$5.95 U.S.

two/sixteen magazine™

Volume 1, Number 3 September-December 1982

The journal for business, professional, and scientific members of the TRS-80 community

"He that publishes a book runs a very great hazzard— since nothing can be more impossible than to compose one that may secure the approbation of every reader." Cervantes DON QUIXOTE

CONTENTS

Editorial 1
Reprint Service 3
Classified Advertising 3
Letters to the Editor 4
Software Revue: PROOFREADER 19
Special User Report: SemiDisk Emulator 24
Novice Column: How to Tell When it's Time to Split 26
MicroShell: UNIX Features for CP/M Users 30
PRO-80 BASIC Correction 32
The BASIC Debate Rages On 33
An RSBASIC Alternative to DOSFIX14 36
Software Revue: GRAMMATIK 37
"Snapp to It", SNAPP-WARES SNAPP III 42
Short Form Directory for the Model II 45
Kill the KillPrompt with KILLABS 47
Reader Survey, September-December 48

two/sixteen magazine

Editor/Publisher Richard H. Young
Associate Editor Barbara S. Albert
Art Director Timothy P. Schleif
Administrative Assistant Barbara M. Powell
Contributing Editors Arnold Fischthal
John Thornton
Roger Conant

Typesetting by Centennial Graphics, Ephrata, PA.

Printed by Science Press, Ephrata, PA.

two/sixteen is published bi-monthly by

two/sixteen

131 East Orange Street, Lancaster, PA 17602

Telephones

Editorial Staff (717) 397-3365
Classified Advertising (717) 397-3789
Display Advertising (717) 397-3364
Subscriptions (717) 397-3364

Annual Subscriptions:

Third Class $30 in U.S.
First Class $36 in U.S.
VISA and MasterCard Accepted

Foreign Subscriptions:

Canada $36 U.S.
Other Foreign $47 U.S.
Air Mail Foreign $67 U.S.
Drawn on U.S. Banks

Send all subscription correspondence, including change of address (with label) to:
two/sixteen
Box 1216, Lancaster, PA 17604

two/sixteen welcomes letters to the editor. All letters sent to the magazine become the property of two/sixteen magazine and are subject to publication unless otherwise requested by the author.

EDITORIAL

INPUT

WHO DOES ALL THIS STUFF?

two/sixteen has added another member to the full-time staff. Barbara Powell began on October 18 as Administrative Assistant (later on she will be doing some in-house software reviews). Along with Barbara Albert, who does nearly all the editing, Tim Schleif, who does all the art, graphics, and layout, and yours truly, who does mostly what he pleases for 80 hours/week and up, this makes a full time staff of four. There are also ten or twenty selfless readers who have volunteered their time to produce articles and reviews which will be of great value to many of you. In addition, there are the dozens of you who have written so many useful and insightful letters that we are unable to find space to print them all.

WHERE ARE WE GOING, AND HOW? (IF SO, WHY NOT?)

Our subscription list now numbers 1600, with a slow but steady growth of about 250 per month. We need 500 per month or more to reach the point where we can go to monthly publication (probably 5000–6000 total circulation). Several forthcoming activities may help us get there:

1. Tandy has promised that Jon Shirley will mention us in his column in their next monthly ***TRS-80 Microcomputer News.*** This will expose us to a large percentage of Model II and 16 owners.

2. We are running a full-page four-color ad featuring a tasteful watercolor by Tim Schleif in the ***80 Micro*** special anniversary issue. Since this issue will have a separate Model II section, it may produce good results.

3. We are obtaining mailing lists of Model II owners from software vendors and list brokers. Experience has shown that these can be effective.

Naturally, we're always interested in suggestions from readers on how to improve circulation.

MAILING LIST SALES.

In addition, to promote further income to be used to expand and improve ***two/sixteen,*** we are offering our mailing li to vendors who sell products which should enhance Model II and 16 computer use (but not to any non-computer mailers). If you want to avoid such mailings, be sure to mark and return the enclosed survey form (it's not necessary to mark it again if you already have told us to omit you, but do let us know if you've changed your mind).

SOFTWARE SALES WITH GENUINE SUPPORT!

Due to overwhelming subscriber approval of the idea, we are beginning to offer software at competitive prices to subscribers. There should be a full page ad in here somewhere. The main idea is that we either use the software extensively ourselves, or we have reviewed it in a thorough manner and can recommend it for at least some worthwhile purpose. Also, our first-hand experience with it helps us support this software on Model II and 16 micros. We often support this software anyway, so we might as well sell it. If you're a subscriber, we'll ship on receipt of your check (without that usual 14-day wait).

We also accept VISA and MasterCard by phone or mail. We can't afford to carry a large inventory, so we've arranged with several wholesalers to "drop-

CHANGE IN *two/sixteen* COVER DATE

When *two/sixteen* began its publishing life, external circumstances forced the initial issue to appear about five weeks behind its original schedule. Other start-up problems (and general disorganization on the publisher's part) have allowed about three more weeks lag to creep in. We could go on pretending that we're going to catch up, but it just won't happen, at least not very soon. So, we're EXTENDING all subscriptions by TWO MONTHS, and combining the Sept-Oct issue with the Nov-Dec issue to produce a Sept-Dec issue (Volume 1, Number 3). You'll still receive the same number of issues in your subscription (six—more if we should be able to go monthly by then). It's just that the issue you will receive in early January will be called Jan-Feb instead of (embarassingly) Nov-Dec.

If you are the typical subscriber, whose subscription began with the May-June, 1982, issue, then your subscription originally expired with the Mar-Apr, 1983 (or April, if monthly) issue. Now your subscription will expire with the May-June, 1983 (or June, if monthly) issue.

We have added another full-time staff member (for a total of four) and leased more space. Further slippages are not expected.

ship" within 24 hours directly to you in case we are out of stock. In addition to the products listed, we can promptly obtain (or have shipped directly to you) almost any software item available. If you have a special software need, I will personally undertake to help you find the right product, whether we sell it or not. We can't be quite as price-competitive as the lowest-priced mail order vendor, but then he may not be able to support the software on a II or 16. Give us your software business. We need your trade. You'll get our support in any case.

LOCATING THAT HARD-TO-FIND SOFTWARE.

Speaking of finding the right software, John Spady, a subscriber in Portland, Oregon, told us about a specialized software search firm. For an annual fee, several computer searches may be conducted for various software needs. Based on John's preliminary results (looking for Model 16 operating systems and software), it looks like it's worth the price. Anyway, if you need some kind of software that you just can't find, call Dennis Beyer at Softsearch, (800) 531-5955, and find out more about their services. Tell him we recommend Softsearch. (We can't report on the 16 software search yet—the results aren't conclusive.)

MYSTERIES OF THE PLAINS.

I asked someone in Fort Worth about the puzzling situation with regard to the Model 16 multi-user operating system. Since about the first of September, Computer Center managers all around the country have been freely saying that the multi-user operating system would be "UNIX-like" and that it was to be released by December, yet Tandy has made no official announcement. This seems to be a new twist in non-announcements, as the Tandy representative simply refused to "confirm or deny" any of it, observing that "the world gets curiouser and curiouser." If Tandy is "running it (UNIX) up the flagpole to see if anyone salutes," they should be aware that many of us are at least standing-at-attention. Quite a few of us would like to see a Tandy-supported operating system with the versatility and universality that UNIX is reputed to possess. Let's hope it comes down the flagpole and into the stores.

CALIFORNIA DREAMIN'?

One California Computer Center manager was quoted as saying that, in addition, by December, Tandy would release a version of CP/M 3.0 (Digital Research's enhanced version of CP/M for 8-bit machines) and also FORTRAN-77 (source and machine-size unknown). This rumor may be a result of sun-stroke and is probably too much to hope for; I couldn't bring myself to incur the disappointment of another refusal to confirm or deny. Just knock on wood.

MORE MYSTERIES OF THE sOFTwAREwORLD.

There are a number of other rumors going around about the Model 16, but there is little support for any (in fact, many don't make much sense no matter how you look at them). Ed Juge has told me that one set of rumors, although very popular in fashionable circles, was absolutely unfounded, so there's no point in retailing those. However, at least five different 68000-experienced software firms have refused to divulge what they are doing with respect to Model 16 software on the ground that they were "still negotiating with Tandy" about a possible OEM software deal. One firm said that they could tell me that they *had* made an OEM deal with Tandy, but that they couldn't tell me any details. A specific type of popular software came to mind based on the historical activities of this particular firm. The firm's representative said merely, "I can't deny that's what we've been good at." In an unrelated part of the conversation, the software executive mentioned that they were "desperately searching for UNIX gurus." I said "Aha!" and he said, "I didn't say that, did I?" In any case, these are all just "straws in the wind," and don't deserve the status of reliable rumor. But with software beginning to appear for the other 68000 machines (Fortune, Motorola, et al), can the Model 16 be far behind?

BIGGER FLOPPIES FOR THE II, OR, WHERE HAVE I BEEN ALL THIS TIME?

We've thought for some time about how handy it would be to have a double-sided disk expansion bay on our Model II, so we could use the two-siders created by our 16 on our II. In fact, the idea of all that storage (3,750,000 bytes in a 3-drive bay) can be mighty appealing, indeed. So we've been poking around and found out that one fellow we know has been running his Model II with a double-sided expansion bay (under CP/M) for a long time. We asked how he had got hold of a double-sided controller board (an outfit in Texas—not Tandy—said they cost $1300, but aren't available, anyway). He said that, as far as he knew, ALL Model IIs come equipped with a double-sided controller, but that some have had the trace cut and require a simple jumper. We are looking into this with considerable interest, and would like to hear from readers who would also be interested in double-sided drives for their Model IIs. If there is enough demand, and good drives can't be reasonably obtained some other way, we are prepared to negotiate an OEM arrangement ourselves. It looks like a two-drive bay ought to cost about $1600, with a 3-drive unit going for about $2100. It is also possible that the Tandy expansion bay for the 16 (two drives, $2098)

will run on the Model II. We'll keep you informed of any progress. Please let us know if you're interested. Put a note on the reader survey form.

HARD DISKS MAY BE GROWING UP SOON.

If the widespread rumors (some from inside manufacturers' plants) have any validity, hard disk capacities will be going up dramatically while prices remain about the same or marginally higher. We've heard several tales of 100MB to 140MB hard disks from major manufacturers at $6000–$10,000 by early next year. For those who prefer to stay with a single vendor (it does reduce the finger-pointing!), we haven't any Tandy-specific rumors. But then, with Tandy's vast spread, hardware has to take a little longer. Mid-83 is probably a safe bet. The problem of backup for these larger disks has not been settled, however. Competing systems include boards to drive video cassette recorders, 9-track tape drives, and independent cartridge tape systems. None seems a clear winner at this time.

SOFTWARE REVIEWS, WHAT AND WHEN?

We are in the process of reviewing a wide range of software products. In fact, some are close enough to completion that they might have been included in this issue except for space limitations. ***two/sixteen*** is printed by one of the leading journal printers in the United States (Science Press of Ephrata, Pennsylvania). Under the usual method of printing smaller journals like ***two/sixteen***, pages are printed in groups (signatures) of eight pages. Therefor, you may notice that our first issue contained 24 pages, the second 32, and this issue has 48 pages. We don't have enough additional material (or cash, for that matter) to add eight more pages. Anyway, here are a few of the products that we are currently trying to kick to pieces:

"Data base" type:
dBASE II
Condor III
Quick-n-Easi
Data Ace
Super
DataStar

Operating systems:
Pickles & Trout CP/M 2.2eD
DOSPLUS II
Aton CP/M

Other items:

ScratchPad (SuperSoft)
NISSCAST (forecasting)
Taranto General Ledger
ISSI General Ledger
ReformaTTer TRSDOS<>CPM
ReformaTTer TRSDOS<>DEC
Microsoft Basic Compiler-CP/M
Various communications packages
Various utility packages
WordStar
MailMerge
SpellStar
SuperSort
CalcStar
FORTRAN-80
STRING-80
AUTOGRAMMER

I've probably missed a few, but that's most of them. Radio Shack is notably missing, but since they provide a fixed-period loan (extendable) rather than the usual permanent review copy, we don't want to get a larger backlog than we can handle in the time permitted. We have recently requested several of Radio Shack's most often mentioned (by subscribers) software products for review, and, if possible, Tandy software reviews will appear in each future issue.

Our practice in reviewing software is to apply it to our regular work whenever practical. We attempt to evaluate software in the same operating environment as the typical user, that is, with access to the manual but not to the manufacturer. Many otherwise fine pieces of software cannot stand this documentation test, and eventually require recourse to the manufacturer for clarification. Unfortunately, the average user is sometimes forbidden to contact the manufacturer directly and is at the mercy of his dealer's (often limited, if existent) knowledge. Between our travails with nearly always inadequate documentation and our efforts to run our business on several different data base systems at once, our records (and ourselves) are often in an unbelievably confused condition. (Your should see the reader survey results, partially entered in three different data base systems, none quite right for it!)

Such reviews take quite a long time, even several months, to complete. But I think you'd rather see a review that reflects the sort of experience you might expect to have with a product than the usual one-or-two day romp with the software that's often published. We've all seen reviews of products we own; it's often surprising how their greatest virtues and worst faults are completely overlooked. Sometimes I'm sure it's some other piece of software they're writing about.

two/sixteen ON THE WIRE!

Here's a final note that I'm really excited about. ***two/sixteen*** has been accepted as a ***NewsNet*** publisher. What's that, you say? Well, you've heard of the late lamented ***Philadelphia Bulletin***. ***NewsNet*** is its surviving offspring, so to speak. To be more enlightening, ***NewsNet*** is an "electronic publisher," specializing in newsletters. Our latest information is that they "publish" over 170 newsletters. Each publication is fully indexed and available on line to subscribers. ***two/sixteen,*** while small for a magazine, is very large compared to newsletters, so our longer items will be abstracted for ***NewsNet.*** Our subscribers will be able to access our ***NewsNet*** editions at the basic ***NewsNet*** connection fee, without surcharge. We plan to enter articles, abstracts, and programs into ***NewsNet*** on a monthly or even more frequent basis, so that items of current interest will be available earlier than possible with our print edition. Of particular interest is your ability to download program material from ***NewsNet*** without fear of typographical error. We'll provide full details in our Jan–Feb issue. We may have material on ***NewsNet*** as soon as early December. Let us know if you want to be validated for access. If you want to find out more about their general service and other newsletters available, call ***NewsNet*** at (800) 345-1301, or, in Pennsylvania, call (215) 527-8030. You can also write to them at 945 Haverford Road, Bryn Mawr, PA 19010. Please mention ***two/sixteen*** if you do. RHY

REPRINT SERVICE COMING SOON!

two/sixteen magazine is pleased to announce that beginning in January, 1983, reprint service will be available for articles appearing in our publication.

For a fee of $1.50 per article, you will be able to order a reprint of articles published in *two/sixteen*. The only limitation imposed on this offer is that reprints must be ordered within 90 days of the magazine's publication date.

For reprint copies, send your request and check to:

COMPendium Reprint Service
P.O. Box 129
Lincolndale, NY 10540

Classified Advertising

A non-dealer subscriber may place one ad per issue at the rate of $7 per column inch.

A dealer may place one classified ad per issue at the rate of $20 per column inch. Dealer classifieds must be for the sale of specific products or services at specified prices.

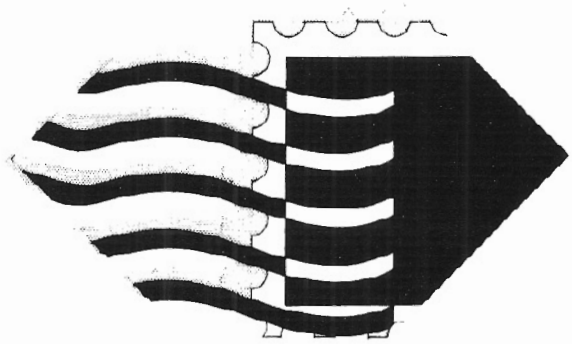
Ads must be for hardware, software, or services which would be of interest to Model II or 16 users (no mystical societies chain letters, book clubs, etc.).

Ads will be set in the same size type as the text in the magazine.

Closing date for classified ads is the 15th day of the month prior to issue date (e.g., the January–February classifieds will close December 15.

We reserve the right to determine what heading an ad should be placed under (e.g., HARDWARE, SOFTWARE, MODEMS, PROGRAMMING), but we will accept advertisers' reasonable requests for specific headings.

Feedback



Our policy is to publish all letters on topics which may be of general interest to our readers. We have included most of the recent letters received so that our readers may see the cross-section of other readers' concerns. Our comments, in italics, follow many of the letters.

We are particularly grateful to the many subscribers who took the time to write to us about their experiences and problems. Several patterns have emerged that will help us provide Model II and 16 users with useful reports.

Several of the following letters ask for assistance beyond the knowledge of the two/sixteen *staff. Readers with solutions to any of these problems are urged to respond to the writer directly with a copy to us.*

FROM: Jerry Lippey, The Lippey Company, 210 S. Bundy Dr., Los Angeles, Calif. 90049:

OUR NEEDS: I do most of our programming—we don't generally purchase application software. Our needs are simple. The Free MBASIC interpreter works okay for our purposes. So far, all of our machine-language programs have been small. I assemble them by hand. Consequently, we haven't found a need for CP/M, a compiler, or even an assembler. (But the RSMII disassembler has turned out to be very useful.)

We expect to stick with TRSDOS unless we run across an unanticipated requirement.

Here are three problems that I would like solved (under TRSDOS):

1.) A convenient way to divide a file. For ex., a program that resets the directory (and alternate) so that one file becomes two, i.e., by assigning a new filespec to records beginning with record # x and terminating the original file at record # x-1. This should operate on files of any record length (fixed—not variable), but we would use it most frequently on ASCII (RL = 1) files.

It's a pain to segment files by copying pieces. Restricting the partition points to where record boundaries coincide with GRAN boundaries is a lesser evil.

2.) Upgrade VISICALC to achieve features of the new Model III version. Neither Software Arts nor Tandy has plans for this. Tandy wants us to wait for the 68000 version. An upgrade that provides only formula edit and "IF" would be satisfactory.

3.) FORTH. This is not required for business reasons. I fooled around with LISP once and enjoyed it. Also, I like RPN and stack-based processing. FORTH would be fun to play with. I haven't been able to locate an implementation for the Mod II.

I would be most interested in magazine articles related to utilities, patches, and so forth which make life easier under TRSDOS. In particular, a map of TRSDOS would be very helpful—sometimes I would like to make small changes.

THINGS TO SHARE: We have several tiny machine-language routines (Using TRSDOS SVC's) that we consider indispensable. Also, we wrote a couple of MBASCI general purpose programs that others may find useful.

ON INFORMATION EXCHANGE. You ought to consider COMPUSERVE as one vehicle. If a local COMPUSERVE phone number is available, this means of transferring files costs only $5 per hour, with no fee up front. It beats the cost of phone calls between L.A and Lancaster. OUR COMPUSERVE # IS 70205,1104.

Incidentally, the TRS-80 SIG there has been my *only* contact with knowledgeable Model II people during the two years we've had this machine.

If you fulfill only half of your objectives with *two/sixteen*, we'll by continuing, grateful subscribers.

In the interest of helping you know your audience (not as a potential author, reporter, or columnist), here's some of my background:

—First program was SOAP for the IBM 650 in late 1950's.

—Coded in assembler for the IBM 1620, 1130, and 1401 and 7090 series. Also in FORTRAN, ALGOL, COBOL (ugh), PL/I, and APL.

—Employed by IBM for 16 years in advanced product development, usually as a software development manager (I was the systems programmer manager for the IBM 1500—an experimental CAI [computer assisted instruction, ed.] system).

—From 1964, involved mostly with education-oriented use of computers—published several papers and edited a book in this field.

—Introduction to micros was via the early IMSAI.

—Subscribe to *80 Microcomputing, 80 U.S. Journal, Computronics Mod II Newsletter* (filled with misinformation).

P.S. I wouldn't have filled these three pages if I didn't have high hopes for a successful new publication. Don't let me down.

(2) I'm inclined to agree with Tandy on the question of upgrades to VisiCalc. Useful financial plans can't be created in 64k. There is not even enough room for a reasonably detailed profit and loss statement by month for one year, much less the many assumptions and calculations which should be included in the same array. Once a "Calc" type product is available for the Model 16, we will upgrade to maximum memory so as to allow planning in reasonable detail.

THINGS TO SHARE: Please send us the routines. Many readers have asked for ways to apply the SVCs.

INFORMATOIN EXCHANGE: Another reader has expressed preference for The Source. We feel that the $100 entry fee may deter readers from signing up with The Source unless they have other specific and significant uses for its services. In any case, we will publish readers' CompuServe and Source numbers in a directory periodically. Send them in.

FROM: Paul Naitoh, Applied Psychophysiology Group, 4446 Marseilles Street, San Diego, CA 92107:

I have just received the premier issue of *two/sixteen*. I would like to thank you and congratulate you for your courage in starting a new magazine for a potentially limited market. I hope that many of 45,000 owners of MOD II would rally around your flag.

As for me, I am willing to help you and your worthwhile venture. I think that I can help you analyze some of the statistical packages available for MOD II. I am also available for you as a reader/reviewer of submitted articles to your magazines, regarding statistical analyses. You will be surprised to learn that the *best* analysis of variance programs is available free by asking its copy from the author of a paper which appeared in *Behavioral Research Methods and Instrumentation*. I am quite familiar with FORTRAN, and I have purchased RS's FORTRAN (Microsoft) and Editor/Assembler. But don't send me any paper which uses the assembler, because I am illiterate on the assembly language!

I do have a question about Patches 2 and 3 which you have published in *two/sixteen* (page 8). These patches did *not* work on my MOD II TRSDOS 2.0a. From what I understood, I could make SYSTEM files accessible by patching in "PATCH SYSRES/SYS A = 1786, F=23, C=00. That is, I can use COPY or MOVE to transfer, say M80, F80 or L80 from one diskette to another. Well, it did not work. After COPY command, I got the error saying that the system program is protected from such a no-no venture! Similarly, (the) PATCH for head-stepping rate change did not work. Do I have a bad system? Or where have I goofed up? Your PATCH 4 works fine. Since I need to move Assembler to a diskette which contains BASIC, I really do like to have an access to SYSTEM program.

You have raised many questions to us, the subscribers, especially about future articles in *two/sixteen*. Your "FUTURE ARTICLES" announcement contains topical areas I am very much interested in. I would also like to add a few more items of my own. Please forgive me to list a low level question about microcomputers, because I am self-taught to use FORTRAN and BASIC, and I have no experience of being formally taught about computer architecture and assembly language, and other high level topics. But as a professional, I do know what is missing in many of the magazines which are supposed to teach us the way of using microcomputers. I really wish that *two/sixteen* will fill the information gaps for a guy like me, who learned computers by "hacking."

First, can you come up with Diagnostic Programs for a variety of MOD II components? I have noticed that the Radio Shack computer technician (who comes to do a preventive maintenance for MOD II) has his own DIAGNOSTIC programs on his CP/M diskette. I know two of these technicians, and both have told me that they use CP/M to do a diagnosis for MOD II because CP/M provides an acid test in comparison with all-forgiving TRSDOS. I don't know how much TRSDOS checks the system, before it gives us the "go ahead." Not that I can repair the system, but it would be nice to know which component really failed the test.

Second, I wish to get an extra disk drive to facilitate a process of backing-up diskettes, and to get a bit more diskette space. But, I don't want to purchase RS's bulky diskette drive with space for three drives (I need only one; besides who has a program or programs to let us do inter-drive diskette manipulations?); but I am also scared of purchasing non-RS diskette drive (8"/just one). You might suggest a purchase of fixed disk drive, but isn't it expensive and don't we have software problems of

copying hard disk to multiple floppies, unless we spend $6000? I think that the readers of *two/sixteen* will be delighted to learn that we have (or don't have) an option to buy one floppy disk drive (or one inexpensive fixed disk drive) from other vendors. I saw so many advertisements about optional disk drives for MOD III, but really none for MOD II.

Similarly, I really wish someone would tell me about MOD II graphics (with or without new RS enhancement), especially a printer which can do the graphics. In an RS advertisement, I saw that new graphics enhancement will use Color COMP's extended BASIC (without color, of course) to drive a printer MOD VIII. Is this the only option we have?

Third, I wonder if anyone ever came up with an analog/digital conversion system for MOD II. When we don't even have a stick to move a cursor around on the monitor scope, perhaps I am asking too much for MOD II. But surely there must be a digitizer for MOD II.

Fourth, I wonder if you can develop a series of articles to explain assembly coding business, say for BASIC or FORTRAN. In BASIC, what can we do in USR function calls, when we don't have PEEK and POKE? A recent article in *80 Micro* described an installation of PEEKPOKE, but that article was but one page long! Another article in *80 Micro* mentioned simply that the authors used PEEK and POKE in programming, but no details are given. I guess what I am asking for is a foolproof step-by-step description of how-to-call machine coded program from BASIC and FORTRAN.

Now about my MOD II: I have a MOD II 64k system with *one* built-in disk drive. No external or hard disk drives. I have a Daisy Wheel II printer.

I use TRSDOS 2.0a system. Since the editor is so poor, I have purchased a full-screen text editor from *Aspen* Software Company. A super product. To do manuscript typing, I have purchased RS's SCRIPSIT 2.0—and do I have a complaint about it!! After I paid close to $400 for it, I still don't have the right-side adjusted porportionally spaced typing. What a disgrace the RS is! I am sure that WordStar can do such proportionally spaced right side-adjusted typing. The RS is also very slow in developing new Daisy Wheel type face, such as a type wheel for mathematical symbols.

I purchased an array of RS (or Microsoft) software, i.e. FORTRAN, EDITOR/ASSEMBLER, and STATISTIC PACKAGE. My FORTRAN is the latest verson, I believe, but it still lacks COMPLEX algebra, while FORTRAN which is sold by Microsoft does have COMPLEX function implemented. And the instructional manuals are atrociously written.

