VOLUME 3 NUMBER 1 MAY/JUNE 1984

ADVANCED COMPUTING™

FORMERLY TWO/SIXTEEN

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

INSIDE:

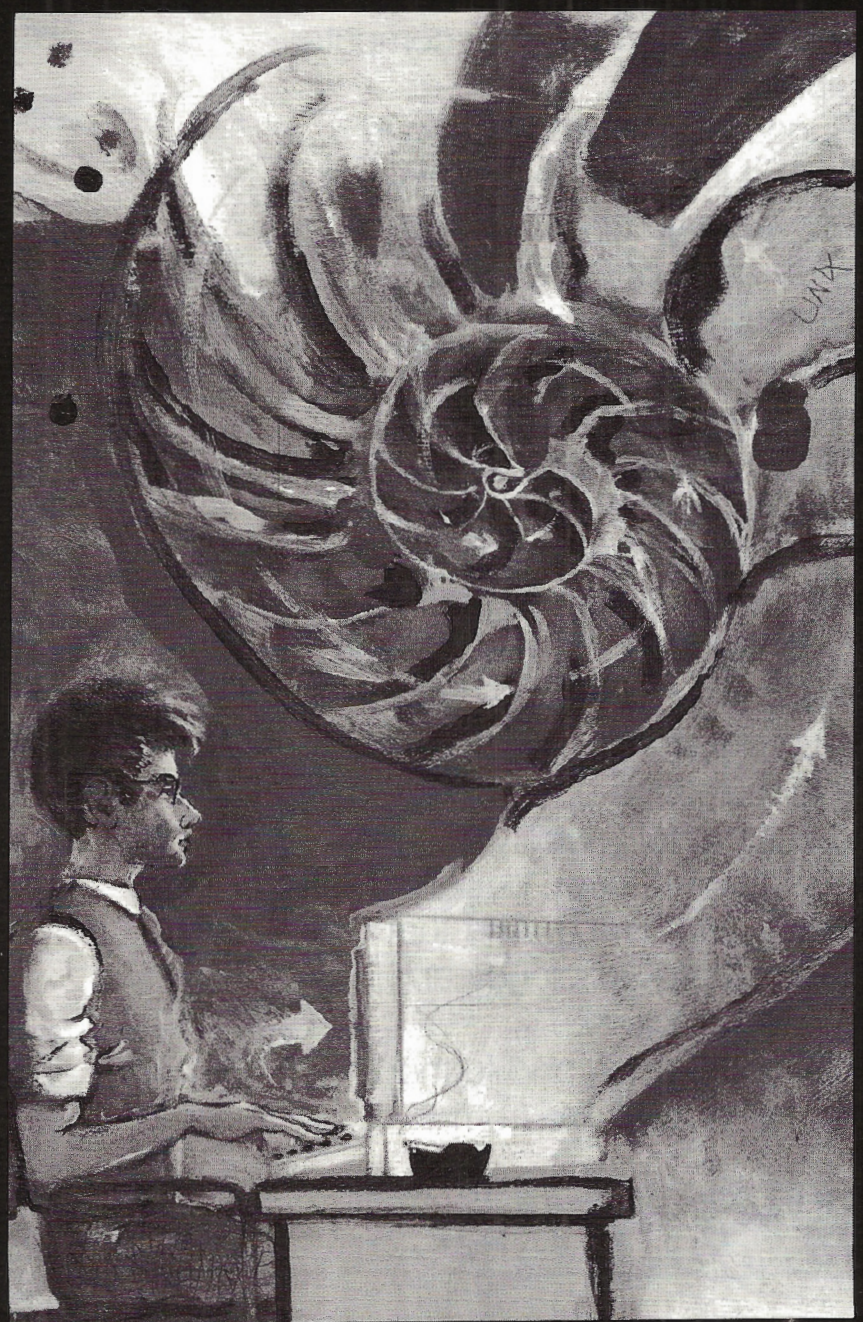
REVIEW OF THE STATISTICIAN
A STATISTICAL TOOL FOR BUSINESS 24

A DISK COPY QUEUE UTILITY
TRANSFER FILES EASILY WITH COPY/BAS 22

XENIX GRAPHICS
USE SCREEN CONTROL SEQUENCES TO CREATE GRAPHICS 28

ARCNET: THE SLEEPING GIANT
SPECIAL USER REPORT 34

PLUS MORE...



THE SCENERY IN XENIX PART II
PERMISSION, SHELLS, AND COMMUNICATIONS 4

COMPUTER FURNITURE BY Data-MATE®

INTRODUCTION

DATA-MATE® computer furniture is carefully designed and engineered to provide higher efficiency and productivity. It is far more than simply a support for your computer equipment.

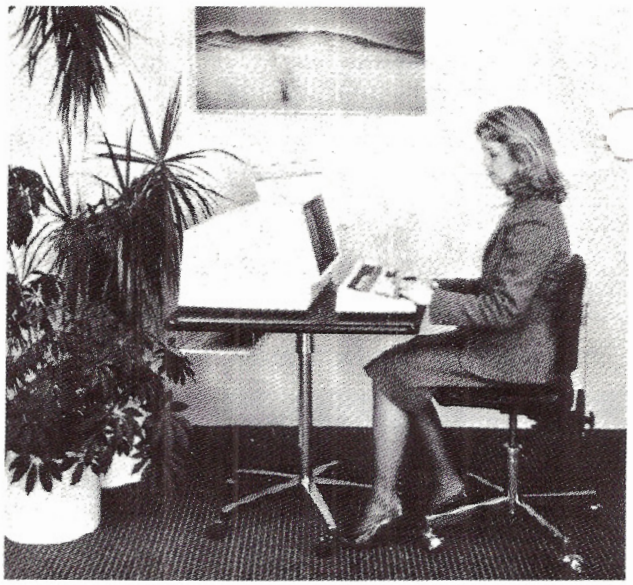
Rugged steel construction is designed for durability. Cable and paper management systems provide efficiency *and* neat, uncluttered appearance. All equipment bays have ample ventilation for heat dissipation. Clean lines blend perfectly with hardware design and function.

Data-MATE® products are in stock for immediate shipment.

Operator comfort is becoming increasingly more important. Features such as height adjustments to reduce operator fatigue and casters for mobility and versatility have been designed into the Data-MATE line. And that means greater efficiency and higher productivity.

Catalog, prices, and further information available from

ADVANCED COMPUTING (717)397-3364



Excellence in computer furniture technology.

VOL. 3 NO. 1 MAY/JUNE 1984

ADVANCED COMPUTING™

STAFF

Publisher	**Richard H. Young**
Managing Editor	**Barbara S. Albert**
Assistant Editor	**Sherry A. Symonds**
Art Director	**Timothy P. Schleif**
Director of Advertising/Circulation	**William R. Holland**
Director of Technical Services	**Mark A. Ingoglia**
Business Manager	**Margaret A. Holt**

Typesetting by **Forry & Hacker Lititz, PA**

Printed by **Forry & Hacker Lititz, PA**

Advanced Computing (ISSN 0743-4278) is published bi-monthly by ***Advanced Computing*** 31 East Orange Street, Lancaster, PA 17602

Telephones

Editorial Staff	**717/397-3365**
Classified Advertising	**717/397-3789**
Display Advertising	**717/397-3789**
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 and Mexico	**$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 ***Advanced Computing*** Box 1216, Lancaster, PA 17603

Advanced Computing welcomes letters to the editor. All letters sent to the magazine become the property of ***Advanced Computing*** and are subject to publication unless otherwise requested by the author.

CONTENTS

EDITORIAL

Notes & Comment — Richard H. Young — 2

XENIX

The Scenery in XENIX (Part Two) — John Esak — 4
Permissions, shells, and communications are the territories explored this issue.

More on Inverted Indexes — Richard Bilancia — 14
Short shell programs can perform simple on-line searches.

XENIX Graphics — Jean-Pierre Radley — 28
Use these screen control sequences for XENIX MBASIC graphics.

SOFTWARE

The Statistician — Paul Naitoh — 5
The statistical tool for better business management.

The RSCOBOL Generator — Ian B. Kazin — 12
RSCOBOL Utilities from David Ray
These two tools are invaluable for any COBOL programmer.

Review of RED and VEDIT — Robert P. VanNatta — 18
Compare the text editors, one new and one updated.

Review of LUCID — Arnold Fischthal — 32
An on-line HELP routine makes LUCID a user-friendly DBMS.

Review of Milestone — Richard Trahan — 42
The project management package for micros.

SPECIAL USER REPORT

ARCNET, The Sleeping Giant — Bro. Gary Eck, S.M. — 34
Network your Model IIs and 16s almost instantly with applications processors, hubs and cables, and file processors.

BASIC PROGRAMMING

A Disk Queue Copy Utility — Todd Heard — 22
File transfer is painless with the user-friendly COPY/BAS utility.

SYSTEMS APPLICATION

One Keystroke Improves Efficiency — William John Elliott — 38
Call your files with fewer keystrokes by using single character file names.

Intelligent Computing delayed

We announced a while back that we would introduce a new magazine titled Intelligent Computing some time this summer, and that it would cover Intel-chip based computers such as the Tandy 2000 and IBM PC. However, we have test-mailed several lists of supposed IBM PC owners and obtained dismal response (mostly because the lists contained only a small proportion of computer owners). So we are postponing publication of Intelligent Computing for a few months until better lists are available. We'll also redefine Intelligent Computing to limit its coverage to the better-than-IBM Intel-based machines. In the meantime, Advanced Computing will cover the Tandy 2000 as well as the II/12/16 line.

A tale of two BUGS

Recently, Barbara Albert, Advanced's managing editor, and Mark Ingoglia, its director of technical services, attended the national Tandy Computer Business Users' Group's second annual meeting in Fort Worth. A couple of weeks before that your publisher attended a monthly meeting of the Chicago area TBUG meeting in Deerfield, Illinois. Some comments on the effectiveness and value of each of these groups may be helpful to readers who may be interested in forming local user groups.

