terminal description. Items not listed are blank or as in the TVI file.

```
Cursor lead in command.....Esc =
Offset to add to line.....32
Offset to add to column.....32
Binary address.....Y (yes)
Clear screen.....Ctrl-Z
Does clear screen
home cursor?.....Y (yes)
Delete line command.....Esc R
Insert line.....Esc E
Erase to end of line.....Esc T
Start highlighting.....Esc G 0 (zero)
End highlighting.....Esc G 2
```

Here are the only two bug tips I can offer for Turbo Pascal. Problems with your own program can affect the editor. If the Turbo editor screen is scrambled, you can clean it by scrolling down one page and then up one page with the ^C and ^R commands.

If your program incorrectly indexes a pointer off the end of a linked list, you may encounter strange "variable not found, file does not exist" messages. A pointer indexed off into outer space may not be shown until your program is run for a second time. I had a program with pointer problems that acted oddly and I couldn't figure out what was happening until I reset the entire Turbo system. Execute an external command to reset Turbo Pascal following a problem by running the eExecute command. After

an execute, Turbo Pascal reloads itself and recovers from the damage done by the wayward pointer.

How to Select Your First Supplementary Pascal Book.

You will discover that the Turbo Tutor covers only the elements of Turbo Pascal. When I finished the Tutor I was unable to organize my own project and I needed ideas and example programs from other Pascal books to pull it all together. Many publishers offer "How to Program in Turbo Pascal" books. Here are suggestions to help you select a supplementary Pascal book.

You do not have to choose a book specifically written for Turbo Pascal. For reading, standard Pascal is just fine. There are scores of Pascal textbooks you can mine for concepts and program ideas.

To select a good Pascal book out of the dozens offered in a bookstore, first select two or three books you like. Your first supplementary Pascal book should devote at least one chapter to pointers and one complete program to demonstrate pointers. If the book does not fully explain pointers, consider another book. Once you have found a few books with enough pointer information, compare how each author explains them.

My Pascal library has only two books, picked from the reference list in Turbo Tutor. According to my program needs, I study one chapter at a time and run the example programs, adapting the ideas for my own program designs.

Extend your personal library by using the public library. The Pascal shelves may seem bare because the best books will be checked out by other students of Pascal, so place reserves to snag the best Pascal books. All dialects of Pascal are close enough to Turbo for study and learning. ■
See page 47 for a complete reference list.

Cheap Disks

Low Cost May Not Mean Low Performance

By Ted Jensen

So, you just spent \$400 for that super software package, copied the diskette and are working away with your backup copy. Suddenly, while using a relatively unimportant utility program on another disk, your disk fails. This is not a major problem. You have a backup copy somewhere, but you begin to think about your commercial program backup copies. What happens if they go bad? Should all your backup programs be copied onto premium diskettes to avoid problems?

You scour through catalogs and ads in magazines. There are sources galore for diskettes ... at all prices, and some even with specifications. You run into one "spec" called *clipping level* and the supplier claims that his disks are superior because they have been tested at a higher clipping level. Should you

*Ted Jensen is an electronics engineer specializing in the field of magnetic recording and is a computer hobbyist. He appreciates suggestions and feedback and can be reached through the KAY*FOG PCBBS (see note at end of article) or by mail at P.O. Box 324, Redwood City, CA 94062. If you wish an answer by mail, please send a SASE.*

pay a premium for disks with superior specifications? What does this actually mean? Will your backups be less likely to fail if you use premium disks?

These are difficult questions to answer, and perhaps an explanation of the tests run on disks and what can happen to backups over time will help you make your decision. In addition, you may want to consider the cost tradeoffs of using higher priced disks.

As an engineer with many years of experience in magnetic recording I had never heard of *clipping levels* until it came up in a discussion on the KAY*FOG BBS. In fact, I had never seen a specification sheet in any box or bag (I buy the cheap stuff by mail order, too) of disks I have purchased. However, I did spend a few years with a design team on a Winchester Drive for personal computers and one of my tasks was to develop specifications for and test the disks used in those drives.

Since magnetic media are mostly the same whether on tape, diskette or hard disk (the major difference being that materials to which the magnetic particles are bonded is mylar for tape and diskettes, and aluminum for hard disks), it wasn't difficult for me to guess what *clipping level* meant. A

little looking through a parts catalog and I found the term vaguely defined in a specification for a chip designed for use in disk drives. It is unfortunate that these words are used to describe a test performed on diskettes since they have a different and more widely understood meaning throughout the general electronics industry. In any case, we will have to accept these words since they are the ones used in the advertisements.

In simple terms, your drive uses a "head" to read disk information. You can think of this as a needle and pickup on a record turntable. The head reads the magnetic information previously written on your disk and converts it into an electrical signal. This signal is processed further and eventually takes on a form suitable for transmission to your computer as bits or bytes which represent the data.

Signal Variation

The size and shape of the electrical signal developed by the head varies for many reasons. First of all, it varies as a result of the information written on the disk and this variation itself represents that information. However, there are other variations that take

place due to head imperfections, mechanical drive characteristics or diskette imperfections. If large enough, these variations lead to the drive electronics being unable to correctly decode the information. At that point, your computer generates an error message telling you it cannot read the disk. Therefore, it is important to minimize the variations (those that are not part of the data).

Coating Thickness

Magnetic diskettes or tapes are manufactured by bonding magnetic particles to a flexible mylar backing material. Characteristics which affect the performance of the final product include, but are not limited to, the particle magnetic characteristics, the particle size, the coating thickness and, most important for the clipping level subject, the coating uniformity. If a tiny, pinhole sized part of the disk does not get coated, the signal level recoverable from that spot is reduced. Thus, if several of these spots exist that are of sufficient size, the signal level will be fairly uniform until that pinhole spot passes underneath the head ... at which point it will *drop*. These are referred to as *drop-outs* in the industry.

Furthermore, if the coating thickness varies over the surface of the disk, the signal amplitude can vary in a relatively smooth manner as the disk rotates. However, this is not generally a serious problem.

Your drive can recover your data by separating these disk related variations from the signal variations due to the real data, provided that the disk related variations are not too large. Typically a drive might be able to successfully ignore disk related variations which did not reduce the amplitude of the real signal to less than 30 percent of the normal output. This number, however, also depends upon a wide variety of factors and varies from drive to drive

even with identical models from the same manufacturer.

Anything one could do to assure that the level of these disk related variations are held within a specified range should reduce error probability. The key word here is *probability*, which I will address later. A disk which is tested to a clipping level of 60 percent assures that the variations due to the disk are small enough that the signal level never drops below 60 percent. That is, the variations are held to a range between 60 and 100 percent. It follows that the higher the clipping level, the less the variation in the signal output and the reduced probability of disk error.

Now for the tough part. How much more should you pay for a disk tested to a 60 percent level as compared to one tested to a 40 percent level? Would you pay 50 percent more? Twice as much? Ten times as much? The way I look at it, there is a high probability that if I buy 25 or 50 "brand X" disks and they all work, whatever tests were run on them were probably sufficient to assure me that they will always work. I have no way of knowing what specification clipping level disks destined for my drives should be tested to, nor, do I believe, do the floppy disk manufacturers.

A Word About Hard Disks

In the case of Winchester drives, the situation is a little different. The manufacturers of the disks for these drives are generally different from those companies manufacturing the actual drives. The drive manufacturer imposes specifications upon the disk manufacturer and continually tests the disks using sophisticated equipment to verify that the disk manufacturer meets the specifications. That is, people who manufacture disks for use in hard drives do not sell them directly to end users (except for removable hard disks).

Bottom Line \$\$\$

Let's go back to floppies. Let's assume that I buy 100 diskettes from each of two imaginary sources, SuperDisk and CheapDisk and that I pay 40 cents each for the CheapDisks and two dollars each for the SuperDisks. Out of all the disks I bought, one SuperDisk won't format and 10 CheapDisk won't format which means that I have paid \$2.02 per disk for the good SuperDisks and 44 cents for the good CheapDisks. I think I got a better buy for the CheapDisks.

More Use ... Better Performance

What about disk failures in the future? That is, as I use the 90 CheapDisks are they more likely to fail than the 99 SuperDisks? There are those who would argue that, in fact, they would, but I don't believe it. The reason is that the first few times I use any diskette, its performance will improve. The disk surface is unintentionally left slightly rough during the manufacturing process, which prevents good contact between the head and the disk. The poor contact degrades disk performance, and as the disk is used and rotates past the head, coating particles are knocked off. This smoothes the surface and improves the contact and performance.

In critical applications of tape recording, such as creating studio masters, new tape is never used without running it through a machine several times to "smooth it out." Therefore, after I have used my CheapDisks several times I feel more comfortable with them than when they were brand new.

How Long Will They Last?

Finally, what about the long term? Will CheapDisks retain the information stored on them equally as

well as SuperDisks, say over a period of 100 Years? Here we are dealing with real unknowns. There are no disks that have yet reached 100 years of age. Magnetic recording using disk-type media is only about 40 years old. Archival data that has existed for long time periods has become a problem in a number of fields ... just ask any librarian about the problems facing the Library of Congress in protecting old books.

There has been some information about magnetic recording in general that may be of interest. In tape, such as your audio or video cassettes, or computer tape as used for mainframe systems, there is a long term storage problem known as *print through*. The magnetic pattern on the tape representing the information emanates a magnetic field, just as with the North and South Poles. This field is small, but still present, and any material susceptible to magnetization will do so in the presence of a magnetic field. This is true even for weak fields if the material is held still within the field for long time periods. All tape is susceptible to being magnetized ... that is its prime purpose in life.

When would on a reel, each piece of tape is tightly pressed against another, and each piece emanates a magnetic field. If the tape is left untouched in this form for several years, some of the information recorded on adjacent pieces mixes. In audio tapes one can hear this as a low level background of the same music played a few seconds earlier or later, particularly where a loud passage is immediately followed by a quiet one.

Normally, a fairly thick jacket surrounds disks, making print through occurrence unlikely. On double-side disks, however, magnetic information on each side of the disks is close to the other, about the space found between tape reel pieces. If I were to guess at the first cause of long term failure, in the sense of not being to recover 100

percent of the material on the floppy, print through would be the culprit.

Recopy Your Floppies

Someone once raised the question of whether it made sense to recopy masters or backups occasionally to make new backups, if the software permitted another copy to be made. My initial answer was that I didn't think it was worthwhile, but having given it some thought, I think it may be a good idea. If degradation occurs over time as an untouched copy sits on the shelf, recopying does restore the information to a more pristine state and acts as added protection against data loss.

As to SuperDisks being better than CheapDisks for archival purposes, I think that little difference exists ... but only time will tell.

How Come So Cheap?

Many factors create price differences between disks. Advertising, packaging and other overhead costs may play a part. Some companies sell primarily to other companies who avoid buying through the mail. The disks are sold because people are willing to pay for it, whatever the reason. In fact, many people selling cheaper diskettes are buying their raw materials from the same source as those selling the more expensive versions.

The entire mass production selling concept, whether for disks, light bulbs, drives, or computers, is that they are produced on a statistical basis. Costs are reduced to the point that the probability of delivering a defective unit to a consumer is acceptably low. This is simply good business, and no company can remain afloat if it demands perfection in a commercial product line. Only governments can afford products which have been tested to the level of a space shuttle, and as we found out even they are not perfect.

Personally, I have always bought

the least expensive disks I could find. Furthermore, I buy single-sided, single-density diskettes and use them in double-sided, double-density drives *with no problems*. On one occasion, I paid more than \$25 for a box of ten diskettes, but it was a Sunday, I needed them and they were the only ones to be found. One of the disks in that box proved to be the first bad disk I have ever run into.

One last comment on probabilities. If the chance of a given disk failing is one in one thousand, under any possible circumstances, the chance of two failing under the same conditions is one in one million. Anyone for making two 45 cent backups instead of one for two dollars? ■

*From the Backup News, A Monthly Publication of (San Francisco) Bay Area Kaypro Users and Programmers, December, 1986. The article was originally published in Articles, The KAY*FOG Online Magazine, P.O. Box 11135, San Francisco, California 94101-7135. The KAY*FOG PCBBS can be reached at (415) 285-2687.*

(How diskettes are jacketed can affect their life. Optimally, the jacket material is thick and completely sealed around the edges with a special cement and then "hot stamped." Some manufacturers of less expensive disks do not perform all parts of this process, which may contribute to a shorter diskette life or data loss. Also, a thinner jacket can allow temperature, accident, magnetic fields, or other factors to affect the disk. Usually you can tell how jacket thickness differs from diskette to diskette ... the thinner ones are "floppier" than the thicker ones. Lastly, the addition of a "hub ring," a reinforcing plastic ring around the center of the disk that touches the hub, may not hurt or help the data, but will help the physical disk to last longer if it is used often ... as in the case of backup disks.--Ed.)

Toolwork's C/80

Some Questions Answered

By David Allen Giunti

Who might be interested in Toolwork's C/80?

Owners of a CP/M computer system who A) like to program at an advanced level, are familiar with at least one computer language and are looking to master another faster system; B) are interested in C by reputation and are seeking working knowledge; and C) are familiar with the resources of their system.

Is there anyone who shouldn't purchase C/80?

Yes! C is not a language for a lazy or careless programmer. It does not protect you from the programming environment and it demands that you keep close track of what you are doing. For instance, if you attempt to access an element beyond the declared bounds of an array, C will look where you asked it to and do what you ask it to do, as opposed to languages like BASIC and Pascal which will issue run time error messages. Also, C/80

would not fulfill the requirements of someone who is looking for an at-home UNIX development system.

Is there a difference between C/80 and UNIX C?

No, the C is almost exactly the same on a VAX as it is in C/80 because the language is remarkably simple and well defined in what is called "The Bible," The C Programming language by Kernighan and Ritchie (Prentice Hall). The troubles that are related to the VAX are with what is referred to as the "standard library." C Language does all of its communication with the world through this library, and there are at least seven different standard libraries. C/80 uses version six of this library, while version seven is now a little more common.

Is there a major difference between libraries?

No, not really, Kernighan and Ritchie only make recommendations about the standard library. The library performs tasks such as file operations and controls (with options) things like the keyboard and screen, however the language itself controls access to these

functions. Once you have a mastery of the language, you will know what to do with these libraries, and can, in fact, build your own library.

How about the language? Above it was stated that C/80 is "almost exactly" the same as C Language on the VAX. What's missing?

In reality, there are very few things missing from C/80, and all the features not implemented can be easily programmed around. For instance, a directive called TYPEDEF, which would allow you to build variable names another way, isn't there. However, this is no loss because it operates much like #define a b, which causes every "a" to be replaced by "b". Part of the #define is also missing, and it does not take arguments. Something like #define SQUARE(A) ((A)*(A)) would replace all instances of SQUARE(n) with ((n)*(n)) and allow you to build a function in line. Calling a function or filling in the blanks yourself solves this little problem quickly.

The bitfield directive (to give names to a specific number of bits), which isn't in my \$350 Aztec C either, can be handled with C/80's full set of bit-

This article originally appeared in the Backup News and is used by permission.

wise operators.

Other than these omissions, C/80 is a full C, with 30 some odd operators for fiddling with bits, logic, math and type conversion. All storage

implemented. If you need an integer larger than $2 \cdot 10^9$, examine your methodology ... this is not a serious defect.

FLOAT, a 32-bit signed floating

larger and slower, but "safer." Because of the extensive library, Aztec C's filing system is more complex.

Documentation

C/80's documentation is about 50 pages long (for the compiler) plus another 10 for the MATHPAK. It makes no pretenses about teaching you to program in C, but gives a summary of the language and explains its library functions. The documentation explains how to operate the compiler, the compiler option switches and the use of the CCONGIG program to set them as default conditions for the compiler.

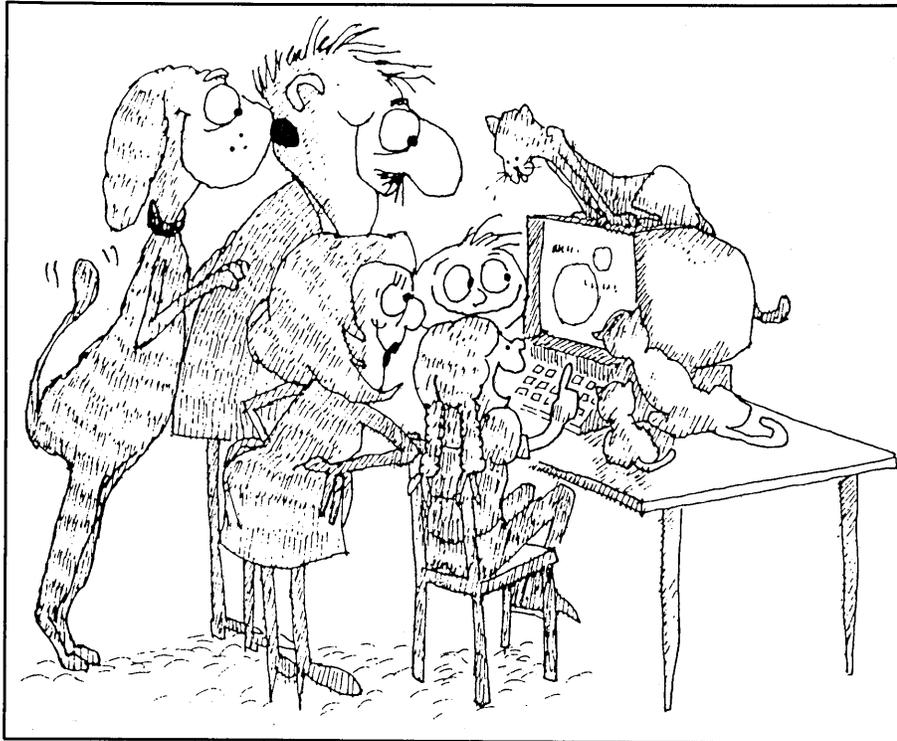
A great deal of attention is given in the documentation to operation with Microsoft's M80 and Digital Research's RMAC assemblers (if you have either of these tools, C/80 can use them to advantage). Options for use of the C/80 assembler, as well as interfacing 8080 assembler subroutines, are well covered. Other routines are also documented:

CTRACE, a neat little routine that allows you to follow the execution of a program, function by function, monitoring the values in the CPU registers.