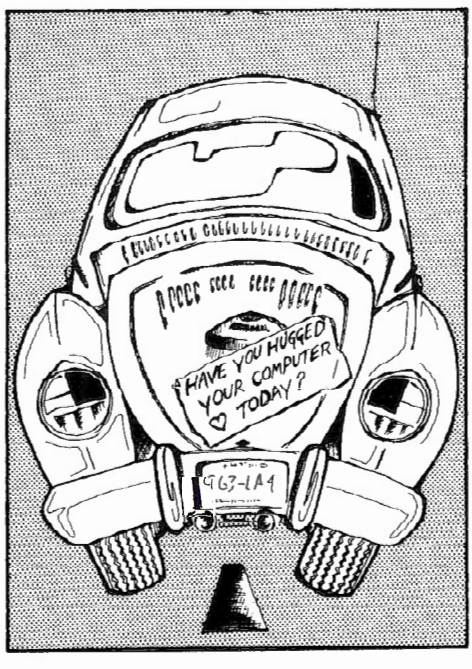
STATISTICS package is not as good as it is packaged. I have an excellent write-up, but what it can do is very limited. To do more statistics, I purchased Quant System's "The Statistician," which is quite satisfactory, except it is rather elementary in analysis of variance and time-series analysis.

From Aspen Software, I purchased PROOFREADER/GRAMMATIK which is quite good, but very limited in the sense that it cannot show incorrectly spelled words in the text of the manuscript. All errors are listed in a table. The Aspen people have written me, saying that I can do better with their own word processing program than with SCRIPSIT, because, with Aspen word processing software, all the typographical errors are shown in the text (as it is done by Radio Shack's dictionary, but RS dictionary requires a two disk system, while Aspen system just one disk.)

My apology for long-winded nonsense, but I wanted to express my enthusiasm towards your magazine. Again, if I can be of any help as an editor/reader or whatever, please write me.

Finally, I encourage you to get all of the advertisers lined up in *two/sixteen*. To me, the advertisements represent information as long as you select them to be relevant to MOD II/16. I am really tired of going through all those pages of *Byte Magazine* and *80 Micro* to find two or three pages of advertisements on MOD II, not to mention almost non-existent articles on the use of MOD II. I have enjoyed working with MOD II, and I hope that your magazine will add more enjoyment to the MOD II owners.

Keep up your good works, and may I wish you the best of luck (and hope that you may make some real profit).



Anyone have a Model II hardware diagnostic system? Please let us know if you do.

Last Fall, we bought a Shugart single drive expansion unit from Texas Computer Systems, (800) 433-5184. We understand that they are out of stock at this writing, but are attempting to obtain more units. The one-drive unit cost about $850 and has worked fine for the last 8 months. It is less than 5" wide, and includes its power supply. A. M. Electronics, 3446 Washtenaw Ave., Ann Arbor, MI 48104, telephone (313) 973-2312, offers Tandon "Thinline" 8" drives for the Model II at $695. Non-RS drives shouldn't be feared. They are usually identical to RS drives. The only differences are that they cost at least 30% less, and they are often more tastefully encased. Failures of disk drives are rare, but if a non-RS drive fails, just about any dealer (except RS) can fix it easily. RS has said they are not competent to repair equipment that has not got their logo on it, even if it is otherwise identical. However, there are encouraging signs of common sense creeping into RS policies, and I believe that it's reasonable to expect that Tandy-RS's warranty and service policies will eventually be competitive with other stores.

Business screen graphics for a computer like the Model II or 16 is just BS and romance as far as I'm concerned. In a business situation, what we all need is some kind of hard copy we can use to explain subtle concepts to others. For printer graphics, I recommend the NEC Spinwriter 7725 with bi-directional tractor. We use this printer under both TRSDOS and P+T CP/M. It cost us $2659 for the printer and $300 for the tractor from Bartlett Systems, Inc. (215) 666-7100, delivered and installed, plus sales tax. This may seem a bit expensive compared to some other fully-formed character printers, but I wouldn't trade it for two of any other (or even four of the under-$2000 daisy wheels). We also have a dot-matrix Epson MX-100 (now costing $750–800), which is a fine printer for its type, but I wouldn't attempt business graphics with any dot-matrix printer. Writing your own graphics routines for the NEC with bi-directional tractor is a piece of cake for even

the most inexperienced programmer, assuming that he has the basic business knowledge to produce a good graphic design.

Digitizers for RS-232 ports exist. I can't locate one right now—they're not often advertised. Perhaps a reader can let you know where one can be obtained. However, before you buy, make sure that you can obtain or develop the software necessary to read the data presented to the port.

Scripsit 2.0: Caveat emptor. RS assures us that Scripsit was never intended to handle right justification of proportionally spaced typing. Two points: (1) make sure you ask about all the features you need before you buy and (2) don't expect a $2000 printer to do everything that a $4000 (list) printer will do.

Daisy wheel typefaces: a subscriber believes he has found a (non-RS) source for additional Ricoh (they are the manufactuer of DW II) daisy wheels. If he is successful in obtaining some, we'll let you know in the next issue.

Microsoft tells us that they don't have a FORTRAN with complex functions for any 8-bit machine. If you wish to do complex algebra, you will probably have to buy a 16-bit machine.

Statistical packages will be reviewed regularly in two/sixteen, *beginning soon.*

FROM: David L. Dickey, 501 N. State St., Ukiah, CA 95482: (1) Looking to buy a black box printer buffer of 32k or more plus switching device to change printer and computers, so I'd like info on these. (2) Very interested in hard disks and using MOD II (to be upgraded to MOD 16?) with 2 extra terminals. Would like information on pros and cons of various hard disks. I'm using over 500 8" diskettes presently. I want maximum storage and *BU* capacity for least price as long as reliability is there. I don't see *BU* in RS hard disk. (3) What are the best CP/M "fixit" programs. Are they any good? Is there any literature explaining CP/M disk structure with reference to "fixit" programs that's easy to understand? Ditto for literature on RACETS, SUPERZAP, etc. (4) Interested in software and operating systems for MOD 16.

(1) There are quite a few printer buffers on the market, and you will find several advertised in almost any general purpose micro magazine. As to the switching box, get the catalog from Black Box Catalog, P.O. Box 12800, Pittsburgh, PA 15241, telephone (412) 746-2910. They carry an astonishing array of switch boxes and other useful aids.

(2) See our hard disk article in this issue.

(3) We would appreciate an article from a reader with respect to the CP/M "fixits."

(4) We regularly talk with several potential operating system developers for Model 16, but as yet have nothing publishable. We'll let you know as soon as we do. The situation looks promising for some useful developments by the end of this year.

FROM: Richard Rothenberg, M.D., Director, Bureau of Chronic Disease Prevention, State of N.Y., Dept. of Health, Tower Bldg., The Gov. Nelson A. Rockefeller Empire State Plaza, Albany, NY 12237:

I received the first copy of the magazine, for which I thank you. I am enclosing the reader survey form, and would like to add that my major use for the Model II is as a number cruncher (of sorts) in the analysis of statistical and epidemiologic information. I use it primarily in conjunction with a larger data base that can be accessed by remote dialup (know a cheap baud modem, by any chance?). With that in mind, I would be particularly interested in material on the following:

(1) Floppy disk units that are plug-in compatible. I've been able to identify one, a Traxx, which costs about $550 for a single side, double density disc drive with case and power supply, and feel as if I'd like to make the leap, but would be interested in knowing what more the market may have.

(2) The use of Mumps on the Model II. There is, actually, a disk that I've sent off for which presumes to run Mumps on a mini (I've actually seen the version for the Apple), but would be interested in knowing more from users who've tried.

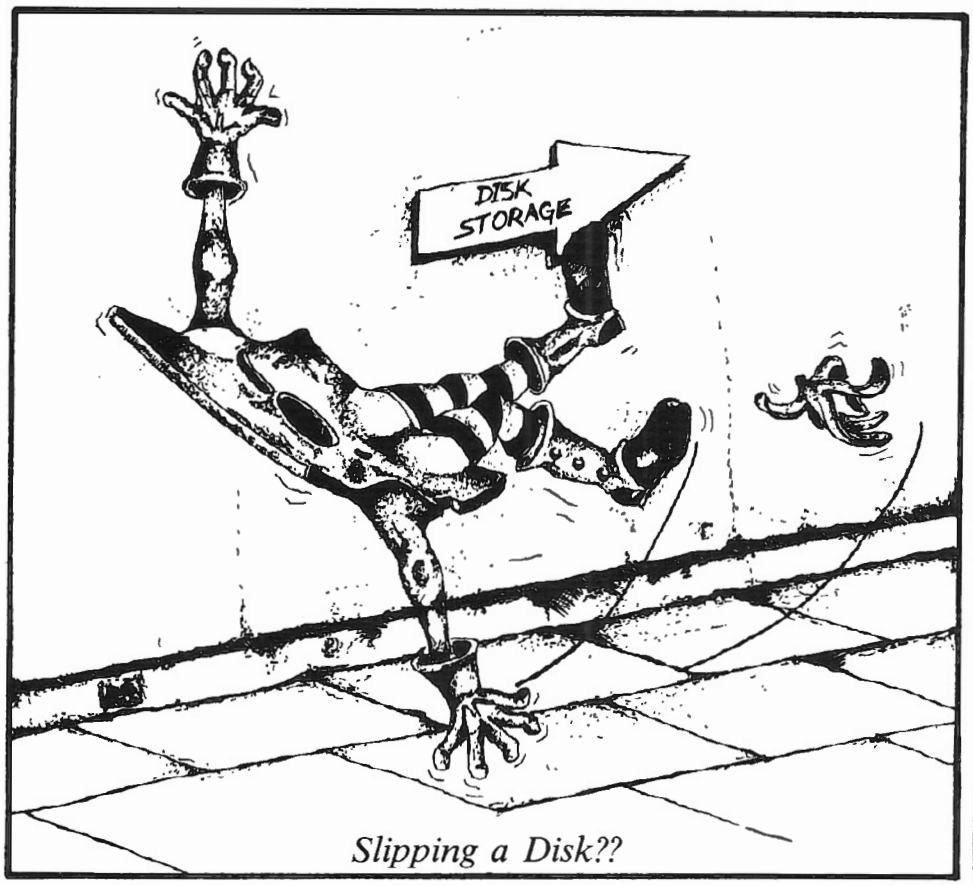
(3) Discussion of dBASE II applications using the Model II configuration. I've seen one or two articles elsewhere, but this is such a powerful relational DBMS that it might be worth some copy.

(4) This is way out, and may in fact do nothing but demonstrate my ignorance, but I was wondering what has been considered in the interfacing of noncompatible cpu's (e.g. Z80 and 6502).

(5) Material on the big networks and data bases (Compuserve, The Source, etc.) would be most illuminating if presented from the user's rather than the seller's point of view. One can sort of play around and find out what they have, but I would bet there are some really nifty ways to use these things that the casual subscriber won't notice (and that the companies might not push). As an aside (and free-associating a bit), are there computer simulation models available for the general public to play with? An example would be a tax model which would permit quick access to the tax outcomes of personal financial decisions. I just can't believe that the megacompanies function without these.

I'm just hoping that you and your staff can keep this thing off the ground. You certainly touched a responsive note in me, a user floundering alone in the wilderness—the local Radio Shack Computer Center told me that I'm one of four people in the Albany area to whom they've sold Model II's for home use. I'm afraid I'm too much of a tyro to be of much help to you, but will try to offer whatever I can.

Your letter inadvertently omitted the



Slipping a Disk??

baud rate of the modem you want. Assuming it's 1200 baud, it appears that several modems will be available soon in the $700 range. RS announced one last month.

(1) If the Traxx you mention is truly plug compatible and includes a power supply, it looks like a good buy. Several others have been available in the past (Siemens, Shugart, Lobo, Tandon) at prices from about $700 to $900 (fully assembled, with case, power supply, and cables), but I haven't seen an ad for any of these recently. See my comments re a prior letter.

(2) Readers: let us know if you use MUMPS.

(3) We are obtaining a copy of dBase II for review and hope to cover it in the next issue.

(4) Actually, almost any two CPUs can be compatible if they are placed in the right hardware and software environment. The Model 16 uses both a Zilog Z80 8-bit CPU and a Motorola 68000 16/32-bit CPU (of course, we have no way yet to judge their performance). In addition, another firm is developing a board for the Model II (or 16) with 8086 or 8088 CPUs on it. More on that in the next issue.

(5) We intend to cover the big networks in future issues, and the free bulletin boards. There are a large number of Remote CP/M Software Exchange Systems which provide free software. We'll cover those as well. General purpose simulation models are very expensive to develop and very expensive to buy. I know of none in the timesharing business, but several TS systems can provide the basic tools necessary to construct simulation models. Don't try it unless you have a lot of time and money.

FROM: Robert D. Aaron, Pres., Aaron/Smith Associates, Inc. (Information Brokers), P.O. Box 7567, Atlanta, GA 30357 404/876-0085:

I am very excited that you are starting this magazine. I have often felt like a ship without a harbor with my MOD II. One or two articles in *80 Microcomputing* just doesn't make it, and *Computronics' Newsletter* isn't very informative. Many best wishes, and I look forward to receiving my first copy.

In addition to your planned articles, I would like to recommend a detailed comparison between the major word processing packages available for the II/16. I am a heavy user of Scripsit 2.0, which I like overall but would like to see specific enhancements (alphanumeric sorting capability, mainly). Other items of interest to me include electronic mail systems, VisiCalc vs. MagiCalc (spelling?—the one from Peachtree Software), telecommunications, and evaluations of items such as this fast backup/fast file copying utility from Racet Computes. Also, any information on how one can manage PEEK and POKE commands on a Mod II would be greatly appreciated, in that many if not most Mod III BASIC programs could run on a Mod II if this could be accomplished.

We will be doing some kind of review of the "Calcs," including the new ScratchPad 2.0 from SuperSoft, which is said to allow 2600 cells with data in each one (a substantial increase over the about 500 cells usually allowed). It should be in the next issue. We have a fast backup utility for review in the next issue, also, and hope to get the dope on PEEK and POKE by then.

FROM: Robert D. Aaron, Aaron/Smith Associates, Inc., P.O. Box 7567, Atlanta, GA 30357:

. . . you asked about other Mod II games. I purchased a marvelous draw poker simulation called DRAW5 from Wilson Software in Camp Hill, PA, telephone 717/737-0280. In it, the computer plays six hands of five card draw (jacks or better to open) and you play one, thus allowing for the better odds that arise when enough hands are dealt. If you fold, the computer's hands play among themselves. Most of the time it is a very believable and effective simulation (I am an avid poker player), and it is written completely in BASIC. It also costs less than $20.00. You might want to look at it.

I am very excited about your magazine. It is about time that someone did what you are doing, and I applaud your first issue as auguring well for the future. I am also very interested in the information exchange idea and would certainly subscribe to it. I do think that a software library of programs donated by members would be great, as well as discounts on commercially available software.

In fact, I have a question for you already. I would like to find a patch to TRSDOS that will produce an underline cursor such as that found on IBM terminals (and as is available in the new Scripsit) rather than the blinking box now used. I think this would be a pleasant cosmetic addition to TRSDOS. Please call, write, or publish the results if you

can find such a patch.

I have other ideas for the information exchange, but I'll wait until it actually exists. Thank you again for the outstanding job you're doing.

P.S. If you like, I'd be willing to write a review of DRAW5 for your magazine. Please let me know.

As aficionados of Model II games for occasional diversion, we tried to raise Wilson Software several times (day and evening) without success, so we are unable to confirm that DRAW5 is still available. However, if it is, we'd like to review from you.

FROM: Norman N. Holland, James H. McNulty Professor of English, State University of New York at Buffalo, Center for the Psychological Study of the Arts, 409 Samuel Clemens Hall, Amherst, NY 14260

As you must have heard from dozens like me, I am delighted that you have begun *two/sixteen*. You are absolutely right when you say that the Model II has been neglected by the regular computer magazines and by Tandy. I hope the 44,000 of us out here are enough to make your venture successful.

You ask for input. I'd like very much to see you publish whatever you can by way of a memory map of Mod II. Tandy has been rather secretive about same, and it would be a great help to me (and, I suppose, others) in transferring programs written for other computers if we knew where the video memory began, where keyboard input goes, and so on.

Along the same lines, I find the instructions in my TRSDOS 1.2 manual on SVC calls quite cryptic, and I have been unable to use them by means of DEBUG. I would be very grateful for an article showing at greater length and more clearly how to use them, how to write programs in DEBUG using them (with examples), how to interface them with BASIC and so on.

I take it my third request is part of the same problem. I've been unable to get a screen dump that would work reliably with my NEC Spinwriter (serial, running on TRSDOS 1.2). Could you?

Any or all of these would be a help to this Model II owner, but there is such a dearth of communication and information that anything you say is bound to be a boon.

Anybody got a memory map? Let's print it. And how about screen dumps?

FROM: K. Hoyenga, 1137 Stacey, Macomb, IL 61455 (Winter Address); Box 1278, Whitehall, MT 59759 (Summer Address):

Problems, Interests, Concerns: (1) Does anyone know how likely I am to have trouble created by transporting my computer back and forth between Montana and Illinois? (Always in the original packing carton and material, of course!) Radio Shack said none, but they wanted to sell me the computer! (2) Interested mostly in data analysis and word processing. Is there any indexing program that can work with a Scripsit 2.0 document (or the ASCII version of same)? Is there any program that will sort a very long file consisting of words followed by numbers—sorting in the words (alphabetically) *and* the numbers)—file is longer than CPU memory. (3) What do electric line surges and spikes do to the Model II? Evidently my built-in filter couldn't cope with the ones I was getting.

You shouldn't encounter any machine problems from transporting your Model II, whatever the distance, but you may wear out the original cartons. If I had to do that, I'd get a specially built case. You can get a Fiberbilt case made especially for the Model II (naturally, it will hold a 16 just as well) from Ikelheimer-Ernst, Inc., 601 West 26th Street, New York, NY 10001, telephone (212) 675-5820, for $165. We saw one at a computer show in NYC in April. It looked very sturdy. They also sell the best collapsible baggage cart we've seen. It's $35; we bought one and wouldn't part with it.

We don't know of any indexing add-ons for Scripsit, but there are at least two for WordStar. They will be reviewed soon.

SuperSort by MicroPro will sort almost anything, including files whose fields are comma delimited. It is the nicest sort package we have seen. Of course, it is CP/M-based. We (and our RS CC) know of no sort utilities for TRSDOS.

In extreme cases, electric line surges are said to do great damage to the chips. There are many line isolators and uninteruptible power supplies on the market. Check the ads in the major general purpose micro magazines.

Please all, and you will please none.
—Aesop
THE MAN, THE BOY, AND THE DONKEY

More Feedback

FROM: Dr. Dennis D. Kelly, Dept. of Psychiatry, Columbia University, 722 W. 168th St., NY, NY 10032:

Your comments in the May-June *two/sixteen* struck home. Your description of the intended composite reader quite accurately described me. I am a scientist, not a computer professional, and I would like to use my Model II to help me do science. Until now the major benefit to my research of owning the Model II has been word-processing, itself a minor revolution. However, after 18 months and reasonable, if not single-minded effort, I still control my experiments with hard-wired solid-state logic circuitry (as I have for 20 years), and I still batch process my keyboard-entered data using large statistical packages on the main-frame at the computer center. I had hoped for more. The power for process control of experiments and for data analysis is obviously there in the Model II, not to mention the 16, but I have not been able to harness it for my research, unwilling as I am to sidetrack my scientific program for an exhaustive computer tool-up.

There have been two identifiable stumbling blocks. Perhaps *two/sixteen* would at some point consider them, for with respect to scientific users of Models II and 16, I do not think they represent idiosyncratic concerns.

First, there are no adequate and reasonably priced statistical packages available for the Model II. In fact, to my knowledge, there currently exist better commercial programs available for the Model I/III than for the Model II. The RS Statistical Analysis MII program is quite poor and, for the professional, probably not worth the effort to learn. Indeed, it does not even pitch the Model II as the equal of most $200 calculators. In particular, the file structures are inflexible, and virtually preclude performing multiple analyses upon the same data file. For instance, a file format appropriate for simple descriptive statistics cannot be subjected, without extensive revision, to an analysis of variance, and vice versa. The CP/M based Microstat is quite expensive. It is also quite limited compared to standard statistical programs supplied with the relatively slow, 16-bit minicomputers. In sum, for the scientist, the cost of packaged statistical programs for the Model II is probably not justified, unless he or she has no other access to a computer center.

Second, I have found very little information on, or commercial hardware support for, interfacing the Model II for process control in real time. How do I get the Model II to run my experiments, and to record my data? Is there a general purpose interface for the Model II that will allow me to close switches at certain times and to record digital data (also switch closures) so that the latter can be used to alter the program controlling the former?

For the type of work you appear to be doing, I suspect that you are correct in suggesting that it might be more efficiently done on a large system. If one is not handy, there are several good time-sharing systems available, including CompuServe's large and friendly DEC-10s, which can even be accessed through the CompuServe Videotex network with your Model II or 16. However, we have a number of subscribers who are expert in the field of microcomputer statistical work, and perhaps they will be good enough to let you know their opinions.

Process control interfaces tend to be custom made. Volume is small and applications vary widely. You need two essential items: (1) hardware that connects to one of the RS-232 ports on the micro and interfaces to (or contains) the switches, analog to digital converter (if required), and manual input keyboard (if required) and (2) software that will operate in real time (the real time part should be easy), interrogates the port (A or B), and writes the result to memory (or disk, but watch out for lost data if you do this while observations are coming in). You might be able to find a real-time analog-to-digital type individual in one of the New York City universities (those that offer Masters in Computer Science degrees), or even through a computer store. The non-RS stores are more likely to be familiar with people having this kind of interest.

FROM: Dennis O'Connor, 141 S. Colonial Avenue, Richmond, VA 23221:

I called you today asking for help in converting TRSDOS diskettes to CP/M format. . . . I may need to have other files converted in the future so if you know of other *two/sixteen* subscribers in the Richmond or Washington, D.C. area that have P & T CP/M, I would like to be able to contact them to arrange for a file copying service.

I was wondering if you could help me with another problem I am having with the Model II using CP/M. I would like to know how to initialize the features for the serial I/O interface, such as baud rate, character bit-length, parity, etc. Under TRSDOS, this is accomplished using the SETCOM command or through a call to the operating system. I have the technical manual for the Model II, but the Radio Shack information says nothing and the Zilog spec sheets are beyond my understanding. I believe what is required is to send a series of command bytes to the status port to initialize and enable the serial interface, but I am uncertain as to what bits and bytes to send. Can you tell me how or where I can find this information?

I am very impressed with the first issue of *two/sixteen*. I am now in the process of deciding which other subscriptions to cancel as there has been very little of value for a Model II in the TRS80 magazines.

Under TRSDOS, you must also give the command

FORMS S

to cause the output to be directed to serial port B. I also give

FORMS X

so that any special characters will be passed "transparently," that is, not interpreted or translated by TRSDOS. If your printer is set up in "standard" form (no automatic line feed after carriage returns), you will need to issue the command

FORMS A

II + II = (BUSINESS DYNAMICS)

DOSPLUS II and the TRS-80 model II. The Dynamic Duo.

The TRS-80 model II was created to help you take care of your business. DOSPLUS II was created to help your TRS-80 help you and your business even more.

DOSPLUS II replaces your O.E.M. dos with an ultra-sophisticated operating system that is 5 or more times faster, more reliable and is totally compatible with all standard business oriented software designed for use with the model II. The result is a computer/operating system that far exceeds manufacturer's specs—with more potential and more capacity to be of service to you.

DOSPLUS II operating systems are available for either your floppy or hard disk drives. The hard drive operating system is compatible with Radio Shack, QuCeS, Corvus Micro Mainframe, VR Data and other popular hard drive sub-systems.

DOSPLUS II...All the quality and dependability you'd expect from DOSPLUS.

For the dealer nearest you call
1-800-327-8724
Dealer inquiries invited.

$249.95

For product information (305) 983-3390

DOSPLUS DOSPLUS first in quality! First in the industry!

To order call toll free 1-800-327-8724

POWERSOFT
A Division of Breeze/QSD, Inc.
11500 Stemmons Fwy., Dallas, Texas 75229
(214) 484-2976 Micronet 70130, 203

MICRO SYSTEMS SOFTWARE, INC.
4301-18 Oak Circle.
Boca Raton, FL 33431

or else TRSDOS will print all the lines on top of each other.

Since there is no way to make these parameters "permanent," we put a BUILD file named STARTUP containing the SETCOM and FORMS commands on each TRSDOS disk and issue the command

AUTO DO STARTUP.

If you are running under P+T CP/M, the SETUP command will display a matrix of output parameters which you can easily modify to your requirements. If you want these parameters to be permanent, that is, to remain active when you next boot on the current disk, you must run IOFREEZE after SETUP. IOFREEZE also "locks in" any parameters changed by SETMISC and elimintes the need to specify the number of drives at cold (initial) bootstrap load.

FROM: Charles L. Nelson, 751-D Pinetree Drive, Indian Harbour Beach, Fl 32937:

Received my first copy of your magazine. Congratulations and best of luck on your new enterprise. The launching of a new company in these economic times requires some guts.

At the age of 59 I am jumping into the world of computers. I feel that there is an enormous amount of "catch up" that I must do to understand how to harness the computer to give me the information I want. Unfortunately, most of what I read sounds as though it is written in a foreign language. I am, therefore, fighting to get an understanding without sufficient background material to understand much of what I read. Optimistically, I hope to dig hard enough to gain the necessary background and knowledge to harness the computer by the end of this year. You can help by suggesting a progressive series of books or articles to read that will take me progressively to higher levels of understanding of "computerese" if possible. What I am looking for is a self taught course: Computer Primer, Computer I, Computer II, Advanced Computer. If you have a suggested reading list, I for one would welcome the suggestion, and I expect that there are a number of other business people that must share my need for "a reading path to follow." There are literally thousands of publications and books. The problem is that many are too technical for the beginner. A selection (or published course, if there is one) of books that progressively introduces an understanding would be most welcome by a large group of businessmen who, like it or not, must join the world of computers.