The Chicago group holds its monthly dinner meetings in the early evening at a north shore restaurant about 25 miles north of Chicago's loop. Almost all of the thirty-or-so members in attendance at the March meeting reside in far north suburbs, and represented about half of the total group membership.

The meeting format began at 6 p.m. with a cocktail hour (including much valuable discussion of computer matters, the one common concern of all), followed by dinner, a formal program which included presentations by local Tandy reps (who were much more "with it" than the average such rep), and an open discussion which continued well into the night. The Chicago group is a good model for other local groups. Perhaps its only serious defect is the location of its meetings. The bulk of Chicagoland Tandy users are located in the "loop" and other non-north places and would have to stop work early to make the meeting at all. Those who rely on public transportation would have to make arrangements for pickup at the train station (after a total trip of an hour or more to get there). Chicago probably needs another similar group which meets in or near the loop, enabling it to attract several times the participation.

All in all, the north Chicago group provides an excellent service to its members, especially in the free interchange of ideas and problem solutions. Business (and other) users in many other areas would do well to emulate its format.

We sent our editor and technical director to the national TCBUG meeting so that their direct contact with Tandy and some of its major users could be enhanced, and we have gleaned the following from their notes and comments (the opinions are mainly our own).

Much of the meeting time was spent on two areas: (1) Tandy presenting its views to the group and (2) presentation of individuals' minor technical problems (like specific Scripsit hang-ups) and Tandy's fielding solutions for most of them. While these two activities have value, and while it is reported that both were handled well, it is our opinion that these are not the essential matters which should occupy the time of a national users' group. In fact, these two activities could be much better handled through a system of good communications in both directions (Tandy <> group members).

Such communication systems were discussed in the 1983 meeting, but nothing effective has been done in the intervening year to make them work. Elements of these systems include (1) a regular national newsletter, (2) TCBUG columns in other Tandy-oriented publications, (3) active support for the development and continuation of local groups, and (4) a mechanism for formal presentation of problems to Tandy and publication of their solutions or comments.

We understand that the 1984 meeting included a good deal of apology and handwringing over past TCBUG failures, and that most of the officers of TCBUG were replaced with people who may have more time or enthusiasm to accomplish objectives. However, the president, Ira Tolmich, appears to have "reappointed" himself for another term.

TCBUG has made some significant contributions to Tandy business users. However, it has achieved only a very small portion of its potential value (to Tandy as well as its users). It's our intention to report regularly on the progress of TCBUG in the coming year, and, if it continues to play dead, to chide its leaders into keeping their promises.

Tandy has been developing a 35 megabyte hard disk system for some months. It hasn't been released because Tandy has not been able to find a satisfactory backup method (floppies take a long time and there are many technical problems with the various "streaming tape" schemes). However, TCBUG members prevailed upon Tandy to release the drive even though floppy backup would be necessary. After brief consideration, Tandy agreed to do so. We don't have an official price or release date, but we'll guess that the price is $3000-4000 and that it will be available by June first (maybe considerably sooner, as the machines are believed to exist now). So TCBUG scored a victory for those who need the capacity without regard to backup method.

Several new products (available or in progress) were mentioned by Tandy representatives:

1. Telexpress' terminal emulator program (TELETERM) will be sold in RS Computer Centers by about May 1. This program is available in TRSDOS (II/12/16, $225; Model 4, $195), MSDOS (2000, $195), or IBM PC ($195) versions. In addition to general terminal emulation, the program allows its computer to act as a terminal to Model 16 XENIX systems and to transfer files in either direction. The program is also available directly from Telexpress, Inc., P. O. Box 217, Willingboro, NJ 08046, (609) 877-4900.

2. Another terminal emulator (Softerm) from Peachtree will also be available through RS. Other terminal emulators may also be added to the RS line to give users a choice of various capabilities.

3. Model 100 accessories, such as a CRT monitor, spreadsheet programs, larger LCD screens, interface for small cassettes, and Scripsit were discussed. Since all of our readers who own Model 100s presumably also own larger systems, we see little practical use for these fairly expensive add-ons. Up-loading and down-loading between a 100 and larger micros is so simple that we expect that almost all Model 100 users will take advantage of the superior capabilities of the larger systems for more sophisticated applications. Tandy says that (in spite of rumors following the price drop) the Model 100 absolutely will not be replaced by a new model in 1984.

4. Tandy 2000 items included Ovation (an itegrated package available in the third quarter at $795), Microsoft's MicroWord (which will show superscripts, bolds, italics, multiple type fonts, etc., on the screen with color), Multi-Mate, Graph Writer, Programmer hardware reference manual ("now in the warehouse"), an external 15 megabyte hard drive (available later this year), a BASIC compiler (this summer), RM COBOL (now available), and Lotus 1-2-3.

5. Dr. John Patterson, head of research and development, said that Tandy was "committed to standard

operating systems - XENIX and MSDOS," without mentioning TRSDOS or CP/M. He also mentioned that they are working on 256K bit RAM chips, peripheral controllers (specifically speech devices), a 32 bit microprocessor, Winchester disk drives, color liquid crystal displays, networks, communications, and multi-tasking (with windows, graphics, and color).

6. XENIX 1.3.2, supporting local printing, should be available soon.

7. A faster 68000/Z-80 board is in development, but no release date was mentioned.

8. Better XENIX documentation is being worked on, but it's not clear whether it will be at additional cost or when it will be available.

9. Third party software of known reliability and merit of application is available through the Radio Shack Express Order program. At present, the only II/12/16 software in the warehouse for sale is Data Acc (a DBMS) for XENIX at $995. WordStar is available at $495 for the Tandy 2000. (However, the combination of WordStar, MailMerge, CorrectStar, and StarIndex will soon be available under the name WordStarPro at $695; separately, they are $1045.)

10. A dictionary for Scripsit 16 (described as a corrector as well as a verifier) has been contracted for but no production schedule is available.

11. RMCOS (Ryan-McFarland Commercial Operating System) is available for the Model 16. It is multi-user (up to 64), multi-tasking, has a multi-level file structure, and is said to run COBOL five to seven times as fast as XENIX. However, it runs ONLY COBOL. It's available with RM COBOL from Tandy Special Markets at $1400.

12. CP/M Plus (a k a CP/M 3.0) is finally available (15 months after its announcement) for the Model 12. It's said to require ROM replacement to bank switch.

13. XENIX products (each requires the XENIX Development System) now available are FORTRAN-77 (26-5451, $699), PASCAL 2 (Oregon Software, 26-5452, $699), and a Macro Assembler (26-5453, $399). There are no current plans for a BASIC compiler.

XENIX users may telephone Tandy XENIX specialists directly in Fort Worth. If you are west of the Mississipi, call Dennis Young at (817) 390-2788. If east, call Dick Peterson or Dave Butler at (817) 390-3001. Young (no kin) was at the conference and attendees were impressed with his XENIX knowledge.

XENIX non-Shack software

We're beginning to receive literature on non-Shack software that runs under Xenix on the Model 16. Established software that was developed in UNIX is being "converted" to run under XENIX.

Software Express: Their primary product (APPGEN) is a complete application generation environment for the implementation of commercial applications (for 2 to 16 users). Also available is the single-user "Financial Series" (General Ledger, Accounts Payable & Receivable, Payroll, and Inventory Systems) and several multi-user Vertical Applications. Their literature looks "top of the line." The Software Express, Inc., 10103 Fondren, Suite 220, Houston, TX 77096, (713) 974-2298.

Sunburst Software LTD: This is another example of UNIX software developed for minicomputers that has been successfully 'ported' to the 16 bit micros. They provide multi-user Accounting software and Management Systems with incorporated record locking. The programs are written in C and therefore upwardly compatible without software replacement. Their programs will soon be available for the Model 16. Their Point-of-Sale system will also run on the Model 16 with XENIX. Sunburst Software, 2696 North University Ave., Provo, UT 84604, (801) 374-5223.

Santa Cruz Operation: As is Radio Shack, they are a distributer for Microsoft's XENIX. They are committed to UNIX as well. Their products for the Model 16 include UNIPLEX (Word Processing Software), INFORMIX (Relational DBMS), and a Macro Assembler and Linker. Unix System Training Tutorials are also available. The Santa Crux Operation, Inc., 500 Chestnut Street, Santa Cruz, CA 95060, (408) 425-7222.

RUBIX is a relational database management system utilizing the power of XENIX. For the non-programmer, queries can be made in English-like syntax using the "Q" language, while the C-interface is available for the advanced programmer. Also available is PREFIX, an application generator for translation processing, and rshell, a user-friendly full-screen command interpretor for RUBIX as well as XENIX. Infosystems Technology, Inc., 6301 Ivy Lane, Greenbelt, MD 20770, (301) 345-7800.

Clinical Data Design: MDX (Medical Data Management) is a software product written for physicians. Several stand-alone modules include Appointment Scheduling, Medical History, Financial Management and Report Generators for those Modules. A Dental Package (DDX) is promised for August. Again, the programs are written in C. Clinical Data Design, 4718 West Lisbon Ave., Milwaukee, WI 53208, (414) 444-9900.

Another UNIX conference

Computer Faire, Inc. announced UNIX SYSTEM EXPO/84. This exhibit and conference (all in one building) is designed to bring together users, vendors, and resellers of Unix operating system products. This naturally will include Xenix products. Approximately 50% of all the products will be related to micros with over 200 vendors expected to be represented.

The objective of the conference program is "to insure that attendees come away with an understanding of how the UNIX system can be used." To assure this goal, Gnostic Concepts, Inc. is assisting in the design of the tutorials and the panel discussions which will feature many UNIX authorities.

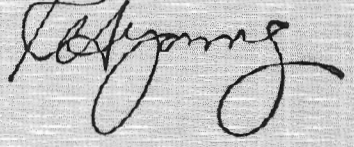
EXPO/84 will be held at the Los Angeles Convention Center on Sept. 11-14. Contact Computer Faire, Inc., 181 Wells Ave., Newton, MA 02159, (617) 965-8350.