CPROF allows monitoring the time actually spent in the routines with a system clock (if present). This program is set up for a Heath system, but can be modified to operate with other systems.

There is a list of reference texts for C Language. Of those given, I would recommend the Kernighan and Ritchie text because it not only defines the language, but is also designed to teach C. However, this book is perhaps a little slim for beginners, so I also suggest books by Purdum and Hunter. Both of these deal with C in a CP/M context.

If you're interested in the C Language, then C/80 is a tool that can be used not only to learn the language, but also to develop a significant CP/M software library. ■



classes and statement forms are implemented just as on a VAX.

The pre-processor functions are complete, with a couple of useful C/80 additions. For example, #ifneed name is used in C/80 to build libraries that will be compiled into your program only if they are needed.

The data types that C/80 supports are:

CHAR, a character that can also be a signed eight-bit quantity (a default that can be changed with a -z switch in the compiler).

INT, a 16-bit integer, either signed or unsigned.

The optional MATHPAK adds LONG INT, a 32-bit signed integer. The unsigned long is not

point number, with seven digits of mantissa and an exponent range of {plus/minus} 38.

DOUBLE is the same as FLOAT in C/80.

POINTERS are 16 bits in this implementation, but resist dealing with them as INTs if possible. They can have absolute addresses and you can compute the difference between two pointers.

You can make structure out of any combination of these elements, and structures of structures, if you need to.

Compared with Aztec C

Aztec C provides the version seven libraries. The .COM files that Aztec generates are a little more complex because of built-in levels of error checking. The code it generates is

Small-C

A Public Domain C Compiler with Notable Limitations

By Erik Westgard

As part of the never ending struggle to develop a perfect programming language, in the early seventies AT&T Bell Laboratories' Dennis Ritchie wrote a language called "C." He wanted a structured, high-level language that would be transportable across different machines and powerful enough to write operating systems.

He ended up with what is now called a portable assembly language. C has a core of instructions common to all versions and a rich set of functions that are often written in C. Instructions are like those for subtraction, assignment and multiplication. Functions, in part, open and close files and output characters to terminals. You need only to bring up the compiler on a new machine, compile the functions and you have a new C system.

Professional programmers and systems houses use C extensively, since transportability and efficiency are key to success in the software market. The new product must run on many different computers and be as fast as possible. The AT&T operating system UNIX is written almost entirely in C, which explains why it has appeared with dozens of different computers in the last few years. For systems programming, C is the language of choice on small systems, edging out assembly language.

For the home and personal computer enthusiast, there are advantages to using C, especially since program speed and memory consumption continue to be significant programming problems. C programs compile into very tight machine code, eliminating most needs for inline assembly language instructions in programs. If, for example, you have a Morrow and an Apple computer, portability is important. The C language is quite standard with only minor differences between versions. Imagine trying to move BASIC graphics programs from your Apple to your Morrow with all the PEEK and POKE statements. C functions take care of all that.

Big, unwieldy programs with lots of GOTO statements

Erik Westgard works as a sales engineer for a computer company in the midwest. In his spare time he races sailboats and writes articles for model railroading magazines.

are a thing of the past. C is designed to force the programmer to write programs as a series of modules, each performing a unique function. One can do a given job just by calling the modules in order, and change the function of a program just by rearranging the modules. Colleges are teaching C and structured programming heavily and the demand for C programming skills increases daily.

Getting Started

There are almost ten C compilers available commercially for CP/M 80. Five of these were reviewed in the August, 1983 issue of *BYTE* magazine, with prices ranging from \$40 to \$500. For those who just want an introduction to the language, several compilers costing less than \$50 are available. MIX C, with an excellent introductory manual, and C/80 are both in this price range.

I wanted to experiment with a low-cost public domain version of C, which generally appear under the name *Small-C*. This version of C first appeared in a 1980 *Dr. Dobbs Journal* article (reprints are still available). I got an early version of Small-C from the BAMDUA (Bay Area Micro Decision Users' Group) library on two diskettes, numbers 20 and 21. I got a copy of the same compiler later from the now defunct MUG.1 user's group in Minneapolis on their disks 25 and 26.

Apparently written by Mike Bernson, my version included the compiler with the source code, an assembler and a linker. You write a C program with an editor like Wordstar in the non-text mode and then run the compiler. If that works, and your program compiles without errors, you run the assembler and then link the module that results in C libraries that form a working .COM file.

Limitations

Armed with very sketchy documentation (a read.me file on the diskette) I was able to struggle through some short programs. Almost at once a number of important limitations and bugs became apparent, so I started making a list and kept going. The first thing I noticed was the lack of a long data type, so I had to use integers for all math. Also

missing from the standard library is the printf function, the standard method for writing output to the screen (like PRINT in BASIC). Needless to say, most of the published sample programs and benchmarks I found could not be compiled. Printf is an important function which prints character data at the terminal like the BASIC PRINT statement. We tried unsuccessfully to adapt a sample small-C printf function from *BYTE* magazine to this package (Christopher Kern, "Printf for the C Function Library," *BYTE*, May, 1981, p.430). There seemed to be too many problems to overcome in the character I/O. What we were able to get running was another version of my tiny terminal program for the MD-2, this time in C. The program uses a while loop to cycle through console status and serial port status tests looking for console input or characters from the host. The vertical line (ASCII 127) is used as an exit character.

Friend and C enthusiast Warren Kinninger used the package for a while and added to my list of limitations. Some of these included the lack of a "do" statement, no unsigned variables and no function return type declaration (it is always "int"). The linker is undocumented and only lists the first reference it could not find instead of all of them.

In sum, this is a very inexpensive compiler that can be used to learn about C. It is instructive to watch the assembly language output and compare it with the program source, as well as to study the the compiler source code to learn about how compilers work. The compiler is not suitable for any extensive programming use, due to the many limitations. It can be very frustrating trying to learn to program with a buggy compiler that only supports a subset of the language. If you are at all serious about C, look at some of the commercial compilers mentioned earlier in the article.

I did most of my C experimenting on an MD-2 with single sided drives. The C compilers do not usually have an editor, and so having the compiler, editor, linker and assembler all on the machine at one time strained the capacity of my single-sided diskettes. I found that all the files I needed could not fit on two diskettes, so I was constantly swapping them back and forth. Double sided drives or a hard disk are needed for any serious C work.

SAMPLE PROGRAM DESCRIPTION

Cterm is a very simple terminal program for the MD2 that can adapt to the MD3 and other models by just changing the values for COMMSTS (Communication Status Register), COMMDAT, the data register for the serial modem and printer port and CONSSTS and CONSDAT, the ports for the serial console (your terminal) status and data. A bit is set in the console status register

whenever you press a key on your terminal, and it is set in the serial port status register whenever a character comes in from a modem attached to the spare serial port on your MD.

The sample program tests these status registers constantly and writes characters from the terminal out through the serial port (COMMDAT) and characters from the modem port out to your terminal (what CP/M calls your console). This is CONSDAT in my program.

To use CTERM, type it using the non-text mode with an editor like Wordstar or New-Word, and then use Small-C to compile, assemble and link-edit the program. Use SETUP.COM on your CP/M disk to set the port speed (my program does not do that for you), connect your modem and type CTERM. If the transmit light flashes on your modem when you press a key, you are ready to dial your favorite BBS system or timesharing service. Type a vertical bar to exit the program when you finish. ■

See page 47 for a complete reference list.

```

/* CTERM.C - A Simple Terminal Program (6/85)
*/
#define NO    0 #define YES 1

/* Status and port definitions */

#define COMMSTS 255
#define COMMDAT 254
#define CONSSTS 253
#define CONSDAT 252

#define RXRDY 2
/* A character is ready to be received */

main() {
    int x, done;
    done = NO;
    while (!done)
        {
            if (inp(COMMSTS) & RXRDY)
                {
                    x = inp(COMMDAT);
                    outp(CONSDAT, x);
                }
            if (inp(CONSSTS) & RXRDY)
                {
                    x = inp(CONSDAT);
                    if ((x & 127) == '|')
                        done = YES;
                    else
                        outp(COMMDAT, x);
                }
        }
}

```

MOR Products

Original *User's Manuals* published by Morrow.

<i>Prod #</i>	<i>Description</i>	<i>Price</i>
601	Correct-It User's Manual	8.00
602	CP/M 2.2 User's Guide (for MD2 & MD3)	12.00
603	CP/M Plus Guides (for hard disk MDs; 5 books)	35.00
604	MBASIC User's Guide	12.00
605	MD Companion Guide (for everyone who isn't compatible with CP/M)	12.00
606	MD User's Guide (for MD2 & MD3)	12.00
607	MD Hard Disk User's Guide	12.00
608	Morrow Software Sourcebook (1983 edition)	8.00
609	NewWord User's Guide	12.00
610	Pilot User's Guide	12.00
611	SmartKey User's Guide	8.00
612	SuperCalc User's Guide	12.00

Service Manuals to help repair or enhance your machine. These technical manuals are skillfully edited from internal Morrow data and were previously only available to Morrow dealers. (Even if you don't do the repair yourself, your repairman probably will need them.)

<i>Prod #</i>	<i>Description</i>	<i>Price</i>
700	Complete manual set (701-710)	80.00
701	Micro Decision Service Guide w/ schematics for Rev. 1.1 & 2.0 (for MD2 & MD3)	25.00
702	Diagnostic II User's Manual (disk sold separately)	25.00
703	Hard Disk MD Service Guide w/ schematics	25.00
704	SA 200 Shugart Mini Floppy Drive Preliminary OEM Manual (for SS drives)	15.00

705	SA 455/465 Shugart Mini Floppy Drive Service Guide w/ schematics (for DS drives)	15.00
706	MDT20 Terminal Reference Guide	10.00
707	MDT20 Service Manual	12.00
708	MDT50 Video Terminal Service Manual	10.00
709	MDT60 Video Terminal Service Manual w/ MT70 Video Terminal Supplement	15.00
710	David Block's original MD Q&A book	6.00

ZCPR3 is a Morrow-bootable CP/M 2.2 replacement by Echelon for the MD3. Complete with two disks and manual, the ZCPR3 kit provides an operating system superior to CP/M 3.0. This kit currently only available for ROM Rev. 3.1. (NOTE: If you order ZCPR3 through **MOR**, you qualify for a discount group purchase price of less than \$500 for a 10-megabyte hard disk made by Advanced Concepts for the MD3.)

<i>Prod #</i>	<i>Description</i>	<i>Price</i>
801	ZCPR3—including: Z-System, 85utilities, help disk and <i>The Z-System User's Guide</i>	59.50
802	Diagnostic II disk (\$45.00 price effective 1/1/87)	45.00
803	ZCPR3 Supplement – utilities and source code for BIOS and utilities (4 disks)	35.00
804	ZCPR3: The Manual	20.00
805	Out-Think–Outline processor	45.00

ROM Upgrade Kits contain instructions, ROM chip and software—everything you need to bring your machine up to date. (Some software that won't run on older machines *will run* with an upgrade.)

<i>Prod #</i>	<i>Description</i>	<i>Price</i>
901	ROM Upgrade Kit for MD3	35.00
902	ROM Upgrade Kit for MD Hard Disks	45.00

Floppies of the Bimonth

With every issue of *MOR*, we compile floppies that contain public domain programs, short articles and other interesting tidbits. Each **Floppy of the Bimonth (FLOB)** costs \$8.00. Listed here are the **FLOBs** for this issue. A list of all **FLOBs** to date is available upon request.

A Note on Using FLOBs

Some of the files on **FLOBs** have been compacted to allow us to put more on a **FLOB**. When you receive your **FLOB**, you will need to return these compacted files to their original form.

Which Files Have Been Compacted?

If a file name ends with "LBR", it is a library. A library is a file that contains a number of related files (e.g., one file might be a program, another file might be the documentation for the program). To use these files you must extract them from the library with a library-management program like NULU, which is available on **FLOB #3301**.

If a file name has a "Q" in the middle of its three-letter extension (e.g., WORDSTAR.DQC), the file has been squeezed. Many programs will unsqueeze files, NSweep is a popular one.

For more information on using **FLOBs**, see *MOR* Vol. 3, #3, p. 55.

Prod.#4101 — FLOB/FEB87 Disk #1:

SMLC210.LBR Small-C C language compiler.

Prod.#4102 — FLOB/FEB87 Disk #2:

SMLCMAC1.LBR Macro assembler for Small-C (1 of 2 needed libraries).

Prod.#4103 — FLOB/FEB87 Disk #3:

SMLCMAC2.LBR Part two of the Small-C Macro assembler package.

E-BASIC BASIC compiler.

SPY-8611.LBR Utility to examine the way a program access the disk drive while running.

Prod.#4104 — FLOB/FEB87 Disk #4:

PASUTIL.LBR Utilities for Turbo Pascal.
TURBOINC.LBR Extensions to Turbo Pascal routines.
TOOLWORK1.LBR Elementary tools for C/80 C compiler

Prod.#4105 — FLOB/FEB87 Disk #5:

PASGAMES.LBR Lots of little games in Turbo Pascal.
TURBSCRN.LBR Screen-formatting utilities for inclusion in Turbo Pascal programs.
T3DH00.DQC Patches to Turbo Pascal 3.00A.

Prod.#4106 — FLOB/FEB87 Disk #6:

COBOL.LBR COBOL programming language.
TURBUTIL.LBR More patches, utilities for Turbo Pascal.
MATH.PQS Mathematisian simulator in Pascal.
STUMP3.PAS Tests the limits of Turbo Pascal.
STUMP4.PAS Another numerical test for Turbo Pascal.
ISLOCAL2.PQS Checks to see if a phone number is local.
HEXCALC.PQS Hexadecimal to decimal conversion.
MOBILITY.LBR Calculates job turnover, mobility rates in heirarchical organizations.

Prod.#4107 — FLOB/FEB87 Disk #7:

ILISP.LBR LISP programming language.
SCI-11.LBR Small-C interpreter; useful for debugging C programs.

Prod.#4108 — FLOB/FEB87 Disk #8:

XLISP.LBR Version of LISP with Smalltalk-like features.
PROLOG.LBR Small introduction to the PROLOG programming language.

Prod.#4109 — FLOB/FEB87 Disk #9:

MDCLCK12.LBR Mike Allen's clock article on page 35 plus clock programs and schematics.
MAILHAND.4X1 Erik Westgard's article on page 38 including MBASIC source code.
PASFAVOR.4X1 Ken Grymala's article on page 14 including source code in "C."
BAM-PBBS.MSG BAMDUA PBBS messages: Discussions on various topics between the BAMDUA BBS callers.
SCTAX86.LBR SuperCalc spreadsheet IRS Form 1040.