I purchased *80 Microcomputing* at the local newsstand and your magazine is the first that I have subscribed to. As a result of your Directory of Microcomputer Periodicals, I am also entering a subscription for "Creative Computing" and "Desktop Computing" as they sound as though they might contain information useful to me.

Now I am going to burden you with my personal needs and desires for a computer, and the action that I intend to take to fill these needs. If you have the time and if you have any suggestions, they will be welcome. My present posture is to charge full speed ahead, recognize that I am going to make mistakes, but charge these off to the "learning course expense" or "educational expense" and make the necessary corrections and end up—hopefully—feeling at home with the computer by 1982—if not at the start at least at the finish.

My principal need for a computer is to control a construction corporation that has been building condominiums for its own account. At the moment, the condo market is dead. We are building nothing. In fact, we are fighting to get rid of inventory with minimum losses. This gives me the time to buy, install and learn to operate computer hardware and software that I hope will be useful when the next wave of building activity starts. This company has had as high as 4 projects under construction at one time of the dollar magnitude of: $500,000; $4,000,000; $600,000; and $400,000. It hopes to do or have the capability to do two or three projects in the $5 million range with several smaller projects (less than a million) at the same time.

It needs to get set up now with (1) the hardware, (2) the software, and (3) the personnel to be able to handle this workload. Several years ago we examined the IBM 34 for general contractors. It looked good but (1) it was expensive; (2) IBM appeared to continue to own you; (3) at the time, we were too busy and spread too thin to change.

When Radio Shack introduced the Model 16, I decided to jump and hope for the best. With the memory and ability to have two additional stations and to network with additional Model 16s and the existence of some Model II software available from at least two known outside sources, it appeared to be an affordable gamble.

I have therefore very recently (June) purchased a Model 16 with 2 disk drives and a hard disk Winchester, Model IV Printer and one dumb terminal with Scripsit and dictionary, Job Costing, and Profile Plus software and have ordered VisiCalc for Model II.

I have to make a decision this week (or soon) whether I (1) buy the main software needed (i.e., General Ledger, Accounts Payable, Job Cost Control, payroll and probably Job Estimating) from (a) Data Automation Services International, Inc. (DASI) out of Gainesville, Florida who specialize in software for the building industry, or from (b) Construction Data Control, Inc. out of Atlanta, Georgia, who also specialize in software for the construction industry or whether I (2) buy Radio Shack software and have a local programmer modify to make do or whether I (3) hire a local programmer to write the programs (Possibly for the 68000) or

whether I (4) locate any better source or alternative through some source (such as yourself).

As soon as possible I need to adopt a software system and either start training the people I have to use it or secure the proper person or persons to operate the system—if the Radio Shack 16 can be made to do the job well. If it cannot, then I need to get what will do the job, sell out what I bought, take the loss and move on. In any event, I want to get the necessary company control via computer as early as I can accomplish this, hopefully during 1982.

My present thinking . . . is to try to set up the Model 16 with 2 floppy disk drives and one hard disk and two dumb terminals in the office and then to add an additional Model 16 in the construction office of each major building site tied back to the main Model 16 with hard disk and any additional hard disks that may be necessary, by modems and telephone lines. This seems to me to be flexible, workable, and still provide the office with positive and immediate information. Most of these construction sites will be within 15 miles of the home office. One of them, however, will be 2,000 miles away in Ecuador. Here the information may be transferred by (1) modem and telephone lines, (2) possibly modem-phone patch to single side band radio, (3) by mailing copies of floppy disks.

In addition to the main use for a computer, it is hoped that it can be used as (1) property management tool for rental property, i.e., small shopping center and apartments; (2) word processing device; (3) creation of internal bookkeeping, thus reducing present cost of outside accountants; (4) general information file locating system; and (5) control of small mortgage company, i.e., amortization schedules, etc.

Any help or suggestions would be appreciated.

In my experience, courses, per se, are of little value. The only *way to learn anything useful about computers is to program useful work on them. Probably the best way to learn is to tackle specific business applications with high level languages (one language at a time until you are comfortable with each). In addition to the manual that comes with a language, you will need a good tutorial manual to help solve programming problems. In my opinion, the best manuals around for several of the high level languages (FORTRAN, COBOL, etc.) are those by Daniel McCracken. You should be able to find these in any good bookstore. The new bi-monthly magazine* DataCast, *available from Wireless Digital, Inc., 345 Swett Road, Woodside, CA 94062, for $18/year, is the best tutorial magazine I've seen. Each issue treats one or two subjects in great depth.*

It's very difficult to suggest specific courses of action with respect to building systems for the Model 16 until there are operating systems available for the 68000 CPU. However, in general, I believe that modifying Radio Shack programs (or any other "canned" programs) is a dangerous alternative. Modifications often take several times as long to perfect as just writing brand new programs. I've long felt that "doing it yourself" is the most efficient way of getting desired results from interactive processing in any specialized business, but this presupposes plenty of time and talent. The next best way is to hire someone who knows your business and has both the interest in computers and the problem solving ability required. Third best: hire a consulting firm which

has experience serving your business. But, however you do it, make sure that the programs can be "maintained," that is, modified to fit future changes as needed: (1) the programs should be written in an ANSI standard high level language, such as FORTRAN or COBOL, so that they can be read by any qualified programmer and can be transported to another model computer if necessary, (2) if an outside firm does the job or if you buy a package, you should be given the source code (in "machineable" form) so that you have control over program maintenance (and aren't forever at the mercy of the outsiders), and (3) get a firm price quote in writing from any outside programming source. Keep a log of all modifications to the initial specifications so that you can defend against cost overruns. Most contract programming costs much more than the estimates, partly due to unforseen changes made by the client (you must expect to pay for those) and partly due to ineptitude on the part of the contract programmer (you shouldn't have to pay for those). I've seen jobs that ran more than five times the original estimate, mostly due to programmer malfeasance, but were ultimately paid for entirely by the defenseless client.

Your general hardware plan seems well thought out and practical. Having had a little experience with attempts at international data transmission, I'd suggest that you follow your Ecuador alternative (3), mailing floppies, for the present. Let others experiment with exotica until it's more reliable.

There are a number of package offerings for each of your final five requirements. Radio Shack publishes an Application Sourcebook which lists such specialized software (unfortunately, the current edition is about 18 months old and the next isn't due until January, 1983). Digital Research, 160 Central Avenue, Pacific Grove, CA 93950, publishes a volume which lists CP/M based systems. Phase I Systems, 7700 Edgewater Drive, Oakland, CA 94621, publishes a similar list of applications which run under their OASIS operating system.

FROM: Vincent J. Brown, Sr. I.E., Zygo Corporation, Laurel Brook Road, P.O. Box 448, Middlefield, CT 06455:

COMMENTS FOR *two-sixteen:*

Since the first issue was mostly introductory, it should be free to subscribers.

I've only had a Model II for about 2 months and am using it in manufacturing. I'm setting up a production analysis system which will include punching in daily production data (P/N, time on/off, employee number, operation sequence, etc.) from which I'll derive total manufacturing time for each item as well as labor efficiency based on a standards file. I've not seen anything like this in any other publication. Would this be of interest to you?

I've also worked out ways of modifying the RS "Profile" software so I could correct errors without starting all over again.

I'm interested in just about anything dealing with the Model II and would be happy to act as a test site for new products, especially in the *manufacturing field* which seems to be a forgotten field.

The Industrial Engeering magazine has published various programs but they are extremely basic and simplistic.

Best wishes.

Free copies would be nice, but we are obliged to pay our employees, printers, etc., just like your business probably does.

Your production analysis system would probably be of considerable interest to some of our subscribers, especially if you can include program code. The code may be too extensive to print in its entirety, but, with your permission, we might make it available to readers in diskette form at nominal cost.

I'm sure that many of our readers would benefit from your Profile enhancements.

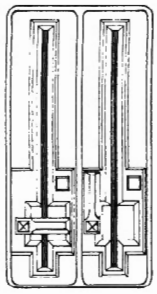
We see very little software listings that are directed primarily at manufacturers. Quite a few of our subscribers appear to be in manufacturing and perhaps some of them will write to you to discuss experiences.

FROM: C. H. Humphrey, 921 Lurline Dr., Foster City, CA:

SOME COMMENTS AND SUGGESTIONS FOR YOUR MAGAZINE:

—*two/sixteen* is a much needed magazine and will provide a much appreciated service if done properly. Go monthly ASAP.

—VERY IMPORTANT: As a general policy, don't knock RS products (hard or soft), even if they're not perfect. It's much more constructive to speak positively about the many strengths; offer corrections or help in circumventing weaknesses. Nobody likes a loser, and you won't help our cause if you constantly cast aspersions at RS (everybody else does that . . . let's be supporters for a change). After all, RS is "our computer company" and it's to



our advantage for them to have as large a customer base for the MOD II/16 as possible!

—Buy a Daisy Wheel printer such as RS DW-II. You will improve your image.

—When you include patches, explain them in more detail; what advantage do they offer? what disadvantage, if any? Questions not answered regarding those patches offered in your first issue:

—Will FORMAT be faster after patch 1?

—Will I be able to list the System after patch 2?

—Does patch 3 affect the external drives or just the internal one? After doing the patch will my drives access data faster?

—How much speed improvement is obtained by eliminating the logo? (ANS: Only a second or so—so why bother? Let's speed up the initialization and save 5–8 seconds.)

—Exactly what switches and questions are bypassed by patch 5?

—NEVER make mistakes in listing patches, especially System patches. (On pg. 8: OCFB should be 0CFB; OD12 should be 0D12).

—Get your typesetter to use slashes through zeros for the sake of readability.

—In fact, the RS S/W I have is quite good. Even the BASIC interpreter, by comparison, is a good one; your article notwithstanding. Weaknesses, if any, are usually attributable to there being functional improvements that could be made, as opposed to bugs, etc. This is partly because it's "tuned" to our machines. So FEATURE RS S/W. Scripsit II, Visicalc, Profile II are all superb products. Put them up as the criteria for judging the advantages (or disadvantages) of competing software. (Why review Micro-Pro's S/W *before* RS's software?)

—Devote more space to way to improve RS software to make it even better (rather than products to replace it with). Get readers to send in their list of desired improvements for a different Mod II software product each month.

—PROVIDE UTILITY PROGRAMS such as disassemblers, basic cross-ref, machine language sorts, disk zappers, etc. I know some of these can be purchased, but I'd rather save my money for application software or hardware improvements. Re: your question on the CP/M to TRSDOS converter: Yes, I'd pay something for it, but it's a utility that someone should donate to you for publication . . . free.

—Publish tutorials on how to get the most out of our computers; how to use the power of TRSDOS (e.g., how, when, and why to use the SVC's, timers, etc.!). The book *Basic Better and Faster* has lots of tricks, but many of them are comprised for the Models I and III. We Mod II users need this kind of literature for our systems.

Sponsor a disk-based assembly language course specific to the Mod II's special features and S/W.

—Establish a reader polling system with results being sent to RS (and published for the benefit of other Mfgs.) so that they can know what kinds of products or product improvements Mod II owners want. Example—Should RS provide:

—a battery supported clock/calendar?

—a programmable audio beeper or bell?

—a better keyboard (for high speed typing)?

—a hardware floating point arithmetic unit?

(My answer to all these is YES.)

—Review the new graphics adapter as soon as possible. Answer questions such as:

—Will it interface with a light pen?

—Will it interface with joy sticks?

—Let's find out the truth about the green-phosphor screen. Is it really easier on the user? Can the standard MOD II CRT be directly replaced with a green one?

—YES to your question about the information exchange. I'm sure it would be worth $20/yr.

SPECIFIC QUESTIONS FOR WHICH I'D LIKE ANSWERS:

—Is there a mod to hardware or software that would prevent a disk crash if the internal disk is inadvertently read with the external drive's power off? This is one of the biggest shortcomings in the MOD II system.

—My Wordstar is version 2.6; my CP/M is Pickles & Trout's 2.2e:

—Has anyone got a "good" installation of WS for MOD II? one that uses the special function and cursor control keys?

—Has anyone got a patch that allows WS to use the DW II's special features instead of handling it like a backspacable TTY?

SOME COMMENTS ON MY ANSWERS TO YOUR READER SURVEY:

—I have had my system for about 1½ years without a single failure of any hardware or software. That's partly why I rate the CPU and Drives a 9. They would get a 10 except MOD II should have a better keyboard and the outboard drives should have in-use lights. I think the machine is highly underrated by the media.

—Daisy Wheel II would be a 10 except it isn't bi-directional. It is otherwise a great printer and its speed isn't bad anyway. HINT: The ribbons can be rewound several times.

—Scripsit 2.0 is an excellent word processor—better than WS for several reasons in my opinion. I rated it 8 because it still has room for improvement. (It doesn't properly justify proportionally spaced characters; doesn't do multi-column formats well; needs graphics capability for form and protected fields for fill-in-the-blanks formats.)

—Scripsit Dictionary is great (fast, large, easy) but it would be better still if it would help with the correct spelling in many cases.

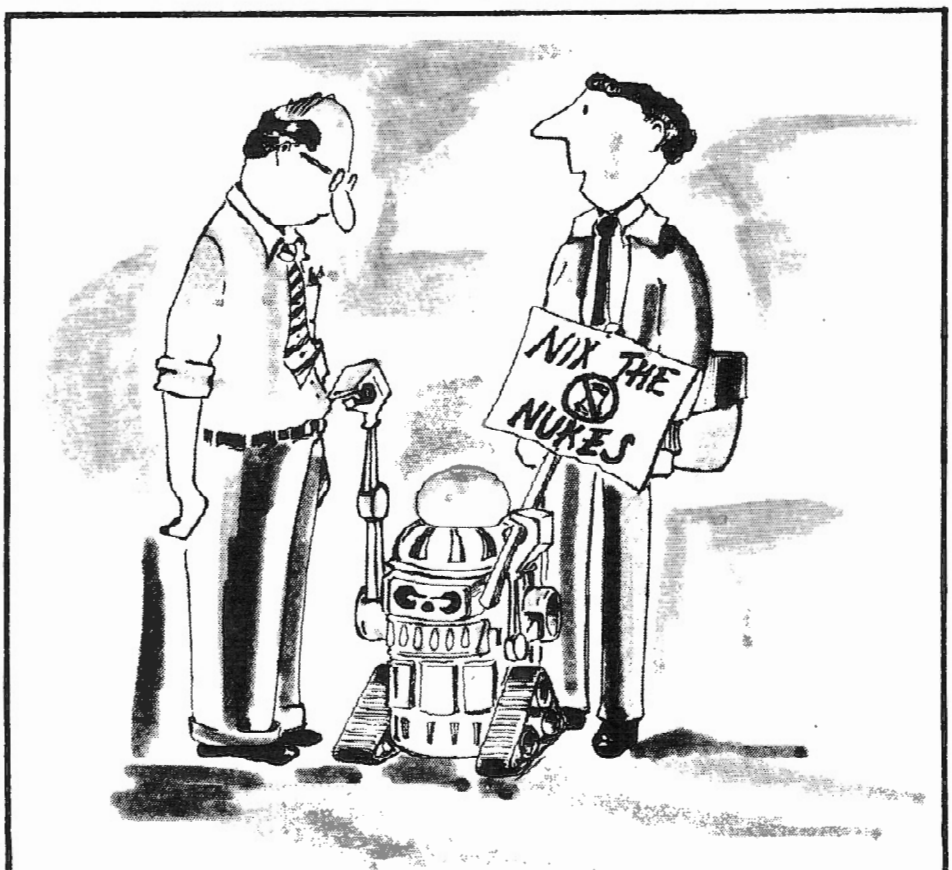
—Visicalc is excellent (my most useful program), but there is room for many improvements—some have already been made for the MOD III version; why not for MOD II?

—Profile II leaves a lot to be desired. It is frustratingly slow for some things and lacks flexibility. The documentation is the poorest I've seen from RS. Nevertheless, it has still proven very useful to me in my business.

—CP/M is faster than TRSDOS but TRSDOS has many commands and utilities not available in CP/M and is "tuned" specifically to the machine and the Z-80A.

—When WS can do a simple thing like go right to a specific page, I'll give it a couple more points.

I agree that Radio Shack sometimes gets unfair knocks, especially in the careless repeating of unfounded and damaging rumors, such as those about the discontinuance of successful products. We will try to avoid this. However, we don't intend to give Radio Shack any kind of "preferred nation" status. Our readers are entitled to know both the



Where was it programmed? University of California at Berkeley. Why?

good and the bad about any product. We intend to publish all non-libelous opinions expressed by readers in letters to us. Most readers are very pleased with the performance of Model II microcomputer hardware, but many are equally displeased with the limitations imposed when using TRSDOS-only software. The Model II is very likely the best CP/M business computer available, and many readers wish that Radio Shack would lend support to CP/M (and UNIX, as well), so that the vast repertoire of available programs could be enjoyed under the Radio Shack banner. Where negative comments are necessary, they will be made. The purpose is two-fold: to inform readers of other opinions and to encourage Radio Shack to broaden their support. Many of us believe that broadened support would result in several times as many sales of Model II and 16 micros, resulting in greater profits for RS and a much richer environment for all of us. We intend that all of our remakrs have constructive effects.

We have a magnificient NEC 7725, which has given our image about all the improvement it can stand. I recommend it to all who want to achieve genuinely professional results for a relatively small additional cost.

People who develop high grade reliable software have little incentive to give it away. In more than fifteen years of heavy involvement with interactive computing, I've seen only a few free *programs that were worth the trouble of loading. Good software requires incredible effort by highly talented people (I'd guess that there are several man years of work, preceded by many years of experience, in a product like the TRSDOS-CP/M ReformaTTer). Developers are entitled to a fair return.*

We continue to solicit readers for contributions, including tutorials. Almost every user is a relative expert on some aspect of Model II or 16 use, and we encourage you to give our other readers the benefit of your own experiences.

We'd like to see a Model II-specific course in assembler. I've seen such courses advertised for other computers and seem to recall a price in the $100 range. Probably 200 or so could be ultimately sold, resulting in a gross of $20,000. The developer would probably get about 20%, or $4000, after expenses. Do we have a reader who would like to take this on?

There is an ad in the current issue for Pickles & Trout's TRISWATCH, which provides an eight-year-battery supported clock AND a beeper. It functions under both TRSDOS and CP/M. W bought one the instant our Model I warranty expired last year, and we'll put one in our 16 as soon as we can safely open its case. It's a bargain at $175. Hayes also has a clock, but it uses up one of the scarce RS-232 ports. With these devices available, it would be reasonable for RS to feel that it's not necessary for them to provide another device in this rather thin market.

We're trying to find out more about the graphics mod. So far as we know, none have been installed as yet. Nothing has been heard about possible light pen or joystick use, but it's certainly a theoretical possibility. All that is required is a hardware light pen or joystick that connects to a RS-232 port and software (in the form of callable subroutines) that will obtain data from the port. It seems likely that the number of graphics units sold will not be sufficient to warrant the cost of developing this hardware and software.

Our Model 16 green phosphor screen is much sharper than our Model II screen, but it seems to be a matter of personal taste as to which is preferable. I much prefer the 16's screen, but on of our local RS CC salesmen is just as stuck on the Model II screen. RS will not retrofit green phosphor screens to Model IIs.

We don't know of a mod to prevent the dread BDOS error and destruction of the first two or three files which occurs when you boot with expansion drive(s) off. However, one simple solution (which I'm sorry to admit I've not yet done) is to buy a simple electrical cable with switched outlets so that a single switch will turn on both the drives and the computer (whose switches are left on all the time). Once upon a time, there were solenoid switch boxes which would turn off a hi-fi system when the turntable shut itself down (this should work OK if computer = turntable and drive = receiver). If anyone knows where these can be found nowadays, please let us know.

You may be able to modify WordStar to use the unique Model II keys, but you will probably need the WordStar Customization Notes to do it. At $429 (discounted), this is a high price to pay just to make your WordStar nonstandard.

If you know what special character the DW II requires, you may be able to patch the printer area of WordStar (following the same approach we used for the Epson) to make use of at least some of DW II's special features.

Yours is the first complaint we've

heard about the keyboard. Please let us know what specifically you would like to see changed.

FROM: William Strating, CPA, Small Business Computer Associates, 126 East 39th Street, Holland, MI 49423:

The first issue of *two/sixteen* has arrived. If your perception of the "composite reader" is indicative of your marketing skill, and that skill is in turn indicative of the balance of your publishing skills, I predict that you have a huge success on your hands if you stick to your commitment to give us "immediate usable information."

Since I so strongly identify with your "composite", it will serve both you and me well to pass along my first reactions to your magazine.

The format of the magazine and the individual articles is excellent. Don't change it just to get fancy; do change it if you think you are improving it. 1., 2., 3., and A., B., C. is hard to improve on for business oriented readers.

You have reserved the right to determine ad headings. I wonder why? I will venture a guess and offer the generalization that you believe that "classification" has a lot to offer "clarification." If that is true, would you give consideration to classifying your published material under these headings (add others if necessary):

a. Programming & System Development
b. Business Applications

It is often difficult for me to understand if a product or an idea is developed to aid programmers or as a business application. Perhaps an indexing scheme would serve the purpose. I have an immense interest in matters relating to business applications and practically no interest in matters relating to programming.

Try to provide a reprint service for certain articles. You will, of course, have to trust your judgment. The only input that I can offer is that small business owners and managers have a lot of respect for the success and failure of their own breed. Articles dealing with specific business experiences should have a wide interest. When you publish them, consider making reprints available.

If you are considering a column which will comment upon and report material and events from the viewpoint of a chief accounting officer of a small business, I offer my services:

a. William H. Strating
b. 126 East 39th Street
c. Holland, MI 49423
d. My work experience includes a staff accountant position in a CPA firm where I functioned as an in-charge accountant. From that position, I went into a controllership of a newly organized small business that in 20 years has grown to a sales volume of 20 million plus. I left that position to start my own business. Two years ago I was introduced to the Model I and for the past 15 months I have been installing Model II (and some Model III) systems in a variety of businesses including manufacturing, meat packing, services, distribution, retailing, etc.
e. My interest is to encourage the development of specific application packages for business. This can only be accomplished through a "clearing house" concept where the seller and the buyer can meet. I believe your concept of an "information exchange" is a good one and that it will be successful. Since much of the reporting, costing, pricing and record keeping in small businesses is organized under the control of the accounting department, it makes sense that an "accountants column" will be complimentary to your information exchange and will enjoy wide readership interest. It would be my intent to make requests through the column from time to time for application programs that businesses need.
f. You as a publisher will need someone who is qualified in the field of small business accounting to consult with and help you evaluate requests made to your information exchange. Your columnist could serve in that capacity during the development stages.
g. Be sure that your evaluation of applicants stresses the importance of finding someone who understands the job to be done well enough to find the tools to do it; this being much more important than finding someone who understands the tool and then "shotguns" for jobs business wants done.

Good luck!! I am looking forward to a rewarding relationship with you either as a subscriber/columnist or as a subscriber only. Why don't you consider publishing a "personal" on Richard H. Young for your charter subscribers? I can't believe that I am the only one who would like to know more about you.

Our reservation of the right to set classified ad headings is simply to forestall the confusion produced when nearly every ad has its own special classification. I've seen a few magazines like that and I don't bother to read the ads. Also, we don't want to place ads that are clearly about software in a classification that is clearly hardware, etc. Most newspapers reserve this classification privilege, and it seems to work out all right. We intend to be reasonable about it.

We will make reprints available at reasonable cost. All of our magazine is on diskette before it is typeset, so we can provide a "reprint" in several alternate forms. In general, we will allow any other publication to reprint any article in two/sixteen, *providing they obtain written permission from us and indicate credit in the heading of the article and the contents page. Exceptions are articles whose copyright is held by others and printed by us with permission. Subscribers are herby given blanket permission to reproduce, for non-commercial use, any article originally copyrighted by us, providing credit is given to its author and to this magazine.*

I'd be delighted to consider you as a columnist in small business accounting. We have practicing accountants, in-house accountants, and several of the Big Eight firms among our subscribers (as a whole, the subscription list represents quality rather than quantity), so I believe that significant attention should be paid to accounting. I wholeheartedly agree with you that this kind of work should be done by people interested in the result instead of the means. Our main criterion for writers, though, is their ability to express ideas in a way that will convey the meaning clearly to the audience. This requires, of course, a sound comprehension of the audience (its present level of understanding and how to raise that level to the objective). So "the proof of the pudding is in the eating." We will judge potential columnists on the basis of their product more than on any prior experience.

I must agree with you that our subscribers are entitled to know something about their publisher. Such information might make it easier for subscribers to communicate with us. So here's a thumbnail sketch. (But this will probably be the last of this. There has been a tendency for some publishers to overdo personal history to the point of embarrassing readers.)

Richard H. Young

Age 49

Certified Public Accountant (Illinois, 1959)

Numerous insurance industry exam designations

Commercial Pilot, Instrument, Land & Sea

Primary previous employment:

Insurance financial executive for 26 years, working for companies (a total of five different entities) of medium and large size. Duties in all areas of finance and accounting, including controller, treasurer, and vice president of three of the smaller companies. Emphasis on financial analysis, especially acquisitions and other business valuing projects. Used timesharing computers extensively for about 12 years (primarily DEC KL-10). Have lived for more than one year in each of: Oregon, Illinois, Alaska, New York (city), Florida, Washington, New Hampshire, and Pennsylvania.

My orientation is toward accomplishing objectives more than toward corporate politics. Even though writing and publishing are not prime aptitudes, this work is much more rewarding (except financially, where we will be doing well to break even in one more year) than any possible work in large corporations. From letters and telephone conversations, I know that this general attitude is shared by many of our readers. May our tribe increase!