Microcomputing infringes Advanced

We've just received a direct mail piece from Microcomputing magazine in which our Advanced Computing trade mark has been infringed several times. In fact, the envelope in which the piece was mailed bore the legend (in red) "Move up to ADVANCED COMPUTING." Since no other business name appeared anywhere on the envelope, it would be natural for a recipient to think the piece came from us. We want to reassure subscribers who may have seen this piece that we are in no way connected with Microcomputing, Wayne Green, or CW. Also, we have retained a trade mark attorney and have sent a request to Wayne Green asking that his company cease using our trade mark.

Machines available

We've received offers to sell us many more machines than we can use, so we've initiated a "machine exchange" service for subscribers. If you have any kind of used II/12/16 equipment (Shack or other) for sale, send us the details. We'll maintain a file of available items. If you need used equipment call us. We'll refer you to the sellers. No charge.



PART II

THE SCENERY IN XENIX

In our March/April issue, Mr. Esak guided us through the "word processing region" of the XENIX scenery, using nroff combined with ed or vi. In this issue, he'll show us around "Permissions," "The Shell," and "Communications," and answer "The Three Most often Asked Questions about XENIX."

BY JOHN ESAK

Let's traverse some XENIX land between directories, files, and permissions.

Permissions

Permission is needed to move around within the boundaries of the user shell $ environment (the root shell # environment is unencumbered by permission levels). Each user has control over his own property; he may allow access, or he may deny it. He can give permission to look but not touch, or he may say, "You can play the game, but you can't see the code."

Such flexibility of permissions is far superior to the password protection system found in other systems. Assuming the root password is suitably protected, as far as I can tell, the protections are inviolable.

Permissions are granted on three grounds: (1) read permission, (2) write permission, and (3) execute permission. These three privileges are granted to three sets of users: (1) the author (user/owner of the file), (2) the author's group, and (3) all others.

As you know, every new user admitted to the system by mkuser is given a group id. This defaults so that all users are added within the group 50 class. You may, of course, change this, but on smaller systems, it may not be necessary.

To see the permission levels of any file, use

ls -l filename

(this is an el, not a one). Or just a plain l. The list of files in the current working directory will be displayed, or you may specify a single file to see.

Each line of the directory listing begins with ten characters. A d as the first character on the line denotes a directory. A -(dash) in the first position represents a regular file. The next nine characters represent the permission levels for three sets of people: the first three represent the user who created the file, the second group of three represent the user's group, and the last group of three characters represent all others, meaning either r, w, or x (read, write, and execute) permission. A dash (-) in one of the positions indicates that permission is denied.

The line

drwxrwxrwx 2 bin 48 Feb 14 19:48 tmp

is the directory read out of a file called tmp last written into on Feb 14 at 19:48 hours. It has 48 characters in it, and here is the important part: everyone is allowed to do everything to it! The user, the user's group, and all others -- all have the three permission types granted.

If the directory listing looked like this

drwxr-xr-x 2 bin 48 Feb 14 19:48 tmp

only the user who created the file would be able to read, write and execute it. The user's group and all others would be denied the ability to change anything in that file. However, they would be able to read it, i.e. cat it, and execute it, if it were an executable file.

Changing permissions can be done in two ways. The first, using the "chmod" (change mode) command, is a little cumbersome, but you should know it. The syntax is

chmod ugoa+-rwx filename

You select who is to be affected (user, group, other, or all), whether permission is to be granted (+) or taken away (-), which permission type (read, write, or execute), and the file this is all happening to. The effects are, of course, immediate, and the next time you look, the directory listing will reflect the changes.

To add permission for yourself to write into a file, you might type

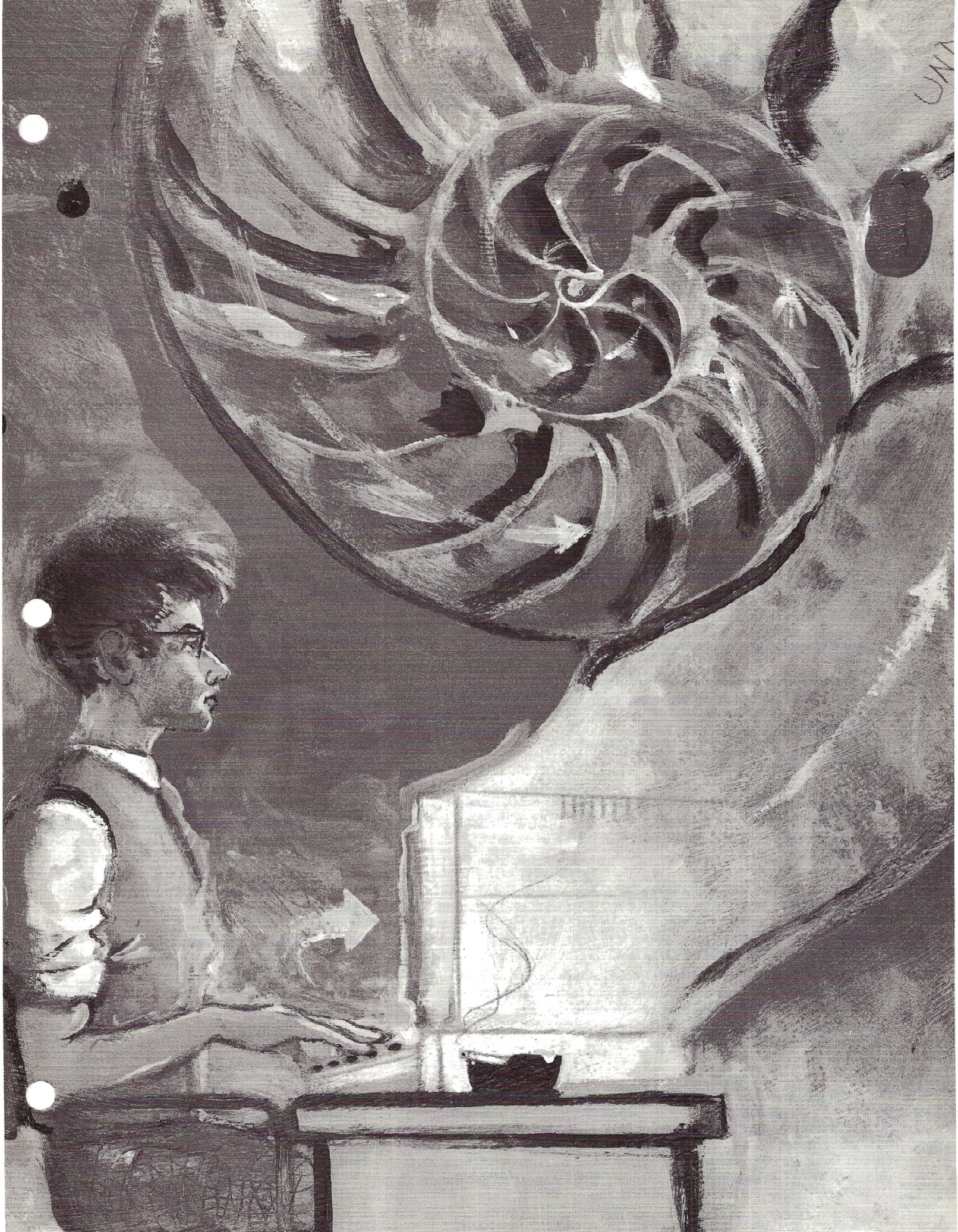
chmod u+w filename

The second and easier method of permission setting involves remembering a simple table. The syntax is

chmod mode filename

where mode is a 3 digit numerical representation of who gets what permission. Here's the table.

0 No permission
1 Execute/Search permission
2 Write permission
4 Read permission



XENIX

Any combination of permissions can be granted from this list, i.e.

5 Read and execute permission (4+1)
6 Read and write permission (4+2)
7 Read write and execute permission (4+2+1)

The three digit number designating the mode is configured in the usual way. The first digit is the user, the second digit is the user's group, and the third digit is everyone else. So

chmod 777 funhouse

would give everyone all permissions to the file "funhouse."

chmod 710 funhouse

would let the user do everything, the user's group execute only (no read and write), and all others would be denied access of any kind.

chmod 511 funhouse

would prohibit the user himself from writing in the file, but he could read it and execute it. All others could only execute it. (This is a good one to keep yourself from accidentally ruining the code on a solidly working program.)

Only the creator of a file can change its mode (grant permission levels). Of course, the super-user can do whatever he pleases to it.

The execute permission applies to programs which are executable. When the file in question is a directory, the x permission becomes a search permission. If it is not granted, the affected party can not list (ls) the contents of the directory. He can, however, still cat out a file in that directory if he already knows what name it goes by (unless that file is also read protected, or if the whole directory is read protected).

Only the person who "owns" a file (the person who created it) may alter its permission levels. If that person leaves or wants to give up the file or program, the chown (change ownership) command must be used. The syntax is

chown username file

This process is easiest done from root to avoid "file: not owner" responses.

The Shell

The shell may not actually be part of the scenery in XENIX, but it is surely its vital center. I wish I knew all about it. (It really isn't an it either. It's a them -- there are several shells.)

The shell is a superstructure around the core or kernel of the operating system which enables almost all interaction, from the simple tasks of searching through the file system for commands, to combining commands and programming to produce virtually any desired results. From program flow, user execution, default processing, etc., to simple, quick job applications, the shell is the most dynamic thing about XENIX.

The shell is a superstructure around the core or kernel of the operating system that enables almost all interaction.

Like the editor vi, the shell requires a concentrated learning effort. I will give you some simple processes to start you on your way to exploring the geography of the shell.

Essentially, each user has his own working copy of the shell at his beck and call. This shell will generate "children" shells (sub-shells) as necessary to do various jobs. Control is passed to these children until the action is completed, then the children die off and the response and control are passed back to the parent shell.

All this is transparent to the user at his keyboard (unless you request otherwise) so the only thing you need to know is that the shell can create sub-shells by "forking" its processing into several avenues. Whether you call an executable program or a text file which contains a list of commands, the shell does whatever it must to get the job done.

When the shell is given a command, it looks for that command in three places. If the file does not exist in any of these three places, the command will fail with an appropriate error message.