Back Issues

A resource to help make your Morrow really useful

Vol. 1, #2, JUN84 (Prod #102)

Concurrent Editing and Printing with WordStar
☆ Customizing NewWord ☆ Troubleshooting
Your Micro Decision ☆ Diskettes Deteriorate
☆ About Serial Communications ☆ Installing
WordStar for Dot Matrix Printers

Vol. 1, #3, AUG84 (Prod #103)

Graphics Package for the Morrow ☆ Freindly
Telecommunications ☆ A Very Simple Modem
Program ☆ Proportional Spacing with WordStar
☆ Using BASIC to Control Your Printer ☆
Patching WordStar for an Epson Printer
☆ Installing WordStar 3.0 for the MP100

Vol. 1, #4, OCT84 (Prod #104)

About Surge and Spike Protectors: I Thought It
Would Never Happen to Me ☆ Public Domain
Software Made Easy ☆ Programming Your
Function Keys with NewWord ☆ More Printing
and Editing Concurrently with WordStar

Vol. 1, #5, DEC84 (Prod #105)

Speeding Up Your Computer ☆ WordStar
Emergency Tricks ☆ Customizing WordStar's
Terminal Defaults ☆ ZCPR3 Makes CP/M 2.2
Shine ☆ Spool-Away with Printer Buffers
☆ Notes from a Smartkey User

Vol. 2, #1, FEB85 (Prod #201)

Trouble-Shooting Your Morrow ☆ Micro
Justification and Prop. Spacing in WordStar
☆ Teaching with Pilot ☆ Toward a More Useful
MD2 (1,876K disk space) ☆ Getting Started
with TURBO PASCAL

Vol. 2, #2, APR85 (Prod #202)

Morrow Drive Error Messages ☆ Saving Time in
SuperCalc with X (eXecute) ☆ The Morrow
Toolbox ☆ Automatic Formatting with PILOT ☆
Uniform, A File Transfer Utility ☆ SUBMIT to
Less Work

Vol. 2, #3, JUN85 (Prod #203)

Modems: What's All the Fuss About? ☆
Getting Started with Modem Programs ☆
Morrow Printer Ribbons ☆ Locked Out of
LogiCalc? ☆ Short Introduction to FIDONET ☆
Adding Z-Clock to MD3 ☆ The Smartest Modem
of All (Visionary) ☆ The CTS MM300 Modem
with MITE

Vol. 2, #4, AUG85 (Prod #204)

Upgrading NewWord: Why and How? ☆
KAMAS, an Outline Processor ☆ EUREKA! A
CP/M Disk Cataloguer ☆ NSWEEP: A Public
Domain Jewel ☆ New Tricks for WordStar ☆
ShadowPrint (SPRINT), a Print Spooler

Vol. 2, #5, OCT85 (Prod #205)

Running a Small Business with a Morrow ☆
Reflections on dBase II, Comparing Gold to
Pearl ☆ dBase II Hints for Morrow Users ☆
Accrual Bookkeeping on Quest, Vers. 1.2 ☆
An Overview of Personal Pearl ☆ Eliminate
Sort Disk Swap when Using Pearl

Vol. 2, #6, DEC85 (Prod #206)

The Co-Processor Board Confusion ☆
Converting Your Morrow Keyboard to Dvorak ☆
Genealogy with Personal Pearl ☆
Telecommunications: National and International
☆ VersaLedger II ☆ Indexing with Correct-It ☆
The Animal Game ☆ Handicapped Users and
the Morrow ☆ Conway's Game of Life

Vol. 3, #1, FEB86 (Prod #301)

Best of Public Domain Software ☆ Debugging:
Overcoming the Fear ☆ Public Domain Games
on the Morrow ☆ A Note on the C Programming
Language ☆ Of Copyrights and "Public
Domain" ☆ Adventure Strategies ☆ Functional
Comparisons of Database Software ☆ Personal
Pearl Hints and Kinks ☆ Doing Returns on a
Computer ☆ Reviews of TAXWORKS, 1040
Plus

Vol. 3, #2, APR86 (Prod #302)

The Basics of Computer Typesetting ☆ More
on Tools for Writers ☆ Some Favorite Public
Domain Writers Tools ☆ Desktop Publishing
with a Morrow ☆ Proportional Spaced Printing
with CP/M ☆ Reviews of WordFinder, Index,
Footnote, Bibliography, Grammatik, wsNote,
Number, FTNOTE, Dataplotter, ElectraFind,
PropStar, MagicSeries, PrintMaster

Vol. 3, #3, JUN86 (Prod #303)

Doubling the Speed of Your Terminal Port ☆
Your 4-Drive Morrow: Adding Two More Drives
to Your MD3 ☆ Connecting a Portable to Your
MD ☆ Exploring the ROM in Your Morrow ☆
Updating Your Morrow with the Latest ROM —
How and Why ☆ About Printer Buffers and
Speed ☆ A User's Guide to Understanding the
FLOBs ☆ How I Quieted My Disk Drives and
Lived Happily Ever After

Vol. 3, #4, AUG86 (Prod #304)

Maintenance for Your Daisy-Wheel Printer ☆
Where to Get Your System Fixed: A List of
Repair Shops Servicing MDs ☆ UnSqueaking
Your Keys ☆ The Man Who's Got the Parts ☆
Keeping Your System Alive: Minor Crises and
How Prevent Them ☆ Reviews of OUT-THINK
and Thoughtline

Vol. 3, #5, OCT86 (Prod #305)

The Wonderful World of ZCPR3 ☆ Writing Your
Own Educational Software ☆ Selecting
Educational Software ☆ The Only Software an
Academic Needs ☆ Reviews of Language
Tutor, three typing programs and Write-Hand-
Man

Vol. 3, #6, DEC86 (Prod #306)

From Morrow to Mac ☆ Simplified Desktop
Publishing ☆ A Homegrown Portable MD11 ☆
Tune in to the MOR PBBS ☆ The Z-System and
Mini-Winnie Hard Disk ☆ Writing with SmartKey

All This information can be yours for \$4.00 per issue.

Shipment outside of North America add US\$ 1.50.

Subscribe/Renew

\$18 — 1 year (6 issues)

\$32 — 2 years (12 issues)

For US first class add \$8 per year. All Canada & Mexico residents add US\$ 7 per year. All other countries add US\$7 per year for surface mail or add US\$16 per year for airmail.

For RENEWALS be sure to include your **Subscriber No.** which is found in the upper right hand corner of your mailing label, preceding the expiration date.

MOR is published bimonthly, starting in February. We must receive your subscription application before the 15th of the month preceding publication in order to begin your subscription with the that issue.

Name _____

Phone (____) _____ Subscriber No. _____

Address _____

City _____ State ____ ZIP _____

Total amount enclosed \$ _____

Check/Money Order VISA Master Card

Account No. _____

Exp. Date _____ Signature _____

Order Form

Name _____

Phone (____) _____ Subscriber No. _____

Address _____

City _____ State ____ ZIP _____

Please include the Product No.

Product No.	Quantity	Description	Price

Send orders to: MOR, P.O. Box 5487 Berkeley, CA 94705 California Residents please add 6.5% sales tax

Subtotal
TOTAL

Check/Money Order VISA Master Card

Account No. _____

Exp. Date _____ Signature _____

IMPORTANT NOTE: All prices include UPS or first class postage to domestic addresses. Due to weight constraints, some orders must be shipped UPS, so please include a street address. Foreign shipments are additional, please inquire.

The Reluctant Programmer

Mom Doesn't Want You To Spend Your Time That Way, Does She?

By Bruce Gowens

When I purchased my MD2 I knew exactly what I was going to do with it. It was going to be a very useful tool, to be used for work and not to play around with games or even think about programming. What's more, I was never, never going to be a hacker. I was a user, not a computer freak or nerd! I wanted to work up spreadsheets for a business and do word processing. That was IT!

Since I was living in Silicon Valley, my acquaintances were techies or hackers (in the traditional meaning of "hacking around with computers" as in "hacking around in boats ... there's nothing so pleasant"). The kind of person that referred to all life in technobabble and couldn't seem to interface (whoops -- I mean connect) one word of English with another. The boring types whose eyes lit up only when they were talking about MVI 05BC into the H register of the Z80 in 8080/8085 assembly language. In other words, the kind that felt

computers were too exciting as toys to be wasted on something as mundane as word processing.

I would not, I swore, become like them. Life is complicated enough without adding layers of obscure and arcane knowledge to my database (whoops -- I mean memory -- err, mind -- err, memory). But "the best laid plans" I found my spreadsheet needs limited and discovered the power (whoops -- usefulness) of databases. Word processing was all and more than I had thought it would be, but after a while I asked why the computer couldn't do a few more things for which there weren't programs.

As I became involved with Personal Pearl databases, I set up and operated a few for some non-profit groups. Partly it was the newness and the potential I had under my fingertips ("when you first get a hammer, everything looks like a nail"), partly because it was a way (I thought) I could help them out. Amazingly, they thought I was a hacker and only talked technobabble.

Although David Farr, then the Music Director of the Junior Bach Festival, referred to me as their "Computerist." (Musicbabble?) Then I realized that Junior Bach could use a custom program for help in planning the concerts. About that time JRT

Pascal was hung out on a bulletin board (the company was belly up) so I thought I'd try my hand at a program using it. Although I had BASIC, of course, a glancing acquaintance a few years previously had convinced me that I didn't want to attempt anything serious with it (an opinion I still hold).

Well... that was a complete fiasco. The program never worked and was never completed. Partly because I wasn't a programmer, partly because I bit off too large a chunk with which to teach myself, but mainly because JRT Pascal is terrible (and buggy and inconsistent). It was enough to reinforce my previous opinion that I did not want to program. Too frustrating ... too much work ... too painful.

But I did have this irritating problem. My Gemini Star printer was supposed to be able to do all kinds of nice things, like condensed print, that I couldn't preset it for. So I asked a programmer friend (Dave Longyear ... he loves to see his name in print) if it was possible to make a little program that would set up the printer when I wanted condensed, etc., and he said he'd help me write one.

That wasn't what I had in mind. I wanted him to write it.

Sometime before that, in a fit of

Bruce Gowens is interested in distributing programs written in Turbo Pascal. Those with programs can send a postcard (ONLY!) with address and phone to Idiot Savant Programs, P.O. Box 5367, Berkeley, CA 94705. DO NOT give a description.

weakness, I had bought a copy of Borland's Turbo Pascal at the West Coast Computer Faire. I was there to see any new and practical business software, of course, and had no interest (of course) in those flashy (hmmm..) new desktop publishing machines with their suave cases (hmmmm...) or the shiny new music synthesizers (drool...).

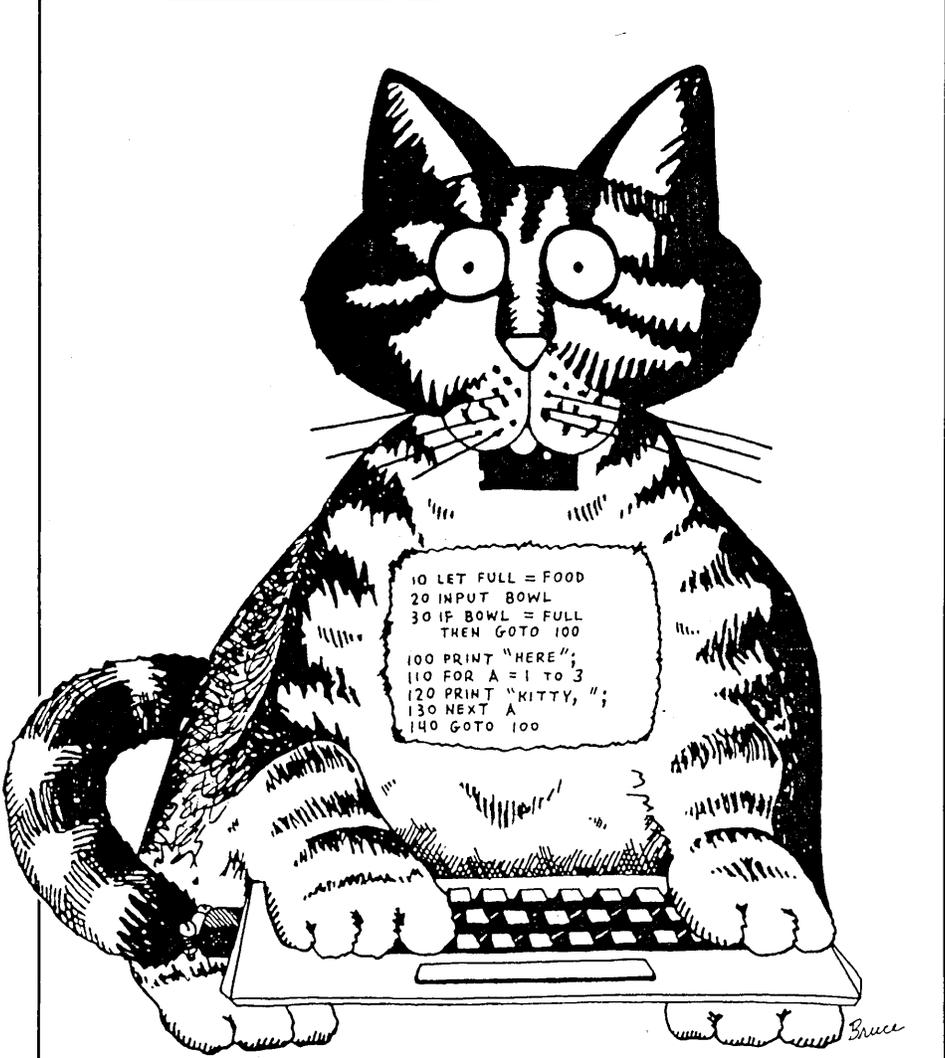
Not that I intended to actually use Turbo, you understand. But maybe if I had it around it would quiet my nagging hacker friends. "Oh, yes, I've got Turbo, and I'll be learning it just as soon as I find the time."

So one day Dave came over and said "You are going to write that printer control program now." Caught! No way out. So we sat down and I started learning to program. Could I get him at the keyboard and finagle him into writing it? No. Flattery didn't work. Bribery didn't work. What's more, he wouldn't even tell me what to write. Instead he sat back and played Polonius (Ophelia's father in Hamlet, who got his just deserts) with such brilliant insights as, "Plan what you want to do before you do it" and "Organize the operations like an outline, then start at the top." I'd ask "How do you do thus-and-so?" and he'd answer, "How do you think you would do thus-and-so?"

There's a computer program that can answer like that. I wanted to control my printer, and what do I get? Psychoanalysis!

But, funny thing, it worked. What's more, the program worked. And totally unexpected was the rush of joy and feeling of accomplishment when I found that I could really control the computer.

Not that it was such a good program. In fact, it stank. But I didn't know that until much later, when I realized that there were so many ways it could have been better. I guess I'll have to revise and clean it up some day, but I still use it. Meanwhile, thank you Dave, for getting me started



Morrow

The new Morrow portable computer. It comes complete with 256 Kittybytes, but no mouse.

-- even if you let me write a lousy program.

The experience was like a breaking dam. Suddenly I could program and it was like getting a hammer all over again. Strangely enough, although I

still knew very little, I was able to pick up or figure out the rest by myself (with the aid of a few books that I've noted in this article).

And now? Well, hacker I am not! You can tell because I do English

good. But a couple of publications are using my program POZ ("Post Office Zones") for their second class mailings, and MOR uses both NoSpaces (in the soon to be released PPF ["Pearl Print Formatter"] that permits WS printer codes for bold, etc.) and MUP (Multi-Up and Cheshire mailing labels). The MOR editor keeps asking when I will release WCT (a word counting program she's been using that's accurate with WS files, handles hyphens properly and counts multiple files) so MOR can review it (soon, I promise). And there are fun little programs such as STW that turns my \$2000 computer into a \$5 stop watch. Not to mention the ones I'm still working on such as SBD (converts a script into a story board) and WiSP (prints WordStar files just like WS, with multi-spaced printing of a single-spaced file and multi-file printing with continuing headers and page numbers). And a data encryption program, and my magnum opus "MM" which does NOT stand for Marilyn Monroe. (Tooting one's own horn is a nasty job ... but it's fun.) I plan to distribute these and more by mail under the trade name "Idiot Savant." (Watch for it!)

So what's it like to be a programmer? Are the blank looks when you talk about programming and the loss of friends worth it? What kind of twisted personality does it require to be a programmer? Can a proper diet when young prevent it from happening? And most important, if you catch the bug should you commit suicide or simply quarantine yourself?

To answer the last question first, programmers automatically quarantine themselves. One programming instructor tells his class that the test for a potential programmer is that if they sit down at 5 p.m. and are surprised that the sun comes up only an hour later (in the winter!) ... they have it. The odd hours and the long periods of concentration seem to be inherent to programming, as even

mainland China has discovered. Despite the regime's discouragement, their programmers started acting the same way to the point that the regime gave up. (Time flies when you're having fun.)

Programming *is* fun. It's a challenge to build the idea, piece by piece, until the program is right. Usually it's more than you first wanted it to be, since it's natural to add features to your original plan as you create it (a popular expression in Multi-Image, the super-size of slide shows, goes: "Multi-Image shows are never completed ... they're just abandoned." It may also be true of programs ... the search for perfection continues forever). There's a sense of power and accomplishment as you make the computer do exactly what you want it to. Of course, if it was easy, everyone would do it. The occasional (maybe not so occasional) frustrations actually add to the enjoyment as you overcome them ("... it feels so good when I stop").

There's often an additional feeling of accomplishment not only because the program works, but because you know you've done a good job. There is style and elegance in a good program. Not something that you can show to your diminishing number of casual friends, but something that you can be proud of nevertheless. When a program really "sings," I smile and say, "Ignore that man behind the curtain!"

The experience most akin to programming that I've experienced was when, at the age of twelve or so, I learned to play chess. Perhaps you've noticed the youngster that becomes totally involved in chess for a while. I believe the fascination is not merely similar ... it is the same. The same kind of thinking as the moves/operations are analyzed step-by-step; as the grand sweep of the game/program is felt and understood; as the consideration of each move/step affects the development of the board/structure in ways that can only

intuitively be known; as strategy is board control/program blocking while tactics are pressures/nesting; and as sudden insight produces the most brilliant attacks on the opponent/problem. The main difference is that chess has a two-dimensional spatial geometry, while the geometry of programs is more like wormholes through a five dimensional universe with everything relating to everything simultaneously (or so it sometimes seems, especially with BASIC).

It helps if you have the so-called "logical mind." The kind of mind that can, when necessary, go from a premise through each step to a conclusion. Psychologists call it "formal thinking." You may not normally think that way; few people do. We generally make intuitive leaps. The logical mind, however, can then go back and see every step in the process even if it jumped from premise to conclusion. Ironically, the plodder that has to go through each logical step will most likely never be a good programmer.

This may seem like a contradiction. Writing the program is indeed a step-by-step process. But simply getting it down is not the whole process. When debugging (finding and correcting errors) the intuitive "sense" of where the problem might be can turn hours into minutes.

Also, intuition and creativity go hand in hand, and programming is a creative activity. Even a simple program can be written in a multitude of ways, and it takes both intuition and creativity to find the best.

Niklaus Wirth has defined what a program is in the title of his book "Algorithms + Data Structures = Programs." An algorithm is the method of solving a problem, and discovering that method is a creative act. It is, you might say, the concept of a program. Just as great art is a concept rather than merely paint (otherwise all art would be great), it is

\$995!*
8 MHZ
XT

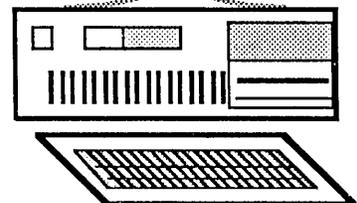


XT includes
RUN/CPM &
NEC V20
FREE!

*Includes
Powerful
Free Software!*

- * DOS 3.2
- * NewWord2
WordStar commands!
- * SuperCalc2
Superb Spreadsheet!
- * SuperFile
Free-form text filing!