FROM: Joseph E. Boling, P. O. Box 16188, Indianapolis, IN 46216:

Reference your ad and the review of your journal in the May issue of *80 Microcomputing*. . . . I have been waiting a long time for a journal that addresses my interests. I run a Model II with P&T CP/M, three disks (the expansion disks are Lobo through P&T) and a C. Itoh Starwriter 45 printer. I intend to upgrade to the Model 16 and add a terminal if I can do that and stay compatible with the software and files I have. . . .

CHECK OUT ASPEN'S SPELLING CHECKER PROOFREADER

Barbara S. Albert

Aspen Software Company in Tijeras, New Mexico, designs and markets software especially for CP/M (it's also for the IBM Personal Computer but that doesn't concern us). Since we use CP/M almost exclusively here at *two/sixteen,* two of their self-proclaimed "state of the art word processing tools" found their way to my work-station for review: **Proofreader** and **Grammatik.** [To set the record straight right at the start, both **Proofreader** and **Grammatik** are trademarks of Aspen Software, and RANDOM HOUSE is a trademark of Random House, Inc.; the t.m. symbol is included by reference throughout this article.]

Proofreader is a spelling checker featuring the RANDOM HOUSE Dictionary. The version we received and the one most compatible with the Model II's capacity is 50,000 words, 100,000 bytes. The manual that accompanies **Proofreader** states several different times that "**Proofreader** is easy to use." And it is, very easy. The procedure required to proofread a document with **Proofreader** is so simple that you should read Aspen's documentation to get yourself "plugged in", but you need not do more than give it a quick once-over. The entire process is menu driven; you are guided through every step of the way so there is never a problem with "what do I do next?" Only two steps are required to check over your document: (1) Finding unknown words with **Proofreader** and (2) Correcting the document with **Proof-Edit** (part of **Proofreader**).

One feature of **Proofreader** eliminates a pet peeve of mine: the user prompts shown on the screen are unobtrusive; you don't feel as if you are getting a complete set of beginner's instructions every time you check a document. Of course you want to be able to figure out what's going on when you first use a new program, but after you've become accustomed to it, having to read the same directions over and over can become a real drag. **Proofreader** prompts you through the process, but you are only shown the directions after the first error. After that, you have to ask for the complete directions again if you forget.

The entire process of checking a document with **Proofreader** takes only a few minutes (less than four in most cases), but those four minutes can make a world of difference to someone who is a poor speller or a careless typist. Most anyone who is remotely conscientious about the appearance of his work knows the importance of an error-free document, whether it be a letter, report, manuscript, memo or whatever. **Proofreader** will certainly help you toward perfection, but it has limitations of which you should be aware before relying on it 100%.

THINGS TO LIKE ABOUT PROOFREADER

Let's start with the **Proofreader's** good features: First of all, as stated above, it's fast. If you ever try to proofread your own work, you know how tedious and slow this process can be. Proofreading your own work can also be very inefficient. In four quick minutes (or less), your document can be checked for unknown words, and you don't have to read your work again if you don't want to. My experience has always been that when I have been working closely with a particular piece of work for a long time, very often I won't see typing errors at all. I know what should be there and what I expect to see, so that's what I see. Lots of embarrassing mistakes slip through that way! Here is a feature you can hook up to your computer that will do away with the tedium of time-consuming and inefficient proofreading.

Second, as I also said above, it's easy. It's so easy, in fact, that I managed to figure out how to use it all by myself, and I have been known to have my problems. All you have to do to get started is invoke **Proofreader** at the command level and tell it the name of the file you want checked:

A> PROOFRDR b:filename

To save precious disk space, **Proofreader** and **Grammatik** are on a disk by themselves; this disk is put in Drive A. Then the WordStar document files are on their customary working disk that is put in Drive B.

The **Proofreader** manual points out:

WordStar users should note that Proofreader can be invoked from the "NO FILES" command menu of WordStar. Specify the "R" command to run a program, then enter "PROOFRDR FILE". . . . You may then check the file with Proofreader, correct it with Proof-Edit, and then return to WordStar.

Proofreader is quick, and it's easy—what more could you want?

Proofreader provides you with lots of options once it has checked your document against the RANDOM HOUSE dictionary. You can:

C — Correct file filename with Proof-Edit
D — Display unknown words on screen
E — Exit from Proofreader
G — Grammatik: check file with GMK
P — Print unknown words on printer
R — Review and edit unknown words
S — Save unknown words in file: filename.BWD

If you are interested in saving time, use the R option first: review and edit unknown words immediately after the document has been proofread. This feature shows you the unknown words, one at a time, in succession. After each word is shown on the screen, you are given the choice of allowing the unknown word to remain as unknown so you can correct it with **Proof-Edit** or eliminating the word as unknown. This command enables you to get rid of words that are really correct from the unknown word list. This is the most marvelous feature because you can save yourself the annoyance of having to tell **Proof-Edit** over and over that certain words are, in fact, not words at all but merely hyphenations or even WordStar dot printing commands. (More about that later—see the section "Things To Not Like"). After all the words that were unknown to **Proofreader** but not to you have been eliminated, proceed to **Proof-Edit** (choice C above).

Another feature I greatly appreciated appears in the **Proof-Edit** portion of **Proofreader.** After **Proofreader** has checked your document for unknown words, use **Proof-Edit** to locate and correct these unknown words. The way **Proof-Edit** does this is the nifty part: it scrolls through your entire document, stopping only when it encounters an unknown word. This way, you can see your error in context. Not only can you read the words that come before and after the questionable word, but you can read over the preceding sentence if you need to in order to figure out what the problem word is supposed to be. From the spelling checker reviews I have read in other computer periodicals, this feature apparently is not standard. How any software developer could think that you can tell what a misspelled or mistyped word is supposed to be when you just see it on the screen and not in any context is beyond me. Fortunately, the folks at Aspen had the foresight to realize that "ghjytuqe" out of context can probably be corrected only with difficulty.

I really like the AUXDICT.TXT feature of **Proofreader;** with it, you can create your own dictionary containing all the words you're never quite sure how to spell correctly. You can also add those words that are special to your own particular industry. The one here at *two/sixteen* is filled with words such as *microcomputing, Tandy, TRS-80, debug,* etc. This auxiliary dictionary will come in so handy so often, you'll wonder how you ever got along without it. **Proofreader** provides you with two methods of adding words to the Auxiliary Dictionary: (1) you can add those words that were marked as unknown by **Proofreader** during a proofreading session and that you commanded **Proof-Edit** to "learn for auxiliary dictionary;" or (2) you can create a separate file in WordStar filled with technical terms or words that you want to spell correctly once and for all. (The **Proofreader** manual provides more than adequate documentation on this feature; it's quite easy to do.)

The other command options offered by **Proofreader** probably have a use for someone, but I wasn't able to determine any practical purpose for them. D/Display Unknown Words on Screen; P/Print Unknown Words on Printer; and S/Save Unknown Words in File: filename.BWD (the "bwd" extension stands for "bad words") all do essentially the same thing—show you a list of the words **Proofreader** found as unknown. Perhaps a school teacher would find some value in a separate list of misspelled words, but for me and the type of work we do here (writing and editing other people's writing), these features were extraneous.

THINGS TO NOT LIKE ABOUT PROOFREADER

To segue right into this section from the last, I don't like **Proofreader** automatically creating a file called filename.BWD, and it does this when you use **Proof-Edit.** (The "Bad Words" file is also created when you select option G/check file with Grammatik.) Who needs a file full of errors? I am reminded often enough of mistakes I make in this world; I certainly don't need this. Also, when working on a disk that is getting full, you do not need to be worrying if this "Bad Word" file is going to send you over the top. The automatic creation of a "Bad Words" file is a feature I would recommend be eliminated.

Aspen's **Proofreader** Manual dismisses in two sentences a problem I consider important: "Since every word processor handles hyphenation differently, **Proofreader** makes no assumptions about hyphenation, and should be used before the document is hyphenated. If a previously hyphenated document is checked by **Proofreader,** each half of a split word will be checked independently." The fact that **Proofreader** cannot distinguish when a word is hyphenated represents a major short-coming in the software design. Maybe all word processors handle hyphenation differently, but word wrap and text reformatting are all rather standard features these days, and both incorporate hyphenation. I think **Proofreader** should be programmed to recognize a hyphen at the end of a line, search for the second half of the word at the beginning of the next line, merge the two parts together, and then analyze the word. Since it does all kinds of other wonderful things, it should be able to do this too.

And another thing: The people who designed **Proofreader** obviously did so with WordStar in mind. There is a special command for WordStar users built into the program. If you use WordStar, a separate mode checks your document with **Proof-Edit.** After checking and editing has been completed, **Proof-Edit** will tell you how many words were changed in length; each change is marked in the text of your document with "#." Then you can exit from **Proof-Edit** and return to your document in WordStar, do a Control -QF to find #, eliminate the #s, and then Control -B to reform margins. Not bad. Because this feature exists, we know the Aspen programmers are aware of and presumably understand WordStar. So why won't **Proofreader** recognize WordStar dot commands?!?! I want to know why I was told repeatedly that ".po 5" is "??? UNKNOWN". And why ".op" is "??? UNKNOWN." This was truly my least favorite aspect of **Proofreader** until I resolved the problem and eliminated the annoyance by putting the WordStar dot commands into the Auxiliary Dictio-

nary. By using this method, I am safe from seeing them in the unknown words list every time. But I didn't think of doing this right away, and until I did, I found this weakness extremely annoying.

WordStar dot commands are not the only feature of WordStar that is incompatible with **Proofreader. Proofreader** gets hung up on "soft hyphens" imbedded in the text. As any WordStar user knows, these soft hyphens do not affect the margin widths, nor do they show up in hard copy text if they are found in the middle of the paragraph due to subsequent paragraph re-forming. But nobody bothered to program this information into **Proofreader;** consequently, the word checker gets hung up on soft hyphens every time. This can really get to you after awhile. The only way I can see to get around this is to do a global search for soft hyphens (use Control -OE) and eliminate them all before running the document through **Proofreader.**

Here's another teeny-weeny complaint, but one that bears reporting to any potential purchaser: the dictionary is very limited. When using a regular dictionary, you take for granted that every form of the word you are looking for will be present. For example, the other day **Proofreader** told me it didn't recognize the word "speller." I asked for dictionary assistance, and the dictionary showed me that it has "spell," "spells," "spelled," and "spelling" but not "speller." And there are other common words that aren't included, such as "kangaroo." I understand that it's physically impossible to have *Webster's Unabridged Dictionary* in disk form, but sometimes I do wish for a larger one with **Proofreader.**

TESTING A DOCUMENT WITH PROOFREADER

Now for a little fun. Rather than check yet another boring business type document for my review, I decided to see how **Proofreader** went for some "real" literature.

Before beginning my discussion of test results, let me state for the record that I have, in fact, subjected many, many business documents (my own and those of others) to the test with **Proofreader.** I would not attempt to publish the results of what many will probably consider a "frivolous" exercise if these results were not 100% compatible with those achieved when testing "serious" business documents.

I was reading Thomas Hardy's novel Return of the Native at the time we decided to conduct the test. The first chapter of this book is rather well-known for Hardy's treatment of part of the landscape—Egdon Heath—as one of the major characters. His description of the Heath is the subject of my test, so here is Book I, Chapter I, Paragraphs 1 and 2 of Return of the Native. (These two test paragraphs have been reproduced here as they were entered into the computer document file; no effort was made to proofread the work for spelling or typographical errors.)

THE RETURN OF THE NATIVE
by
Thomas Hardy

Book First "THE THREE WOMEN'
Chapter 1: "A Face on Which Time Makes But Little Impression"

A Saturday afternoon in November was approaching the time of twidsflight, and the vast tract if unenclosed wild known as Egdon Heath embrowned itself mment by moment. overhead the hollow stretch of whitish cloud shvtting out the skY was as a tent which had the whole heath fsfdweor its floor.

THe heaven being spread with this pallid screen and the earth with the darkest vegetation, their meeting-linc at the horizon was clearkly marked. In such contrast the heath wore the appearance of an installment of night which had taken up its place before itdss astronomical hour was come: darkness had to a great extent arrived hereon, while day stood distinct in the sky. Looking upwards, a furze-cutter would have been inclined to continue worsk; looking down, he would have decided to finish his faggot and go home. The distgfdant rims of the world and of the firmament seemed to be a division in time no less than a division in matter. The face of the heat by its mere complexion added half ansdd hour to evening; it could in like manner retard the dawn, sadden noon, anticipate the frowning of storms scarcely generated, an d intensify the opacity of a moonless midnight to a cause of shaking and dread.

To begin proofreading, invoke **Proofreader** at the command level and tell it to check the file test.txt:

B> PROOFRDR b:test.txt

Proofreader moves through its functions in logical progression:

"—Reading File"
"—Sorting File: Number of Different words Found 155"
"—Checking Master Dictionary"
"—Checking Auxiliary Dictionary"
"Number of Unknown Words Found: 19"

For **Proofreader** to check a 155 word file, it took 1 minute and 20 seconds, and it found 19 errors. For purposes of this test, I asked **Proofreader** to "P/Print Unknown Words on Printer," and here they are:

ance ansdd ated clearkly distgfdant egdon embrowned fsfdweor furze gener heath hereon itdss mment moonless shvtting twidsflight whitish worsk

A quick glance over this list shows that I recognize some of these words, but obviously **Proofreader** did not. Note: **Proofreader** does not distinguish between capital letters and lower case letters. If this is important for your use of a spelling checker, **Proofreader** isn't for you.

You can also see by looking over this list that **Proofreader** puts this unknown words list into alphabetical order; the words are not sent to the printer in the order in which they appear in the text.

Thus, the hyphenated word "gener-ated" shows up in two parts on the list: "ated" comes first and further down the list we find "gener." This can be confusing.

After the unknown words were printed, I asked for "R/Review and Edit Unknown Words." By using this feature, I can eliminate some of the garbage on that list. For instance, I know that "ance" is half of a hyphenated word so during this "Review and Edit" session, I tell **Proof-Edit** that I do not wish to keep "ance" as an unknown word. The word "ansdd" appears as if it might be a legitimate error so I tell **Proof-Edit** to keep it. "Ated," and "gener" are also hyphenations, so they are eliminated. "Clearkly," "distgfdant," "fsfdweor," "itdss," "mment," "shvtting," "twidsflight," and "worsk" are all words I wish to keep as unknown; I recognize my typing when I see it! Those words remaining on the list are words that are "real" words; however, they are not included in **Proofreader**'s dictionary: "egdon," "embrowned," "furze," "heath," "moonless," and "whitish" are all words that I recognize, but **Proofreader** does not. Since I am sure of the spelling on these words, I eliminate them from the unknown word list. However, should you have questions about spelling, keep the words on the Unknown List and then check them when you move on to **Proof-Edit.**

When you ask for **Proof-Edit** (C/Correct file test.txt with Proof-Edit), you will be given three additional options:

C: Correct unknown words interactively
M: Mark unknown words in file with #
W: WordStar mode: correct + mark if change width

I use the W/Wordstar mode and find it to be quite helpful. Every time a word is changed in your text, **Proof-Edit** marks the change with #. When you are finished checked and correcting, you can go back into WordStar, remove all the #s, and reform the paragraphs that were changed.

At this point in the test, we're in the Proof-Edit/WordStar mode. This feature scrolls through the text and stops at words that have been marked "unknown" by **Proofreader.** The first word stopped at is "twidsflight." (I am rather surprised that the first unknown word is not "Thomas." A check into the dictionary [using "LOOKUP", another part of the **Proofreader** package] shows us that, in fact, proper names are included. When checking for "Thomas," I also found "Thompson" so common last names are included, too.) **Proof-Edit** provides several options for correcting this unknown word:

A — Accept word for rest of session—don't learn
C — Correct word, prompt will follow
D — Dictionary Help. Find word in dictionary
E — Exit—Accept word and discontinue checking
L — Learn word for auxiliary dictionary
O — One time acceptance of word

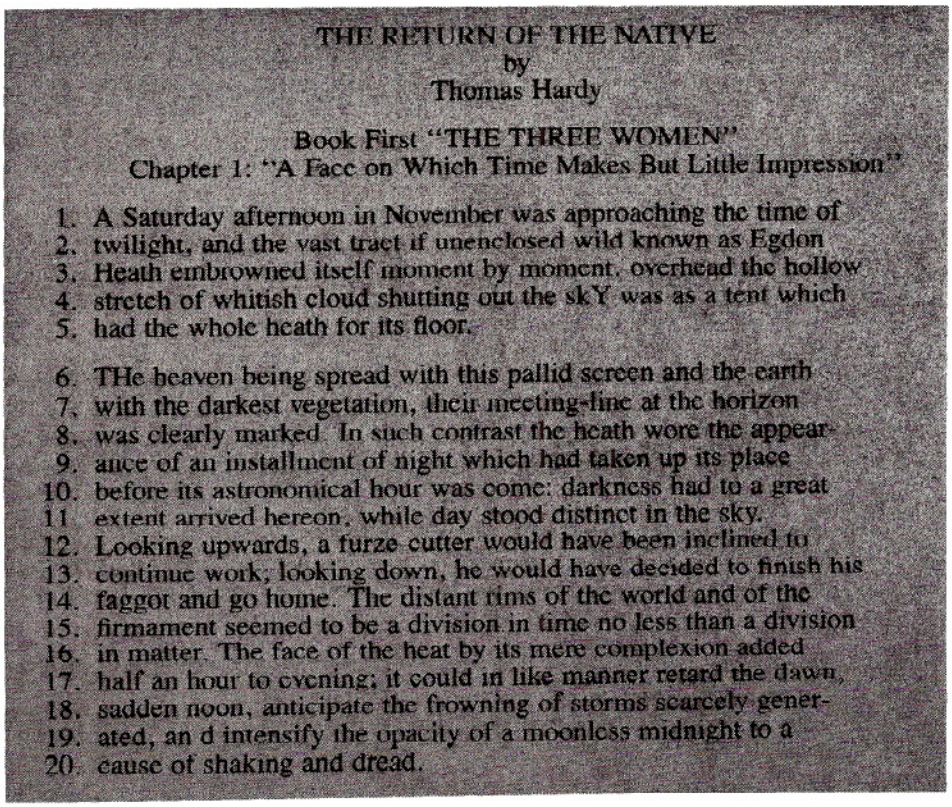
This list of options is only displayed the first time **Proof-Edit** encounters an unknown word. Thereafter, you must ask for H/Help and **Proof-Edit** will once again show you this menu.

The word "twidsflight" reminds me that I really meant to type "twilight" so I press C/Correct Word. **Proof-Edit** tells me to: "Enter Exact Replacement." I carefully type in "twilight" and move onto the next unknown word: "mment." Once again, I chose "C" and replace the error with the correct word: "moment."

If you make a typing error when entering the "exact replacement" or if you enter a word that is not in the dictionary, you will get a loud "BEEP" from the computer and a warning from **Proof-Edit:** "*** REPLACEMENT NOT IN DICTIONARY!" You do receive a chance to redeem yourself; all is not lost.

After several more corrections of this type, we come to "ance", the second half of the hyphenated word "appear-ance." You will notice that **Proofreader** recognized "appear", but "ance" got flagged as an unknown. We know that "ance" is correct and we want to use it, so in this instance, choose option O/ One time acceptance of the word. If this word were going to appear again anywhere in the document, you should use A/Accept word for rest of session—don't learn.

The word "moonless" shows up as unknown. That's hard to believe, so I ask for D/Dictionary Help to check on it. The Dictionary Help shows: "moon," "moonbeam," "moonbeams," "mooned," "mooning," "moonlight," "moonlighted," "moonlighter," "moonlighting," "moonlights," "moonlit," "moons," "moonshine," "moonshiner," "moonstruck," "moonwalk," and "moonwalks," but not "moonless". Since I am fairly certain that "moonless" is, in fact, a legitimate word, I once again hit O/One time acceptance of word.


THE RETURN OF THE NATIVE
by
Thomas Hardy

Book First "THE THREE WOMEN"
Chapter 1: "A Face on Which Time Makes But Little Impression"

1. A Saturday afternoon in November was approaching the time of
2. twilight, and the vast tract if unenclosed wild known as Egdon
3. Heath embrowned itself moment by moment. overhead the hollow
4. stretch of whitish cloud shutting out the skY was as a tent which
5. had the whole heath for its floor.

6. THe heaven being spread with this pallid screen and the earth
7. with the darkest vegetation, their meeting-line at the horizon
8. was clearly marked. In such contrast the heath wore the appear-
9. ance of an installment of night which had taken up its place
10. before its astronomical hour was come: darkness had to a great
11. extent arrived hereon, while day stood distinct in the sky.
12. Looking upwards, a furze cutter would have been inclined to
13. continue work; looking down, he would have decided to finish his
14. faggot and go home. The distant rims of the world and of the
15. firmament seemed to be a division in time no less than a division
16. in matter. The face of the heat by its mere complexion added
17. half an hour to evening; it could in like manner retard the dawn,
18. sadden noon, anticipate the frowning of storms scarcely gener-
19. ated, an d intensify the opacity of a moonless midnight to a
20. cause of shaking and dread.

When the last word has been corrected, **Proof-Edit** will tell you that you have reached

END OF FILE
—WS Correction Mode—
8 words changed width and were marked with #. Be sure to go back and remove the marks and rejustify text as needed. Proof.Edit finished.
Final options:

D to delete bad word file test.BWD
E to exit to CP/M
G to check B:test.txt with Grammatik

I always delete the bad word file because I haven't as yet found a need for it. Then I move on to choice G/check file with Grammatik. This document-checker is alot of fun; we'll review it separately. For now, I'll simply E/Exit to CP/M and give the thoroughly corrected document one last check.

As you will notice, this work is far from perfect despite our extensive and careful checking with **Proofreader.**

In Line 2, "the vast tract if unenclosed wild" should be "the vast tract *of* unenclosed wild." **Proofreader** was unable to detect this as an error because "if" is a legitimate word. Aspen's Manual states: ". . . words are only checked to see if they are in the dictionary. **Proofreader** cannot find mistakes such as 'from' misspelled as 'form'."

In Line 16 another error of this type occurs: "The face of the heat. . ." should be "The face of the heath. . . ." These are the types of errors that no word-checking system is capable of catching. Until such time as a software program is developed that can actually *understand* the writer's implied intent and his stated meaning, errors like those in the example will have to be corrected by a human.

In Line 3, the first word of the sentence, "overhead," is not capitalized. Since **Proofreader** treats upper and lower case letters the same, there is no way for it to detect an error such as this one—or the next error in Line 4 ("skY") or the next one in Line 6 ("THe").

In Line 19 "an d" represents yet another class of errors undetected and undetectable by **Proofreader.** "An" is certainly a recognizable word, and "d" as a "one letter word" is considered correct. Once again, this error is beyond the scope of a spelling checker.

OTHER PROOFREADER FEATURES

Proofreader contains a feature that allows you to add words to or delete words from the Master Dictionary. You can also add to or delete from your Auxiliary Dictionary. If you wanted to, you could merge the two dictionaries together to create a third new and distinct entity.

Proofreader can be used as a regular dictionary—separate from its spelling-checking abilities. Invoke the dictionary at the command level by typing in

A> LOOKUP

and then entering the word you want to look up. **Proofreader** will display the 20 or so words found in the dictionary near the word you are seeking. For example:

Looking for: computer

compulsory	compunction	compunctions
computability	computable	computation
computations	compute	computed
computer	computerization	computerize
computerized	computerizes	computerizing
computers	computes	computing
comrade	comradely	comrades

This LOOKUP feature can be very useful if you don't have the space to store a dictionary near your work area, or if you prefer to have the computer do as much of your work as possible.

THE FINAL WORD

The final word on **Proofreader:** it's not perfect, but it is helpful. It will never replace a human's ability to recognize and indentify what words mean and thus whether they are correct or not in a given context. But it is a place to start with the job of catching and correcting errors in typed work.

THIS MONTH FEATURING

SemiDisk Disk Emulator

Thomas Evan Keiser
Attorney-At-Law
3020 Sheffield Drive
Norristown, PA 19403

We are pleased to introduce a new column: Special User Report ***which is slated to continue as a regular feature. The*** Special User Report ***column will present a subscriber's opinion concerning a product he/she is particularly excited about (or perhaps disgusted with).***

Our intention for publishing this column is to provide a forum for you, our readers, to contribute to the effort of circulating as much information about the Models II and 16 as possible.

If you are interested in contributing to the Special User Report, ***submit your opinions in regular letter form, and please, keep the length to between 2-5 typewritten pages. We look forward to hearing from you!***

I have recently purchased [from SemiDisk Systems, P.O. Box GG, Beaverton, Oregon 97075] a 512k SemiDisk Disk Emulator for my Model II and would like to describe the experience for you and your readers.

First, some background information: I run a small service bureau on a part-time basis (one client in the bakery business) processing route delivery information and generating invoices, inventory reports, payroll and commission reports and other statistical data. Although I am assuredly not a programmer (I find BASIC perplexing), I have been able to create all the software for this endeavor using Quic-N-Easi (tm) from Standard Microsystems, 136 Granite Hill Court, Langhorn, PA 19047. My software now occupies about 400k on CP/M and is ideally suited for use with the SemiDisk for several reasons, discussed below.

Typically, my weekly data file contains about 1000 sales transactions of 128 bytes each, or 128k total. These are organized by Quic-N-Easi in an indexed sequential file which is keyed according to store number and date—12 characters in all. Thus, any application program can step through this file and read transactions from Monday, Store A00000 to Friday, Store Z99999 regardless of the order in which the data were entered and without any need for sorting. Similarly, any day's delivery may be instantly retrieved by merely specifying the store number and the date. Quic-N-Easi also permitted design of a CRT screen format to match the paper transaction slip and several coding and cross-checking features which allow the operator actually to key in a transaction in less than four seconds.

Herein lies the rub: after the data file reaches a size of 60k or more, keying the data begins to take less time than writing it to the disk. This problem is compounded by the fact that my program reads a customer file and brings certain information onto the screen as part of the keying process; thus, there is both a read and a write associated with each transaction.