The first place the shell looks is in the present working directory of the operator who called the sh command. If it is not found there, the shell adds the prefix /bin/ to the command and tries again by searching for the command in the /bin directory. If it is not found there, the shell substitutes the prefix /usr/bin/ to the command and tries a third time, by looking for the command in the /usr/bin directory. If found, it is executed; if not, it isn't.

This sequential search is defaulted to these three directories in this order by the PATH variable for this user's shell, something you probably do not want to change unless you know what you are doing. This sequential search through the three directories for the called file is only performed if the shell command is not preceded by a /. If it does start with a /, for instance /fort, then the shell just makes one attempt to find the called file and stops if it isn't found.

As a rule, I place all my global or system-wide shell processes (those used by users) in the /bin directory and all my personal shell sequences in the /usr/bin directory. This helps me locate files a little faster when I want to revise them, and keeps the /bin directory and my own directories from getting cluttered. Also it allows everyone to access the shell routines if I want them to. If I leave them in my directory, no one else will be able to get at them easily. (Remember the shell searches only three places for the file. If you call it and it is in your current working directory, let's say your HOME directory, you will get the shell process, but others will not. They would have to use the full path name of the file to access it.)

Building a simple shell is easy. Here is an example. The /usr/games directory contains a file called "fortune." If you want to access this program, which does nothing more than give you a random fortune, type

/usr/games/fortune

This is a fairly cumbersome line to type in each time you want a fortune, so let's summon the shell.

Create a file called f using ed. Put the command listed above in the file and

close it. The exact sequence would be

```
ed f
a
/usr/games/fortune
.
w
q
```

Now you have a file which contains the command to call up a fortune, but you can not run it. If you do an

```
ls -l f
```

you will see that it has permissions of

```
-rw-r--r--
```

No person can "execute" it (x), but the shell can!

```
sh f
```

will produce the desired results.

To make the utility even easier to use, we can grant execution privileges to everyone without their having to invoke the shell by doing the following:

```
chmod 555 f
```

This gives everyone both read and write permission as discussed earlier. Now type f all by itself, and the execution takes place.

But if another user tries an f, and he is in his home directory, he will get message "f: not found." The shell looked for f in those three places, the user's present working directory, the /bin, and the /usr/bin, and couldn't locate it because it is sitting "hidden" in your directory.

If you want people to use your shell scripts (notation for shell procedures), you must move the file into either /bin or /usr/bin. You still have to change the permissions to a 555 mode (assuming you want everyone to have access). Other users must have both read and execute to perform execute, since the shell needs to be able to read the file in order to run it. Do not however, give anyone write permission to the file.

/usr/games/fortune is a one line command put into a shell script. You may put more than one command in these types of files. For example, you might have a file called "startup" which contains the lines

```
who
lc
cd /usr/yourname/memos
```

Then if you type

chmod 500 startup

you will have a file, executable only by you, that will give you a listing of who's signed on, then a listing of your present working directory in columns, and then move you to your memo file (assuming you have a memos file).

You may leave the mode alone (as it is defaulted) and type

sh startup

and the same things will be done. This is a personal shell operation, so you probably would not want to move this to /bin or /usr/bin.

The shell can be used to generate the XENIX counterpart of a TRSDOS DO file made with the BUILD command, although the XENIX shell is far more sophisticated in that it lets you pass arguments to the commands, such as

lpr filename1 filename2

and so forth. When used in conjunction with your .profile file, which is read and acted upon when you sign in, much automatic processing can be accomplished.

The shell can accomplish processes like if-then and and/or functions. A little programming knowledge is necessary, but the results are well worth the study.

The sequence

```
if command
then command
else command
fi
```

will first process the if command. If it fails (has a non zero) output, the next command (the then group) gets executed; however, if the first command worked, then the else command is initiated. (fi is the closing for the if command).

Also, the "while/do" command is very useful. Its syntax is

```
while command list
do
        command list
done
```

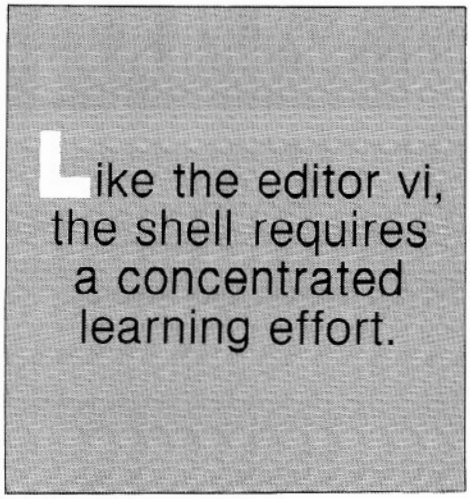
The first line is executed, and if the last command in that line has an exit status of 0, then the second (do) line is executed. This sequence gets repeated until the first command line returns a non zero exit status. The second or do loop can be repeated or looped by replacing the while with until.

A trap procedure, which is necessary to generate good shell processes, is most usually concerned with the number 2 (generated by an interrupt <BREAK>) and the number 1 (generated by a hang-up during processing). The syntax would be something like

trap 'command arguments' signal

For instance,

trap 'rm$tmp* >/dev/null; exit' 2 1



would, upon a break or a hang-up, cause the removal of any temporary files which might have been created by whatever process was going on. If no files were created, the output, which would be an error message, gets funneled into the null device (an absolute nowhereland in the environment of XENIX, from which no character ever returns). The shell then exits to the last parent shell process.

Writing good shell procedure is not something you will do your first day. I find myself continually learning better and more elegant ways of doing things with it. A good way to learn is to study the sample shell procedures in the Fundamentals book from the Development package, but a better way is to buy lots of easy to understand books on the UNIX system and read and try.

The trsshell, another shell built into XENIX by Radio Shack, is invoked by typing tsh and exited by typing <CTRL><D>. It contains almost all of the familiar TRSDOS commands. If you would rather type DIR to get a listing of the current directory instead of ls, go for it.

I think this shell is only good for tw things. First, it has CLS, the old standby clear screen command, and second, it is the only way you can save to floppies or other hard disks successfully. Other than that, it is for sissies.

There is also the C shell which is solely for programmers and very hip system operators. Take the C learn package before you play with this one.

I don't claim to be an expert! I would love to hear how to clear the screen from the keyboard. <CTRL> L is close but no cigar. If anyone knows how, other than writing a C program for cursor control, please let me know.

Tandy recently provided a series of <ESCAPE> sequences that do affect the cursor but clear the screen wasn't one of the provided entities.

The best of the offering was the sequence that turns the cursor into a block instead of the teeny, weeny underscore! It can be accomplished as a command by typing

<ESC><[><ctrl-><SPACE><q>

(<ctrl-> is control key and the - key pressed simultaneously, of course.) This gives you a block cursor and is so nifty I put it into a file and created a shell script to cat the file. Putting this into my .profile file which gets initiated as I login works great to save my already failing eyesight.

Communications

Many people are trying to access their consoles down phone lines, a process easily accomplished with three changes. First, enable the tty port you will be using:

enable tty01

This is exactly the same as going into the /etc/ttys file and placing a 1 in the first character position of the desired terminal.

Second, make the tty responsive to either 1200 or 300 baud by putting a 3 the second character position of the d sired tty in the same file /etc/ttys. XENIX will switch between 1200 and 300 baud until it finds what is being sent to it; then it provides a login: down the line.

Third, set the termcap of the tty in question in the /etc/ttytype file. TTTY01, 2, 4, 5, and 6 are defaulted to the ADDS25 characteristics which the DT-1 terminal can be made to emulate. If that is what you will be using there is no need to make a change.

I recently acquired an excellent program I would like to highly recommend to anyone who needs to access XENIX from various computers, especially a model II, 12. The package is called TELETERM (EM) and is from Telexpress in Willingboro, NJ. 609-877-4900. ($225)

This TRSDOS package is a very sophisticated terminal emulator and communications program, complete with a word processor (on/off line). It comes with an emulator file for several different terminal types, but of particular interest to me is the TRS16 emulator. When implemented, you can sign on to XENIX from any model II, 12 or 16 and have full use of every console function, including arrow keys, etc. You can download and upload files, operate programs, do direct to disk transfers, and much more, especially in the areas of RS232/VIDEO/PRINTER trapping and filtering. Everything works like a champ, including operation of Profile 16 and Multiplan down the lines. At 1200 baud it is actually quite acceptable.

If you use an emulator or convert a DT-1 to something other than a ADDS25, you need to change the appropriate line in the /etc/ttytype file. Cat the file and it will be evident what needs to be done. Since I often sign on to tty01 with a trs16 emulator, I change that line to read

trs16 tty01

Be sure to use only designations that have entries in the /etc/termcap file. Do a

more /etc/termcap

to get a look at this important file. You can even set the tty ports to output to a model 100!

The Three Most Asked Questions

By far the most oft asked question is: "How do I kill a background process?" The answer is a two step procedure: (1) find the PID#, and (2) kill it.

In case you are unaware of how to put something in the background in the first place, you place an ampersand (&) at the end of the normal command line. A command such as

nroff test >test.n &

will cause the process to be done in the background, allowing you to move on to something else.

Suppose someone enters the command

lpr editlist &

and then needs to stop the printer because he has found that something wasn't updated into the list yet.

To kill the process, you must know its PID#. This is a code number the computer assigns to all processes being acted upon. To find out the number, type

ps

The shell can be used to generate the XENIX counterpart of a TRSDOS DO file made with the BUILD command, although the XENIX shell is far more sophisticated.

and the response will be the processes you have placed in activation, and it will look like this

```
PID  TTY TIME  CMD
115  01   0:06  -sh
231  01   0:02  nroff test
```

The list can be much longer depending on what is going on. In any case, the information you need is the first number, the PID#.

In order to kill a process, type

kill PID#

This will usually work. If not, you can make it a more forceful request by adding a switch. The kill command defaults with a switch of -15, but some programs cannot be killed with a -15 kill level (for your own protection). If you absolutely want to kill a process, use a -9 switch, i.e.

kill -9 PID#

This works like Raid; it kills bugs dead.