\$1695!*
10 MHZ
AT



*Complete
Features!*

- * Speed: 4.77 & 8 MHZ
- * Memory: 640K
- * Pwr Supply: 135W
- * Keyboard: AT Style
- * Floppy: Two 360K
- * IBM-Compatible
- * Slots: Eight
- * Clock: Battery
- * Ports: Parallel
Serial
Game

**15 DAY MONEY-BACK
GUARANTEE!**
**1 FULL YEAR
WARRANTY!**

- * Speed: 10 MHZ
- * Memory: 1 Mb
- * Pwr Supply: 200W
- * Floppy: 1.2Mb
- * IBM-compatible
- * New IBM-5339 style Keyboard
- * Slots: Eight
- * Clock: Battery
- * Ports: Parallel
Serial
Reset Switch

21 Mb with 1 flpy... *Only \$1395**
45 Mb with 1 flpy... *Only \$1995**

with Fast Access
Seagate drives

30 Mb with 1 flpy... *Only \$2295**
45 Mb with 1 flpy... *Only \$2695**

For a complete system, add one of the monitor/graphics card packages listed below

MONOCHROME

- * Hi-res 14" Monitor!
Amber or Green, Swivel
- * 80 col X 56 rows!
For Word Processing
- * 132 col X 44 rows!
For Spreadsheets
- * 740 X 348 Graphics!
Hercules compatible
- * Parallel Printer Port!

Only \$295!

CGA COLOR

- * Hi-res 14" Monitor!
True RGB color, Swivel
- * 640X200, 320X200!
Up to 4 out of 16 colors
- * Fine-line Char Set!
For colorful Text
- * IBM-Compatible!
High quality CGA
- * Parallel Printer Port!

Only \$595!

EGA COLOR

- * Ultra Hi-Res NEC
Multi-Sync Monitor!
TTL and Analog RGB!
- * 650 X 350 Color!
Up to 16 out of 64 colors
- * EGA,CGA,MGA &
Hercules compatible!
- * IBM-Compatible!
- * Hi-res text in color!

Only \$995!

RUN/CPM Z80
5 MHZ Z80, 64K, full CP/M
compatibility!
Only \$199!

300/1200 Modem
Hayes compatible, includes
software!
Only \$199!

Phone: 415 - 489 - 5566
Tech Support calls
9am to 1pm PST only

WestWind
COMPUTER
33447 Western Avenue
Union City, CA 94587

60 MEG Tape
Your choice - Everex
Internal or External Tape!
Includes Tape drive, controller
card, cartridge, cables, manual
and software.
Only \$1295!

Prices & specifications subject to change without notice
* Base systems do not include graphics.

the algorithm rather than the code that makes a great program. (As paint follows concept, code follows algorithm.) Notice that Dr. Wirth left the language and code completely out of his definition. Dr. Wirth has not only designed six or more computer languages, most of which are in heavy use, but he is internationally respected as one of the "deepest thinkers" on the nature of languages and programming. His latest language is Modula-2, a further development of Pascal. Modula-2 is available for CP/M from Workman Associates, and I've been holding my breath ... and turning blue ... waiting for Borland to release their promised version. Maybe MOR will let me review it ... write and tell the editor!

I've been using the word "language" here, but I haven't slipped into technobabble. We call computer languages "language" because they have a syntax, a vocabulary, punctuation and parts such as nouns (the names of variables, constants and types), verbs ("do" and a lot of implied verbs), conditionals ("if," "while," "until") and even more grammatical clutter ("then," "and," "or," "in" and the symbols used as words and phrases such as "greater/lesser than," "is the same as," "make the same as" and so on). Computer languages differ from the evolved human languages (such as English and Japanese) in that computer languages have the special purpose of giving commands. One doesn't write love poetry in a computer language (the nerds may try). And computer languages are consistent (making them easier to learn) and logical.

But back to the personality twists that cause normal people to degenerate into programmers. You don't have to actually love detail, but it helps. More important, I think, is a certain perfectionism that permits you to tolerate the detail. There is a lot of detail, but programming is not for bean counters.

Despite all I've said to discourage

you, you may still decide to try it. Where should you start? I strongly recommend Borland's Turbo Pascal. Languages such as Pascal and BASIC are really just an idea. Like English, they are rules inscribed on air. To make the idea useful someone has to create a program that understands the language and translates it into the binary instructions that the computer "understands" (this is called a "compiler," "assembler" or "interpreter" depending on the way it works). So JRT made a terribly bad Pascal compiler, and Borland made a great one. Somewhere in between is UCSD Pascal.

Niklaus Wirth developed Pascal to teach the principles of programming. As such, it is easy to do things right the first time and avoid the frustrations and errors almost inherent in other languages. BASIC, on the other hand, is a very poor learning language. It teaches bad programming habits, inherently permits logical mistakes to be made, is difficult to understand, more difficult to debug, and makes it very difficult to keep track of and separate different operations in a large program. Programmers speak of "spaghetti code," meaning it is so entwined that you can't follow it. BASIC encourages spaghetti code with its "goto" statements and inflexibility about inserting lines, while Pascal helps keep the structure clear. That's one reason why they call Pascal a "structured language."

As for other languages, C permits machine-level operations that can create confusion for beginners and Modula-2 is for large programs. Both are easy to learn once you know Pascal.

A Pascal program reads almost like an English outline. The punctuation is a little different, but in hardly any time you will be reading (and writing) it.

With Turbo Pascal, you write a text file using the editor part of the program. You could use WordStar in the non-document mode if you wished,

but the Turbo editor makes it easier. Then you compile the program with four quick keystrokes. If the program is small enough to fit in memory this works just as easily as running a BASIC program from its editor. Otherwise the compiler creates a ".COM" file on your disk and runs that, still just as easy as any BASIC editor.

The Turbo editor works, for the most part, just like WordStar. Unlike BASIC, when you are done you have a ".COM" file that runs just like any other ".COM" file. Simple programs are simple to write, yet Turbo permits the most sophisticated techniques and giant sized programs when they are needed. If Turbo finds an error when it is compiling or running in memory it returns you to the very place in your code where the error was found, along with a message telling the kind of error. If running from a ".COM" file, a similar and easy process finds the place.

Programs written in Turbo Pascal compile fast ... and the programs run the same way. Much faster than BASIC. And should you, in the future, get an MS-DOS or Macintosh computer, your source code can be transferred over, compiled, and run on them (with, perhaps, a few small changes).

Borland offers two other packages that go along with Turbo Pascal; "Turbo Tutor" and the "DataBase Toolbox" (they also have a "Graphix Toolbox" and an "Editor Toolbox," but neither of these is available for CP/M).

I recommend Turbo Tutor. As a tutorial it is nowhere as useful as the books I recommend below, but it does have one big advantage. There's a disk with all the program examples ready for you to use or modify. And these are useful programs, routines and examples.

The DataBase Toolbox has a database and a sorting program that you can modify, as well as GINST.

GINST is a General INSTallation program generator. You give it your program name and it automatically creates a terminal and screen installation program for your program. Also included is documentation for the installation program that you are permitted to copy and use, although it needs the attention of an editor (the kind that understands English). GINST is needed if you are going to distribute your programs. Unfortunately no way is provided to modify the installation programs thus created so they will set defaults and so on (but you can make a second installation program with Turbo that will do that). In general, the DataBase Toolbox is a bit advanced, and unless you need GINST or have database programming as your immediate goal you can delay picking it up. Borland does have a package deal on all three items, however.

As to whether diet is a factor, I expect the reputation programmers have for living on Twinkies and Coca-Cola is because they can get and consume these without interrupting themselves. I personally prefer coffee (sweet), beer and cigarettes so that I get the four basic food groups of caffeine, sugar, alcohol and nicotine.

I hope you've gained a sense of what programming is like, and a little advice to get you off on the right foot. Your friends may not understand your enthusiasm and may suddenly consider you a fool (what's so exciting about that?) or a genius (everyone knows it takes a genius to program a computer). Or they may simply be confused (like a new house mate that had a friend "that can speak Pascal"). I know it's fun for me, and it keeps me from hanging around on the street.

And Mom? She just loses parity (whoops - gets confused) when I talk about programming, and wonders where she went wrong.

Recommended books:

"Pascal From BASIC" by Peter

Brown, Addison-Wesley Microbooks 1982, paper \$12.95, 175 pp w/ full index. If you've used BASIC, you want this one. Even if you haven't, it is still worthwhile as a tutorial. It's an amusingly written comparison of the two languages with the differences and advantages of Pascal highlighted. Teaches Pascal basics as well as some programming style and principles.

"Programming in Pascal" by Peter Grogono, Addison-Wesley 1980 (revised), paper and hardbound, 363 pp w/ index. A textbook that starts at the beginning and gets advanced at the end. Clear examples and explanations. Serves well as a reference book. This book taught me more than the others.

"The Pascal Handbook" by Jacques Tiberghien, Sybex 1981 (may be later editions), paper \$19.95, 485 pp, book arrangement makes an index unnecessary. A reference worth having right at hand as you learn. The syntax and structure of Pascal are clearly diagramed with explanations, although it is not a tutorial. Sybex has some other books on Pascal, and I expect, without examining them, that they are as good.

"Pascal With Style: Programming Proverbs" by Henry F. Ledgard, Paul A. Nagin and John F. Hueras, Hayden 1979, paper \$9.50, 210 pp w/ index. Don't run out and buy this book until you've done a little Pascal programming. It's a bit of "deep think" about principles, and you won't be ready for it immediately. Then, however, it will answer some of the questions about efficiency and programming style that you will have started wondering about.

Data Structures and Algorithms by Niklaus Wirth, article, Scientific American, Sept. 1984, v251 i3, pp 60-69. Although this may be slightly advanced for the rank beginner, it is an excellently written, clear and concise discussion of underlying principles with an unusually clear explanation of the use of pointers ... a subject most find a little difficult to grasp at first.

Worth photo copying at your local library if you can't find a back issue. Lee McKusick tells me that this is an excerpt from "Algorithms + Data Structures = Programs" by Niklaus Wirth, Prentice Hall 1976 (may be later edition), hardbound \$36.98, 366 pp w/ index, which I have not seen.

"Introduction to 8080/8085 Assembly Language Programming - a Self-Teaching Guide" by Judi N. Fernandez and Ruth Ashley, John Wiley & Sons 1981, paper \$12.95, 303 pp w/ an apparently complete index that doesn't always help. Although I do not recommend you start programming with assembly language unless you are a teen-age genius, this book will help you understand hexadecimal, octal and binary numbers (little of which is needed with Pascal, but is still useful) how to do the math to convert one to another and a bit of how your CPU actually works. As the subtitle indicates, this is a self-teaching guide; actually a workbook. It first explains, then offers questions and problems. Each lesson is short and clear, and questions refresh your memory with further drill. As a reference book it falls flat, and as the title indicates, it is introductory. It will not, by itself, teach you assembly language programming. If you like puzzles or math, you'll love it.

NOT recommended:

"Pascal User Manual and Report" by Kathleen Jensen and Niklaus Wirth, Springer-Verlag 1974 (second edition), paper \$9.50, 198 pp w/ index. Niklaus Wirth invented Pascal, and this is the seminal work with which it was introduced. Therefore you will often find it recommended as the prime Pascal book. Do not be fooled! It is neither a tutorial nor a reference book, but is more of a scientific paper for computer language designers. As such, it can be unintelligible to even the intelligent computer user. ■

See page 47 for a complete reference list.

Good Timing

A Nuts and Bolts Approach to Building Your Own Real-Time Clock

By Mike Allen

For a long time I had wondered to what good use I could put the I/O expansion header (that strange looking double row of 20 pins each) in my revision two MD3. Then in the March 1986 *BYTE*, Ciarcia's Circuit Cellar took a look at real-time clocks.

"Aha," I said to myself, "just what I need." So I took a look at the schematic for the clock that used a 8255 Programmable Peripheral Interface and the MSM5832 clock. It looked to me that all I had to do to make it work with the I/O expansion connector was to invert the RESET-(reset not) signal.

So off I ran to my junk box (I have high class junk!) and my catalogs to see where I could get what for the cheapest price. It seemed that JDR Microdevices (1224 S. Bascom Ave., San Jose, CA 95128) had the best prices on all the critical parts. Off went an order to them. Had I purchased the parts that I dug out of my junk box from local sources I would have

Mike Allen is a regular MOR writer with a special knack for the "nuts and bolts" approach to computers and software.

spent a total of \$18.69 ... I spent a little less.

The circuit is based on the MSM5832 clock chip from OKI. The thing is an old chip and is slow but has the advantage of being cheap. It will run on as little as 2.2 volts and draws minimal power in the standby mode, ideal for battery back-up.

The MSM5832 is organized around a four bit bus structure (remember the 4004?). It has four address and four data lines. There are also READ, WRITE, HOLD and Chip Select lines. The functions of READ, WRITE and Chip Select are obvious. The HOLD line is strange ... it stops the clock! I suppose there are uses for it when reading, but I couldn't find any. While setting the clock, it is used to stop the clock until you want it to start. That's useful.

The addresses to the MSM5832 from 00H through 0CH are used to address 13 four bit registers within the MSM5832. These registers have four bit values as follow:

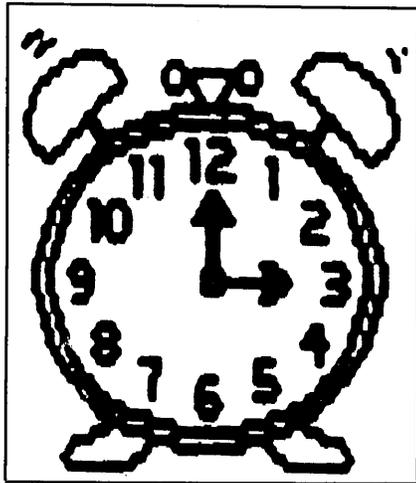
(NOTE: These differ from the table in Ciarcia's article. Through hard experience I have found these are correct.)

ADDRESS FUNCTION
RANGE

00	1s of seconds	0->9
01	10s of seconds	0->5
02	1s of minutes	0->9
03	10s of minutes	0->5
704	1s of hours	0->9
05		
bits 0,1	10s of hours	0->2
bit 2	AM(0)/PM(1)	0->1
bit 3	12(0)/024(1) hr format	0->1
06	Day of the week	0->6
07	1s of day of month	0->9
08		
bits 0,1	10s of day of month	0->3
bit 2	regular(0)/leap(1) year	0->1
09	1s of months	0->9
0A	10s of months	0->1
0B	1s of years	0->9
0C	10s of years	0->9

To read the MSM5832 you set the READ line high and send out an address. Then you wait six usec and read the data. Writing is more difficult. You send out the HOLD line, wait 150 usec then send out the address. Wait six usec and then send out the data. Set the write line high for one usec and then back low. Drop the hold line when you want the clock to start.

Since it is so slow, it cannot interface directly with the Z80 bus. To give you an example, it takes six usec for the data to be ready after the address is sent. This is too slow for the bus that is available in the MD. The way around this is to use a 8255 to buffer the address, data and control lines. I took the circuit directly from the Ciarcia article, using the bottom four lines of the 8255 A port as bidirectional data lines to the MSM5832, the bottom four lines of the B port as the address lines to the



MSM5832 and the top four lines of the C port as the control lines to the MSM5832. In this way all the timing can be established in software. For instance, I can send out the address on the B port, waste some time and then read the result on the A port. It's inefficient 8255 use, but who cares. I decided to use the IO30- signal for the 8255 chip select. This means that port A is at I/O location 30H, port B at 31H, port C at 32H and the control for the 8255 at 33H.

The only critical part in the circuit is the 1N34A diode. If you must substitute, make sure you use a germanium diode with a forward voltage drop of 0.5 volts or less. I used a piece of perf board that I had lying around to build the clock. The 3x5 perf board that I call out in the parts list is a little large, but it can be

cut down. I used five minute epoxy to glue down the sockets and the battery holder. I stuffed the various components where it looked like they fit the best and then point-to-point wired everything using #26 solid wire. It got tedious, especially soldering the 40 conductor ribbon cable to the board. Then I installed the 40 pin connector on the end of the ribbon cable. I found the best way to do this was to put a couple of blocks of wood in a vise and then use it to squeeze the connector onto the cable. The wood helps to distribute the force evenly across the connector and also protects the connector itself. Probably the easiest way to mount the board is with some foam tape that is sticky on both sides. Radio Control modelers use it often. It's called "servo tape" and can be found at most any hobby shop.

After a couple of days of trying to induce blindness in my already weak eyes, it was time to try it. I plugged it in, being careful to make sure that I had the connector oriented correctly, and started checking voltages. Being a chicken at heart I had not installed the chips in the board yet. Everything seemed like it was what it should be, so I turned off the MD, plugged in the chips and turned the power back on.

There it sat. The MD worked fine, but I wasn't sure what the clock was doing, if anything. Back to Ciarcia's article. He had a BASIC program that was supposed to allow you to read and set the clock. I pulled out my pristine BASIC disk (I program mostly in "C" or Pascal) and entered the program. Well, things looked OK. I could set and read the clock, although in a rather useless format. Then I noticed that the seconds weren't setting properly ... actually, though, they were. What Ciarcia failed to mention is that you can't set the seconds! Writing to the seconds and tenths of seconds resets them to zero. Now I understood that! The next thing I noticed was that, according to Ciarcia, the days of the week were supposed to be a number

between one and seven. I was checking things out like leap-years and 12 and 24 hour time formats and I had set the day of the week to seven. It ticked over and went to zero! Checking some other articles that I have on the MSM5832 clock I found out that the day of the week goes between zero and six.

So, I started writing programs to make pretty time and date displays on my MDT60. Things were going swimmingly until I noticed that every time I pressed the reset button, I lost about 30 seconds. Investigating further I found out I was losing the time from when I pressed the reset button and when I called up the program to read the time. Looking at the schematic more closely with my trusty Intel data book at my side I discovered that when the 8255 was reset all the I/O lines go to the input mode. This meant that the hold line which is normally output from the 8255 to the MSM5832 was being pulled high and staying there until the 8255 was programmed by the read time program. It made me wonder if Ciarcia had ever even built the circuit.