I struggled with this problem (and a cranky operator who wanted to go fast!) for several months. Finally, I opted for keying the data onto a simple sequential file and transferring it, record by record, onto an indexed file at week's end. Although this pleased my operator by making disk I/O faster than she, it added an extra step to my busy Saturday processing schedule and prevented any mid-week search of the file. A hard disk would have solved the problem, but at a cost far exceeding its benefit.

Finally, I discovered an ad for SemiDisk and sent for the user's manual. After reading it several times and asking friends what they thought, I decided to chance an order even though parts of the manual were Greek to me and despite a clear implication that the unit would never perform properly unless the user rewrote the BIOS for his version of CP/M.

Upon receiving the SemiDisk circuit board, I was surprised to find no detailed pictorial instructions for its installation. However, upon removing the two screws that secure the top cover of the Model II, the problem immediately resolved itself: there, in plain sight, were a number of similarly shaped circuit boards, each plugged into slots in a "mother board" and into which the SemiDisk was obviously designed to go.

After booting up CP/M, I typed in the command

SEMIDISK.COM

to enable the so-called self-installing driver. This command contains several options, including the name by which CP/M will address the SemiDisk (I use "E:"), the size of the SemiDisk and the "switches" for certain software functions such as automatic format of the SemiDisk on cold boot, etc.

After entering this command, the message

Zero, are you sure (YES, NO)

appeared on the screen. I later learned that this was to permit a choice of formatting the semi-disk on cold boot (YES) or skipping this step (NO), but at the time I wasn't sure who Zero was.

The message

Please wait

then appeared on the screen for about five seconds followed by

SemiDisk Active E:

I was pleasantly surprised to find that DIR and STAT commands addressed to drive E: came back with appropriate responses, and even more surprised at the speed at which "PIP E:=A:*.* [VO]" worked.

Next, because my SemiDisk had arrived in mid-week, I decided to try converting my Monday and Tuesday transaction file from its sequential format onto an Indexed Sequential file on the SemiDisk. Normally, this would require reads from the sequential file alternated with writes to the indexed file and would take about 20 to 30 minutes. However, with both files located on the SemiDisk, the job took an incredible 2 and ½ minutes!

Similar time savings have been achieved on other programs, depending upon how much disk access is required. In any case, my operator is delighted (disk I/O is lightning fast now), and I am as well because my Saturday processing job has been really cut down to size. Now if anyone can sell me a reasonably priced, rugged 200-400 lpm line printer, I'll be able to take up golf on Saturday afternoons.

The SemiDisk System brochure states: "An added advantage of the SemiDisk is the capability of battery back-up operation. With the low power consumption of the SemiDisk, it's easy to provide power-fail protection." Thus, SemiDisk data may be protected against power failure by connecting a gel-cell to two terminals on the circuit board. The gel-cell is then charged by the Model II's power supply during computer operation. (Note: the battery for power-fail protection is not included in the purchase price of the SemiDisk System. Jim Beara of SemiDisk Systems told *two-sixteen* that the battery for back-up protection is not included in the SemiDisk package; only the capability for battery back-up is built in. The reason for this, he explained, is that everyone has different back-up protection needs so the size of the battery to be inserted is left up to the user.)

All in all, I am quite pleased with my purchase although it is apparent the price will come down as the price of 64k chips declines. Nonetheless, even at today's price, SemiDisk is far cheaper and considerably faster than a hard disk; also it is not wearing itself out every revolution and has no head to "crash." In fact, the speed and flexibility of SemiDisk with Model II and existing CP/M based software such as Quic-N-Easi may well make this combination superior to anything available for the Model 16 for years to come. Personally, I'm planning to purchase another SemiDisk for my backup Model II rather than upgrade to, or purchase, Model 16s.

HOW TO TELL WHEN IT'S TIME TO SPLIT

Barbara S. Albert

One day as I was saving a file (with WordStar ^KD), a message flashed across the top of the screen:

***WARNING: DISK FULL

Since I had never seen this message before, my initial reaction was to ignore it and hope it would go away because I didn't know what to do about it.

However, I had an idea what that message meant, and I didn't want to run the risk of losing any of my precious data. To find out what was going on, I did what I always do when I encounter an unknown situation. I turned to my guide and mentor, Dick Young: "Help!" This month's Novice Column is the result of his sage advice and wise counsel as he helped me through another of

BARBARA'S ADVENTURES IN COMPUTERLAND

"Speak English!" said the Eaglet. "I don't know the meaning of half those long words, and what's more, I don't believe you do either!"

The first thing he told me to do when the "Disk Full" message came through was check for back-up files (Filename.BAK) in the directory. These back-up files are automatically created when a file is saved in WordStar using ^KD or ^KS or ^KX. As you probably already know, back-up files can't be edited, if you try, you'll get the message:

Can't edit a file of type .BAK or .$$$
—rEname or cOpy before editing.

Back-ups come in handy if you accidentally erase the original file, leaving only the back-up*, or if you somehow damage an important part of your file and don't discover the error until a subsequent edit**. Usually, the "Disk Full" message is transmitted while saving a file because the .BAK file takes up exactly as much disk space as the original file. So if you are working close to the edge, a ^KS, ^KD or ^KX will probably result in the "DISK FULL" warning.

*I do this occasionally when I mean to erase the back-up. I press "y" to delete at the WordStar No-File Menu and instead of typing in Filename.BAK, I'll accidentally type in Filename.ART (".ART" is the extension I use for "articles"). There are two ways to correct this problem. The first is to copy the .BAK file using "O" at the No-File Menu. WordStar will ask you: "Name of file to copy from?" Type in:

filename.BAK

Then it will ask you: "Name of file to copy to?" and you enter the name of the original file that was accidentally erased:

filename.ART

The other way to resurrect your accidentally erased file is to rename the .BAK file using "E", also at the No-File menu. WordStar will ask you: "Name of file to Rename?" Answer:

filename.BAK

Then you will be asked: "New name?" Type in:

filename.ART

Knowing this technique can save you some anxious moments if you accidentally erase your original file instead of the back-up.

**What would you do if you called up an important document file and realized that you had inadvertently erased an essential section during the last edit? You went to look at page 3, and it wasn't there! There is a way to salvage this work. Quit the damaged file using ^KQ, and copy ("O") or rename ("E") the .BAK file at the No-File Menu. You cannot rename or copy to the original file name; you'll have to change it to something different. Then delete ("Y") the damaged file. Your "missing" page should be back in place in the new file, but this file now has a different name than your original. You could re-name it, but the procedure has suddenly gotten very involved.

A simpler way to resolve this situation is to use the CP/M command PIP. At the command level, enter

A>PIP filename.LTR = filename.BAK

PIP will over-write the damaged file (filename.LTR) with the data contained in the back-up file (filename.BAK), which file includes your accidentally misplaced information. No need to re-name; no need for four or five steps of procedure.

My directory showed at least four files with a .BAK extension, and Dick said, "Kill them."

"It sounded an excellent plan, no doubt, and very neatly and simply arranged; the only difficulty was, that she had not the smallest idea how to set about it;. . . ."

Here's an easy way to erase them all at once using CP/M. If you are in WordStar, exit to the command level. Type in:

A>ERA *.BAK

and that's all there is to it! (Don't forget the space between "era" and the asterisk.)

Even after eliminating the .BAK files, a check on the amount of space left on my disk (using STAT at the command level) showed that the disk was very close to being full.

A general rule-of-thumb—depending on the size of your disks and the application being used—for determining how full you should allow your disks to get is to stop adding new data when the disk becomes three-fourths full. When a 596K disk reaches 447K, that's enough.

The disk configuration I was using—CP/M, single sided, double density—holds 596K user-accessible characters (bytes), and I was up to 550K. Time to split.

BREAKING UP DOESN'T HAVE TO BE HARD TO DO

The very first thing you have to do before tackling the task of splitting a disk is back up the full disk exactly as it is. Take the backed-up disk and put it away someplace where no coffee or Coke will get spilled on it or where the dog and/or the kids can't get at it. You'll be thankful later! This is a crucial safety measure designed to save your work (and your neck) in case the break-up doesn't go as planned. (See Fig. 1.)

FIGURE 1

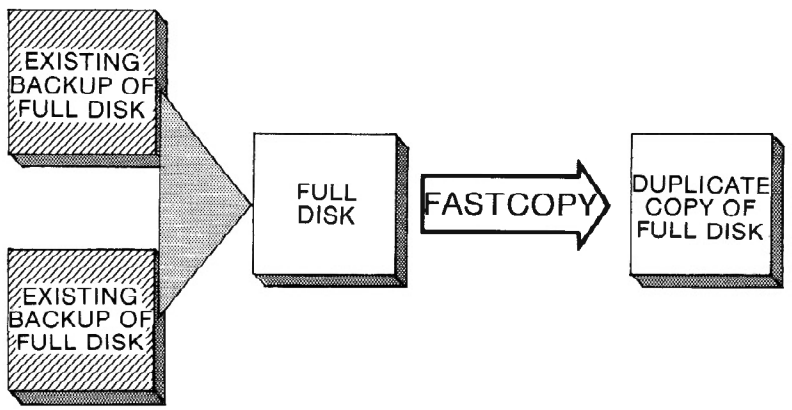


Fig. 1 shows the disk situation as it should be before any splits are made. The center disk represents your full disk, greatly in need of division before data is lost. The two back-up disks on the left are the existing back-ups of the full disk; they are two-thirds of the generation of which the full disk is the other third. And the disk on the right is an exact duplicate copy of the full disk, the one you will put away somewhere safe for at least a week until you are positive that all your data survived the division intact.

With a duplicate copy of your original disk safely stashed away, we may now consider the actual division of your disk. You, and you only, have to make the decision concerning how you want to divide your files. Hopefully, there was a logical arrangement involved when you named your files so now that it's time to move them, you can say, "Well, I'll move all my letters to X Company onto the new disk but keep all my letters to Y Company on the original disk."

I moved all my correspondence files to the new disk and kept all the magazine articles on the original. No one can tell you how to do this; you're the only one who really knows how you want things organized (or not organized, as the case may be).

TWO WAYS TO DIVIDE UP DISKS

THE FIRST WAY

"What is the use of repeating all that stuff," the Mock Turtle interrupted, "if you don't explain it as you go on? It's by far the most confusing thing I ever heard!"

"Why," said the Dodo, "the best way to explain it is to do it."

Actually, there are two ways to divide full disks. One way is to Fastcopy everything from the full disk onto another disk and then delete what you don't want.

For example: Your full disk (#1) contains four files:

A.LTR
B.LTR
Y.LTR
Z.LTR

STEP I: Fastcopy all four files onto another disk (#2).
STEP II: On Disk #1, erase A.LTR and B.LTR.
STEP III: On Disk #2, delete Y.LTR and Z.LTR.

When finished, your #1 disk contains Y.LTR and Z.LTR and your #2 disk contains A.LTR and B.LTR.

STEP I:

DISK #1 FASTCOPY	DISK #2
A.LTR	A.LTR
B.LTR	B.LTR
Y.LTR	Y.LTR
Z.LTR	Z.LTR

STEP II and III:

DISK #1	DISK #2
~~A.LTR~~	A.LTR
~~B.LTR~~	B.LTR
Y.LTR	~~Y.LTR~~
Z.LTR	~~Z.LTR~~

RESULT:

DISK #1	DISK #2
Y.LTR	A.LTR
Z.LTR	B.LTR

This method of splitting a disk is not the preferred method. The files are not packed together tightly, leaving "holes" in the two disks that have been created. When a disk has "holes" in it like this, every new file you open will be placed in the first available hole, resulting in

an incredible hodge-podge in the directory. It's best to have all the system files contiguous (in CP/M these are the ones with the extensions .COM, .OVR, .DEF, and .HEX), and then the document files can be added on systematically. This way the computer doesn't have to search all over the disk for the right file every time you enter instructions; i.e., it eliminates track-jumping.

THE RIGHT WAY

The preferred method for splitting a disk is the second of the two. Here, you PIP all the files you have decided to transfer onto a minimum system disk. Minimum system disk is one of those phrases that takes its meaning from the person using it; it has no standardized meaning. In this context, it means a disk with nothing on it except the operating system residing on Track 0 and the essential CP/M executable files (those with the .COM extension).

"When I use a word," Humpty Dumpty said, in rather a scornful tone, "it means just what I choose it to mean—neither more nor less."

To accomplish the break-up using this method, all you do is PIP each of the document files you wish to transfer to the new disk. Here, B is the destination, A is the source, and "filename.LTR" is the file you wish to transfer.

A> PIP B:=A:filename.LTR

Repeat this maneuver for each of the files you have decided to move.

When you have finished PIPing, you should determine that all your files copied properly by checking STAT at the command level. You can check the number of bytes remaining on the entire disk by entering

A>STAT

or you can check the number of bytes occupied by each file on the disk by doing

A>STAT *.*

or you can check the space taken up by one file only by typing

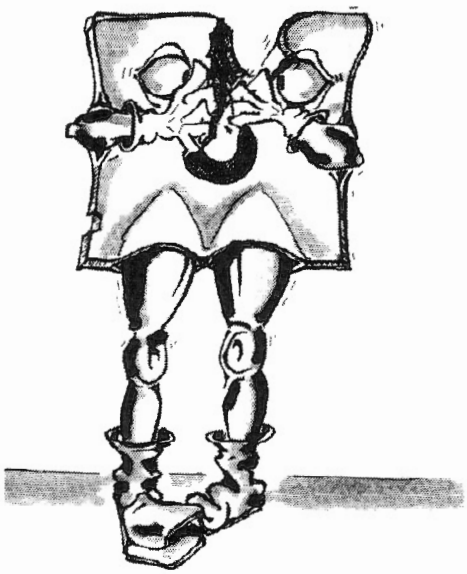
A>STAT filename.LTR

It's important that you have the same number of R/W characters on both disks. You will probably never see a discrepancy, but you never know—one of your files might not have copied. The motto of this column could be stated as a tired-but-true cliché: It's better to be safe than to be sorry.

After you have made sure that everything has been copied just the way you want it, then you can erase duplicate document files on your original disk (Disk A). These duplicates would be the ones you just finished PIPing onto Disk B.

I was confused when Dick told me to erase document files on my original disk. I thought I had physically moved them from A to B using PIP. But PIP is not a moving device; it is only a copying device. It copies files from A to B, giving you an exact copy on both disks. So you need to erase.

"Curiouser and curiouser!" cried Alice.



IT'S BACK-UP TIME ONCE AGAIN

At this point in the proceedings we have two disks—well, actually, it's four disks, but more about that later. The two we're concerned with now both have the operating system and all commands, but A has one set of document files and B has a different set of document files. As discussed in this column last month, you cannot do business with only one disk; you have to have back-up copies.

If you stayed with me last time in the back-up article, you'll do just fine here. Get organized before you start. You're going to need four more disks, all of them formatted only.

If you're not sure how to tell whether a disk is formatted or not, put it in Drive 0. If it boots, it's formatted and has an operating system in track 0. You don't want it to have anything more than that on it so ask for the directory listing:

A>DIR

If you get "NO FILE," the disk can be used for copying on to.

We store our disks in file boxes made especially for disks, and plastic dividers separate the different categories of disks within each box. Before I started the back-up process, I marked the contents of the disk I was not going to be copying this time (Disk B) and put it into its own compartment of the file box.

I also marked the contents of the disk I was going to be copying (Disk A), and designated a special compartment for it in the file box. This may sound overly cautious, and it probably is, but there is nothing more confusing than having six disks lying around when all of them are more or less the same. How do you tell which is which? I got hopelessly tangled up once, and that was enough.

LIFE IN THE FASTCOPY LANE

Ready? Put the original disk (A) in Drive 0 and the formatted disk in Drive 1, and FASTCOPY from Disk A to Disk A(1), i.e., from Drive 0 to Drive 1. When FASTCOPY is completed, remove the backed-up disk (Disk A(1)) from Drive 1, label it, and put it away safely in its compartment in the file box. Then insert the next blank formatted disk in Drive 1, and FASTCOPY once again from Disk A to Disk A(2).*

*I should point out that FASTCOPY is the acid test to determine whether or not the split was truly successful. When I did my split, I took the disk I had copied on to (Disk B), checked the DIR (that was fine) and WordStar (that was fine, too). But when I went to FASTCOPY (one of the CP/M commands that had been PIPed), I got a BDOS Error message. Essentially what I had was a useless disk. I could work on it, but why would I want to? Once the information was on the disk, there was no way for me to back it up. I should also point out that when I checked the disk before the disastrous FASTCOPY attempt and decided everything was OK, I erased the back-up disk I had so carefully put out of harm's way back at the start. Needless to say, we had an awful

mess, but Dick managed to straighten everything out for me *once again*.

"That was a narrow escape! said Alice, a good deal frightened . . . , but very glad to find herself still in existence. . . ."

When the first FASTCOPYing process is completed, you will have three identical disks: A, A(1), and A(2). Label them so you will remember what is on them, and file them away in their own labeled compartment in the file box.

Time for step two: Remove Disk B from its position in the file box, and insert it into Drive 0. Insert one of the blank, formatted disks into Drive 1 and then FASTCOPY from Disk B to Disk B(1) (or Drive 0 to Drive 1, if you prefer). When FASTCOPY is completed, remove disk B(1) from Drive 1, label it and file it away properly, and then insert Disk B(2) into Drive 1 and FASTCOPY from Disk B(1) to Disk B(2). Repeat the same procedure you did with the A disks.

Now, to return to the issue raised back at the beginning of this section—about the actual number of disks you had before you began backing-up. Why did I say you have four disks when you are thinking you only have two? The original disk, the one that got too full and had to be divided up, was part of a generation of three and had its own two back-up disks. These two disks are now obsolete and can be returned to circulation.

But the back-up you made of your original disk before you started dividing—remember the one I told you to hide somewhere?—should be archived for at least a week so that you can be absolutely certain your division went without a hitch and all your files copied intact. (See Fig. 2 for a summary of the entire procedure.)

ARE THERE ANY QUESTIONS?

For those of you who asked why a series of three disks is necessary for proper back-up instead of only two, here's the reason. If you have only a main, working disk and one back-up, and you are backing-up your main disk onto your one-and only-auxiliary disk and thus have two disks in the computer at once—what happens if the power goes off? You might lose everything. But if you are working with a generation of three disks, you will be protected against power failure or computer failure since you won't have all three of your disks in the computer at once. Most disk damage occurs in the computer, not from external sources.

"Well, I never heard that before," said the Mock Turtle; "but it sounds uncommon nonsense."

And so we end another strange and wonderful adventure in computerland. When I first began, one file was truly a marvelous experience. Now I have six disks!

"For you see, so many out-of-the-way things had happened lately that Alice had begun to think that very few things indeed were really impossible."

FIGURE 2

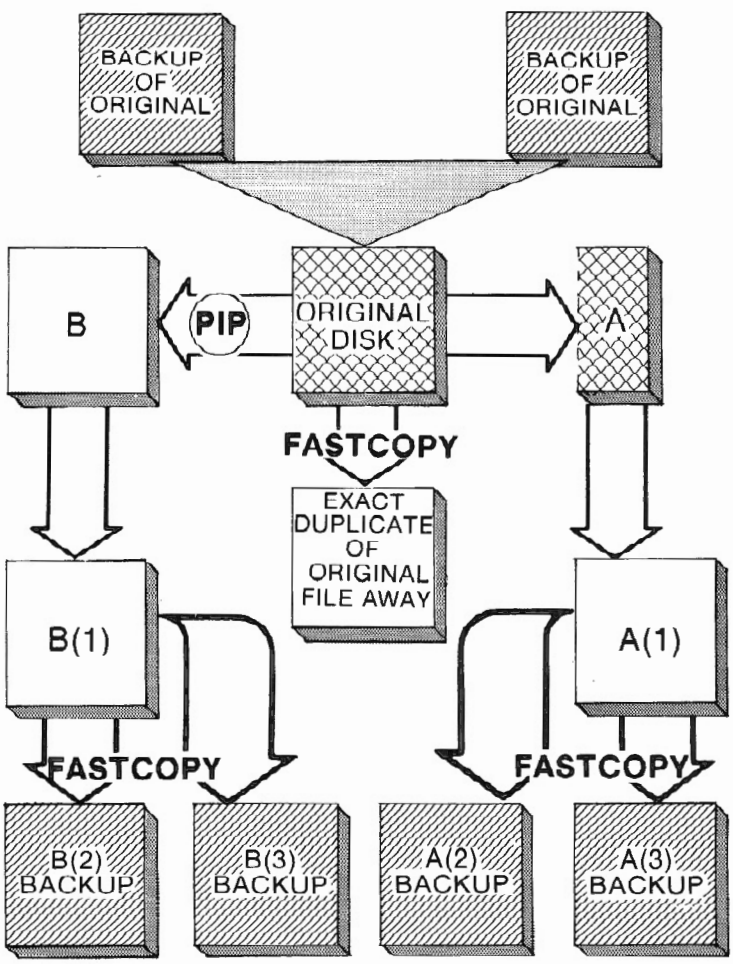


Fig. 2 shows a general outline of the entire division process. Starting with the original full disk and its two back-up files, first FASTCOPY an exact copy of the full disk and put it away for safekeeping. Then PIP off onto a minimum system disk—Disk B—all files you wish to move. Disk A in the illustration is the residue of the original disk after the other files have been PIPed. The concluding step is to back-up the two new disks so that a new generation is established for both. The back-up of the full disk is archived until such time as you are positive the split went okay; i.e., no files were damaged or lost in transit, and the two back-ups of the original full disk can be entered back into circulation.

MicroShell: UNIX Features for CP/M Users

Richard H. Young

MicroShell is a new "facility" which gives CP/M users amazing new capabilities. In effect, MicroShell "sits on top of" CP/M, adding functionality to commands by providing sophisticated redirection of input to programs and output from programs. For example, CP/M has no facility for placing file directories into files or for printing file directories, but the command

dir > files

to MicroShell will cause the output from DIR to be placed in the file "FILES," instead of going to the console. It will go to both the console and the file if the command is

dir >+files

FILES can then be printed, e.g.:

pip lst:=files[p60]

Command lines to MicroShell are formed just like commands to CP/M, except that MicroShell allows additional arguments. All CP/M commands function in exactly the same way as before (except that: 32 user areas are allowed, instead of 16; there is no warning message when the command "era *.*" is given; and certain commands which produce "synchronization errors" with MicroShell, e.g., FORMAT, require special treatment). Also, multiple commands may be typed on one line, separated by semicolons, e.g.:

dir > files;pip lst:=files[p60];era files

Output can be appended to existing files. For example,

stat *.* >> files

would append detailed file status to the directory information previously placed in FILES.

Input to programs can come from files (when the program is expecting console input). The command

prog1 < input1

runs the program PROG1.COM and causes input expected from the console to be read from the file INPUT1.

Command files can be invoked, similar to SUBMIT with XSUB, from MicroShell by simply giving the file name. To execute the commands in file CMD.SUB, type

CMD

Argument substitution can be performed in the same way as in the SUBMIT command. If CMD.SUB contained the following

mac $1 $2; load $1;era $1.hex

and the following was typed to MicroShell

cmd test $PB AA

then MicroShell would pass the following commands to CP/M:

```
MAC TEST $PB AA
LOAD TEST
ERA TEST.HEX
```

Unlike SUBMIT, MicroShell does not abort if parameters are missing, but merely closes up the file. If in the above example the following was typed to MicroShell

EXAMPLE 1	EXAMPLE 2
cmd test $PB	cmd test

then MicroShell would pass to CP/M:

MAC TEST $PB	MAC TEST
LOAD TEST	LOAD TEST
ERA TEST.HEX	ERA TEST.HEX

One of MicroShell's greatest extensions to CP/M is its ability to drive programs like WordStar from command files. As an exercise in utilization of MicroShell command files, we developed a multi-sorted file directory in printed form. Examples display the code, the original STAT report, and the resulting presentation of three parallel reports sorted (1) by file name and extension, (2) by size and filename, and (3) by extension and filename. The software used in this application is

CP/M 2.2e, Pickles & Trout
MicroShell 1.21
WordStar 3.0
SuperSort 1.6

Obviously, FORTRAN is a more elegant and practical way than WordStar to perform the editing. But it's kind of fun to watch WordStar hacking away at the file all by itself. And this example demonstrates the high degree of control obtained with MicroShell.

MicroShell's normal prompt character is a percent sign (%), but it can be changed to any string of up to 40 characters, and may include the currently selected drive and user number. My preference is

d%>

where d is the currently selected drive, e.g.,

A%>

This looks enough like the CP/M prompt to be comforting, but it contains the % to indicate that it is a MicroShell prompt.

If you have a Pickles & Trout clock board or other device which will sound a "bell" when ^G is issued, you can include the

bell in the prompt string. This is a useful signal of job completion when the operator is doing other work while a long job is running.

MicroShell can be used as a machine language disassembler. Assume the following code in a file named DISASM.SUB:

```
ddt $1.com > $1.asm
L100 $2
G0
```

If this is executed by

```
disasm test 1000
```

then MicroShell will pass to CP/M:

```
DDT TEST.COM
L100 1000
G0
```

and will place the assembly language output in TEST.ASM. This output can then be modified and reassembled.

In order to better understand the "synchronization error" referred to above, we have obtained the following explanation from MicroShell's developer:

> MicroShell depends on a program making "standard" calls to CP/M, i.e., either BDOS or direct BIOS calls are acceptable. Some programs, notably public domain programs and some system utility programs (like FORMAT), make assumptions about the CP/M configuration that are non-standard. They may "poke around" inside CP/M's command processor which MicroShell replaces or depend upon a CP/M warm start to reload CP/M after they terminate. MicroShell prevents "warm starts" to speed processing with the resulting incompatibility with this small group of programs. A facility is provided in MicroShell to exit MicroShell to run these programs followed by automatic reloading of MicroShell if desired.

We have used this latter facility regularly and have had complete success with it.