You can get a long listing of PID information by typing a longer ps request:

ps -alx

This request will give you all processes running, not just your own, and will contain lots more very useful information. Two of the new columns added in this response are PRI (priority) and NICE. PRI is the priority assigned by the computer to the process, and NICE is the niceness level. It may sound funny, but these assignments can be manipulated to speed things up.

Nice is the value the computer uses to set the PRI (priority) level concerning how an operation gets worked on. It is called nice, since it is "nice" to not tie up the background processing with high priority processes when they really aren't. Foreground processes (both your own and those of other users), like ls and cat, will go much faster when the computer knows it can let the background actions slide a little to get the foreground done quicker.

Let's say you've put an nroff process in the background using

nroff test >test.n &

If you check the NICE level and the PRI level using "ps -alx", the NICE levels will probably match (perhaps 20), and the PRI levels will be near each other (perhaps 53 47). This means all processes, even background ones, are being treated about equally by the CPU.

To put the same nroff process in the background "nicely", you would use

nice nroff test >test.n &

Check the assigned values now, and they will be different. The "niced" background process might be a 30, while others are 20. The PRI value of the "niced" process might be 70, while others are at 23 and 42, and so on.

The second most asked question is "How do I get the &#@!(* printer to stop printing giant names at the beginning of each printout, then rolling up the paper, printing the request, and rolling up the paper some more?"

The printer does this so that on large systems, there is no confusion concerning where to direct the requested output. On your Radio Shack system with its upper limit of nine terminals, but more usually only three, this is not a big problem.

The printer output marked this way is called a "banner," and these banners are generated by a switch that can be either enabled or disabled.

To disable the banner switch, log in as root and then

```
cd /etc/default
ed lpd
s/1/0/
w
q
```

The file lpd has only one line in it. If you cat /etc/default/lpd, it will read

BANNERS=1

which indicates the switch is set to on. To turn it off, the 1 must be changed to a 0.

Once changed, printer output will be printed immediately, with no waste of paper. However, it will still roll to top of form when done with each printout.

The third most asked question is: "How can I communicate from terminal to terminal while online?" This is the easiest question of all, so I like getting it. You use the

write user1 user2

command.

If you enter only one user's login name, he is the one who gets your message. If you add more names, they all receive the message. After entering the write command, it will appear as if nothing has happened, but everything you type from that point on -- until you press a <CTRL> D -- goes to the specified user. If you do not hit the <CTRL> D, he will get each line you terminate with an <ENTER> until you do.

What's more, he can also and simultaneously issue a write command to you so you can talk back and forth. When you issue a <CTRL> D, he receives an EOF (end of file) notation. He must also end with a <CTRL> D or his end of the connection will not be broken.

The written material which appears on the receiving terminal does not show up in the actual work.

If you do not wish to be bothered with messages, you may use the msg switch. You type

msg -n

and no messages will come to you. Conversely,

msg -y

allows them to come in again.

Acknowledgements
Invaluable assistance was gained from Mr. Frank Dupree of LCL Systems, Washington, D.C. and Mr. Jorge Gardos of Radio Shack, Paramus, N.J. ■

The Radio Shack COBOL Generator and David Ray's RSCOBOL Utilties are two invaluable tools for any serious COBOL programmer.

The RSCOBOL Generator

BY IAN B. KAZIN

If this program cost $300 instead of $995, or if it included several more features it noticeably lacks, I would give it an excellent mark. As it stands, if you want to write a lot of COBOL programs, I would recommend purchasing the Program Generator. But don't expect to be happy about the cost/benefit ratio in comparison with other programs you may own.

A program generator takes the drudgery out of writing COBOL programs. To the uninitiated, there is nothing so verbose as COBOL. A simple program to print a simple report can run well over 500 lines of code. While this is wonderful in terms of making COBOL a virtually self-documenting language, COBOL programs can take a very long time to write and debug.

The RSCOBOL Program Generator (referred to as RSCPG from here on) directly addresses these problems. After defining your files in a way the program can later use, describing the relationship between files and the nature of the program, and laying out input screens or report formats, RSCPG will produce bug-free programs in minutes.

A COBOL programmer's dream? Almost.

The RSCPG will generate the following types of programs:

- Data entry with screens you can define
- Data entry without screens
- Backup copy of a data file under a different name
- Simple reports with or without column totals
- File maintenance
- File maintenance and reporting

RSCPG does certain things very well. Data validation is as good as you can reasonably expect. Date validation is perfect. You can validate data against other files, against a minimum or maximum value, or against certain values specified by you. This is really one of the stand-out features of the program. I have yet to think of a way of validating data input that the program will not handle.

Using the RSCPG is also a great way to keep track of your data descriptions since every time you build a program using the program generator, not only will the basic architecture of the program be the same, but there will be no doubt that data types will be accurate.

Once the program is generated, you can be certain of two things: (1) unless you have OCCURS or other clauses that RSCPG cannot handle correctly, the program will compile; and (2) once compiled, the program will run, unless you have made mistakes in defining ISAM keys.

Often a bit of work is needed to perfect the program. If you make use of OCCURS clauses, you will have to add subscripts and some code to get it to compile. The number of hours saved, however, can be very substantial. A simple 500 line report program, for example, might take between 15 minutes and an hour to complete as opposed to 3-6 hours.

The program is very good; it does not, however, deserve to cost almost a thousand dollars. Some things could stand a lot of improvement.

While you can define files that use the OCCURS clause, RSCPG will not write programs that implement them.

The methods of inputting data into RSCPG are not very efficient, and it is possible to waste much effort because of simple mistakes that cannot be corrected without re-entering the data.

The documentation is horrible. It consists almost entirely of a tutorial featuring a simplistic case that does not make use of several of the available features.

The code generated by the RSCPG is not optimized in terms of saving words or instructions, except for certain generalized subroutines used for such things as date validation and printing.

While RSCPG is equipped with an editor, it is not very sophisticated. Worse yet, RSCPG will build files that are too big to be edited by its own editor.

What can you do to edit those programs? The only answer I have found is to spend more money on another program Radio Shack calls a Program Editor and sells for $79. Unfortunately, the Program Editor will work only under TRSDOS 2.0 and not under 4.2

and is packed with as many features as its price indicates. If you want to work with double-sided diskettes on a Model 12 or 16 (since RSCPG can take up a tremendous amount of space when you deal with many complex data files), you will have to go through the following procedure to make a change in your program:

1.) FCOPY the program to a 2.0 diskette (1 minute)
2.) Load the 2.0 operating system (30 seconds)
3.) Start the Program Editor (as long as 5 minutes)
4.) Edit the program
5.) End the program edit (as much time as starting)
6.) Load the 4.2 operating system (30 seconds)
7.) FCOPY the program back to the 4.2 diskette (1 minute)
8.) Compile the program and then find that you have made a stupid mistake which is going to take as long as 13 minutes to change, even without counting the editing time itself.

When I spend $995 for a program like this, I object strenuously to having to spend more money and waste my time using such a complicated means of editing a program.

The Reports Generated are not limited in terms of the number of lines that can appear on a particular report, but they are limited in terms of the number of data items that can be used. An additional problem is that there is no facility to cause selective processing, control breaks, subtotaling, or many other features standard to most serious reports you will want to generate.

A great program generator should be more powerful than a great data base management system (DBMS), as they both perform the same basic function; e.g., to help in the development of systems. Program Generators are more powerful because you can delve into the operational programs and make modifications that you usually cannot do in a DBMS.

Unfortunately, the RSCOBOL GENERATOR is not nearly so powerful as many DBMSs on the market and is thus a bit of a disappointment. Worse, RSCPG costs a lot more that a good DBMS.

RSCOBOL UTILITIES

Hats off to David Ray ! ! !

His RSCOBOL utilities are a beautiful piece of work that add some important features for users of Ryan-McFarland COBOL. I have found so many applications for them that I wonder how I ever managed without.

The functions which operate within COBOL programs include:

Disk Utility Program
Makes a copy of a file from within a COBOL program
Renames existing files
Deletes existing files
Reads diskette IDs

CRT Utility Program
Displays characters of business graphics
Returns cursor position data to the program
Returns ASCII value of the character at a location
Protects part of the screen against scrolling
Prints a copy of the screen on a printer
Allows use of the HOLD key within programs

Printer Utility Program
Initializes the printer driver (similar to FORMS)
Returns current printer driver status
Selects Serial or Parallel printer
Resets line counter
Starts or Ends auto linefeed mode
Sends individual characters, including control codes

Keyboard Input Routine
For data entry and validation of signed integer fields, dollar and cent fields, date fields, and simple character fields

Print Headings Subroutine
For printing and keeping track of page headings and numbers

All the functions are easy to implement and seem to work perfectly. Better yet, the programmer has gone the extra mile to provide versions of each utility to allow your programs to run more quickly and/or save user memory.

One means of achieving this is the ability to run some of the routines directly from machine language programs, rather than through a COBOL subprogram. Another means employed to save user memory is the optional use of "high" memory (used only for certain system functions not generally needed while running programs) to store the utility's code.

Although I may not have explained how this is done very well, Mr. Ray does. The documentation leaves no doubt as to how to implement the various functions in their several versions.

What more can I say? I give this very important program a rave review.

The RSCOBOL Utilities are available from:
David Ray, C.P.A.
1301 Northwest Hwy., Suite 210
Garland, TX 75041
214/840-9531
Price: $120.00
Documentation: $10.00

Creating Inverted Indexes Using TRS-XENIX Utilities

Here are a few deceptively simple TRS-XENIX Shell programs that can read files, build and invert key word indices, and then quickly perform fast searches.

BY RICHARD A. BILANCIA

One of the fascinating aspects of the UNIX operating system (or in our case the TRS-XENIX Development System) is the wealth of tools or utilities that are available.

Three of these tools are utilities to read a file or files, build an index of the key words, invert that index, and then provide an extremely fast method of searching the index and retrieving the record or document from which the index was originally created.

You can find the official discussion of these tools in chapter six, "Formatting Papers," of the TRS-XENIX Text Processing System (from the TRS-XENIX Development System) in an article by M.E. Lesk entitled "Some Applications of Inverted Indexes on the UNIX System."