After much head-scratching I decided that the easiest approach was to force the Chip Select to the MSM5832 to be low until the 8255 could be programmed. This was done by using another part of the 74LS04 that I was using to invert the RESET signal. I used it to invert another I/O line from the 8255. This line would be pulled high (and inverted low to the MSM5832) until the 8255 could be programmed and that particular line commanded low. As long as the MSM5832 is not selected, it keeps time but ignores all the command lines. Just what I needed. The clock ignores all commands until the 8255 is programmed properly!

Aligning the clock is easy and tedious. Just see how much you are off in a given period of time (say a day) and then adjust the trimmer capacitor slightly; less capacity if you are running slow, more if you are fast. If

you have access to a frequency counter there is a faster way. Send the clock an address of 0FH. Connect the counter to pin nine of the MSM5832 and adjust the capacitor until you read exactly 1024 hz.

As an exercise, I decided to see if I could write programs in all the different languages that I had to read and set the clock. I was able to read and set the clock in 8080 assembly language, MicroSoft BASIC, MIX "C" and Turbo Pascal 3.0. I was unable to write routines in PILOT since I could find no instruction to read an I/O port although there is one to output to an I/O port. The programs are rather brute force, but they work. I also wrote an insert for use with BYE5.

About this time a buddy of mine with a Radio Shack Color Computer

(at which I had been scoffing greatly) showed me a public domain schematic drawing program that he had. It was really a nice effort and worked well, if a little slowly at times. It ended up creating nine disk files which were screen dumps of different schematic parts. We took a look at the program that took those files and sent them to the printer and decided that it would be easy to write an 8080 program to do the same thing. So, I sat down in front of the CoCo and drew the schematic. We then transferred the files to my MD by modem and I sat down to write the CADPRINT program. Many hours of confusion later, I had a program that could take the screen dumps generated by COCOCAD on a CoCo and dump them to my Epson MX-80 to print the schematic.

The bottom line is that I now have

a file called MDCLCK12.LBR that contains all the programs I have written for using the clock, a parts list, this article and the CADPRINT program for an EPSON or OKI ML92/93 printer and the nine files that contain the schematic. There are also a couple of nice programs to set and read the clock by John Osnes who also made the mods to CADPRINT for the OKIs. Hopefully some people out there will look at the CADPRINT program and make modifications so that it can be used with other printers. I do hope this happens, because I plan on using CADPRINT with other hardware articles in the future. ■

MDCLCK12.LBR is available on the MOR BBS or on FLOB #4109.

Real Time Clock Parts List

For Morrow Micro-Decision MD2/3 Rev. 2 and MD5 Up (8/26/86)

QTY.	DESCRIPTION	PRICE	SUPPLIER	QTY.	DESCRIPTION	PRICE	SUPPLIER
CHIPS				RESISTORS			
1	74LS04	0.16	JDR	1	10x13 DIP (14 PIN)	0.85	JDR
1	MSM583	2.95	JDR	1	10K	0.05	LP
1	8255-5	1.89	JDR	1	100 OHM	0.05	LP
SOLDER-TAIL SOCKETS				CONNECTOR			
2	14-PIN (0.11@)	0.22	JDR	1	IDS40	1.99	JDR
1	18-PIN	0.15	JDR	CABLE			
1	40-PIN	0.30	JDR	1FT	40 COND. RIBBON	0.72	JDR
1	3V. BATT HOLDER	1.49	JDR	BOARD			
CRYSTAL				1	3"x5" PERF. BOARD	1.50	LP
1	32.768 KHZ	0.95	JDR	TOTAL \$18.69			
BATTERY				JDR=JDR MICRODEVICES, 1224 BASCOM AVE., SAN JOSE, CA 95128			
1	3V. LITHIUM	3.95	JDR	LP =LOCAL PURCHASE OR FOUND IN MY JUNK BOX! THE PRICES, HOWEVER ARE REAL.			
DIODES							
1	1N34A (GERMANIUM)	0.20	LP				
1	1N4002 (SILICON)	0.06	LP				
CAPACITORS							
1	4.7 UFD/25WVDC	0.16	LP				
1	20 PF	0.10	LP				
1	5-35 PF TRIMMER	0.95	LP				

A Customized Mailing List

Simple Basic Programming Eases Your Work

By Erik Westgard

Writing computer programs is hard work. The effort doubles for those of us who are "word people," and don't enjoy mathematical challenges.

Keeping a mailing list is something you can do with either word processing software or a database package. You will have to set up a narrow, carefully paginated document format or a merge system to use a word processing package for a mailing list and to have it print properly on labels. The same is true with a database. You'll need to set up a unique format the first time, and then use that format again each time you want to add or run labels.

What if you want to hire a temporary employee to maintain the list or add names during holiday rush? Do you have time to teach that person all they need to know about your word processing package or database to do the job without getting into trouble?

Large businesses go through this all the time. They need specific tasks done with minimal operator training and high accuracy levels. Often they go to the expense and trouble of tailoring programs to run on a large computer or even a series of microcomputers. Similarly, you can hone your data processing skills and run your own business better by occasionally writing custom programs. This article shows you how to write a simple menu-driven mailing list program.

The most important part of any programming effort is to

Erik Westgard works as a sales engineer for a computer company in the midwest. In his spare time he races sailboats and writes articles for model railroading magazines.

identify business needs. This simply means determining what function needs to be performed by the computer. It has to be something very specific. Think of the steps involved in planning to build a house. Just deciding that you want it "big and comfortable and under 100,000 dollars" is not enough. You must say how many bedrooms you need, how big the living room should be, and whether or not you want a fireplace.

The same is true when programming a computer. What's in a name and address? How many lines? How long is each line? Are you dealing with character data or numbers? Just write out a sample name like this:

Ms. Martha Robinson
Acme Manufacturing
2100 Industrial Court
Anytown, MN 55203-0000

Four lines should be enough. There is a limitation in BASIC on CPM 80 computers that you can only use a total of 128 positions for a given record, which for the program leaves 32 characters for each of three lines. Write the information down on paper.

- 1) (32 spaces) Name (WHOS)
- 2) (32 spaces) Address1 (AFS)
- 3) (32 spaces) Address2 (ASS)
- 4) (32 spaces) Address3 (ATS)

The first line is for the name, and is called (WHOS). The dollar sign in BASIC stands for "string variable," which is a made up value or any kind of character data of a given

length. The address and or company name is divided into three lines called (AF\$), (ASS) and (AT\$), short for address first, second and third.

Now that your data is defined, you must be specific about your program structure. Do the following steps in order:

- 1) Type the names and addresses into the computer
- 2) Store them permanently on a diskette
- 3) List the names on the screen
- 4) Print mailing labels
- 5) Add, delete or change names or addresses

If you know a programming language, you can look at the five step list and the four fields to write a program, just as if you know French you can find of *Paris Match* and read aloud an English translation. Assuming you have minimal programming ability, it is helpful to find an example program and change things around a bit. Open your "Microsoft BASIC 80" book, and turn to page 4-1, an appendix called "Microsoft BASIC Disk I/O." This chapter tells how to store your data on a diskette. First, you will open a space on your diskette called a file, then you will type your names into memory, and then you will have BASIC write those names out to the disk file and close the file. To access the names already out on the disk, you need simply to open the file, read the records until you find the name or names you are looking for, send them to the printer, and close the file. It will seem simple once you have tried it.

If you can find the time, read the whole chapter. Otherwise refer to the program and section beginning on page 4-6, "Random Files." Listed below is the first part of the program. Anytime you don't understand a statement refer to the reference section of the book, starting on page 2-1.

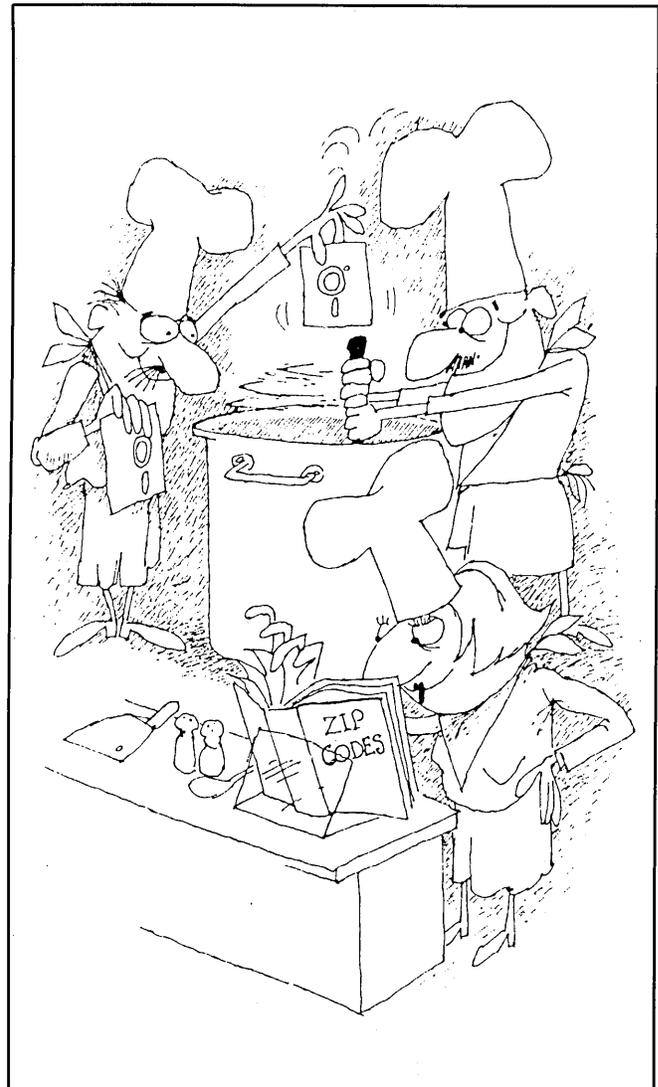
```
10 ON ERROR GOTO 0
20 REM *A simple mailing list. (C)Erik Westgard 1986
30 COUNT% = 1
40 PRINT "OPENING A:MAILING.DAT"
50 OPEN "R", #1, "MAILING.DAT",128
60 FIELD #1,32 AS WHO$,32 AS AF$,32 AS ASS,32 AS AT$
```

At this point, you have set a name count to one in line 30 and opened your mailing list file as file number one with 128 total characters per record, or the total of the name and three address lines.

Next you'll open the file and read in the data one record at a time to find out how many names you have stored in the file. This is important if you have to add any on at the end, which happens often. This time, you will use a programming convention called a "loop" to help the work and to keep track of the count. Read one record in from the diskette, check for a name in the name field, and if one is

there, get another record. You are at the end if you read a blank name field.

```
70 FOR COUNT% = 1 TO 100
80 GET #1, COUNT%
90 IF ASC(WHO$) < 33 THEN 110
100 NEXT COUNT%
110 PRINT COUNT%-1;" Names found"
```



Next you can write out a menu, so the person running the program can pick a function. You'll even keep the function numbers you used above. You will store the names as soon as they are entered, so skip the step called "store names" above.

```
120 PRINT:PRINT "Please select a function"
130 PRINT:PRINT "1) Enter new names"
```

```

140 PRINT "2) Close files and exit"
150 PRINT "3) List the names on the screen"
160 PRINT "4) Print mailing labels"
170 PRINT "5) Update the master list"
180 PRINT: INPUT "Enter a selection";S

```

Next look at the selection entered, and jump to sections of the program which will perform needed function. You'll use a series of ON-GOTO statements which jump to the first line number on one, the second one on two, etc. You could use a series of IF-THEN statements, but it would take five of them.

```

190 ON S GOTO 220,1090,390,510,730
200 PRINT "Please re-enter your selection number"
210 GOTO 180

```

If an invalid number is entered, say a six, assume that you don't understand and jump back to the question again. Now you are set to add names, you first function. For this, ask for the name and address one line at a time, and when you are done and satisfied with the entry you will join all four lines into a 128 character record. Then you will write that record to the disk, adding one to the record count. Note in line 220 there is a "REM," or "remark" statement to put a comment in the program to identify each program subsection.

```

220 REM *** ADD NEW NAMES ***
230 PRINT "Add a name. Use <Enter> for blank lines."
235 PRINT "Max 32 letters per line. Use no commas.": PRINT
240 INPUT "Enter full name";N$
250 INPUT "Enter street address";A1$
260 INPUT "Apartment or box information";A2$
270 INPUT "Enter city";C$
280 INPUT "Enter state or country";SAS$
290 INPUT "Enter zip code if any";Z$
300 INPUT "ALL OK (Y OR N)";OK$
310 IF OK$="N" THEN 230
320 STZ$=C$+CHR$(44)+" "+SAS$+" "+Z$
330 LSET WHO$=N$: LSET AF$=A1$: LSET AS$=A2$
340 LSET AT$=STZ$
350 PUT #1,COUNT%: COUNT% = COUNT%+1
360 PRINT: INPUT "ANOTHER NAME (Y OR N)";GAS$
370 IF GAS$="Y" THEN 230
380 GOTO 120

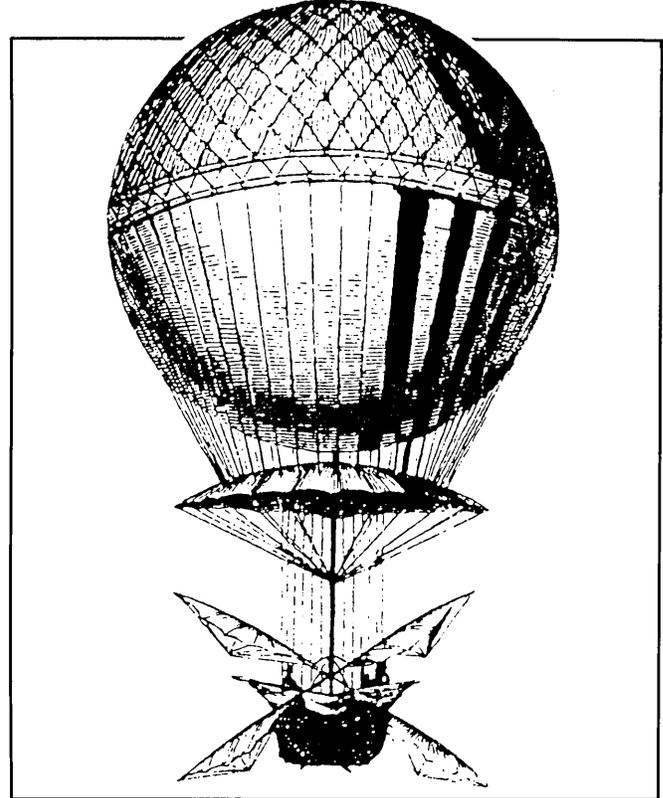
```

Use the LSET statements to set the variable information typed in to equal the values line WHO\$ in the record written out to the disk. The PUT statement writes the completed record. Finally, add one to the record counter and write the code for function three, listing the names on the screen.

```

390 REM *** LIST NAMES AT THE TERMINAL **
400 FOR L%=1 TO COUNT%-1
410 GET #1,L%
420 PRINT WHO$
430 PRINT AF$
440 IF ASC(AS$) < 33 THEN 460
450 PRINT AS$
460 PRINT AT$
470 PRINT
480 NEXT L%
490 INPUT "Press <Enter> to continue";GOS$
500 GOTO 120

```



In this section you merely step through all the records in the file, getting them one at a time. Then write each non-empty record to the screen. When all the records have been written, go back to the main menu. Next you will print out the labels.

```

510 REM *** PRINT SECTION ***
520 PSTAT=INP(245)
530 REM * TEST IF PRINTER IS READY - MODIFY IF NEEDED
540 IF PSTAT = 240 THEN 560
550 GOTO 570
560 INPUT "The printer is not ready. Press ENTER";PSS$
570 INPUT "Enter label length in lines";LLONG
580 FOR P%=1 TO COUNT%-1
590 LINES=0
600 GET #1, P%
610 LPRINT
620 LPRINT WHO$

```

```

630 LPRINT AF$: LINES = LINES + 3
640 IF ASC(AS$) < 33 THEN 660
650 LPRINT AS$: LINES = LINES + 1
660 LPRINT AT$
670 LINES=LINES+1
680 FOR BL=LINES TO LLONG
690 LPRINT
700 NEXT BL
710 NEXT P%
720 GOTO 120

```

In this section you again stepped through all the records on the disk and printed out to the line printer all the non-blank records. You kept track of how many lines per label were printed and added blanks at the end as needed. Line 640 checks for and does not print blank lines in the address. Lines 520 to 560 are a test to see if the parallel printer is ready on a newer MD-2. Otherwise, BASIC hangs up the computer if something is wrong with the printer. You can modify or leave out this section altogether, as the actual print section starts at line 570. Measure your continuous perforated labels you would use with a ruler. Assume six lines per inch, which is the standard line spacing for most computer printers. (Sometimes eight lines per inch is optional, though is not supported by this program.) Doing the updates occurs next.

```

730 REM *** UPDATE NAME OR ADDRESS ***
740 PRINT:PRINT "Update area. Enter name to change or delete"
750 PRINT "Enter a 99 to quit."
760 PRINT:"Enter full name";FINDIT$
770 IF FINDIT$="99" THEN 120
780 HOWLONG=LEN(FINDIT$)
790 FOR LOOK%=1 TO COUNT%
800 GET #1,LOOK%
810 IF LEFT$(WHOS$,HOWLONG) = FINDIT$ THEN 840
820 NEXT LOOK%
830 PRINT:PRINT"Name not found. Please re-enter":GOTO 740
840 PRINT WHO$: PRINT AF$: PRINT AS$
850 PRINT AT$
860 PRINT "Enter changes or 22 to delete name"
870 INPUT "Enter new name";NNS$
880 IF NNS$="22" THEN 1010
890 INPUT "Enter new address";ADD1$
900 INPUT "Apartment or box info";ADD2$
910 INPUT "Enter city";CT$
920 INPUT "Enter new state or country";SS$
930 INPUT "Enter new zip code";ZZ$
940 INPUT "OK? (Y OR N)";OKN$
950 IF OKN$="N" THEN 870
960 CISTZ$=CT$+CHR$(44)+" "+SS$+" "+ZZ$
970 LSET WHO$=NNS$:LSET AF$=ADD1$
980 LSET AS$=ADD2$:LSET AT$=CISTZ$
990 PUT #1,LOOK%
1000 GOTO 120

```

```

1010 REM * DELETE A RECORD *
1020 PRINT "**** WORKING ****"
1030 FOR ER% = LOOK% TO COUNT%
1040 GET #1,ER%+1
1050 PUT #1,ER%
1060 NEXT ER%
1070 COUNT% = COUNT%-1
1080 GOTO 120

```

In the update section you try to find a name record by searching the whole file, looking for a match in the names. If one is found, you get the record, make any changes, and write (PUT) the record back out to the disk. Deleting records is more complicated. If you just delete the record in the middle of the file, the count made during the first program run would stop at the empty record. Get the next record after the one you wish to delete and write it on the deleted record. Then add one to the count and repeat the process.