A general discussion of MicroShell does not do it justice. Once you begin to use it, you will discover many valuable and astonishing things that can be done with it, and you will use it constantly. It quickly becomes an indispensable tool.

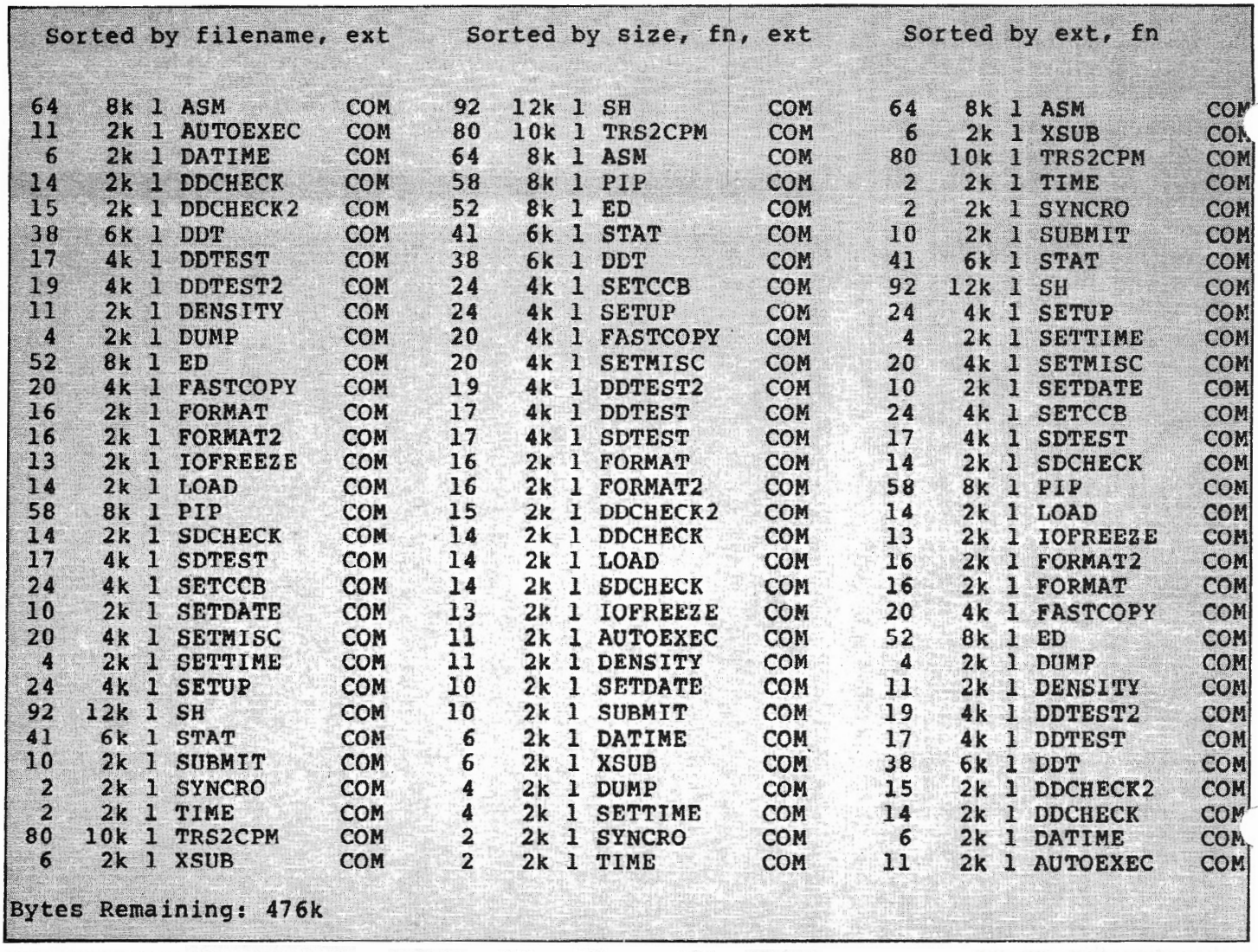
We will be doing further testing of MicroShell in additional applications which are difficult or impossible without it, and will report on these applications in future issues.

At $150, MicroShell is a very good buy. It will considerably enhance the value of CP/M, word processors, and other software you already own, as well as being a very useful tool in its own right.

MicroShell Version 1.21, $150
New Generation Systems, Inc.
2153 Golf Course Drive
Reston, VA 22091
(703) 476-9143

```
;STAT1.SUB - PRODUCES STATUS REPORTS OF DRIVE B: FILES  SORTED BY:
;       FILENAME
;       SIZE
;       EXTENSION
;RUNS UNDER MICROSHELL IN DRIVE A:.
;RHV, 7/12/82
+V
STAT B:*.*>STAT
ERA STAT.TMP
WSN
FB1NSTAT
^QA On B:
;
GN
^[^QR^QFRemaining
1
^QS^KB^X^KK^KNSTAT.TMP
^[^QR^QFRemaining
1
^Y^Y^[^QA R/W B:
,
GN
^[^QA.
,
GNB
^[^QA
,
GN
^[^QA
,
GNB
^[^QA
,
GN
^[^QA
,
GNB
^[^QA
,
GN
^[^QA^N,
^N
GNB
^[^QR^QFRecs
1
^Y^[^KK
SORT
IN      60 CR
MERGE   STAT
OUT     STATUS1,CR,K-OUTPUT
KEY     #1,5,RIGHT-JUSTIFY,#2,5,RIGHT-JUSTIFY,59,59,#3,1,59,59,#4,10,59,59,#5,3
GO
SORT
IN      30 CR
SORT    STATUS1
OUT     STATUS2,CR
KEY     1,5,DESC,14,27
GO
SORT
IN      30 CR
SORT    STATUS1
OUT     STATUS3,CR
KEY     25,27
GO
ERA STATUS9
WSN
FB1NSTATUS9
^KRSTATUS1
^QC^KRSTATUS2
^KB^QC                          ^KK^KN^QR^QD ^KV^QK^QS^KB^KN^QC^KK^KY^QR^QC^KRSTATUS3
^KB^QC                          ^KK^KN^QR^QD ^KV^QK^QS^KB^KN^QC^KK^KY^QR
^E.MT 5
^QC
  ^KRSTAT.TMP
^KDPSTATUS9^[X
ERA STAT
ERA STAT.TMP
ERA STATUS1
ERA STATUS2
ERA STATUS3
-V
```

```
  Recs   Bytes   Ext Acc
    64      8k     1 R/W B:ASM.COM
    11      2k     1 R/W B:AUTOEXEC.COM
     6      2k     1 R/W B:DATIME.COM
    14      2k     1 R/W B:DDCHECK.COM
    15      2k     1 R/W B:DDCHECK2.COM
    38      6k     1 R/W B:DDT.COM
    17      4k     1 R/W B:DDTEST.COM
    19      4k     1 R/W B:DDTEST2.COM
    11      2k     1 R/W B:DENSITY.COM
     4      2k     1 R/W B:DUMP.COM
    52      8k     1 R/W B:ED.COM
    20      4k     1 R/W B:FASTCOPY.COM
    16      2k     1 R/W B:FORMAT.COM
    16      2k     1 R/W B:FORMAT2.COM
    13      2k     1 R/W B:IOFREEZE.COM
    14      2k     1 R/W B:LOAD.COM
    58      8k     1 R/W B:PIP.COM
    14      2k     1 R/W B:SDCHECK.COM
    17      4k     1 R/W B:SDTEST.COM
    24      4k     1 R/W B:SETCCB.COM
    10      2k     1 R/W B:SETDATE.COM
    20      4k     1 R/W B:SETMISC.COM
     4      2k     1 R/W B:SETTIME.COM
    24      4k     1 R/W B:SETUP.COM
    92     12k     1 R/W B:SH.COM
    41      6k     1 R/W B:STAT.COM
    10      2k     1 R/W B:SUBMIT.COM
     2      2k     1 R/W B:SYNCRO.COM
     2      2k     1 R/W B:TIME.COM
    80     10k     1 R/W B:TRS2CPM.COM
     6      2k     1 R/W B:XSUB.COM
Bytes Remaining On B:  476k
```



Sorted by filename, ext

64	8k	1	ASM		COM
11	2k	1	AUTOEXEC		COM
6	2k	1	DATIME		COM
14	2k	1	DDCHECK		COM
15	2k	1	DDCHECK2		COM
38	6k	1	DDT		COM
17	4k	1	DDTEST		COM
19	4k	1	DDTEST2		COM
11	2k	1	DENSITY		COM
4	2k	1	DUMP		COM
52	8k	1	ED		COM
20	4k	1	FASTCOPY		COM
16	2k	1	FORMAT		COM
16	2k	1	FORMAT2		COM
13	2k	1	IOFREEZE		COM
14	2k	1	LOAD		COM
58	8k	1	PIP		COM
14	2k	1	SDCHECK		COM
17	4k	1	SDTEST		COM
24	4k	1	SETCCB		COM
10	2k	1	SETDATE		COM
20	4k	1	SETMISC		COM
4	2k	1	SETTIME		COM
24	4k	1	SETUP		COM
92	12k	1	SH		COM
41	6k	1	STAT		COM
10	2k	1	SUBMIT		COM
2	2k	1	SYNCRO		COM
2	2k	1	TIME		COM
80	10k	1	TRS2CPM		COM
6	2k	1	XSUB		COM

Sorted by size, fn, ext

92	12k	1	SH		COM
80	10k	1	TRS2CPM		COM
64	8k	1	ASM		COM
58	8k	1	PIP		COM
52	8k	1	ED		COM
41	6k	1	STAT		COM
38	6k	1	DDT		COM
24	4k	1	SETCCB		COM
24	4k	1	SETUP		COM
20	4k	1	FASTCOPY		COM
20	4k	1	SETMISC		COM
19	4k	1	DDTEST2		COM
17	4k	1	DDTEST		COM
17	4k	1	SDTEST		COM
16	2k	1	FORMAT		COM
16	2k	1	FORMAT2		COM
15	2k	1	DDCHECK2		COM
14	2k	1	DDCHECK		COM
14	2k	1	LOAD		COM
14	2k	1	SDCHECK		COM
13	2k	1	IOFREEZE		COM
11	2k	1	AUTOEXEC		COM
11	2k	1	DENSITY		COM
10	2k	1	SETDATE		COM
10	2k	1	SUBMIT		COM
6	2k	1	DATIME		COM
6	2k	1	XSUB		COM
4	2k	1	DUMP		COM
4	2k	1	SETTIME		COM
2	2k	1	SYNCRO		COM
2	2k	1	TIME		COM

Sorted by ext, fn

64	8k	1	ASM		COM
6	2k	1	XSUB		COM
80	10k	1	TRS2CPM		COM
2	2k	1	TIME		COM
2	2k	1	SYNCRO		COM
10	2k	1	SUBMIT		COM
41	6k	1	STAT		COM
92	12k	1	SH		COM
24	4k	1	SETUP		COM
4	2k	1	SETTIME		COM
10	2k	1	SETDATE		COM
20	4k	1	SETMISC		COM
24	4k	1	SETCCB		COM
17	4k	1	SDTEST		COM
14	2k	1	SDCHECK		COM
58	8k	1	PIP		COM
14	2k	1	LOAD		COM
13	2k	1	IOFREEZE		COM
16	2k	1	FORMAT2		COM
16	2k	1	FORMAT		COM
20	4k	1	FASTCOPY		COM
52	8k	1	ED		COM
4	2k	1	DUMP		COM
11	2k	1	DENSITY		COM
19	4k	1	DDTEST2		COM
17	4k	1	DDTEST		COM
38	6k	1	DDT		COM
15	2k	1	DDCHECK2		COM
14	2k	1	DDCHECK		COM
6	2k	1	DATIME		COM
11	2k	1	AUTOEXEC		COM

Bytes Remaining: 476k

PRO-80 BASIC CORRECTION

In the article entitled "Review of PRO-80 BASIC," which appeared in our July-August issue, our reviewer misunderstood the sorting syntax as described in the manual (we must frankly state that we were also unable to derive the correct command format without reference to PRO-80 Systems — the manual cries out for clarification). Bob Snapp called to let us know that the published sort times were absurdly high, and that we owed his competitor a correction. Cursory review indicated that we should have realized that these times were higher than could be obtained with a crude sort in BASIC. We asked our regular contributor, Arnold Fischthal, to run some tests. He produced the following times (in seconds) which are more representative of the results which should be obtained from PRO-80 BASIC sorts:

String Size	Array size			
	100	200	500	1000
10	0.38	0.88	3.00	6.63
15	0.50	1.40	3.63	8.25
20	0.75	1.63	4.25	

Arnold also did a comparative study of sort times under Snappware, with the following results:

String Size	Array size			
	100	200	500	1000
10	0.50	0.50	2.00	4.38
15	0.50	0.88	2.70	5.63
20	0.50	1.00	2.88	

These studies are based on one-dimensional arrays which began in reversed order, with samples constructed so that the comparison had to go almost to the end of each string before obtaining a "greater than" or "less than" result. Each sort was performed eight times and the average of the results was taken.

Because of the extremely fast sort times and the difficulty of observing such short intervals, real application times may be slightly different. For all practical purposes, however, it suffices to say that both of these sorts are amazingly faster than anything that could be done in conventional BASIC. The timing differences between them will not be material in any ordinary situation.

We apologize to Ron Ryen (PRO-80 Systems) for publishing the unrepresentatively high times on his BASIC sorts.

THE BASIC DEBATE RAGES ON

The May/June issue of *two/sixteen magazine* contained an article entitled "Lamentations on the Radio Shack BASIC Interpreter" (pp. 2–3). The responses to this article have been numerous, and they are still coming in as new subscribers receive their first copies. Little did we envision the controversy we would stir up, although perhaps we should have had some idea. Any article that begins "To make my unbiased position perfectly clear at the outset . . ." is bound to unleash a whirlwind, and ours continues to blow and swirl around us.

Many of the letters in the "Letters to the Editor" Column contain readers' opinions on this BASIC issue, but we have decided to print the full text of two especially relevant responses. Obviously, these letters represent only two of the varying viewpoints we've received. However, the authors of these letters devoted their respective letters entirely to the question of the value of RSBASIC, and as you will notice after reading them, they address the article from two very opposing points of view!

We also wanted to print the letter from Roklan Corporation so it can never be said that we only print letters portraying us in a favorable manner.

FROM: ROKLAN CORPORATION,
10600 W. Higgins Road, Suite,
200, Rosemont, Illinois 60018:

I was very sorry to read in your inaugural issue the extremely low regard you have for the group of hard working professionals known as programmers. To think, all those programmers wasted 5, 10, 15 years of their lives learning their profession, having their skills, developing their techniques. According to your magazine, they should have mastered all that in a few weeks! "Most people can attain a minimally useful proficiency in a day or two. Fairly complete mastery might require a few weeks." What a quote. While professional BASIC programmers are in fact a small percentage of all the professional programmers currently making a living in the field, the next line quoted from your magazine assures alienation of all of them: "Other more powerful languages, such as Fortran, may take as many months to master."

This compares to being able to become an attorney or a doctor by just reading a book. Imagine, in your spare time over the next few months, you too can become a doctor or lawyer. They would laugh at you. You would be informed that it takes years of hard work and training and dedication to become a member of such an esteemed profession. But of course, anyone can be a master programmer in a few weeks.

The author of the article "Lamentations on the Radio Shack BASIC Interpreter" apparently believes he has mastered programming. Here is a small amount of information from a professional programmer to show this self-made master how little he really knows of BASIC.

1.) An interpreter does not convert from "Source Code" to machine language. An interpreter, such as BASIC, executes machine language routines contained within the interpreter as required by the line of code currently being interpreted. The extremely slow execution of many programs is usually the result of the "parsing" techniques of the individual interpreter. The processing and unprocessing of the line of BASIC and the amount of pointers and cross referencing done internally also determine speed and efficiency. A good working knowledge of BASIC can allow a good programmer

to overcome many of these shortcomings.

2.) No individual manufacturer to date has produced a BASIC that is compatible with any other manufacturer's BASIC, so why single out RS for special criticism?

3.) Aside from an archaic way of referring to main memory (core), there's nothing here good programming techniques can't cure. While it's true that a good program does need to be well documented, no programmer that I know of would include that documentation in the final running program. The place for final comments is in the file copy of the program and in the accompanying documentation. Not in the computer's memory. (Core refers to a type of nonvolatile magnetic memory which fell into disuse with the advent of semi-conductor memory.)

4. & 5.) Contrary to popular belief, subroutine libraries are alive and well and living in the disk files of professional programmers everywhere. Modular programming techniques and appropriate software tools eliminate these objections. Cross reference utilities are available to BASIC programmers just as they are available to assembly language programmers.

6.) See 3.) above

7.) Even Radio Shack sells a separate program editor. No law says you must use only the editor contained in BASIC. RS calls their editor TED, Tandy Program Editor. If you are serious about programming, get the right tool for the right job. No assembler or compiler I know of contains an editor. You must use a separate editor. With BASIC you get a choice.

8.) The field statement in RS BASIC is almost identical to the format statement in Fortran. It only needs to be executed once per program to be effective. Recommendation: put your field state immediately after the OPEN statement, not next to the GET.

9.) A little math practice here goes a long way. When was the last time your accountant used logs in computing your business records?

One final note: it is extremely helpful for people who wish to use computers to know something about programming. But remember, driving the family car does not qualify you for the Indy 500. When there is serious work to be done, acquire the products or services of the qualified professional and don't practice brain surgery on the kids.

P.S. As for "Patching TRSDOS: Fear Not!" I seriously hope nobody read it.

FROM: RODNEY L. WRIGHT, USREP/JECOR, STADAP BOX 235, APO NY 09038:

The premier issue of *two/sixteen magazine* contained an opinion article entitled "Lamentations on the Radio Shack BASIC Interpreter." It is seriously marred by its opening paragraph in which the unnamed author states: ". . . in my opinion, supplying a BASIC interpreter with a business oriented computer should be a felony."

As admitted later in the article, BASIC is easy to learn, easy to debug and has great string handling capabilities. It is very easy to use for small jobs and is most convenient for small "quick and dirty" jobs. Since I have not found that convenience in any other language, BASIC will continue to be one of the essential tools in my programmer's toolbox.

Further, the unbundling of the BASIC interpreter from the hardware would not be likely to produce any meaningful decrease in cost for a Model II/16. It would just increase the market for separately priced BASIC interpreters. Such interpreters typically cost around $350.00. The cost of the BASIC program product plus the hardware would exceed the present cost of the bundled "package" of hardware and interpreter, since the software license fee would be spread over fewer units by the manufacturer. It is generally acknowledged that the common availability of BASIC has had much to do with the explosive growth of sales for microcomputers.

I experienced one more reaction to the opening paragraph. The author stated that the Models II and 16 are "business oriented." The inference is that owners of these machines are not interested in scientific applications, games or other non-business applications. Frankly, my primary reason for owning the Model II is for business, but I'm also a real person who, occasionally, likes to have fun or to use the computer for experimentation, education, or for writing to magazines. These machines are general-purpose processors useful both for a good game and for a data based management system. The dull-gray image of "business computers" is too limiting in scope to describe these machines and their users. Calling the Models II and 16 just "business-oriented" discourages potential program developers from exploiting the full range of their potential.

Despite my opinions about the first paragraph, the rest of the article is essentially correct. BASIC does have severe limitations. In addition to the disadvantages mentioned in the article, the user should consider the following:

10.) The efficiency of the BASIC program is inversely related to its size. As the size of the program is increased, the space available for arrays, variables, and strings is decreased. As the space decreases, there is a related decrease in the performance of the interpreter because it has to store each character of the program, even remarks, and to decode them each time they are encountered in the running of the program.

11.) BASIC discourages good structured programming practices:

a.) The use of spaces between keywords increases code readability and maintainability. Unfortunately, this also increases the program size, thereby imposing a performance penalty on the programmer who attempts to produce readable code.

b.) The use of single statement lines and indentation of code to reflect program structure greatly improves program readability and maintainability. But, once again, interpreted BASIC penalizes good "style" because the resultant program is larger and slower.

c.) The use of section labels, explanatory notes and on-line remarks helps to insure that the code is clear, readable, and maintainable. The resulting structured program is larger and slower with the same performance penalties noted above.

d.) The use of long (i.e., meaningful) variable names is essential to writing good, clear, and understandable programs. Even though Radio Shack BASIC permits long names, only the first two characters are significant. As a result there is less flexibility in assigning those names.

e.) The use of unique variable names for each variable imposes a further performance penalty. BA-

SIC stores each variable in a table that is sequentially searched and interpreted each time that a variable is used in the program. The bigger the table, the slower the search. Large BASIC programs with unique variable names for each function are, therefore, less efficient and poorer performers than those that reuse the variable. If, however, the variables are reused to improve performance, the uniqueness is lost. As a consequence, the program may be difficult to read and maintain.

f.) The lack of the DO-WHILE and DO-UNTIL structures in most BASIC interpreters encourages the use of the GOTO unstructured statement.

g.) The lack of symbolic labels for program lines or statements makes the resultant code less readable.

12.) Because BASIC is interpretive, it does not decode instructions until execution time. As a result, syntax errors may not be discovered until the program is executed. The case of the "hidden syntax error" is all too familiar to BASIC programmers whose programs include seldom used code (such as error code handling routines). Murphy's law says that you will find the undiscovered error at the worst possible time. Compilers, on the other hand, decode the entire program before any execution occurs, thus syntax errors are discovered and corrected early.

13.) The limited editor available in Radio Shack BASIC has several severe shortcomings:

a.) It has no cross-reference utility. A complete cross-reference utility is needed for serious programming. Good cross-reference utilities list the location of all constants, all references to line numbers, and all references to variables. For each reference, the cross reference utility should also indicate the type of variable (string, integer, single or double precision) and how it is used (simple variable or array indexed variable). The cross reference utility should also identify the location of program keywords and strings. (For example, the cross reference utility should tell the programmer which program lines contain OPEN statements).

b.) Radio Shack's renumbering utility is incomplete and slow. It needs to provide the ability to renumber (and thereby move) an imbedded section of code. For example: Renumber old lines from 1542 to 1621 using new numbers beginning at 1000 and incrementing by 2.

c.) The editor also requires full length commands such as "LIST" when "L" would do or "LLIST" when "P" (for Print) would do.

d.) The editor does not provide a simple means to move through the code of a program. It should be possible to scroll through a program using the arrow keys built into the program and to display or edit the current program line with a single keystroke, but this is not provided in the stock Radio Shack editor.

14.) There is no ability to directly read (PEEK) and write (POKE) to main memory.

Some of the deficiencies of Radio Shack BASIC have been corrected by third party program packages available from software vendors. The best examples I

know are the Snapp II and III packages available from Snapp, Inc. of Cincinnati, Ohio. Snapp's modifications to the Radio Shack BASIC interpreter perform all of the functions mentioned above and more, but the cost of these augmentations to Radio Shack's BASIC approximate the cost of a professional compiler-based program package and are most needed when the programming task is too large to be handled in BASIC anyway.

So we come back to my toolbox. I believe we should use interpreted BASIC for small, quick and easily solvable problems but should recognize its inherent limitations and use the more sophisticated higher-level tools for more complex tasks. A good rule of thumb is that BASIC programs should not be used where they exceed a single page of structured code.

AN RSBASIC ALTERNATIVE TO DOSFIX14

Luis S. Shuey
574 W. 550 N. St.
St. George, UT 84770

In his comments on the various DOSFIX files printed here last time (See Vol. 1, #2, p. 27), Bob Snapp explained the function of DOSFIX14: "In our opinion, the FORMS command harasses the computer operator with too many B.S. questions. This one eliminates all the B.S."

One of our readers, Luis S. Shuey, has submitted the following Assembly Language Program to pass along. Mr. Shuey writes: "SNAPP DOSFIX14 is, of course, useful, but in RSBASIC programs I use the following."