Unfortunately, as is quite common with much of the UNIX documentation, Lesk's article is not a complete, clear discussion of these utilities. Accordingly, I spent considerable time experimenting and constructing the applications that follow.

These applications will illustrate a simple search method I use on-line to inquire about phone numbers maintained in a separate file and the complete indexing of all text articles on a system.

Each of the simple shell programs presented below to implement these applications has only a few lines of code. However, the user must also be able to use either of the Xenix editors, ed or vi, in order to create the various files.

Telephone Index Installation

This example of an index to a telephone listing shows how to build an index to the individual records in a single file. You will probably note that since this is a very simple retrieval procedure, it could probably be done much easier with "grep" or "fgrep". However, the illustration should be sufficient to provide a basis for more complicated applications, especially if the object file is very large.

To install the system, follow these steps:

1.) Using either of the Xenix editors, ed or vi, enter and save the two programs "make1" and "search1" exactly as shown in Figures A and B. After creating and saving the files and exiting from the editor, type the following shell commands to make the files executable:

chmod +x make1
chmod +x search1

FIGURE A - make1

```
:               'a program that makes an inverted index of the telephone'
:               'numbers in a file "telnos"'
:               'Copyright 1984 - Richard A. Bilancia'
rm Index*
sed -e 's/$/\\
/' < telnos > tel
/usr/lib/refer/mkey tel | /usr/lib/refer/inv -v -h997
```

```
:               'a program that reads an inverted index of the telephone'
:               'numbers in a file "telnos"'
:               'Copyright 1984 - Richard A. Bilancia'
/usr/lib/refer/hunt Index
```

FIGURE B - search1

2.) Using one of the XENIX editors, create a file called "telnos." This file is a free form file with each line (record) containing a phone number, with or without an area code, and a name, address, and/or personal note. I recommend that the telephone number be entered after the text and aligned with an appropriate number of tab characters, but it isn't necessary. A hypothetical sample file appears in Figure C.

Using The System

Using the system is as easy as its installation. The steps are:

1.) To create a file of keys, and invert the file of keys to form the index, type:

make1<ENTER>

The first time you execute the program, don't be alarmed when you see the following message:

rm: Index* nonexistent

In a few seconds, if the file is not too large, the first step will be completed, and you'll see a message similar to:

40 key occurrences, 20 hashes, 10 docs

The following four files will have been added to your current working directory:

Index.ia	the entry file
Index.ib	the posting file
Index.ic	the tag file
tel	a double spaced copy of telnos

2.) To search the index and file interactively, type:

search1<ENTER>

At this point the program will be waiting for your input, so you can type in the first six letters of any word in the telnos file. (This key must be in lower case and no more than six characters long. See the Lesk article for more details.) For example, if you typed

nation<enter>

using the telnos file in Figure C, you would see

Radio Shack National Parts	(817) 870-5662
National Bureau of Standards Time	(303) 499-7111

on the screen.

Similarly, if you typed two keys on a line the search would return only those lines (records) containing *both* keys. For example, if you typed

nation time<enter>

using the same telnos file in Figure C, you would see

National Bureau of Standards Time	(303) 499-7111

on the screen.

You can continue to request as many searches as you like, and when you're finished, type a <CTRL-D> to return to a shell prompt.

FIGURE C - telnos

```
AT&T Info Systems (Morristown, NJ)       (201) 898-2000
Apple - Cupertino switchboard            (408) 996-1010
Radio Shack Customer Svc (Communication) (817) 338-2394
Radio Shack Customer Svc (Hardware)      (817) 338-2394
Radio Shack Customer Svc (Software)      (817) 338-2390
Radio Shack Customer Svc (Xenix)         (817) 338-2392
Radio Shack Customer Svc (accounting)    (817) 338-2391
Radio Shack National Parts               (817) 870-5662
National Bureau of Standards Time        (303) 499-7111
```

3.) Another way to search the index, using the same examples as above, is:

echo nation | search1

and,

echo nation time | search1

Several variations are possible, so use your imagination.

Understanding The Code
make1

lines 1-3: These are comment lines. The colon prevents the remainder of the line from being executed, but it is still interpreted for correct syntax. Accordingly, I have enclosed the comments within single quote marks to ensure correct syntax.

line 4: This is a remove command to eliminate previous versions of the three index files before proceeding further.

lines 5-6: The stream editor program "sed" is now used to insert a blank line between every line in the telephone numbers file telnos and create another file called "tel". Note that the sed program is contained within single quotes to prevent the $, signifying the end of the record to sed, from being interpreted by the shell. Sed is actually a substitute command, as indicated by the s. Sed substitutes the value of a "newline," protected by a /, at the end of every line as indicated by the $.

line 7: This line actually calls the two utilities "mkey" and "inv", using the file tel as the input to mkey with its output piped into inv. The -v option is for the verbose mode, and the - h997 indicates the hash table size.

search1

lines 1-3: These are the comment lines.

line 4: This line calls "hunt", the program that retrieves matches based upon the input keys.

Document Index Installation

This example of an index to a group of documents shows how to build an index to several different files.

To install the system, there is only a single step:

Using either of the XENIX editors, ed or vi, enter and save the two programs "make2" and "search2" exactly as shown in Figures D and E.

FIGURE D - make2

```
:               'a program that makes an inverted index of all the'
:               'text files in a system'
:               'Copyright 1984 - Richard A. Bilancia'
rm Index*
du -a $HOME | sed 's/[0-9]*       //' > tmp$$
/usr/lib/refer/mkey -w -k100 -n400 -f tmp$$ > Index.id
/usr/lib/refer/inv -v -h4999 < Index.id
rm tmp$$
```

```
:               'a program that reads an inverted index of all the '
:               'text files in a system'
:               'Copyright 1984 - Richard A. Bilancia'
/usr/lib/refer/hunt Index
```

FIGURE E - search2

After creating and saving the files and exiting from the editor, type the following shell commands to make the files executable:

chmod +x make2
chmod +x search2

Using The System
Using the document indexing system is as easy as using the telephone indexing system. The steps are:

1.) To create a file of keys, and invert the file of keys to form the index, type:

nice make2 &<ENTER>

This slightly different syntax will execute the program in low priority and in the background, allowing you to do other things in the meantime. As above, the first time you execute the program don't be alarmed when you see the following message:

rm: Index* nonexistent

Depending on the size and number of the files in your directory, the entire job may take thirty to forty minutes. The job will be completed when you see a message similar to:

44582 key occurrences, 3821 hashes, 722 docs

The following four files will have been added to your current working directory:

Index.ia	the entry file
Index.ib	the posting file
Index.ic	the tag file
Index.id	the keyword references file

2.) In order to search the index and file interactively, type:

search2<ENTER>

At this point the program will be waiting for your input, so you can type in the first six letters of any word or words you wish to select as indicated in the first example above. Note that only the first 1000 characters are retrieved. This can be controlled by the -l option of hunt.

You can continue to request as many searches as you like, and when you're finished, type <CTRL-D> to return to a shell prompt.

You can also use the echo command and pipe the output into search2 as illustrated above with search1.

Sometimes it may be easier or more appropriate to obtain the file name[s] of the file[s] containing a reference to a key, rather than the actual text of the file[s]. When this is the case, the following command can be used:

fgrep "keywrd" Index.id

A listing of the full path names of the files containing the keyword, followed by all of the keywords identified for that file, will be printed on the standard output.

Understanding The Code
make2

lines 1-3: These are the comment lines.

line 4: This is a remove command to eliminate previous versions of the three index files before proceeding further.

line 5: This line creates a list of all files in the home directory of the user. The "du" (disk usage) command with the -a option will print the size and file names of all files in the directory name ($HOME in this case). Any other directory name can be substituted to fit your needs. The output is then piped into a sed program that strips the size of each file from the front of every record and places the output in a temporary file.

line 6: The mkey program makes a list of the keys in all files identified by the previous line stored in the temporary file, and places the output in a file "Index.id". See the Lesk article for an explanation of the options used on this line.

line 7: The inv program reads the Index.id file and creates the inverted index files needed by hunt.

line 8: This line contains a remove command to delete the temporary file created in line 5.

search2

lines 1-3: These are the comment lines.

line 4: This line calls hunt, the program that retrieves matches based upon the input keys. ■

This review covers two exciting products, VEDIT and RED. One is newly improved, and the other is just plain new.

RED And VEDIT

BY ROBERT P. VanNATTA

VEDIT comes from CompuView Products in Ann Arbor, Michigan. The Z80-8080 (CP/M 80) version has variants for almost every known CP/M and MS-DOS compatible computer. CompuView has two special bit-mapped versions for the Models II/12/16, one which will work only on Pickles and Trout CP/M, and the other which works on all versions of CP/M, P & T included.

VEDIT is one of the few programs I have used that works equally well on Lifeboat, ATON (both versions), and P & T CP/M without the necessity of a reinstallation; its terminal patch bypasses the operating system and directly accesses the hardware features of the II/12/16.

The P & T specific version utilizes the special P & T function calls which is why it is precluded from working on competitive versions of CP/M. The reason for the special P & T installation, according to VEDIT's author Ted Greene, relates to some compatibility problems with the P & T clock calendar board. I don't use the clock calendar board, and since I am forever shuffling from one operating system to another, I naturally prefer the generic version.

Another of VEDIT's variants is the stripped down version provided as a free-bee with P & T CP/M. A mere 36 editor functions are provided on the "free" version, as compared to a full 50 functions provided on the current version (1.16).

The CP/M-86 version has also been tested on my Veritas dual processor board (DPO). VEDIT86 may be menu installed for the DPO; however, the installation is not bit-mapped. The result is a work-alike with the 8-bit version. I found no indication that VEDIT86 had any difficulty performing on the Veritas board, but the visual effect associated with rapid scrolling on the DPO was less than wonderful.

VEDIT now includes automatic word-wrap, search & replace from the edit mode, and horizontal scrolling.

Screen updates seem to take forever for two reasons: first, it is unreasonable to expect a serial terminal implementation to show the speed of a memory-mapped program; and second, I have concluded that the Veritas board performs video functions at a somewhat casual pace. I like snappy screen updates, and VEDIT on the Veritas board is slow.