The final function is to close the file and exit the program. If you exit the program without a Control-C, the file will not close and data loss may occur.

```

1090 REM *** CLOSE AND EXIT ***
1100 PRINT "CLOSING FILES...."
1110 CLOSE #1
1120 END

```

```

10 ON ERROR GOTO 0
20 REM * A simple mailing list. (C) Erik Westgard 1986
30 COUNT% = 1
40 PRINT "OPENING A:MAILING.DAT"
50 OPEN "R", #1, "MAILING.DAT",128
60 FIELD #1,32 AS WHO$,32 AS AF$,32 AS AS$,32 AS AT$
70 FOR COUNT% = 1 TO 100
80 GET #1, COUNT%
90 IF ASC(WHO$) < 33 THEN 110
100 NEXT COUNT%
110 PRINT COUNT%-1;" Names found"
120 PRINT: PRINT "Please select a function"
130 PRINT:PRINT "1) Enter new names"
140 PRINT "2) Close files and exit"
150 PRINT "3) List the names on the screen"
160 PRINT "4) Print mailing labels"
170 PRINT "5) Update the master list"
180 PRINT: INPUT "Enter a selection";S
190 ON S GOTO 220,1090,390,510,730
200 PRINT "Please re-enter your selection number"
210 GOTO 180
220 REM *** ADD NEW NAMES ***
230 PRINT "Add a name. Use <Enter> for blank lines."
235 PRINT "Max 32 letters per line. Use no commas.": PRINT
240 INPUT "Enter full name";N$

```

```

250 INPUT "Enter street address";A1$
260 INPUT "Apartment or box information";A2$
270 INPUT "Enter city";C$
280 INPUT "Enter state or country";SA$
290 INPUT "Enter zip code if any";Z$
300 INPUT "ALL OK (Y OR N)";OK$
310 IF OK$="N" THEN 230
320 STZ$=C$+CHR$(44)+" "+SA$+" "+Z$
330 LSET WHO$=N$: LSET AF$=A1$: LSET ASS$=A2$
340 LSET AT$=STZ$
350 PUT #1,COUNT%: COUNT% = COUNT%+1
360 PRINT: INPUT "ANOTHER NAME (Y OR N)";GA$
370 IF GA$="Y" THEN 230
380 GOTO 120
390 REM *** LIST NAMES AT THE TERMINAL **
400 FOR L%=1 TO COUNT%-1
410 GET #1,L%
420 PRINT WHO$
430 PRINT AF$
440 IF ASC(AS$) < 33 THEN 460
450 PRINT ASS$
460 PRINT AT$
470 PRINT
480 NEXT L%
490 INPUT "Press <Enter> to continue";GO$
500 GOTO 120
510 REM *** PRINT SECTION ***
520 PSTAT=INP(245)
530 REM * TEST IF PRINTER IS READY - MODIFY IF NEEDED
540 IF PSTAT = 240 THEN 560
550 GOTO 570
560 INPUT "The printer is not ready. Press ENTER";PS$
570 INPUT "Enter label length in lines";LLONG
580 FOR P%=1 TO COUNT%-1
590 LINES=0
600 GET #1, P%
610 LPRINT
620 LPRINT WHO$
630 LPRINT AF$: LINES = LINES +3
640 IF ASC(AS$) < 33 THEN 660
650 LPRINT ASS$: LINES = LINES +1
660 LPRINT AT$
670 LINES=LINES+1
680 FOR BL=LINES TO LLONG
690 LPRINT
700 NEXT BL
710 NEXT P%
720 GOTO 120
730 REM *** UPDATE NAME OR ADDRESS ***
740 PRINT:PRINT "Update area. Enter name to change or delete"
750 PRINT "Enter a 99 to quit."
760 PRINT:"Enter full name";FINDIT$
770 IF FINDIT$="99" THEN 120
780 HOWLONG=LEN(FINDIT$)
790 FOR LOOK%=1 TO COUNT%
800 GET #1,LOOK%
810 IF LEFT$(WHO$,HOWLONG) = FINDIT$ THEN 840
820 NEXT LOOK%
830 PRINT:PRINT"Name not found. Pleae re-enter":GOTO 740
840 PRINT WHO$: PRINT AF$: PRINT ASS$
850 PRINT AT$
860 PRINT "Enter changes or 22 to delcte name"
870 INPUT "Enter new name";NN$
880 IF NN$="22" THEN 1010
890 INPUT "Enter new address";ADD1$
900 INPUT "Apartment or box info";ADD2$
910 INPUT "Enter city";CT$
920 INPUT "Enter new state or country";SS$
930 INPUT "Enter new zip code";ZZ$
940 INPUT "OK? (Y OR N)";OKN$
950 IF OKN$="N" THEN 870
960 CISTZ$=CT$+CHR$(44)+" "+SS$+" "+ZZ$
970 LSET WHO$=NN$:LSET AF$=ADD1$
980 LSET ASS$=ADD2$:LSET AT$=CISTZ$
990 PUT #1,LOOK%
1000 GOTO 120
1010 REM * DELETE A RECORD *
1020 PRINT "**** WORKING ****"
1030 FOR ER%= LOOK% TO COUNT%
1040 GET #1,ER%+1
1050 PUT #1,ER%
1060 NEXT ER%
1070 COUNT% = COUNT%-1
1080 GOTO 120
1090 REM *** CLOSE AND EXIT ***
1100 PRINT "CLOSING FILES...."
1110 CLOSE #1
1120 END

```



TABLE OF LATEST MORROW REVISIONS

PC Board / Computer	ROM Rev.	CP/M Bios Rev.
Rev. 1.1 MD2 or MD3	1.3	1.6
Rev. 2.0 MD2	2.5	2.3
Rev. 2.0 MD3	3.1	3.1
Any hard disk MD	1.9	2.0

Revision 2.0 Printed circuit (PC) boards have a paralel printer port through the card edge connector and use SETUP to set baud rates on all serial ports. The revision number is engraved on the board.

Computers and the IRS

By Serge Timacheff

Planning to use your computer as a tax deduction? Keep good records. The IRS will want to know more than just how much your system cost.

Taxpayers rarely use their home computers just for business purposes. While it is perfectly legitimate to deduct business related computer usage, including all those times you played Zork or chess isn't exactly ethical.

Joe Calderaro, Technical Coordinator of the San Francisco IRS District office, says "Taxpayers should ensure that all usage is well documented to the extent that each time the computer is used, a record shows the hours of business usage, the purpose, and the date. Without that, the IRS may well decide that he's just using it for pleasure."

The more evidence you can show that you spent time in business related work, the better. Word of mouth is seldom sufficient.

Let's say you get audited. What about putting together all the information the night before the audit, backtracking and constructing it from memory using twenty different pens and slips of paper? "It seldom works," says Calderaro. "It's easy to tell if a timely diary is kept. It is better to have

a log you keep beside the computer."

Make sure, if you plan to take deductions for your computer, that you document the business use. If you are self-employed, keep a log showing what client or prospect for whom you are working. Office supply and computer stores sell logs just for this purpose. If you are working for an employer, the use must be *required* by that employer and satisfy IRS Revenue Ruling 86-129, which includes the "required as a condition of employment" section.

To get more information about using your computer as a tax deduction, contact the main library in your area, any tax information source or write to any IRS District Disclosure Office. ■

Special Credits

Cover Drawing.....Robert Johnson
 Drawing, page 30.....Bruce Braverman
 Production Assistant.....LeMel C. Williams
 FLOB Manager.....Steven Wartofsky
 Caricatures, pages 5, 7,
 10 & 12.....Robert Johnson

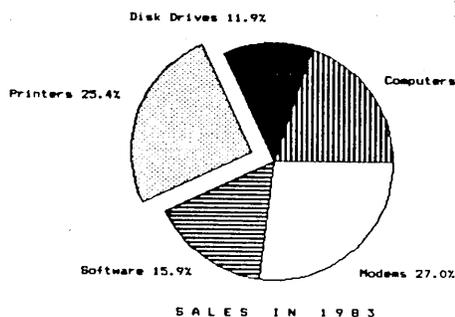
GRAPHS WITHOUT GRAPHICS?

... a superb program accompanied by an exceptionally clear manual at reasonable cost.

-- Morrow Owners' Review, April/May 1986

The quality of the graphs produced by DataPlotter is better than anything I had seen before....
 I highly recommend this program to anyone who needs to make high-quality graphs.

-- Profiles Magazine, 1985



No need for color monitor or graphics board.
 Make graphs on dot matrix printers.

Easy to Use. No Programming.
 CP/M 2.2, 3, 80, or 86, MS-DOS or PC-DOS.
 Excellent Manual. Most disk formats.

DataPlotter™

Line Graphs & Scatterplots . . . \$69
 Bar Graphs & Pie Charts . . . \$69

Both for \$99

(Prices include manual)

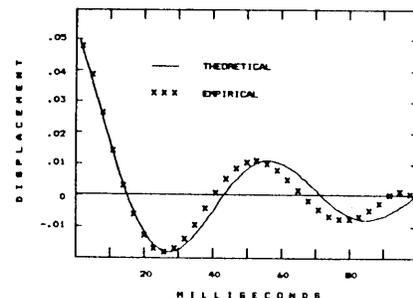
Add \$3 shipping.

\$8 outside US and Canada.

Specify type of Printer.

Lark Software™
 131 N. Leverett Rd.
 Leverett, MA 01054

(413) 773-8687 Visa, M C



Make Those Function Keys Work!

By Gordon Northrup, MD

This article is for a select audience ... owners of the MDT-60 or the MT-70 terminals with an earlier ROM than 3.1. If you have ROM 3.1, you also have KEY.COM and instruction sheets called READ ME FIRST! The ROM number appears on your screen when you boot up. When I bought my MD-3 and MDT-60 the function keys didn't work, but I was enthusiastic and went ahead to define them with SmartKey. When I held down the cursor keys, though, unwanted characters such as "L" crept into the text. For all my efforts, I couldn't fix it. Fortunately, Morrow did, and it's available through MOR as the ROM 3.1 kit which includes CBIOS 3.1 (a corresponding part of CP/M), KEY.COM, DEFAULT.KEY and other good things.

Morrow designed the ROM 3.1 system so that the user didn't have to know that CP/M existed. The Co-Pilot Menus did it all, including function key definition for NewWord and SuperCalc ... though not for the other bundled software and minimally for CP/M. You can do the same, courtesy of MOR, and can also revise the key definitions or write them for any other program. This is only for the function (F) and arrow keys plus SETUP, HELP, HOME, ERASE and TAB. It includes three meanings for each key: For example, the left arrow by itself moves the cursor one space to the left, the shifted left arrow moves the cursor one word to the left, and holding down the CTRL key and tapping the left arrow moves it to the line's left side. KEY.COM only loads a key definition file into CBIOS; if the file were named NEWWORD.KEY, you would enter A>KEY NEWWORD at the CP/M prompt. A NEWWORD.KEY program looks like this shortened sample as a non-document with NewWord or WordStar.

```
#_FST ; FAST, See KEY.COM NOTES
UP ^E ; The up-arrow key is UP
~UP ^Z ; ~UP means shifted up arrow
^UP ^R ; ^UP means CTRL up arrow
; DOWN and RIGHT omitted
LEFT ^S ; Type the control character
~LEFT ^A ; (the caret (^)) as a shifted
^LEFT ^ ; 6 key.
F1 ^R ; Up one screenful
~F1 ^C ; Down one screenful
; You can leave out ^F1
```

Gordon Northrup, MD, is a child psychiatrist and is editor of a professional journal, Residential Treatment for Children and Youth. He would like to hear from potential authors of papers on computers in such programs. Write to him directly at RR 1, Box 698, Lee, MA, 01238.

```
F2 AUXDICT.TXT ; Enter up to 127 Characters
~F2 G Northrup_RTN RR1, BOX 698_RTN Lee, MA. 01238
; _RTN is for RETURN, _ESC is ESCAPE, _TAB is TAB
; Comments come after a semicolon anywhere on the line.
; an empty line is OK
~TAB ^OI
^TAB ^ON ; Et cetera
HELP ^J ; Et cetera
TERM _ESC[ ; Set numeric keypad applications
CHAIN NW.COM ; Go to NewWord
; End with a RETURN
```

This program works, though DOWN and RIGHT, etc. aren't defined. Morrow sent out the full NW.KEY and SuperCalc key files with KEY.COM, but they are not included in the ROM 3.1 kit. NW.KEY (but not SC.KEY) is described in KEY.COM NOTES; the SuperCalc key is described in the READ ME FIRST! documentation. The files are easily written and defined with the directions in KEY.COM NOTES, but there may be a problem in making them work. KEY.COM was an advance. It required user memory space (RAM) and was fine for Co-Pilot menus and when used with the Morrow key definition files. The key definition file is limited to 1000 characters (bytes) which doesn't leave much space for addresses, etc. My information sources give differing figures for the number of characters per definition: READ ME FIRST! says up to 127, while KEY.COM NOTES says up to 240 ... but the point is almost academic, the space is so small. Also, the program a friend and I wrote often didn't work. I didn't find the cause, so it could have been our faults, but some of the problems (like the address crawling across the screen at two characters per second) were probably due to an _FST problem, described in KEY.COM NOTES. Incidentally, KEY.COM works only for the MDT-60 and MT-70.

In contrast, SMARTKEY defines any key, is unlimited in size, allows key definitions while running another program and runs trouble free. Presumably, that's why Morrow bundled SMARTKEY; most of us would just want the KEY.COM definitions out of the way. This is done with a "null definition file" that undoes the default KEY.COM definitions. It contains only two RETURNS, unless you want to set the numeric keypad application mode to get 42 more keys to define (see MDT-60 Terminal Manual, pages 3-5), as in the sample program above. Create it on one line in your word processor in a non-document file called NULL.KEY. Type "TERM_ESC[" followed by two RETURNS, then save it. After you boot, run A)KEY NULL, then SMARTKEY and your SMARTKEY definitions and your program. You'll be delighted by the way your cursor scoots along.

The Basic Approach

Programming with Basic

by Jesse Adams

People often ask the question with a sneer: "Why use BASIC? It's so primitive!" Maybe that's why so many of us do *and it works!*

After mastering the simple things like strings, number variables, formatting and other necessities, much more within Basic's capabilities exists that is easily understood ... unlike Pascal, Lisp, C, Fortran and, worst of all, Assembler. These languages all have their uses and in many ways are superior to garden-variety Basic, but the big advantage to the home computer user is that, with only a little familiarity, Basic says what it means and the self-taught user can make it up and debug it. Basic variants are usually transportable, meaning you can take a Basic program, whether in MBasic, GW-Basic, or some other variety (even the seldom used BAZIC which was bundled with Morrow computers) and with minor changes run it with whatever Basic interpreter you happen to have. Try running UCSD Pascal with a Turbo compiler ... it's not quite so easy.

Another complaint is that Basic is slow. What is "slow?" Does it mean having to wait a few seconds while something happens inside the interpreter? What it comes down to is that Basic suits most "appliance" operators well. You don't have to be a professional programmer to use it.

Basic Differences

Like the CP/M manual, the MBasic manual won't enlighten you without knowing ahead of time what you want to find out. However, the *MBASIC Handbook* from Osborne/McGraw-Hill

Jesse Adams, retired Navy surgeon, uses an MD3 and an Otrona. WordStar is his most used program.

is indispensable. Although I have not made all programs in it work, it has helped me often and offer many helpful examples. To translate Basic designed for other machines or systems, David Lien's *The BASIC Handbook*, from CompuSoft Publishing, is another invaluable text. Trying to make an elaborate, PC-Basic program work on an MBasic interpreter will take some trial and error, but PC and GW-Basic are quite similar to MBasic, all three having emanated from Microsoft. The PC and GW editions just have more extras to offer. A minor difference, among others, is that in Morrow MBasic PRINT CHR\$(26) is used to clear the screen and home the cursor, but GW-Basic, PC and some others use CLS.

If you want to give a particularly juicy MBasic program to someone with a non-Morrow computer, a preferred way exists. You'll have better luck if you force the save in ASCII. The simple and customary SAVE command will commit the program to disk:

```
SAVE "FILENAME
```

... saves the program in a compressed binary format.

```
SAVE "FILENAME",A
```

... saves the basic program in ASCII text. This time you need the second set of quotes and the comma A forces the save in ASCII. This takes more memory and it may run imperceptibly slower -- but the Basic interpreter still runs and edits. Conversion to PC or MS-DOS is simple and it's also possible to work on an ASCII program with a word processor.