This program uses the MicroSoft Assembler, Macro-80. In addition, it incorporates supervisor calls for those of you who have expressed an interest in learning more about them.

```
00010     .COMMENT*
00020     NAME:  INFORM/MAC
00030     PROJ:  LIBRARY
00040     DATE:  09/02/1982
00050     PGMR:  L.S. SHUEY
00060     COMP:  TRS-80, MOD II
00070     SYST:  TRSDOS, 2.0A
00080     LANG:  EDIT-80
00090     DISC:  TRSDOS
00100     SIZE:  3AH, 58
00120
00130     ------------------------------------------------------------------
00140     AN EXTERNAL ROUTINE TO INITIALIZE THE  PRINTER  FORMAT FROM RSBASIC
00150     ------------------------------------------------------------------
00160     ON ENTRY TO INFORM FROM RSBASIC, REGISTER CONTENTS ARE:
00170           (SP) = RETURN ADDRESS
00180           (BC) = PAGE LENGTH, LINES PER PAGE, MAX CHARACTERS PER LINE
00190           (DE) = DECODING ROUTINE
00200     ON DECODING RETURN:
00210           DE IS ADDRESS OF PAGE LENGTH
00220                 ADDRESS OF LINES PER PAGE
00230                 ADDRESS OF MAX CHARACTERS PER LINE
00240           A IS RETURN CODE
00250           B IS ARGUMENT TYPE
00260     ENTRY CONDITIONS FOR PRINIT:
00270           A = 17 (SVC CODE FOR PRINIT)
00280           B = PAGE LENGTH
00290           C = LINES PER PAGE
00300           D = MAX CHARACTERS PER LINE
00310     ------------------------------------------------------------------
00320     TO ASSEMBLE, USE "M80 INFORM=INFORM"
00330     TO LINK, USE: "L80 -P:HHHH,INFORM,INFORM-E-N" (HHHH=HEX LOCATION)
00340           (PRESS ENTER TO IGNORE "ORIGIN ABOVE LOADER MEMORY,....")
00350     ------------------------------------------------------------------
00360     *
00370           CSEG
00380     INFORM:                         ; ROUTINE STARTS HERE
00390     ;-----------------------------------------------------------------
00400     ; SAVE PARAMETER DECODING ROUTINE ADDRESS AS DERAD
00410           LD      HL,DERAD          ; LOC OF DECODE ROUTINE TO HL
00420           LD      (HL),E            ; CON OF E TO LOC (HL)
00430           INC     HL
00440           LD      (HL),D            ; CON OF D TO LOC (HL)
00450                                     ; DERAD NOW DECODE ROUTINE ADDRESS
00460     ;-----------------------------------------------------------------
00470     ; CALL DECODE ROUTINE FOR ADDRESS OF PAGE LENGTH
00480           LD      HL,CONT           ; CONTINUATION ADDRESS TO HL
00490           PUSH    HL                ; PUSH TO STACK
00500           LD      HL,(DERAD)        ; CALL FOR DECODE ROUTINE
00510           JP      (HL)              ; JUMP TO DECODE ROUTINE
00520     CONT:  LD      A,(DE)            ; A NOW PAGE LENGTH
00530           LD      (PAGEL),A         ; PAGEL NOW PAGE LENGTH
00540     ;-----------------------------------------------------------------
00550     ; CALL DECODE ROUTINE FOR ADDRESS OF LINES PER PAGE
00560           LD      HL,CONT1          ; CONTINUATION ADDRESS TO HL
00570           PUSH    HL                ; PUSH TO STACK
00580           LD      HL,(DERAD)        ; CALL FOR DECODE ROUTINE
00590           JP      (HL)              ; JUMP TO DECODE ROUTINE
00600     CONT1: LD      A,(DE)            ; A NOW LINES PER PAGE
00610           LD      (LINES),A         ; LINES NOW LINES PER PAGE
00620     ;-----------------------------------------------------------------
00630     ; CALL DECODE ROUTINE FOR ADDRESS OF MAX CHARACTERS, GET AND PUT TO
00640           LD      HL,CONT2          ; CONTINUATION ADDRESS TO HL
00650           PUSH    HL                ; PUSH TO STACK
00660           LD      HL,(DERAD)        ; CALL FOR DECODE ROUTINE
00670           JP      (HL)              ; JUMP TO DECODE ROUTINE
00680     CONT2: LD      A,(DE)            ; A NOW MAX CHARACTERS IN
00690           ADD     A,1               ; ADD 1 FOR AUTO CARRIAGE RETURN
00700           LD      D,A               ; D NOW MAX CHARACTERS TO PRINIT
00710     ;-----------------------------------------------------------------
00720     ; LOAD FOR SVC AND CALL
00730           LD      A,(PAGEL)         ; A NOW PAGE LENGTH
00740           LD      B,A               ; B NOW PAGE LENGTH
00750           LD      A,(LINES)         ; A NOW LINES PER PAGE
00760           LD      C,A               ; C NOW LINES PER PAGE
00770           LD      A,17              ; A NOW SVC FOR PRINIT
00780           RST     8                 ; CALL PRINIT
00790           RET
00800     ;-----------------------------------------------------------------
00810     ; DATA SECTION
00820     DERAD: DW      00                ; WILL STORE DECODE ROUTINE
00830     PAGEL: DB      0                 ; WILL STORE PAGE LENGTH
00840     LINES: DB      0                 ; WILL STORE LINES PER PAGE
00850     ;-----------------------------------------------------------------
00860           END     INFORM
```

ASPEN'S DRAMATIC GRAMMATIK: TAKES YOU BEYOND SPELLING CHECKING

Barbara S. Albert

Aspen Software Company in Tijeras, New Mexico, has produced not only a very handy and extremely practical spelling checker called **Proofreader** (see review this issue), they have also come out with a companion for **Proofreader** called **Grammatik.**

Grammatik is a "document checking system" designed to take you "beyond spelling checking." (Comments in quotation marks are from the **Grammatik** manual by Aspen.) After you have run your finished work through the system using **Proofreader** and corrected the errors using **Proof-Edit, Grammatik** will check for typographical errors that didn't qualify as "unknown words" with **Proofreader.**

At the same time, **Grammatik** will check your work for stylistic errors. Yes, your very writing style can now come under the scrutiny of computerized checking.

The Aspen manual says:

> By 'style' is meant a writer's particular choices for words and sentence forms. Although what is good or bad style is a subjective judgement (sic), particularly for word choice, there are some standards that experts agree lead to good style. While a document that conforms to the rules of 'good style' is not guaranteed to be coherent and readable, a document that fails to follow the rules is more likely to be confusing and difficult to read. Style analysis performed by **Grammatik** includes checks for specific phrases commonly recognized as being poor or wordy usage, as well as compilation of statistics about word and sentence length.

As a writer, editor, and former English tutor/student, I was particularly fascinated with this aspect of **Grammatik.** Before I say anything more about the wonders of **Grammatik's** syntactical-checking abilities, let's look at **Grammatik's** other abilities which are equally as important but not nearly so exciting.

WHAT GRAMMATIK CAN DO FOR YOU

Grammatik checks for typographical errors; specifically, it checks for those errors common in "computer generated" text, such as doubled words and punctuation marks and inconsistent capitalization. It also checks to make sure that the first word of a sentence is capitalized, that quotation marks and parentheses are balanced, and that punctuation is placed inside quotation marks. These features are truly invaluable!! If **Grammatik** did nothing more than check for these types of errors, it would be worth its weight in gold for this "sloppy typist" whose work always ends up with capital letters in the middle of words and spaces in the most unlikely places.

Grammatik also comes with a feature called **Profile** that "profiles" word usage. **Profile** lists each different word used in a document and indicates how many times it was used. By **Profiling** a document, you can keep a check on overworked words. Very often a writer will not realize how many times he uses a particular word, but alert readers will notice such a weakness.

GRAMMATIK'S COMMAND MENU OPTIONS

Unlike **Proofreader, Grammatik** is not especially easy to use the first time. I found the command menu to be extremely confusing, and quite a few tries were required before I got **Grammatik** to do what I wanted it to do. The manual goes into a lengthy explanation of the menu, but it neglects to tell the befuddled first-timer that only one command can be entered at a time. An overwhelming display of options is offered by the command menu, but if you wish to activate more than one, you must enter each one individually and then return to the menu after each entry. This

is not really a problem once you discover how it works, but I kept getting error messages when I tried to call up three or four of the options in one entry.

Grammatik does offer an astonishing array of things you can do to a document:

1.) You can have **Grammatik** read any number of dictionaries to check your work. All of **Grammatik's** checking is done via dictionaries, and it comes with two dictionaries of its own: *Phrases.GMK* and *Sexist.GMK*. *Phrases.GMK* is the main phrase dictionary used by **Grammatik** that contains over 500 phrases and words used to analyze your work. It includes words and phrases, error categories, and suggestions for alternatives. *Sexist.GMK* is a dictionary of about 100 words that are gender specific and possibly sexist. You may find this dictionary useful if you are interested in removing "gender specific" and "sexist" terms from your writing. (More about this later.)

2.) At the command level, you can select the types of errors you want **Grammatik** to show you. The default setting shows every error. But if you decide you don't want to see anymore errors of type "M" ("commonly misused words"), then you can command **Grammatik** to eliminate type "M" errors from its listing.

3.) **Grammatik** provides an option for WordStar users that allows formatting commands to be ignored. For example, any line beginning with the character "." can be ignored.

4., 5., 6.) Several different ways of looking at errors are offered by **Grammatik.** You can have errors listed in a file; you can have errors printed on the printer; you can have your entire document—with all errors marked—copied to a separate output file. You can also select a combination of these options.

7.) You can turn off echoing of your document on the screen when using **Grammatik.** This option could be used if you had the errors being printed on the printer and being copied to a file, and you didn't want to have to see them on the screen, too.

8.) When **Grammatik** encounters an error, it tells you what the error is, what kind of an error it is, and also offers you suggestions for correction or more effective use. At the command menu, you can elect to have the suggestions eliminated.

9.) When **Grammatik** encounters an error, it pauses in its checking to allow you to (a) decide whether or not to have this error marked in the output file, or (b) quit checking and exit to CP/M. If you don't want either of these choices, press RETURN and **Grammatik** continues checking. At the command level, you can eliminate the pause in checking; this allows **Grammatik** to check a file without pausing.

10.) Perhaps the best option in **Grammatik's** command menu is "Read Configuration File." You can set up a configuration file containing all the options you want to use routinely. This saves the time it takes to go through the menu, decide which options you want to use, enter each one individually, and then begin checking. My configuration file contains the commands: read dictionary *Phrases.GMK*, Ignore any sentences beginning with ".", List errors to output file with errors marked, and Print errors on printer.

With ten options to choose from, perhaps you can understand why I found the command menu bewildering the first few times I tried to use **Grammatik.** However, with ten options to choose from, very little escapes the checking ability of **Grammatik,** and this in turn insures a better finished product for you.

TESTING GRAMMATIK

In my review of **Proofreader,** I tested two paragraphs from Thomas Hardy's novel, Return of the Native. After the document had been thoroughly checked by **Proofreader,** several errors still remained because they were outside the scope of **Proofreader.** Note particularly the errors in lines 3, 4, and 6.

Let's see how **Grammatik** does with these two test paragraphs.

Grammatik and **Proofreader** reside on a separate disk from my **WordStar** disk, so to begin checking the test file, I type in:

A>GMK B:test.txt

When **Grammatik** is loaded, the following message appears on the screen:

Aspen Software Grammatik (tm) CP/M V1.81
(c) (p) 1981 Aspen Software Company

File Not Ready: CONFIG.GMK
<ENTER>when ready OR new file name OR
/ to abort

This message had me baffled when I first started with **Grammatik** since the directions in the Aspen manual section entitled "Starting Up The First Time" clearly state: "The 'C' command or 'CONFIG.GMK' should *not* be used

THE RETURN OF THE NATIVE
by
Thomas Hardy

Book First "THE THREE WOMEN"
Chapter 1: "A Face on Which Time Makes But Little Impression"

1. A Saturday afternoon in November was approaching the time of
2. twilight, and the vast tract if unenclosed wild known as Egdon
3. Heath embrowned itself moment by moment. overhead the hollow
4. stretch of whitish cloud shutting out the skY was as a tent which
5. had the whole heath for its floor.

6. THe heaven being spread with this pallid screen and the earth
7. with the darkest vegetation, their meeting-line at the horizon
8. was clearly marked. In such contrast the heath wore the appear-
9. ance of an installment of night which had taken up its place
10. before its astronomical hour was come: darkness had to a great
11. extent arrived hereon, while day stood distinct in the sky.
12. Looking upwards, a furze-cutter would have been inclined to
13. continue work; looking down, he would have decided to finish his
14. faggot and go home. The distant rims of the world and of the
15. firmament seemed to be a division in time no less than a division
16. iin matter. The face of the heat by its mere complexion added
17. half an hour to evening; it could in like manner retard the dawn,
18. sadden noon, anticipate the frowing of storms scarcely gener-
19. ated an d intensify the opacity of a moonless midnight to a
20. cause of shaking and dread.

when first running **Grammatik.**" However, the first thing I see when I try to use **Grammatik** is a message that appears to be a warning of some type: "File Not Ready. <ENTER> when ready OR new file name OR / to abort." I didn't know how to make Config.GMK ready so I couldn't hit <ENTER>; I didn't have a new file name to give it so I couldn't enter a new file name; and I certainly didn't want to "abort" (what a horrible word). "Abort" means: "to cause to be delivered prematurely;" or "to cut short in the early stages by proper remedies" (Webster's New International Dictionary, 2nd ed.). Neither of those two definitions sounds like what I wanted to do but what choice was there? So I hit "/" and *voila!* the main command menu appeared. As a bit of advice for the beginner: be on the alert for this message. The first thing **Grammatik** shows you is not very friendly.

Once at the command level, I selected the following options for the test: Read two dictionaries *(Phrases.GMK* and *Sexist.GMK),* Output file with errors marked, and Print errors on printer. After all options have been selected and entered, the actual checking process is started by typing in "//" which means "Done—begin checking."

The actual checking process is fast. **Grammatik** checked my test document, containing 257 words, in less than 60 seconds, and that includes the time it took to print the errors on the printer. Not bad. Of course, the test only contained seven problems. It's most likely other documents will contain more problems; consequently, **Grammatik's** checking time will increase. With **Grammatik,** the length of the document determines the length of the checking process.

As **Grammatik** checks your document, the text is displayed on the screen so you can read along. This is nice. You have to be able to see your errors in context or they are meaningless.

Grammatik pauses at every error it detects and gives you a chance to see what it considers to be an error. At this level, your only options are (1) to press RETURN to continue checking, (2) to press "N" to not mark the **Grammatik**-detected error on the output file (use this option if you do not agree with **Grammatik**), or (3) to press "Q" to quit checking. You cannot do any correcting at this point in the proceedings. Since the errors detected in the test document are being sent to the printer and to a separate output file (test.ERR), they are easy to find and correct.

The error messages sent by **Grammatik** are interesting. Not only are you told where the error is and what type the error is, you are also given suggestions for correcting it. This is the message printed by the printer. (See below.)

When **Grammatik** has finished checking, you have a hard-copy printout of the suggestions for correction flashed on the screen during the checking process.

The first "error" on the list from the test document is one the Sexist dictionary picked out. The use of the word "women" is suspect because it is a gender specific term. Well, let's hope it's gender specific and stays that way! I don't want to come down too hard on Aspen because I think they have a good

-------> women
* At sentence 1 — 'G' — Gender specific term
* Suggestion: persons

-------> overhead
* At sentence 2 — 'C' — Capitalization error
* Suggestion: 1st word of sent

-------> skY
* At sentence 2 — 'C' — Capitalization error
* Suggestion: mixed cases

-------> THe
* At sentence 3 — 'C' — Capitalization error
* Suggestion: mixed cases

-------> he
* At sentence 5 — 'G' — Gender specific term
* Suggestion: revise

-------> his
* At sentence 5 — 'G' — Gender specific term
* Suggestion: revise

idea here. *BUT*—and this is a big but—they can't seriously believe that words like "women" and further down on our list "he" and "his" should not be used in stylistically correct English!

Some of the other sexist words contained in the the *Sexist.GMK* dictionary are: "bellboy"—they suggest you substitute "attendant;" "boy"—their suggested alternative is: "child;" a "busboy" becomes a "clearer;" a "businessman," a "cameraman," and a "lady" are all reduced to a "person;" the word "manhours" shouldn't be used—"personhours" (one word!) is the preferred form; a "steward" and a "stewardess" become indistinguishable as an "attendant;" and the words "macho" and "manly" should be avoided altogether (no alternatives are even suggested for such anathema!). As a supporter of "women's lib," I am all in favor of anything that advances women beyond the ranks of housewifely drudgery, but this is going too far! How colorless, dull and *lifeless* will our language be if such words as "macho" and "stewardess" are removed. How tedious to have to say "personhours"—and how ridiculous. The *Sexist.GMK* dictionary included with **Grammatik** is fun to play around with (if you have the time to spare), but it has no place in a serious business application.

The second error, located in sentence 2, is a type "c" or capitalization error. **Grammatik** always checks the first word of a sentence to make sure it is capitalized. In this case, the first word in the sentence is *not* capitalized so the suggestion offered by **Grammatik** is that it should be. This is one of the errors that slipped by **Proofreader.** And the next two errors also went undetected by **Proofreader** but were caught and categorized as capitalization errors by **Grammatik:** "skY" in sentence 2 and "THe" in sentence 3.

Grammatik does not count lines; it only counts sentences. This can create problems when trying to locate errors in the original document if you didn't use the option that prints the text in an "output file with errors marked." The text below shows how the test document appears in the file: test.ERR. All errors detected by **Grammatik** are marked with ?# and the error type ("c" for capitalization, "g" for gender specific, etc.).

THE RETURN OF THE NATIVE
by
Thomas Hardy

Book First "THE THREE WOMEN"?#g
Chapter 1: "A Face on Which Time Makes But Little Impression"

1. A Saturday afternoon in November was approaching the time of
2. twilight, and the vast tract if unenclosed wild known as Egdon
3. Heath embrowned itself moment by moment. overhead ?#cthe hollow
4. stretch of whitish cloud shutting out the skY ?#cwas as a tent which
5. had the whole heath for its floor.

6. THe ?#cheaven being spread with this pallid screen and the earth
7. with the darkest vegetation, their meeting-line at the horizon
8. was clearly marked. In such contrast the heath wore the appear-
9. ance of an installment of night which had taken up its place
10. before its astronomical hour was come: darkness had to a great
11. extent arrived hereon, while day stood distinct in the sky.
12. Looking upwards, a furze-cutter would have been inclined to
13. continue work; looking down, he ?#gwould have decided to finish his?#g
14. faggot and go home. The distant rims of the world and of the
15. firmament seemed to be a division in time no less than a division
16. in matter. The face of the heat by its mere complexion added
17. half an hour to evening; it could in like manner retard the dawn,
18. sadden noon, anticipate the frowning of storms scarcely gener-
19. ated an d intensify the opacity of a moonless midnight to a
20. cause of shaking and dread.

Correcting the document with WordStar becomes quite simple; all you have to do is a Control -QF and Control -L to find every ?#, correct the error if you want to, reformat the line, and your work *should be* perfect. After all corrections are made and the text reformatted, you should rename the test.ERR file so it becomes your main file.

The paragraphs used in this test didn't truly reveal the extent of **Grammatik's** style-checking abilities. I've been using it for about three weeks now on all the articles I edit and all the work I write myself, and I haven't seen all the error types listed in the manual.

For example:
Error type A is "Archaic usage"—"used for words that are archaic and have fallen out of common usage."
Error type I is "Informal usage"—"some words, such as 'ain't', do not belong in formal writing."
Error type K is "Awkward usage"—"some phrases, such as 'and/or' are awkward and best revised."
Error type O is "Overworked or trite"—"Some phrases, while not being wordy or incorrect, are overworked."
Error type T is "Trademark"—"Certain words, such as 'Xerox', are trademarks."
Error type W is "Wordy phrase"—"the largest error category contained in the default dictionary is 'w'. Phrases such as 'all of' or 'a number of' can usually be replaced with more concise terms such as 'all' or 'several'."

With all the checks and controls available from **Grammatik,** we should expect nothing less than perfection. Aspen's ads in the various computer publications proclaim:
"**Grammatik + Proofreader** = No Errors"
Is this true?

Not really. Many of the same types of errors **Proofreader** was unable to detect are undetectable by **Grammatik;** i.e., "if" instead of "of" (line 2 in the test), "heat" instead of "heath" (line 16), "an d" instead of "and." Three errors in 257 words is not particularly accurate. And I did not purposely try to find errors to stymie **Grammatik;** these were simple, ordinary, every-day typing errors that occur all the time. If you have a particular problem with transposed

letters ("if" for "of", "form" for "from", etc), you can add these as an error type to the *Phrases.GMK* dictionary, and **Grammatik** will flag these words as errors. Having the checking process stop at "if" or "of" every time, however, could get really boring really quickly.

One of my favorite of **Grammatik's** many features is the summary displayed at the end of the error messages. This little box tells you all kinds of interesting statistics about your work, as you can see:

```
Summary for B:TEST.TXT / Problems detected: 6

# sent: 8 ; # words: 0257
avg sent len: 32.1 ; avg word len: 4.4
# questions: 0 ; # imperatives: 0
short sent (< 14 wds): 0 ; long sent (> 30 wds): 3
longest 66 wds at sent # 1 ; shortest 23 wds at # 2
to be's: 7 ; prepositions: 35
User category totals:
   NONE
```

The reasoning used for the inclusion of these statistics is that ". . . documents with many long sentences and polysyllabic words tend to have stylistic difficulties." Also, the number of times you use common prepositions and forms of "to be" can also provide an indication of style. The Aspen manual instructs the reader to compare the total number of "to be's" and "prepositions" to the total number of sentences. "A high ratio may indicate overuse of these word classes and suggest a possibility for revision."

The final line of the summary, "user category totals," is provided for you to program in your own problem areas. If you know that you have a tendency to overwork certain words and shouldn't use them quite so often, **Grammatik** will accurately count for you how many times these problem words were used. Then you can decide whether to revise or not.

Profile is another way to keep track of your word usage. Here is the **Profile** printout for the test paragraphs:

Notice that each different word in the document is listed according to how many times it was used. Since "words that appear only once are most likely to be misspelled," **Profile** provides the means for you to check your spelling one last time. You can also make sure that you have not overworked certain words.

SHOULD YOU BUY GRAMMATIK?

```
Aspen Software Grammatik(tm) Word Use Profile V1
Total number of different words: 174

********** 1 **********
added      addressee  afternoon  ance       anticipate            appear     approac
arrived    art        aspen      astro      at         ated       aug        author
be         been       before     being      book       bsu        but        cause
chapter    clearly    cloud      come       comple     continue   contrast   could
cutter     d          darkest    darkness   date       dawn       day        decided
dis        distant    down       dread      earth      egdon      embro      evening
extent     faggot     finish     firmament  first      floor      frowning   furze
gener      go         great      half       hardy      he         heat       heaven
hereon     hing       his        hollow     home       horizon    if
impression            inclined   installment           intensify  it         itself
known      less       line       little     makes      manner     marked     matter
meeting    mere       mid        moonless   mt         n          native     no
nomical    noon       november   on         op         opacity    out        overhead
pallid     paragraphs            place      po         retard     return     review
rims       sadden     saturday   scarcely   screen     seemed     shaking    shutting
spread     stood      storms     stretch    subject    such       taken      tent
test       than       their      this       thomas     three      tinct      tion
tract      twilight   unenclosed            up         upwards    vast       vegeta
while      whitish    whole      wild       wned       women      wore       work
world      xion

********** 2 **********
as         division   face       for        have       hour       looking    moment
night      sky        with       would

********** 3 **********
an         by         had        heath      time       which

********** 4 **********
its        was

********** 5 **********
and

********** 6 **********
in         to

********** 10 **********
a

********** 11 **********
of

********** 23 **********
the
```

This discussion and the test was intended to provide at least a cursory introduction to Aspen's **Grammatik.** However, the main reason a magazine reviews products is so it can recommend whether readers should purchase or avoid the reviewed product. With **Proofreader,** I had no problem: if you ever misspell words, go ahead and buy it.

I do recommend purchasing **Grammatik,** but the recommendation to buy is more qualified. It does catch some typographical errors, and that alone is more than useful. But the style checking feature has not been helpful for me. It really isn't fair for a professional writer to review such a product. My job is to be aware of style and syntax and know the rules for correct usage; I don't need a piece of software to tell me the difference between "affect" and "effect" or that "a number of" represents wordy usage. For the small businessman who struggles with every letter he has to produce or for the junior executive who wants to impress his superiors with his writing style, **Grammatik** is probably worth every penny it costs.

In a recent ad, Aspen claims that **Grammatik:**

- .Detects typos, punctuation errors, capitalization errors, unbalanced parentheses and quotations marks
- .Analyzes your writing style for average sentence and word length, word use profile and more
- .Finds non-standard usage and spelling errors
- .Finds overworked and wordy phrases using customizable dictionary containing over 500 entries
- .Errors optionally marked for easy correction, sent to printer or displayed on screen.

This is all absolutely true. But don't expect **Proofreader** to be able to figure out what you *really* want to say or what you *really* mean. Don't completely substitute it for your own proofreading. And don't rely on it too much; you could be sorry.

What I'm waiting for is the program

that will tell you whether a subject and verb are in agreement, whether a comma is needed in a certain place, whether a pronoun and its antecedent agree, or whether a semicolon or colon is the proper punctuation. Now that will be a true "grammar checker!" As the Aspen manual informs its readers:

> ". . . there are things that Grammatik will not do. Specifically, Grammatik does not know anything about the meaning of words. Anything that requires an understanding of meaning, such as subject-verb agreement, cannot be detected. Such software technology is still some years away."

It is my most fervent hope that such software technology is not *too many* years away; the entire future of our language may be at stake! Admittedly, the English language is difficult to master, and even the best of us have problems unravelling complex syntax on occasion. Proponents for drastic revisions to the "King's English" abound in every sector, even academia where one supposes the traditional idiosyncrasies and characteristics of the language would be cherished and preserved.

However, if Aspen Software could come up with a "real" grammar-checker, one that would make the mysteries of our language more understandable and less mysterious for everyone, the type of suggestions as exemplified by the following could be eliminated permanently:

> Two English teachers at Eastern New Mexico University have come up with a solution to the problem some students have been having mastering the rules of English grammar and punctuation. They've received a grant to produce a series of programs explaining it on public television.
>
> Their plan is so simple it's a wonder nobody thought of it before. The professors suggest we abolish all those silly rules.
>
> We can start, they say, by doing away with the apostrophe. Thats simple enough, theyre pesky things anyway. Then, they say, we should get rid of commas. Good idea theyre hard to remember clutter up the page messy. Then they tell us spelling isnt importint not essenshal for comprehenshion just slows us down looking in the dicshunary an all that.
>
> Grammer an speling are according to these too experts part of an eleetist atiude that confusing students making them mixedup an nervus about expressing theirselves and anyway rules is illogical and arbitrary.
>
> Boy itsa good thing weve got experts like those in this coutry or wed be in reel terible shape.

(Our special thanks to the Keene Sentinel in Keene, NH for permission to reprint this article. Thanks also to InforWorld [Vol 4, No. 37, Sept. 20, 1982] for giving us the idea.)

"SNAPP TO IT" SNAPP-WIRE'S SNAPP-III MAKES THINGS HAPPEN

Dr. Arnold H. Fischthal
Advanced Data Design, Inc.
184-08 Tudor Road
Jamaica Estates, NY 11432

Snapp-Ware has developed an entire series of products that ease the burdens of software development. In this issue we will look at SNAP-III Extended Builtin Functions, language enhancements for the Model II BASIC Interpreter. SNAPP-III adds a collection of over 30 useful commands to the BASIC language, all designed to greatly extend its convenience and utility.