I'm warmly enthusiastic about the new version of VEDIT. Outstanding new additions on the latest version include automatic word-wrap, search and replace from the edit mode, and horizontal scrolling. However, two things kept me from making greater use of VEDIT for program work: management of lines longer than 80 characters, and the difficulty in learning to use it.

I usually set up my programs in a 132-column format, and put the code in about the first 60 columns and the remarks beside the code beginning at about column 70. Using this technique and a horizontal scrolling editor like WordStar, the comments are normally out of sight but may be easily displayed when the need arises. Unfortunately, if you use this technique and don't have a horizontal scrolling editor, the display is a mess.

VEDIT comes through in a blaze of glory. It not only has horizontal scrolling like WordStar's, but also includes explicit commands so you can expressly scroll horizontally.

The literally hundreds of choices available on the installation menu when configuring VEDIT are overwhelming and make it difficult to learn to use. Versions prior to 1.16 had a bug in the keyboard definition routine that prevented me from defining multiple lead-in keys; all two-stroke commands had to use the same first key (normally ESC), a frustration because it hindered my efforts to define the keyboard to be a WordStar workalike.

This problem was cured in version .16, and two lead-in keys may be defined. I chose to define the ^Q as the alternate lead-in key which permitted me to follow WordStar conventions for most of the WordStar "Q" menu, as well as the single stroke cursor movement commands. In this fashion, I was able to get around twenty-five of the most commonly used edit commands to match those of WordStar. I left the remainder either undefined or mnemonically defined with escape sequences which gave me an immediately usable product that has made the process of learning the other commands less painful.

In another positive step to make VEDIT user-friendly, a HELP facility has been provided that lists out many of the various commands supported by VEDIT. For example, you can adjust even trivial things, such as the size of the cursor or its rate of blink by entering a few keystrokes in the command mode.

VEDIT Command Mode

WordStar users may have difficulty relating to the idea of a "command mode" because WordStar doesn't have one. If you have ever used Spellbinder, Wordmaster, or ED, you already know what a command mode is. If you haven't, you must leave the edit mode and enter the command mode in order to accomplish cursor and text movement.

VEDIT's command mode presents itself as a "user-friendly" asterisk on the bottom of the screen. You are expected, without the help of menus, to enter an obscure command sequence in a precise and rigid format. The result will be either marvelous or a disaster, depending on your skill.

VEDIT supports the use of macros in the command mode. As an example of the power of the command mode macros, a sample macro is provided on the distribution disk that will load an assembly language source file written in Zilog mnemonics, shuffle through it, and convert it to Intel mnemonics.

VEDIT Edit Mode

Some common tasks require use of VEDIT's edit mode. For example, suppose you want to save the file you are working on and resume editing at the same position. With WordStar, you would do it with a ^KS^QP. With VEDIT, you have greater complications as there is no way to leave a file marker that will survive a file save. You must, therefore, note from the status line the current line number. With this information (assuming, for example, you wish to resume editing on line 500) you would hit the following sequence:

<ESC><ESC>ea<ESC>500l<ESC>v<ESC><ESC>

The first two <ESC>s get you into the command mode. The "ea" saves the file and moves the pointer to the beginning of the file. The "500l" advances the pointer to line 500 and the "v

<ESC><ESC>" puts you back into the video mode. Multiple commands batched together and separated by a single escape is typical syntax.

Although I have been using WordStar almost exclusively for nearly four years, I am going to wean myself from WordStar for my programming work. The addition of the horizontal scroll and simplified search and replace have finally tipped the balance in favor of VEDIT.

WordStar has some bad habits when used as a program editor. It tends to be ponderous at times and will sneak characters into the file with the high order bit set on occasion. As a single 18k program without overlays, VEDIT suffers from neither of these faults. It is light and fast. Delightfully, it even contains a specific command to strip those dreadful high-order bits from a file. I shall never regret saying goodbye to the "PIP toggle(Z)" command.

Vedit Limitations

VEDIT is not what you would call a word-processing program. Rather, it is a very powerful editor which lacks the print formatting routines necessary to make it suitable for general purpose word processing. It has print facilities a programmer likes, such as the ability to mark a block and send it to the printer, but it does not do right-hand justification or sophisticated formatting. A printer macro intended to number pages and insert page breaks is provided; however, if you want to do any serious printing where appearance is important, you must either add a stand-alone print formatting utility (there are several on the market), or use some other program that has one built-in.

VEDIT is a memory editor, which means it works very well as long as the entire file will fit in memory at one time. Once the file will no longer fit into memory (over approximately 35k), its behavior is less pleasant. As long as you work forward in the file, it works tolerably well; however, I am convinced that it handles large files more inconveniently than WordStar. VEDIT supports backward file buffering also, but the bottom line is stick with WordStar or consider using RED if you must work with files of several hundred k.

Seeing RED

RED is the product of one man, Edward K. Ream, who doesn't hide behind a fancy corporate name and make fancy claims about his product.

Provided in source code, not compiler version, RED contains two floppy disks and a printed reference card. One floppy disk has the documentation; the other has the source code. You have to make some minor configuration decisions and then utilize your C compiler to create a usable program.

The source code provided will compile using the BDS C compiler, the Aztec CII compiler, or the Digital Research

RED is the product of one man who doesn't hide behind a fancy corporate name and make fancy claims about his product.

CP/M-68k compiler. RED provides what is known as portable code. The source code is the same, but, depending on the compiler you use, you get a program that will run on the 68000 under CP/M-68k, or a traditional CP/M-80 program. RED is the first program I have been able to obtain that runs on CP/M-68k.

A submit file that compiles and links all modules into a single 40k program comes with RED. You run through a configuration routine to include information such as your terminal control codes and keyboard command keys, and then start a submit sequence that takes about an hour to run. This one hour compile and link time has convinced me that the DRI compiler is SLOW.

RED is billed as a full-screen editor, but, in fact, it has only twenty-two commands available from the visual mode, and, should you choose to assemble the program with the powerful EDIT mode available (it is a quasi command mode), you must forfeit seven of these.

RED, like VEDIT and Wordmaster, also has a command mode which performs search, file, and block operations. The minimal functions expected of a program editor are present, and block operations include COPY, MOVE, DELETE, PRINT, and WRITE (to disk).

The major limitation of RED is that all operations are line oriented. Unlike the more powerful products, such as VEDIT or WordStar, you cannot arbitrarily set a flag to mark a block. Blocks are identified by line number. You must identify the file line number by moving the cursor to the line in question and noting the line number displayed in the prompt line. A command such as "move 10 20 30" will move file lines 10 through 20 inclusive and insert them after line 30.

RED has no provision for editing lines longer than your screen is wide (80 characters). As distributed, RED will load files containing lines up to 200 characters in length; however, only the first 79 characters display. There is no horizontal scrolling or line wrapping. Prior to compilation, you may (by making a trivial change in the source code) set the maximum line length to anything you want, but you can still only display the first 79 characters of each line.

RED isn't great and never will be. It is, however, absolutely the first CP/M 68k program that I have been able to find. Additionally, it offers a great educational opportunity because you can examine and tinker with the source code if you want to learn the the "C" language.

Most importantly, RED clearly demonstrates the brute force we can expect of programs designed for CP/M-68k. I am still working on modifications to the code to optimize RED for the Model 16, but one simple change allowed RED to expand its buffer are to include all available Model 16 memory, which in my case is 256k.

In a sample case, I generated a file 10000 lines long (314k in size). I could

move forward or backward about 5000 lines without a disk access. That would be about 150k of file buffer which isn't too bad for a 256k system. Try that in your 8-bit computer!

More amazing is the ability at block moves. A command to move lines 8000 to 10000 backward and insert this block after line 50 was accomplished in about 2.5 minutes. This is about the same time required to execute a ^QC command in WordStar on the same file. (WordStar, of course, simply explodes if you attempt to make such a large block move.)

RED is very easy to learn to use. I have spent the better part of a year attempting to master the use of VEDIT and am still very uneasy when it comes to attempting something that must be done in the command mode. With RED, I found myself comfortable within a few minutes because it has relatively few commands.

I am doubtful that RED will ever be anything but a hacker's toy, but then I am not sure that it is intended otherwise. Those of us who run CP/M-68k on a Model 16 are lucky enough to be able to retreat to CP/M-80 when we need an editor.

If it weren't furnished in source, I would be less enthusiastic. It provides a modest opportunity to study the workings of the C language and the C compiler, and at the same time have and use a workable program editor. RED (or for that matter VEDIT) is not suitable for an office environment wordprocessing program; but as a program editor, its modest cost and lack of competition auger in its favor.

Conclusions

RED is interesting. If you have CP/M-68k you ought to have it, just because it is there. My guess is that you probably won't use it much, but the 200k of source code could provide a lot of fodder for a determined hacker.

The full version of VEDIT as implemented on the II/12/16 is a definite winner as a programmer's editor. If you have access to Pickles and Trout 2.2m, take a look at its clipped version. If it piques your curiosity, don't hesitate to spring for the full version.

There are both CP/M-86 and MSDOS versions of VEDIT. These versions have the same features as the 8 bit version. Future versions for the 8086 are promised; they will spread themselves out into all available memory, and that should give VEDIT a powerful advantage on the 8086-8, particularly when they get their also promised multiple-file editing features running.

At the moment, however, VEDIT seems to stay in one of those segments of the 8086-8 that provides little advantage over the 8- bit version. As mentioned earlier, the slow screen-display characteristics of the VERITAS board and the lack of bit-mapping make the CP/M-86 version a sorry duck compared to the 8-bit one.

VEDIT is available from:
CompuView Products
1955 Pauline Boulevard
Ann Arbor, MI 481034
313/996-1299
CP/M version costs $195.00
MSDOS version costs $150.00

RED is available from:
Edward K. Ream
1850 Summit Avenue
Madison, WI 53705
608/231-2952
Cost $95.00

A Disk Copy Queue Utility

User-friendly COPY/BAS provides an easy method to copy files from one disk to another.