Before addressing word processing,

you should know something about GW-Basic and PC-Basic. Much of PC-DOS is inside the machine in ROMBIOS and PC-Basic is ROMed; that is, the simpler part of it is inside the machine in an IBM memory chip and so the clones have to take some other route. Advanced Basic, or BASICA, also IBM proprietary, is on disk but must use what's in ROM and the combination makes possible some elaborate effects: music, graphics, color and other extras. IBM clones and others using generic MS-DOS rather than PC-DOS have to do it all on disk and GW-Basic is Microsoft's alternative for the unwashed.

Basic Genealogy

GW-Basic has common parentage with PC-Basic and MBasic (or Basic-80), but differences between CP/M and PC/MS-DOS become troublesome. CP/M ASCII equivalent numbers go from 0 to 127, but PC/MS-DOS ASCII uses numbers from 0 to 255, the additional range being used for graphics and exotic functions. If you convert a CP/M Basic program via a conversion utility to PC/MS-DOS you'll find that "=" has become "XOR," which is acceptable for assembly language but doesn't mean anything to GW-Basic. There are other switches like that and the way to get around them is to save the CP/M Basic in ASCII before attempting a translation. *

Going the other way, from PC/GW-Basic to CP/M Basic, is harder. They can be saved in ASCII too, but you frequently find those programs "condensed," blank spaces and REMs removed, lines combined, and with instructions like "LOCATE," "BEEP" and "COLOR" embedded. You

can edit those out for CP/M and substitute PRINT TAB(nn) and PRINT CHR\$(7), but for "COLOR," you're out of luck. The only way to break up the lines is to use the insert command and a line feed, tabbing to make it look acceptable. If you're clever and determined, you can probably make it work on CP/M and the MBASIC interpreter will try its best. There are good, public domain PC/GW-Basic programs that are not available for CP/M and should be.

Word Processing Tricks

You can use WordStar or another word processor to write Basic programs *if* you use a non-document mode (which you call with N in WordStar). The D command in WordStar or NewWord lets you write and everything looks fine until the Basic interpreter tries to read it. The non-printing characters are either regular or control characters (e.g., ^S), except that they have the "high order bit" set, used in formatting the document. They are interpreted by Basic as direct commands and programs containing them won't load or run.

Why would we want to use a word processor for programs when the Basic editor does so much? For example, let's say you decided to use "structured" code for quick reference and ease of debugging. Your line numbers from 10 to 500 are used to initialize the program, print opening statements and a menu and call the subroutines. Lines 1000 to 1990 do something special and lines 2000-2990 do something else. Now, say you've discovered a bug or you want to change something in the middle of the program and still preserve the "structured" arrangement. You don't want to use RENUM in the Basic editor since that would renumber the program from the designated line right on through to the end and destroy your careful organization. You can't SAVE just part of a Basic program.

With your MBasic program saved in ASCII, return to the SYSTEM, call the .BAS program up on WordStar or NewWord using the non-document mode. Try to keep from editing out those seemingly useless spaces that have appeared between lines of code.

Mark the "bugged" part of this example contained in lines 1000 to 1590 with ^KB and ^KK at the beginning and end of the block. Do a ^KW to write that block to a new file which we'll call MIDPROG.BAS. If you're bold, you can do a ^KY to wipe the block from the WordStar screen. You still have the original code, but now part of it is in another file, MIDPROG.BAS. Do a ^KQ or ^KD to get out of WordStar, enter MBASIC "MIDPROG, and there it is. Now you can use the MBasic editor to do whatever you want in that smaller portion, RENUM 1000,10, and it's fixed. SAVE "MIDPROG.BAS",A and go SYSTEM to return to the CP/M A> prompt. Go back to the non-document mode in WordStar and put your cursor wherever line 1000 should be. Press ^KR and identify MIDPROG.BAS as the file to be read in and you are ready to discontinue WordStar with ^KD and X with minimal pain. Naturally, you made a backup on another disk in case the process failed.

Making Backups

Suppose you're now using random access files to store data you want to manipulate with your Basic program. Again, you want to back it up just to be safe. However, when you used PIP or DISK or NSWP to make the random file copy, all you got on the new disk was a phantom. The directory says it's there, but the data file itself contains 0 K. When you try again, the same result occurs. What happened? Sequential files are stored in ASCII and can be copied like any other text file, but a random access file is stored in packed binary, not ASCII, and your

attempts to copy produced an unidentical copy.

In order to create a random file copy, use BACKUP, which copies track by track and byte by byte onto the new disk. You can use PIP or other utilities for the other data.

Here's something else that occurred when I tried to store several records in a random file. Textbook examples all use a number (actually, part number, part ZIP code) as a pointer to specified stored data in a record. It has to be a whole number and preferably a single byte. I wanted to use names as identifiers, each for a unique record within the same random access file, but could not. Sequential files are generally no problem; but not for random files. Some programs use a "B-Tree" or "Hash-Code" sort to do it, but are procedures that use up large amounts of code.

After a lot of cogitation there seemed to be a way. I called it an "algorithm" to make it sound important. It's cumbersome to type in, but MBasic never stumbled so long as I got my parentheses balanced. (See Figure 1).

```
NAM=INT(ASC(MID$(NAM$,1,1))*
ASC(MID$(NAM$,2,1))*
ASC(MID$(NAM$,3,1))*
ASC(MID$(NAM$,4,1)))/
(ASC(MID$(NAM$,5,1))*
ASC(MID$(NAM$,6,1)))
```

Figure 1 - The algorithm expressed as one long Basic command.

NAM\$ is the name I wanted to use and NAM is an integer to use as the record number. Basically, I encrypted a name as a number. The number is then used as a pointer to a record in a random access file, so that the record can easily be retrieved.

What the convoluted instruction does is to take the product of the ASCII values of the first four letters, divide by the ASCII product of the fifth and sixth, and take the integer of

the dividend as the identifier for that record.

NAM is a number while NAM\$ is an alphanumeric string. I had to use six characters for NAM\$ including spaces to avoid duplications; one 3650 is undistinguishable from another 3650.

Try: SMITH_ (don't forget the space to make it six characters). We get 11417.255 and the integer is 11417, Smith's data record number. That's a big number for the computer to handle without exponentiation, so you might want to divide the whole thing by an odd number like 333 before you take the integer. Balance the parentheses to avoid a syntax error. If that's too complicated to type in, use a simpler algorithm. (See figure 2).

```
NAM=INT(((LEN(NAM$))*  
ASC(MID$(NAM$,4,1)))/3)
```

Figure 2 - A simpler algorithm.

SMITH would now be: 46.6666 with the integer dividend 15. Of course, any other five letter name in which the fourth letter is "T" would give the same integer result, but everything can't be simple. Use an odd number for the divisor. 35, 36 or 37 as the number to be divided would result in 3 in each case with an integer dividend of 10, whereas 3 as the divisor would produce 11, 12 and 12 ... better, but not entirely unambiguous. You could use: SMITH, J. for a different LEN. If you keep in mind that either the number of characters or the fourth (or whatever) character has to be different for each NAM\$, you will stay out of trouble. If you use uppercase when you enter NAM\$, you have to use uppercase to call the record. An uppercase "S" has a value of 83, but a lowercase "s" has a value of 115 and the integer dividend wouldn't be the same.

What you can do is to get out of the main program with SAVE and

then NEW and write a program called DRILL.BAS or some other name that allows you to try out various things and when the results are acceptable, LLIST it, go back to the main program and type it in.

Building the Program

I wanted files of different secondary tide and current stations and the likelihood of getting the same combination twice is small for a short list. Anyway, it allowed me to construct a cheap and dirty file database in MBasic to compute tides and currents at secondary stations and to manipulate the data to get tide height or current at any time in the reporting period. The program runs to nine pages for currents and six pages for tides and the only machine delay is a short one in the printer when the disk drive has to go to a file for data.

Using "structured" code - and no flowchart except in your head - each segment can be worked out in modules as if building a radio set -- build the power supply first, the audio section next, then the IF section and finally the RF section; testing and revising each until it works on the back of the previous module until finally you have a radio. You don't have to make "spaghetti" code out of it. Basic is logical and usable.

Programs always require debugging. An invaluable aid, besides your crude LLIST which types right across the perforations of fanfold paper, is another commonly available, public domain program, MBXREF.BAS. This gives a cross-reference to every line number and variable that prints out as a titled, dated and paginated listing. It only works with ASCII saved programs.

Basic can be used in programs to create .COM files, but then you need an expensive MBASIC compiler. I hope this leaves you encouraged to continue with the "BASICS." ■

Reference List

Use this reference list to locate books on programming.

- Brown, P. (1982). *Pascal from basic* Addison-Wesley Microbooks.
- Cain, R. (1980, September). A runtime library for the small C compiler. *Dr. Dobb's Journal*, pp.4-15.
- Cain, R. (1980, May). A small-C compiler for the 8080's. *Dr. Dobb's Journal*, p.5.
- Clark, D. (1984, May). Two more versions of C for CP/M. *Byte*, p.246.
- Duntemann, J. *Complete turbo pascal* Scott, Foresman and Company.
- Fernandez, J. N., & Ashley, R. (1981) *Introduction to 8080/8085 assembly language programming - a self-teaching guide* John Wiley & Sons.
- Grogono, P. (1980). *Programming in pascal* (rev. ed.) Addison-Wesley.
- Harbison, S., & Steele, G. (1984) *A C reference manual* Prentice Hall Software Series.
- Harris, W., & VanderWood, J. (1983-1984). *Pilot user's guide* Morrow, Inc. (Now available from MOR).
- Hendrix, J. (1984). *The small-C handbook*. Reston Publishing.
- Jensen, K., & Wirth, N. (1974) *Pascal user manual and report* (2nd ed.). New York: Springer-Verlag.
- Jensen, K., & Wirth, N. (1974) *Pascal user manual and report* (3rd ed., rev. for the ISO Pascal Standard, prepared by Mickel, A. B., & Miner, J. F.). New York: Springer-Verlag.
- Joyce, J. (1983, August). A C language primer. *Byte*, p.64.
- Joyce, J. (1983, September). A C language primer. *Byte*, p.289.
- Kem, C. (1979, December). A user's look at tiny C. *Byte*, p.96.
- Kern, C. (1983, August). Five C compilers for CP/M-80. *Byte*, p.110.
- Kern, C. (1981, May). Printf for the C function library. *Byte*, p.430.
- Kemighan, B., & Ritchie, D. (1978). *The C programming language* Englewood Cliffs, N.J.: Prentice-Hall.
- Ledgard, H. F., Nagin, P. A., & Hueras, J. F. (1979). *Pascal with style: programming proverbs* Hayden Books.
- Purdum, Jack Dr. (1982). *C programming library* Que Corporation.
- Purdum, Jack Dr. (1983). *C programming guide* Que Corporation.
- Purdum, Jack Dr. (1983). *C user's guide* Que Corporation.
- Starkweather, John A. *A user's guide to pilot* Englewood Cliffs, N.J.: Prentice-Hall, Inc.
- Tiberghien, J. (1981). *The pascal handbook* Sybex.
- Wirth, N. (1976). *Algorithms + data structures = programs* Englewood Cliffs, N.J.: Prentice-Hall, Inc.

Reviews

SOFTWARE REVIEW	
Product:	Nevada Pilot
VERSION:	Version 6.0
MANUFACTURER:	Ellis Computing, Inc. 5655 Riggins Court Suite 10 Reno, Nevada 89502 (702) 827-3030
TESTED ON:	MD2/MD3
PRICE:	\$39.95
MEMORY REQUIRED:	104K in addition to files created

By Jesse Adams

I dabbled with the PILOT program bundled with my MD2 and found it interesting, but then forgot about it until I saw the AUTOFORM.PIL program article in the April, 1985 MOR issue. It worked well, except for some reason the M: command in the program didn't seem to operate and the program kept repeating until I rebooted. Eventually, I worked around it with another command. The language looked simple enough, so I thought I could use it without much difficulty. Although I suspected there was more to PILOT than the Morrow program manual indicated I found little evidence of activity until the two MOR articles came along.

Later I was attracted to an ad in *Byte* by Ellis Computing, Inc. for NEVADA PILOT and other products. The price was right at less than \$30 and has since gone up to \$39.95, which is still affordable. I've paid that much for things that didn't work ... not happily, mind you, but often enough so that this didn't seem to be an unacceptable risk.

Not expecting too much for the price, considering what "big" software houses demand for products that don't always meet advertising promises, I ordered an MS-DOS version of the Ellis PILOT (UTAH PILOT) to run on my Otrona. Shipping turnaround was fast. The disk includes a configuration program which alters PILOT to fit a wide variety of terminals as well as several sample programs that include demonstrations of various PILOT features.

The package worked like a brand new love affair, and my

Jesse Adams, retired Navy surgeon, uses an MD3 and an Otrona. WordStar is his most used program.

order went in promptly for NEVADA PILOT to run on my Morrow. Not only did it work, the Ellis version towers above the Morrow version. Unlike the one with the trusty Morrow, Ellis PILOT (both the MS-DOS and CP/M versions) includes a built-in editor, which means you don't have to bother with getting in and out of your word processor program as you write and then debug and run PILOT programs. To me, that alone is worth the price.

You say you want more for your money? Well, it's copyable and rather than threatening you for giving away so much as a comma, Ellis simply requests "Please do not distribute the full author version of the CONFIG program." Now there's a courteous way of putting it. But that's not all...Do you get the implication above? You can install the full PILOT-editor version for your own unlimited use as author, or install only the student version which creates programs with commands that cannot be altered. The full screen text editor isn't included and the instructional program won't permit the addition of student graffiti. Sure, words can be altered with any text editor, but you can't compile it to run with any but the original commands. The design impresses me greatly.

As I suspected, the PILOT language is more extensive than the Morrow manual indicates, but not as complicated as many other languages and it seems ideal for getting the computer novice used to programming before jumping into Basic, Pascal or the outer darkness of assembly language. It seems much less difficult to use than NewWord or WordStar, and for creating a PILOT program NEVADA PILOT's editor is entirely adequate.

Although PILOT does simple math (addition, subtraction, multiplication, division) it works only in integers. As a language designed for creating instructional programs, it serves the purpose admirably and if it's kept at an elementary school math level, who can fault that? Having done finger exercises with the relatively limited number of commands in PILOT, simplifies the progress by a novice to Basic or Pascal.

If you acquire NEVADA PILOT, I suggest you also get *A User's Guide to Pilot* by John A. Starkweather, the originator of the PILOT language, published by Prentice-Hall, Inc. (\$12.95) for a more in-depth discussion and tutorial.

For two-computer families, Ellis produces the MS-DOS UTAH series, (Pilot, Basic, Pascal, Edit, Cobol and Fortran) all at the same modest price.

The Ellis series of NEVADA and UTAH programs includes the bargains we non-commercial, non-tax-deductible computer users have been looking for and hardly expecting with the software shake-out looming on the horizon. ■

SOFTWARE REVIEW

PRODUCT: **Mix C Compiler**

MANUFACTURER Mix Software, Inc.
2116 East Arapaho
Suite 363
Richardson, Texas 75081
(800) 523-9520
(800) 622-4070 (within Texas)
(214) 783-6001

TESTED ON: MD3

PRICE: \$39.95
\$29.95 for programmable editor

By Lee McKusick

The Mix C compiler enables Morrow Micro Decision owners to write programs in the popular, elegant and transportable C computer language. I bought the Mix C compiler because I'm taking a C programming course, and I can confirm that Mix C is an excellent gateway for learning C. Overall, Mix C has a very good manual and tutorial, useful error messages, and a compiler that performs as promised.

The 120 page tutorial describes the C language with 55 program listings to enter, compile and run. Each program runs completely and in all it took me about 40 hours to finish. The tutorial provides about a 50 percent understanding of the C language, since it is passive reading and programming by imitation. It made me feel ready to go ahead, get lost, play around and create a program independently. A model of good technical writing, the tutorial systematically displays C programming ideas, one idea at a time, nicely salted with the repetition of difficult ideas.

The Mix Manual is splendidly written. Although never named, it seems that only one author wrote it. The manual explains and illustrates with examples listing everything in the C language.

The Mix C compiler package is sure to become a

Lee McKusick uses a Morrow MD3-P to enhance his computer science skills. He is developing a career in technical writing, typesetting and programming in Pascal, dBase and C.

standard text and software package for hundreds of college level "Introduction to C Programming" courses.

Mix C compared with Turbo Pascal 3.0 and the Turbo Tutor Program

In terms of quality and value, I consider Mix C and Turbo both excellent. Still, points of comparison demand attention. Using NewWord as an editor and a submit file to automate compiling, Mix C compiles, links and runs a 30 to 150 line program in 1.5 minutes. Turbo Pascal is considerably faster.

When Mix C compiles, the source code displays on the screen. When Mix encounters an error, an error line with a pointer shows its location, and the errors are all written out to a file. This is more informative error reporting than Berkeley Unix and many older compilers. Still, Mix C is not as integrated as Turbo Pascal. Mix combined with the Mix editor may prove to be somewhat better.

On the BAMDUA computer bulletin board (call (415) 644-3882, 300/1200/2400), Mix C user Mike Allen reports that some early judgments label Mix C as a very correct compiler. The judgments are that Mix C also generates rather large code for simple programs. Finally, Mix C compiles the SIEVE speed test but executes the program slowly.

I can confirm these observations, and add that because I am a learner, speed and code size matter little to me. I compiled the SIEVE test and found it slow. I set SIEVE aside until I understood enough C to judge the program and tinker with it.