The product is easily installed on any system disk. Even though several patches must be made to the BASIC interpreter so the new commands introduced by SNAPP-III will be recognized, total installation time, which includes reading the manual to make sure you are doing it correctly, is under two minutes. The software takes up no user space in RAM or on disk.

The new SNAPP-III commands are accessed in one of two ways:

(1) variable = FN builtin-function-name (argument[s])

or

(2) SCMD verb

When one of the new commands is accessed, the routines needed to carry out the command are pulled in from disk and stored in non-user accessible RAM. If the same command is used again without accessing another command from SNAPP-III, there will be no second disk access.

SORTING:

I have found the most beneficial routine to be the MACHINE LANGUAGE SORT. This feature alone is well worth the price of the entire package for those of you who need it. For example, suppose several arrays have been DIMed: A$(2000), B$(2000), RN%(2000). And suppose data has been entered into these arrays so that A$ holds last names, B$ holds zip codes, and RN% holds the file's record numbers where this information is stored on disk. Further, suppose that it was necessary to dimension up to 2000, but only the first CT of them hold data. By entering the line

II = FNSRT("+A$,+B$,RN%",CT)

the entire information block will be sorted first by name (the A$), then within a name (maybe Smith) by zip code (the B$). Adding RN% means the RN% ar-

ray goes along for the ride but does not take part as a sort key. It took only 12 seconds to sort 1000 twenty character strings that began in completely reversed order.

As this example illustrates, the FN is used to access the sort routine. In most such cases, the II acts as a dummy variable whose value after the FN is not used.

BUFFERS:

As you move from BASIC program to BASIC program, you soon realize that when one program requires only two file buffers and another program requires seven file buffers, then seven buffers must be made available when entering BASIC. This is an annoying limitation inherent in BASIC that results in the waste of RAM accompanying unused buffers. However, with SNAPP-III, this need no longer by the case.

By issuing the command

SCMD CLEAR 3,3000,61000

at the beginning of a program, you will not only reserve 3000 bytes of string space and set HIMEM to 61000, but you will also reset the number of file buffers available to 3 WITHOUT needing to issue a SYSTEM BASIC! By the way, typing PRINT FNFILES will print the number of buffers currently available.

STRING COMPRESSION:

Have you ever wished for a disk stretcher to turn your 8" disk into a 9" disk so you could store more data? If so, this next routine is for you. It can save approximately 25% of disk space by packing strings into a smaller space. The command FNPK actually compresses every four bytes of string into three.

For example,

B$ = FNPK$("ABCDEFGH")

will yield a six byte encoded form of "ABCDEFGH" to be stored in B$. Then

A$ = FNUPK$(B$)

restores it so that A$ = "ABCDEFGH"

However, even a feature this good isn't perfect: Only characters in the range 20H to 5FH can be compressed. This restriction, unfortunately, eliminates the lower case alphabet and certain special characters.

DATE ARITHMETIC AND DATE PACKING:

The command FNPDAT performs a dual function: like FNPK, it conserves disk space. In addition, it allows you to do arithmetic on dates.

B% = FNPDAT("10/01/81")

will give B% the value 11596. This feature allows the date to be sorted as an integer using only two bytes in the process.

D$ = FNUDAT$(11596)

will cause 10/01/81 to be stored in D$. In effect, FNPDAT "packs" a date into two bytes and FNUDAT$ "unpacks" it.

Converting dates to integers allows arithmetic operations on them. For example, adding 30 to 11596 and then unpacking will yield the date 10/31/81, which is 30 days later. The function FNXDAT$ yields the current date in the form MM/DD/YY as entered into the system at boot-up time.

TIME ARITHMETIC:

The function FNXTIM$ gives the current time in the form HH.MM.SS. A related function, FNETIM$, calculates the difference between two times; the result is given in the form HH.MM.SS. In conjunction with FNXTIM$, FNETIM$ can be used to calculate elapsed time within a portion of a program.

PARTIAL SCROLLING:

SNAPP-III contains several commands that affect the CRT. SCMD ROW LN fixes the upper LN lines of the screen so that only the bottom 24-LN scroll upwards. Issuing a CLS countermands that order. SCMD "CURSOR" can adjust the cursor size (from 1-10 scan lines) and can change the speed at which the cursor blinks. SCMD "LMSGOFF" and SCMD "LMSGON" turns BASIC's long error messages on and off.

XX$ = FNVIDEO$(A,B,C)

causes the first C characters on the video display starting at row A, column B to be stored in XX$. This is good if you want only certain portions of the screen sent to the printer. However, inverse on the screen might not be accurately reproduced on the printer if you use FNVIDEO$.

PEEK AND POKE:

For those of you who like to peek and poke around inside of RAM, SNAPP-III has six different commands for you to use:

(1) To read the contents of a certain byte—FNPEEK
(2) To extract two bytes from a specific memory location—FNPEEKW
(3) To poke a specified value into a specified memory location—FNPOKE
(4) To poke a specified word (2 bytes)—FNPOKEW
(5) To read up to a 255 character string from RAM—FNPEEK$
(6) To poke up to a 255 character string into RAM—FNPOKE$

A particularly convenient use of FNPEEK I have found is FNPEEK (&H0F6B). This tells me the number of lines printed on the current page so that I will know when to issue a page feed (with the appropriate heading and page number) by simply testing this value instead of counting every single line sent to the printer.

SOME OTHER USEFUL COMMANDS:

The following commands are also resident in SNAPP-III:

SCMD RUN filespec,N: runs the program 'filespec' and sets the number of file buffers available to N.

SCMD OPEN "E",N,filespec: opens the sequential file 'filespec' as file number N in extended mode, i.e. data can be added to the end of a sequential file without destroying the data already there.

SCMD ERASE: erases all arrays with this single command.

FNID$: reads any or all of the disk IDs (0–3).

FNUC$(A$): converts any lower case letters in A$ to upper case. This is good for use in conjunction with FNPK$ to make sure all elements are upper case.

FNLC$(A$): converts any upper case letters in A$ to lower case.

SCMD "VDOFF": turns the video display off. This is useful to avoid looking at a flickering screen while a program is updating data on the CRT.

SCMD "VDON": turns the video display back on.

FNMAX (A,...): returns the maximum value of the numerics from the user supplied list.

FNMIN(A,...): returns the minimum value.

FNFMT$(...): will arrange specified items into a string of a specified format, i.e. the result is a string with the data correctly formatted in that string.

SCMD DELETE: allows the deletion of lines from a BASIC program during execution without returning to the Ready prompt when it is done.

SCMD "SCON": alters the string comparison logic so that shorter strings are padded with blanks in comparison operations.

SCMD "SCOFF": countermands SCMD "SCON".

Occasionally when using SNAPP-III you may encounter the drawback which exists because the routines are stored on disk. This can occur when two of these new commands (or some of the commands from other Snapp-Ware products) are used in close proximity to each other. When the second command is used, the routine pulled in from the disk might overlay the routine of the last command used so that, for example, in continuous disk file reads it is not inconceivable to have:

(1) READ DISK FILE (move head over there)
(2) Perform SCMD (causing head to find routine)
(3) Perform Builtin Function (causing another disk head move)
(4) READ DISK FILE again . . . etc.

You will not witness such an occurrence on all occasions where two commands are used together but only when their routines are stored in the exact same place in RAM. I would like to point out, however, if only lines one and two above are repeated, there will be no repetitive head movement. Using the same command immediately after itself causes no second disk read.

I would like to repeat here what I said in my previous review of SNAPP-WARE products ("A Review of Snapp-Ware's SNAPP—II BASIC", p. 6, July/August issue): the customer service offered by Bob Snapp and his company is excellent! Questions will be answered; comments, compliments, and/or criticisms will be accepted—all with extreme courtesy and usually by Bob himself.

SHORT FORM DIRECTORY FOR THE MODEL II

David R. Canning
P.O. Box 55
Oak Creek, CO 80467

Does your Model II spit out a directory faster than you can read it? Does the file name you really wanted to see scroll off the screen before you can find the HOLD key? Do you have to read the screen twice just to find the file name you're looking for? Would you like to see all the files on a disk displayed on a single screen? If you answered "yes" to any of these questions, my Short-Form Directory is for you.

The program following this introduction, XDIR, will produce a Short-Form Directory (like the Model I), with the listing in alphabetical order. File names will be printed to the screen five across, for a screen capacity of 115 file names before scrolling. Since you can't get more than 96 user files on a disk, you can forget about hitting the hold key.

As written, XDIR is tailored for a system with two drives. To make it fit your system, change line 1140 to CP the number of drives you have on line (byte @ 4019 hex).

For those without an assembler, the program can be keyed in with DEBUG using the hex digits in Table 1. Use the following procedure:

Under DOS READY, enter the command:

CLEAR

Enter: DEBUG ON
Enter: DEBUG
Press: M
4000
the F1 Key

[REMEMBER: Don't hit the "Enter" key when you're in DEBUG]

Enter: hex digits from 4000 to 407F
As per Table 1

If you make a mistake, you can move the cursor with the arrow keys and correct it.

After you enter the byte @407F, the curser will jump back to 4000.
At this point:

Press: F2 key
Press: M
4080
the F1 Key
Enter: hex digits from 4080 to 40B7
(as before)

After byte 40B7:

Press: F2 key
Then press: the letter O
(to exit DEBUG)

At DOS READY, type the following command:

DUMP XDIR START = 4000, END = 40B7, TRA = 4000
(To dump the program to a disk file.)

The Short-Form Directory can now be displayed any time in DOS by entering

XDIR

and answering the prompt for the desired drive number.

Note: Since XDIR occupies memory shared with BASIC, it cannot be invoked by a SYSTEM command from BASIC.

```
00900 ;          *** XDIR4/TXT ***
00910 ; Displays Short-form Directory Alphabetically
00920 ; By David R. Canning, Oak Creek, CO
00930 ;
01000   ORG     4000H
01010 DMES      LD      HL,MSG1         ;Display Prompt
01020   LD      B,8
01030   LD      C,20H
01040   LD      A,9
01050   RST     8
01060   LD      A,1             ;Clear Stored Keystrokes
01070   RST     8
01080 GETC      LD      A,4             ;Get Drive # (ASCII)
01090   RST     8
01100   JR      NZ,GETC         ;No key hit - Try again
01110   LD      A,B
01120   LD      (CRET+1),A      ;Store Drive #
01130   XOR     30H             ;Conv't to Binary
01140   CP      2               ;n (2) = Max Drive #
01150   JR      NC,DMES
01160   LD      B,A             ;RAMDIR Call
01170   LD      A,53
01180   LD      C,0
01190   LD      HL,BUF1
01200   RST     8
01210   JR      NZ,ERM
01220   LD      HL,BUF1         ;Find end of DIR Buffer
01230   LD      A,(HL)
01240   CP      3AH             ;= :
01250   JR      NZ,ERM2
01260   LD      B,0
01270   LD      C,34            ;34 = Length of DIR entries
01280 SRCH      ADD     HL,BC
01290   LD      A,(HL)
01300   CP      3A
01310   JR      Z,SRCH
01312   SCF
01314   CCF
01320   SBC     HL,BC           ;Set HL to Start of Last entry
01330 SRT       LD      A,56            ;SORT Call
01340   LD      IX,BUF1
01350   PUSH    HL              ;LD DE w/
01360   POP     DE              ; HL
01370   LD      B,1
01380   LD      C,34
01390   LD      H,0
01400   LD      L,15
01410   RST     8
01420   JR      NZ,ERM
01430   LD      B,1             ;Clear Screen
01440   LD      C,1
01450   LD      A,7
01460   RST     8
01470   LD      D,0
01480   LD      E,34
01490   LD      HL,BUF1
01500 OPT       LD      A,(HL)
01510   CP      3A
01520   JR      NZ,DONE
01530   INC     HL
01540   PUSH    HL              ;Change ":d0D"'s
01550   PUSH    DE              ; in DIR Buffer
01560   LD      DE,CRET         ; to Spaces
01570   LD      B,2
01580   LD      A,49            ;STSCAN Call
01590   RST     8
01591   LD      (HL),20H
01592   INC     HL
01593   LD      (HL),20H
01594   INC     HL
01600   LD      (HL),20H
01610   POP     DE
01620   POP     HL
01630   LD      B,14            ;VDLINE Call
01640   LD      C,9
01650   LD      A,9
01660   RST     8
01670   DEC     HL
01680   ADD     HL,DE
01690   JR      OPT
01700 ERM       LD      HL,MSG2
01710   LD      B,14
01720   JR      PMSG
01730 ERM2      LD      HL,MSG3
01740   LD      B,10
01750 PMSG      LD      C,20H
01760   LD      A,9
01770   RST     8
01780 DONE      RST     0
01790 ;
01800 ;
01810 MSG1      DEFM    'DRIVE #?'
01820 MSG2      DEFM    'DIR READ ERROR'
01830 MSG3      DEFM    'SORT ERROR'
01840 CRET      DEFM    ':0'
01850 BUF1      DEFB    0
01860 ;
01870   END     DMES
```

```
4000  21 95 40 06 08 0E 20 3E 09 CF 3E 01 CF 3E 04 CF   * ! . @ . . .   } . 0 } . 0 } . 0 *
4010  20 FB 78 32 B6 40 EE 30 FE 02 30 E4 47 3E 35 0E   *   { x 2 6 @ n 0 ~ . 0 d G } 5 . *
4020  00 21 B7 40 CF 20 5C 21 B7 40 7E FE 3A 20 5B 06   * . ! 7 @ 0   \ ! 7 @ ~ ~ :   [ . *
4030  00 0E 22 09 7E FE 3A 28 FA 37 3F ED 42 3E 38 DD   * . . " . ~ ~ : ( z 7 ? m B } 8 ] *
4040  21 B7 40 E5 D1 06 01 0E 22 26 00 2E 0F CF 20 33   * ! 7 @ e Q . . . " & . . . 0   3 *
4050  06 01 0E 01 3E 07 CF 16 00 1E 22 21 B7 40 7E FE   * . . . . } . 0 . . . " ! 7 @ ~ ~ *
4060  3A 20 31 23 E5 D5 11 B5 40 06 02 3E 31 CF 36 20   * :   1 # e U . 5 @ . . } 1 0 6   *
4070  23 36 20 23 36 20 D1 E1 06 0E 0E 09 3E 09 CF 2B   * # 6   # 6   Q a . . . } . 0 + *
4080  19 18 DB 21 9D 40 06 0E 18 05 21 AB 40 06 0A 0E   * . . [ ! . @ . . . ! + @ . . . *
4090  20 3E 09 CF C7 44 52 49 56 45 20 23 3F 44 49 52   *   } . O G D R I V E   # ? D I R *
40A0  20 52 45 41 44 20 45 52 52 4F 52 53 4F 52 54 20   *   R E A D   E R R O R S O R T   *
40B0  45 52 52 4F 52 3A 30 00 00 00 00 00 00 00 00 00   * E R R O R : 0 . . . . . . . . . *
40C0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   * . . . . . . . . . . . . . . . . *
40D0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   * . . . . . . . . . . . . . . . . *
40E0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   * . . . . . . . . . . . . . . . .
40F0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   * . . . . . . . . . . . . . . . . *
4100  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   * . . . . . . . . . . . . . . . . *
```

KILL THE KILL PROMPT WITH KILLABS

by Roger Conant
1108 SEO
UIC
Box 4348
Chicago, IL 60680

```
Type in:

                        DEBUG ON

                         DEBUG

(Reminder:  Don't Press Enter in DEBUG!)

Press:

                           M
                         3000
                        F1 Key

Enter:

              hex digits from 3000 to 307F
                   (as per Table 1)

Press:

                        F2 Key
                     (The Letter) O
                   (Turns DEBUG Off)

In response to TRSDOS READY prompt, type in:

     DUMP KILLABS (START=3000 END=307F TRA=2800 RELO=2800)
```

```
DUMP KILLABS (START=3000 END 307F RELO=2800 TRA=2800)

TRSDOS READY

KILLABS NONE                        :COMMAND TO KILL A NON-EXISTENT FILE

-- ERROR -- FILE NOT FOUND

TRSDOS READY

DIR                                 :LOOKING FOR NAME OF FILE TO KILL

FILE NAME! UPDATED ! TYPE! REC.LEN! # RECS! SECTORS
   JUNK  ! 09 08 82!  F  !   256  !   1   !   1

TRSDOS READY

KILLABS JUNK                        :COMMAND TO KILL A LEGITIMATE FILE
                                    (FILENAME JUNK

TRSDOS READY                        (NOTICE NO PROMPT!

DIR JUNK                            :CHECKING TO SEE IF FILENAME JUNK
                                    :IS  GONE

FILE NAME! UPDATED ! TYPE ! REC.LEN !# RECS ! SECTORS

TRSDOS READY                        :FILENAME JUNK HAS BEEN DELETED
```

TABLE 2

Should you be annoyed, as I am, by the TRSDOS prompt given whenever you want to delete a file, try the following small program, KILLABS, that does away completely with the unnecessary prompt associated with the KILL command.

Once the KILLABS program is entered, you will be able to kill a file without any prompt at all. If KILLABS cannot find the file you instruct it to delete, you will get an error message. See Table 2 for a print-out showing KILLABS in action.

KILLABS rings the printer bell as an error warning. This sequence of code—06073E12CF—can be replaced by 0000000000 if you don't have a printer bell.

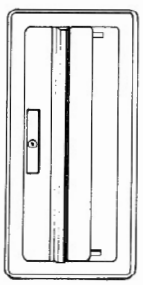
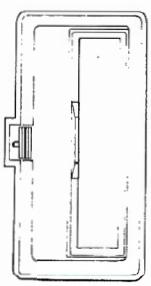
```
3000  01 09 00 09 54 5D 3E 29   CF 20 01 C9 47 F5 21 31     ....T]>). ..G.!1
3010  28 3E 34 CF 06 07 3E 12   CF 21 25 28 06 4E 0E 0D     (>4...>..!%(.N..
3020  3E 09 CF F1 C9 2D 2D 20   65 72 72 6F 72 20 2D 2D     >....-- ERROR --
3030  20 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00     ................
3040  00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00     ................
3050  00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00     ................
3060  00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00     ................
3070  00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00     ................
   PC   SP  SZHPNC  AF   BC   DE   HL   IX   IY   AF'  BC'  DE'  HL'
 2800 21FE 000000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
```

TABLE 1

Reader Survey Volume 1, Number 3 Sept–Dec, 1982

Please return to **two/sixteen,** Box 1216, Lancaster, PA 17603

Based on comments we have received from subscribers, it appears that certain hardware is producing considerably more service problems than others. In this survey, we ask those readers with Model II expansion drives to comment on their reliability. More than one type of drive has been used in expansion bays. Please indicate the number of drives you have in your expansion bay under the diagram of the type drive you have. Indicate also your rating of the drives' reliability on a scale of 0 to 9 (0 = totally unreliable, 5 = average, 9 = perfectly reliable). We will forward the results of this survey to Tandy for their comment.

	"push-button"	"latch"
		
Number	_____	_____
Rating	_____	_____

Comments _____

Would you be interested in a double-sided expansion bay for your II? _____

We also seek further data about operating systems used. Please rate each system you use on the 0 to 9 (useless to perfect) scale:

Computer:	Model II	Model 16	Comments:
TRSDOS 1.2	_____	_____	_____
TRSDOS 2.0	**a** _____	**b** _____	_____
TRSDOS 4.	**0** _____	**1** _____	_____
TRSDOS II		_____	_____
TRSDOS 16		_____	_____
DOSPLUS II	_____	_____	_____
CP/M:			_____
P&T 2.2e	_____	_____	_____
P&T 2.2eD	_____	_____	_____
Aton _____	_____	_____	_____
Lifeboat _____	_____	_____	_____
FMG _____	_____	_____	_____
Turbodos _____	_____	_____	_____
Other _____	_____	_____	_____
OASIS _____	_____	_____	_____
Other _____	_____	_____	_____

Optional responses (fill in all, any part, or none):

Name: _____

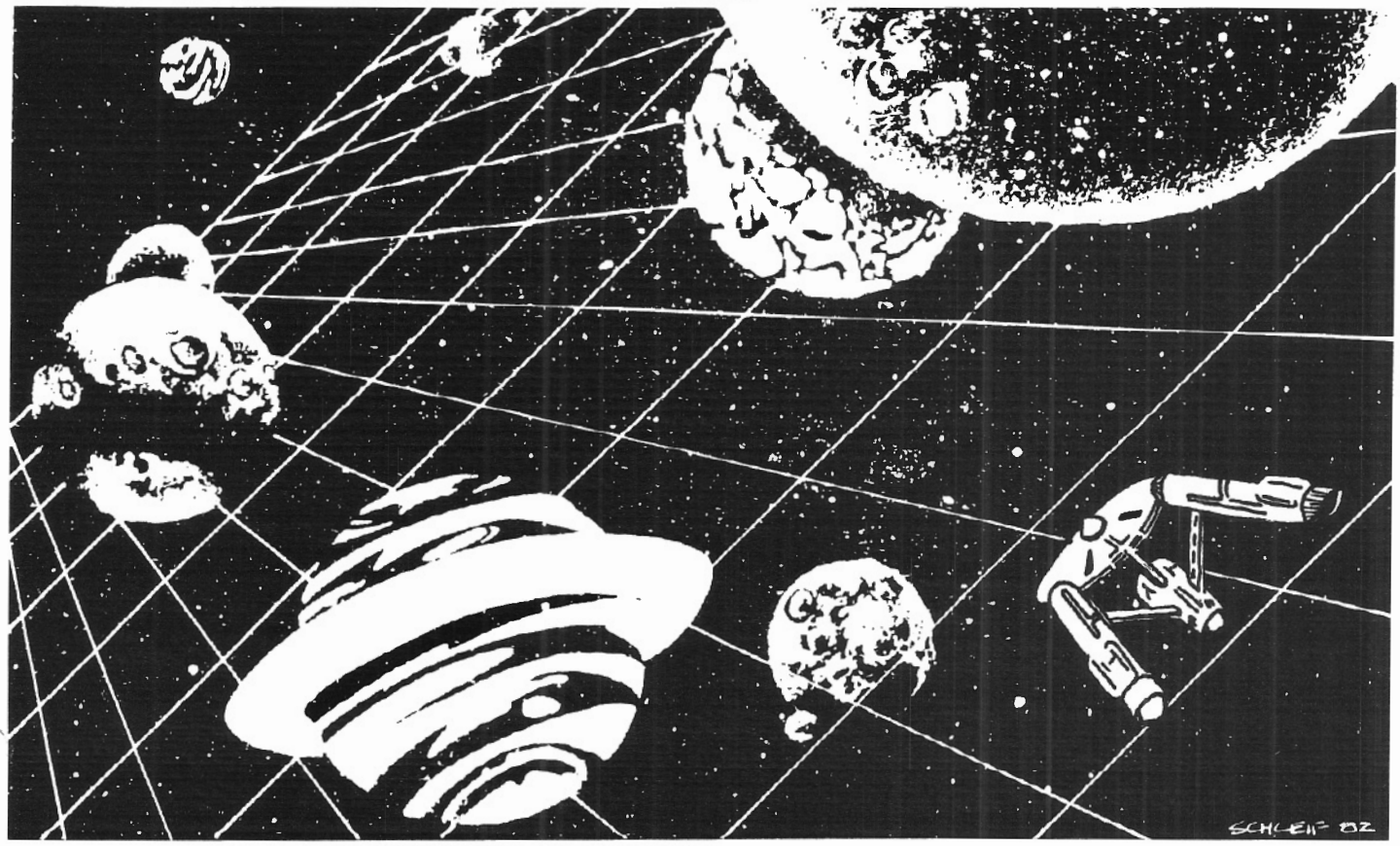
Address: _____

City, State, Zip: _____

Occupation: _____

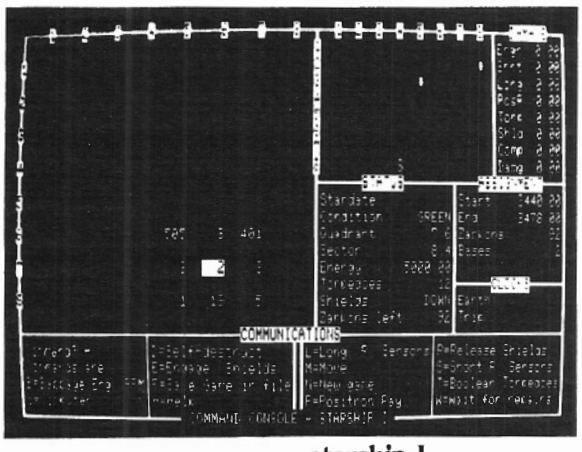
Our mailing list may be made available to selected vendors who provide only products which are useful to owners of Model II and 16 microcomputers. If you DO NOT wish to receive such mailings, please check here: _____

STARSHIP I



STARSHIP/I (copyright registered) is a save-the-galaxy type game based on the venerable STARTREK games which have been played on large mainframes for more than ten years. It is specifically designed for the RS Model II and features a constant console display with no scrolling. Each game begins in a randomly arranged galaxy of 64 "quadrants," each containing 64 "sectors."

STARSHIP/I is a "strategic" game (as opposed to "tactical" games which involve reflex action but little thought). However, STARSHIP requires *quick* thinking, as all events are timed. Because STARSHIP is written in the RS BASIC interpreter language, users can modify the source code or use sections of the code as models to learn techniques which can be used to produce games of their own design.



starship I

STARSHIP comes with a disk instruction file which can be listed on your printer. It can also be listed on the screen at the beginning of a session. In addition, STARSHIP contains a few undocumented "surprises" at advanced levels of play (there are ten levels of difficulty). These surprises may be analyzed by reviewing the source code, then modified or deactivated if desired.

Games may be saved in files at any point for later play or for "duplicate" games in which several players start with the same layout and position and compete for the highest ending score.

STARSHIP/I was sold for several months at $50 through leading computer magazines. It is now available to **two/sixteen** subscribers at the special price of $30, including air mail (anywhere) in protective Kangaroo mailer.

(THIS PAGE INTENTIONALLY LEFT BLANK)