If you want to develop a program, or get control over your Morrow, which language is best: Turbo Pascal or Mix C? I say, first look for friends and books. Both Turbo and Mix are powerful tools, so let these resources guide your first choice. Look for enthusiastic friends and published programs that appeal to you.

Another angle on choosing a language is this rule of thumb: Copy published source code programs. Copying can cut your first prototype time to one third of what it would be if you wrote everything from scratch.

The Mix C Package

The Mix C Compiler comes with a manual that includes the sections "Getting Started," "Tutorial," "Reference Manual," "Functions" and "Tools," totalling almost 450 pages of information. The two Mix C disks come in an Morrow MD-2 format (ssdd 186K per disk), totalling 330 K of software and 21 files.

Also included is the C compiler and runtime file claimed to compile all C instructions as defined in The C Programming Language by Kernighan and Ritchie, a second small C compiler and runtime file for programs that don't

use long or float data types, a menu driven linker that runs from the command line and that gives you more control when making .com files, files of source code for most of the standard functions, a 14K file for functions the source does not supply, and three utilities to shrink object code, speed up object code and convert object code to ASCII format.

The software license for Mix C does not require the purchaser to sign or mail a form. The main point of the license terms is: The software is not to be used on more than one computer. The manual indicates that compiled programs developed with Mix C may be sold without any royalty payment to Mix.

Using Mix C with NewWord as a Programming Editor

NewWord and probably Wordstar adapt well for preparing C source programs. Using an MD-3 system with 380K disk drives, this is an outline of some adjustments I made while getting my system to productively run Mix C.

You must use the non-document mode when preparing C source programs. Document mode editing sets the eighth bit in the last character of each word, which confuses the Mix C compiler. Mix C programs that have the eighth bit set can be cleaned up using the CP/M PIP utility with the [Z] zero parity bit option. If PIP.COM is on A: and BADBITS.C is on B:, then you can filter BADBITS.C into a fresh file labeled CLEANBIT.C using a special instruction line:

```
A:PIP B:CLEANBIT.C = B:BADBITS.C[Z]
```

To modify NewWord tab stops for Mix C using NewWord 2.14 in the non-document mode expands tab stops to four spaces when the files are edited with the ".C" file type. The Mix C compiler expands tab stops to eight spaces. Programs are more consistent if NewWord is modified to agree with Mix C.

To modify NewWord, you will need to run NWINSTAL.COM and alter the obscure EDCOND area. EDCOND means "edit conditions." Find the User Area Listing, which is an Appendix on page C-23 of the NewWord Encyclopedia (not the Nuts and Bolts Tutorial).

Once you are running NWINSTAL.COM, the EDCOND code is selected from Patch Menu #4. The EDCOND code string before and after changing the default tab expansion from four to eight spaces, for Non-Document editing of files with a ".C" suffix, is easily recognized:

```
ASCII :                               C sp sp 3 end
Before: 50 41 53 01 FF 50 4C 49 03 FF 43 20 20 03 FF
After :  " " " " " " " " " " " " " " " 07 "
```

Check your CP/M manual for information about SUBMIT.COM. Submit is good for automating the tedious compile, link and run cycle. Your key definition program can also eliminate typing associated with writing C programs.

Using Submit Files with Mix C

The CP/M operating system on the Morrow computer includes SUBMIT.COM, a program that allows you to repeat lengthy Mix-C compile-and-run tasks. This utility saves typing efforts greatly. When developing programs using the Mix C compiler, you can set up a submit file and cause the computer to automatically compile your C source, link the intermediate code, and run your COM file.

Suppose you have the Mix C compiler on drive A (filename CC.COM), the Mix Linker on drive A (filename LINKER.COM), and your C source text on drive B. Let the C source text be called LEECRASH.C. You can compile, link and run LEECRASH manually with the keystrokes listed below:

```
A>CC B:LEECRASH
```

Wait a few minutes for it to compile. The compiler makes B:LEECRASH.MIX, and linker presumes extension is .MIX

```
A>LINKER B:LEECRASH
```

Wait 30 seconds for linker. Linker makes B:LEECRASH.COM

```
A>B:LEECRASH
```

Try out the compiled program and decide if you like how it runs.

```
A>NW LEECRASH.C
```

Restart NewWord with your source.

The same three minutes of work can be done automatically with the submit command:

```
A>SUBMIT CNRUN LEECRASH
```

Go get a cup of coffee, then come back and watch it run.

Submit requires you to make a command text file (called CNRUN in these examples) containing keystrokes to be submitted to the computer. I use the name CNRUN simply because it reminds me of "Compile aNd RUN." Make a file called CNRUN.SUB on a word processor. CNRUN then pops up in examples. Contents of the CNRUN text file include a simple list of commands:

A:CC B:\$1
A:LINKER B:\$1
B:
B:\$1
A:
A:NW \$1.C

A submit file allows parameters. "\$1" is a parameter. The word LEECRASH replaces \$1 when the command string SUBMIT CNRUN LEECRASH is executed.

Parameters permit one submit file to operate with different program titles. For instance, the same CNRUN file would compile and run NUTHRPRG with this command string:

SUBMIT CNRUN NUTHRPRG

Points to Help Avoid Submit Problems

Command text file: No blank lines are allowed in your command text file, and comments cannot be entered in the command text file. Make text straight ASCII text as in the NewWord nondocument mode.

Submit program: SUBMIT.COM must be on drive A. If the floppy disks are full, SUBMIT won't run and it issues no error message. SUBMIT writes a temporary file on the disk that continuously seizes the A prompt until all the commands in the text file are run.

SUBMIT provides a memory resident extension, XSUB. You can feed keystrokes to a program that expects keyboard responses within XSUB. I have used XSUB useful for automating instructions to the Mix C LINKER.

Your Digital Research CP/M 2.2 Operating System User Reference Manual for the Morrow MD-3P describes SUBMIT on pages 25-27. ■



The Z System



By Echelon, Inc.

You Never Had It So Powerful!

Ready to Run!

Now you can get a complete, ready-to-go replacement for CP/M and your MD3 ... the ZCPR3 kit.

The ZCPR3 kit runs all conventional CP/M programs, and includes:

- ZCPR3 for Morrow
- ZRDOS
- Z System User Guide
 - an 80 page text
- On-Line Help Disk

Extended command processing Searches other disks and user areas when typing CP/M commands with PATH function.

Library Files Group together commonly used utility programs into a one single entry file.

Three different menu systems Create custom menu-driven "front-ends" for your computer. No more "A" prompts!

All for only \$59.50 from MOR Order Product #801 on page 28 of this issue!

ZCPR3 greatly enhances your computing abilities on the Morrow Computer. Here's your chance to improve your computing and work.

If you're really serious...

Get the ZCPR3 Supplement package ... including four disks with the BIOS and Utilities source code.

All for only \$39.50 from MOR. Order Product #803 on page 28.

ZCPR3 - THE MANUAL is the definitive guide to ZCPR3, with 350 pages packed with useful information. Order product #804 on page 28 for only \$20.

Classified Ads

PLACING A CLASIFIED? Private party ads cost \$2 per line. Commercial ads cost \$6 per line. ALL ads that sell something for profit are considered commercial. Specifications: 50-character maximum per line, 10-line maximum per ad. Include a check or money order with your name, address and phone number (make sure we can distinguish your ad from your other information). Send a SASE if you want a copy of your ad.

MD3, ROM 2.5 MDT60 terminal, CP/M 2.2, NewWord, Personal Pearl, LogiCalc, Correct-It, BAZIC, Microsoft BASIC 80 and Turbo Pascal. All with manuals and distribution diskettes. Freight paid. Asking \$700. 406/251-3926.

MD2, FREEDOM 50 terminal, WordStar, Correct-It, LogiCalc, BAZIC, Personal Pearl, BASIC, Pilot. All original documentation and master disks in excellent condition. PLUS communications software and all back issues of *Morrow Owners' Review*. \$595.00 OBO. Linda Socol, 6645 Brewster Court, Cypress, CA 90630, 714/891-6126.

FOR SALE: MD2 64K. Two floppy drives. Excellent condition. All usual software, plus many more (Eliza, SmartKey, LISP etc.) Freedom 100 terminal. \$395 or best. (Also have a nonworking MD2 I'll throw in for nothing.) Call Michelle Bekey, 213/392-3009. West L.A.

MD16 w/MT70 for sale—standard packaging—excellent working condition. Purchased 9/85. \$1500. Will ship COD. 206/757-2711 or 206/755-9806. P.O. Box 100, Burlington, WA 98233. Also Okidata 92 and 84 printers \$500 for pair. Total package \$1950.

MD3 w/MDT50 terminal, software Excel. cond. \$849. Starwriter F10 40 serial printer w/tractor, \$795. For info call Gerry Shacter 213/567-8758.

PIVOT II, 640K, internal 300/1200 Hayes compatible modem, battery, charger, cordura case, telephone and printer cables, all manuals, PC-DOS, NewWord, RAMdisk driver, and P.D. including PibTerm. 100% working. \$1495.00. Bill Vogel, 1252 So. 115th Street, Seattle, WA 98168. 206/242-9761 eves.

We're not selling toothpaste or bubble gum or softdrinks, in which there's no difference in the basic product from one year to another except packaging. We're selling technology, and there's a difference. Look at Detroit. In the early days, the auto industry was run by engineers and technical types. It was only later, to squeeze every last dollar of profit, that they brought in the beancounters. And as soon as they did that, Japan came, and there wasn't anything left.

...From *Quotations From Chairman Morrow*, Copyright 1984, Morrow Designs, Inc.

PROBLEM WITH NewWord ON MD 5,11,16,34.

If you run CP/M 3.0, Revision 1.9 with ROM 1.9, at least two versions of NewWord, 1.19 and 2.14, will not write files properly. The only thing you get are the useless .SAS OR .SB\$ files. Solution: Upgrade to CP/M 3.0, Revision 2.0. Call the MOR office.

List of Advertisers

Advanced Concepts.....	4
Civil Computing.....	4
Jacobs Computer Services.....	inside back cover
James River Group.....	outside back cover
Lark Software.....	43
Spectre Technologies.....	inside front cover
Westwind Computers.....	32

COMPUTER FACTS = MORROW PROBLEMS SOLVED

That's right.

Jacobs Computer Services deals in Facts. Computer Facts. Whether you're seeking reliable advice or reliable repairs, we'll take your problem seriously. We don't pretend that computer repair is an art, to us it's a science. If your computer doesn't work, we'll find out why and get you working again. Fact. No nonsense. Over the telephone or on site we'll furnish support for you; connecting modems and terminals, installing software, solving your hardware problems and helping you to understand CP/M. Now you know to call Jacobs Computer Services when you need help.

That's a Fact.

JACOBS COMPUTER SERVICES CLASSIFIEDS

CLASSIFIEDS

FORSALE: New and used MORROW equipment.

\$100 mother boards with serial & parallel I/O, clock: New Morrow WB14 (14 slot) only \$149.

Liberty-Freedom flybacks—new \$60.

\$100 memory: TERRIFIC DEAL!! New Morrow 256k RAM boards (not for MicroDecision) only \$195. Ram Disk software ordered with board(s) just \$39.

New Morrow MDT 20/Lear Siegler ADM 20 Terminals just \$269.

Good used MDT 60 terminals \$219.

DECISION 1 OWNERS: Speed up CP/M! Micronix too slow? We have RAM DISK software for single user CP/M. Hardware remains able to run Micronix at any time. (RAM DISK and Micronix won't run at the same time.) \$200. Demo \$3.00

Terms: California residents add applicable sales tax. All items plus shipping. Cash or COD (no Visa or MC). Guarantees: 90 days new, 30 days used.

COMPUTER SERVICES CLASSIFIEDS

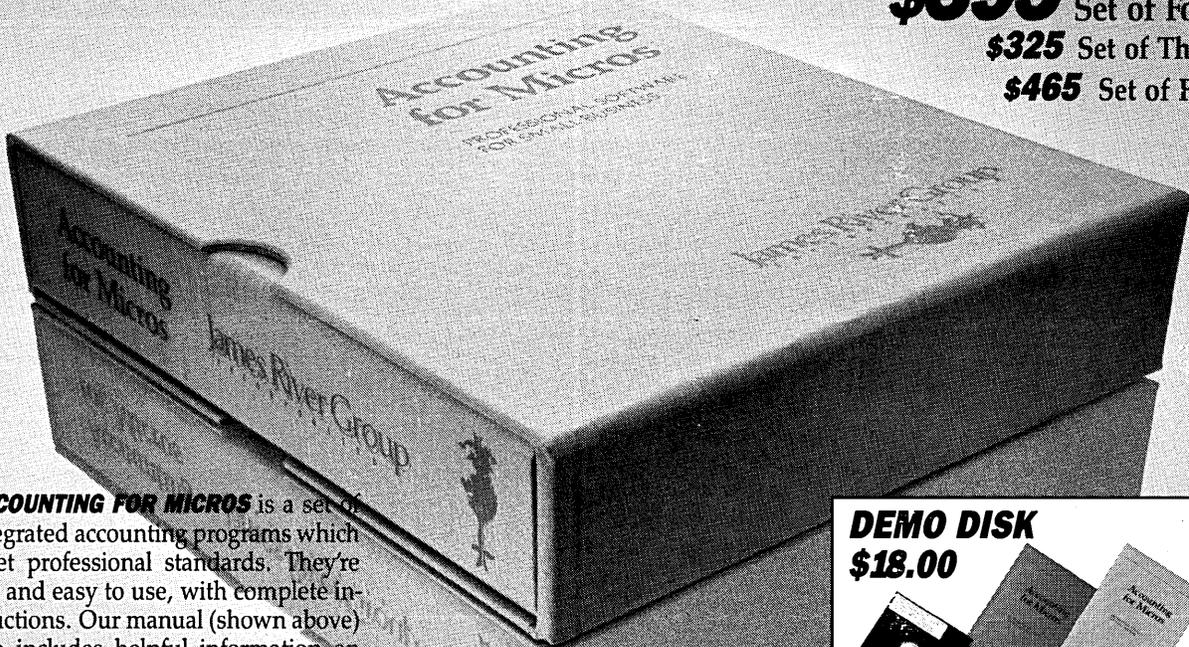
Morrow • Lear Siegler
S100 • Liberty Freedom
Micro Decision • Decision 1
ADM20 • MDT50 • MDT60
MDT70 • Freedom 50
Pivot • Freedom 100 • Drives
Hardware • Software
Sales • Support
Service

JACOBS
computer services
P.O. Box 6092 • Albany • CA 94706

Call (415) 724-2446
anytime, day or night!

ACCOUNTING FOR MICROS

\$395 Set of Four
\$325 Set of Three
\$465 Set of Five



ACCOUNTING FOR MICROS is a set of integrated accounting programs which meet professional standards. They're fast and easy to use, with complete instructions. Our manual (shown above) also includes helpful information on bookkeeping and computers.

GENERAL LEDGER \$125
 Allows up to 1,000 accounts & 1,000 transactions/month. Retains mo/end balances for Last year, This Year and Forecast. Includes Cash Disbursements, Cash Receipts and General Journals. Reports include Balance Sheet, Income Statement, Annual Summaries and Journal Reports.

ACCOUNTS RECEIVABLE \$125
 Allows up to 2,500 customers and 1,000 invoices per month. Invoicing can access Inventory Module. Keeps customer names and addresses. Invoice prints on plain paper or any pre-printed form. Statements can be printed at any time.

INVENTORY \$125
 Allows up to 4,000 parts. Keeps 3 month history of unit sales as well as year to date. With AR, can be used as point of sale system (prints invoices, handles cash). Reports include Inventory Value and Stock Report, Internal and Customer Price List.

ACCOUNTS PAYABLE \$125
 Allows up to 500 vendors and 600 invoices/mo. Records invoices and handwritten checks. Prints computer checks on any pre-printed form. Keeps vendor names and addresses.

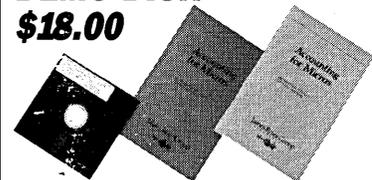
PAYROLL \$125
 Will handle up to 100 employees with six deductions per employee. Deductions may be determined as fixed dollar amounts or percentages, or referred to a table for automatic look-up. Tax tables are easily entered, or purchased separately.

SET OF FIVE \$465
SET OF FOUR \$395
SET OF THREE \$325

RUN ON MOST CPM AND MSDOS

Apple CPM	IBM PC, XT, PC jr, AT	Sanyo (all)
Columbia	Kaypro (all)	Tandy (all)
Compaq	Morrow (all)	TeleVideo
Corona	Osborne (all)	Zenith 100 & 150
Eagle (all)	Panasonic	8" CPM
Epson QX-10	Radio Shack CPM	Other compatibles

DEMO DISK
\$18.00



Try all 5 programs above (GL, AR, AP, IN, i'R). Order our DEMO DISK for \$18.00 (includes shipping). Condensed versions of the programs give you the "feel" of data entry and access. Includes sample reports and instructions. Specify machine.

TRASHMAN (NEW) \$145
 The "Catch-All" program. Files any type of information for quick access. Name or subject oriented with 15 lines of notes per name. Use TMAN as a mailing list, filing system, notebook, etc. Can be used alone or with data from our other programs.

Try **TRASHMAN (only) DEMO \$16**

HOW TO ORDER: Please specify machine and disk format. You can pay by check, by VISA or MasterCard (we need your expiration date and card number), or by UPS COD (add \$2.50 COD charge). Our price includes shipping. Minnesota residents, add 6% sales tax. We ship most orders the same day.

or **ORDER BY PHONE: 612-339-2521**

James River Group
 INCORPORATED



1-800-654-6903

125 North First Street
 Minneapolis, MN 55401