BY TODD A.C. HEARD

How often have you wanted to select an arbitrary group of BASIC files for transfer between disks (or drives) on the TRS-80 model II/12/16/16B? The apparent lack of a software tool to invoke a system utility to process a queue of files for transfer was the inspiration for COPY/BAS, a utility providing one relatively painless way to copy files from one disk to another.

A user-friendly process is needed to automate the tedium of transferring an arbitrary queue of files between two disks. The process should support the selection of an arbitrary set of files and should handle all the details of BUILD file construction, DO process execution, and manual (optional) file deletion upon task completion. Sound like a chore? It is. After you struggle through TRS-80 system support software a few times, you will be easily convinced that there should be a better way. I have found that COPY/BAS, an easily used process for automating the creation of a queue of file names and the implementation of the multiple file process, may be what you need.

COPY/BAS is a monolithic program consisting of the following sequential parts:

1.) System function access
2.) COPY/BAS label and copyright identification
3.) Screen label format
4.) Prompt-driven interactive editor
5.) BUILD file constructor process
6.) COPY process
7.) File close and deletion

These elements are found in the listings of Figure 1 and will be described in more detail below.

```
                A Disk Copy Queue Utility
10 '*************************************************************************
20 '*                                                                       *
30 '*    COPY/BAS:  DISK FILE COPY UTILITY                                  *
40 '*                                                                       *
50 '*    AUTHOR:  TODD A. C. HEARD                                          *
60 '*    VERSION NO. 1.0                    COPYRIGHT 1983 INTRONEX INC.    *
70 '*    Tue:  Aug 9, 1983     TIME:  00.05.55                              *
80 '*                                                                       *
90 '*************************************************************************
100 CLEAR 2000
110 'INITIALIZE HEADER
120 A$=DATE$:B$=MID$(A$,1,3):C$=MID$(A$,4,3):D$=MID$(A$,7,2):E$=MID$(A$,9,4)
130 CLS
140 'BORDER LINES
150 PRINT @(0,0),CHR$(128)
160 PRINT @(8,0),CHR$(131)
170 FOR I=1 TO 78
180 PRINT @(0,I),CHR$(150)
190 PRINT @(8,I),CHR$(150)
200 NEXT I
210 PRINT @(0,79),CHR$(129)
220 PRINT @(8,79),CHR$(130)
230 FOR I=1 TO 7
240 PRINT @(I,0),CHR$(148)
250 PRINT @(I,79),CHR$(148)
260 NEXT I
270 'HEADER TITLE
280 PRINT @(2,5),"COPY/BAS:  A DISK FILE COPY UTILITY"
290 PRINT @(3,0),
300 PRINT @(4,5),"AUTHOR:  TODD A. C. HEARD"
310 PRINT @(5,5),"VERSION NO. 1.0                    COPYRIGHT 1983 INTRONEX INC."
320 PRINT @(6,5),B$;":  ";C$;" ";D$;", ";E$;SPC(4);"TIME:  ";TIME$
330 PRINT @(7,0),
340 PRINT
350 FOR I=1 TO 1000:NEXT I
360 CLS
370 PRINT "COPY/BAS";SPC(33);B$;":  ";C$;"  ";D$;", ";E$;SPC(4);"TIME:  ";TIME$
380 '
390 'INITIALIZATION
400 '
410 CLEAR 200
420 '
430 'OPEN BUILD FILE
440 '
450 OPEN "O",1,"XCOPY"
460 DIM A$(100),B$(100)
470 PRINT
480 PRINT
490 INPUT "INPUT SOURCE DRIVE (0-5)";S
500 INPUT "INPUT DESTINATION DRIVE (0-5)";D
510 '
520 'IDENTIFY SOURCE AND DESTINATION DISK DRIVE
530 '
540 S$=MID$(STR$(S),2,LEN(STR$(S)))
550 D$=MID$(STR$(D),2,LEN(STR$(D)))
560 PRINT
570 PRINT "ENTER DISK FILE NAMES TO TRANSFER.  TYPE 0 WHEN COMPLETE."
580 PRINT
590 I=1
600 INPUT A$(I)
610 IF A$(I)="" THEN 600
620 IF A$(I)="0" THEN 650
630 I=I+1
640 GOTO 600
650 I1=I-1
660 '
670 'IDENTIFY BUILD FILE IDENTIFICATION CODE
680 '
690 PRINT #1,CHR$(208)
700 '
710 'PRINT COMMANDS TO BUILD FILE
720 '
730 FOR I=1 TO I1
740 B$(I)="COPY "+A$(I)+":"+S$+" TO "+A$(I)+":"+D$
750 PRINT #1,B$(I)
760 NEXT I
770 PRINT #1,"PAUSE Enter Source Disk"
780 PRINT #1,"KILL XCOPY"
790 CLOSE
800 SYSTEM "DO XCOPY"
810 END
```

FIGURE 1 COPY/BAS Disk File Copy Listing

COPY/BAS, as illustrated in Figure 1, is a straightforward sequential process. Lines 100-130 access system information from which the page header is constructed. Lines 130-340 set up the header border line display. Lines 350-380 display the COPY/BAS program label for a suitable time. Lines 390-470 handle housekeeping tasks and open the XCOPY file for output. Lines 480-660 provide the prompt-driven interactive editor. Lines 670-760 construct the BUILD file and transfer the print commands for the file queue. File manipulation, file deletion, DO execution, and closure are handled by lines 770-810.

In brief, to use COPY/BAS, load the program under TRSDOS BASIC with the capability of handling at least one file. Run the program and respond to the user prompts.

Figure 2 contains a detailed listing of the interactive prompts and the text displayed during COPY/BAS execution. User response is underlined for clarity. Verification of transfer is illustrated in Figure 3.

The user should approach any file manipulation program with due caution. Try out the utility on scratch files and become thoroughly familiar with it before you depend upon it. The COPY/BAS utility has been found to repay the time required for its development through efficient, trouble-free file transfer. Once you learn to use it you will not want to be without COPY/BAS in your software tools kit.

For information regarding executable copies of COPY/BAS, send a self-addressed, stamped envelope to Intronex, Inc., P. O. Box 515, Menlo Park, CA 94026. ■

```
Ready
>LOAD "COPY/BAS"
Ready
RUN
COPY/BAS                    Sat:  Jan 7, 1984   TIME: 00.00.53


INPUT SOURCE DRIVE (05)? 0
INPUT DESTINATION DRIVE (05)? 0

ENTER DISK FILE NAMES TO TRANSFER.  TYPE 0 WHEN COMPLETE.

?H1/BAS
?0

Mount SOURCE disk
Press ANY key to conbtinue CR
Mount DESTINATION disk
Press ANY key to continue CR
Copying Record 00720
PAUSE Enter Source Disk
Press ANY key to continue cr

XCOPY:0 Delete? (Y/N/Q)..Y      *** File Deleted ***

TRSDOS READY
```

FIGURE 2 Step-by-step Operating Procedure. User Response is Underlined.

```
Ready
PRINT H1/BAS (V)
10 '*****************************************************************************
20 '*                                                                           *
30 '*     H1/BAS:   THIS IS A HEADER PROGRAM                                    *
40 '*                                                                           *
50 '*     AUTHOR:   TODD A. C. HEARD                                            *
60 '*     VERSION NO. 1.0          COPYRIGHT 1983 INTRONEX INC.                 *
70 '*     Sun:  Jan   2,  1983     TIME:  00.09.13                              *
80 '*                                                                           *
90 '*****************************************************************************
```

FIGURE 3 Verification of H1/BAS File Transfer by COPY/BAS.

A REVIEW OF

THE STATISTICIAN

BY PAUL NAITOH

Why do some microcomputer users need a statistical package? For those of you who use micros for managing a small business, statistical packages in the market seem to offer esoteric manipulations of numbers with no particular benefit for your business. You might ask whether statistics can help you to manage your business better.

The Statistician is a package of user-friendly, menu-driven programs providing you with statistical tools that are useful in making business decisions. One catch is that this statistical package, as all other statistical packages now on the market, provides only statistical tools. The Statistician is not a program designed to teach statistics. It does not hold your hands and show you how to use these tools. You must know what statistics is before you can use the package for making business decisions.

Non-business users of micros, such as engineers, medical professionals, and educators will find that The Statistician fills a need for statistical evaluation. The Statistician is interactive, versatile, and friendly. Best of all, it's available immediately; no one will ask you to get off a remote terminal because someone else has a higher priority job to complete. With your own micro and The Statistician, you will be spared the psychological trauma of rushing to get your job done.

Program Requirements

The program runs under TRSDOS, CP/M and XENIX and is also available for the IBM PC. I have run it with my one-drive 64K Model II under TRSDOS 2.0a and Pickles and Trout CP/M 2.2eD. A printer is necessary for obtaining the results of stepwise regression analysis. The results from other programs can be routed either to the screen or the printer, with printout automatically formatted to achieve ease of understanding, as well as economy in paper.

Features

The package has eleven classes of analyses, given in the menu for easy selection. They are:

Regression Analysis
Data Transformations
Descriptive Statistics
Time Series Forecasting
Random Samples
Generate Random Variables
Non-Parametric Statistics
Data Sgrt
Utility
X-Y Plot
Correlation Analysis

Regression analysis and time-series forecasting are outstanding routines not seen in any other micro packages. Regression analysis offers considerably more analytic power than Radio Shack's "Statistical Analysis" package (Cat No. 26-4540). The Statistician consumes about 20K bytes of memory for the regression program in its uncompressed form, but the manual claims that the large number of cases can be analyzed.

Here's an example for one, three, and twelve independent variables:

# of Independent Variables:	1	3	12
TRSDOS (64K; Free RAM, 9.5K)	450	300	110
CP/M (64K; Free RAM, 10.5K)	500	320	120
XENIX (if 100K available for BASIC)	4000	2600	1000

When the BASIC program is compressed, we can run a larger number of cases than shown in the above table, an impressive figure. In most business applications of multiple regression, we will be comfortable with what this package offers in working memory space. For a linear model extensions of multiple regression to the analysis of variance (ANOVA), the program's ability to accept up to 50 or more independent variables is a highly desirable feature.

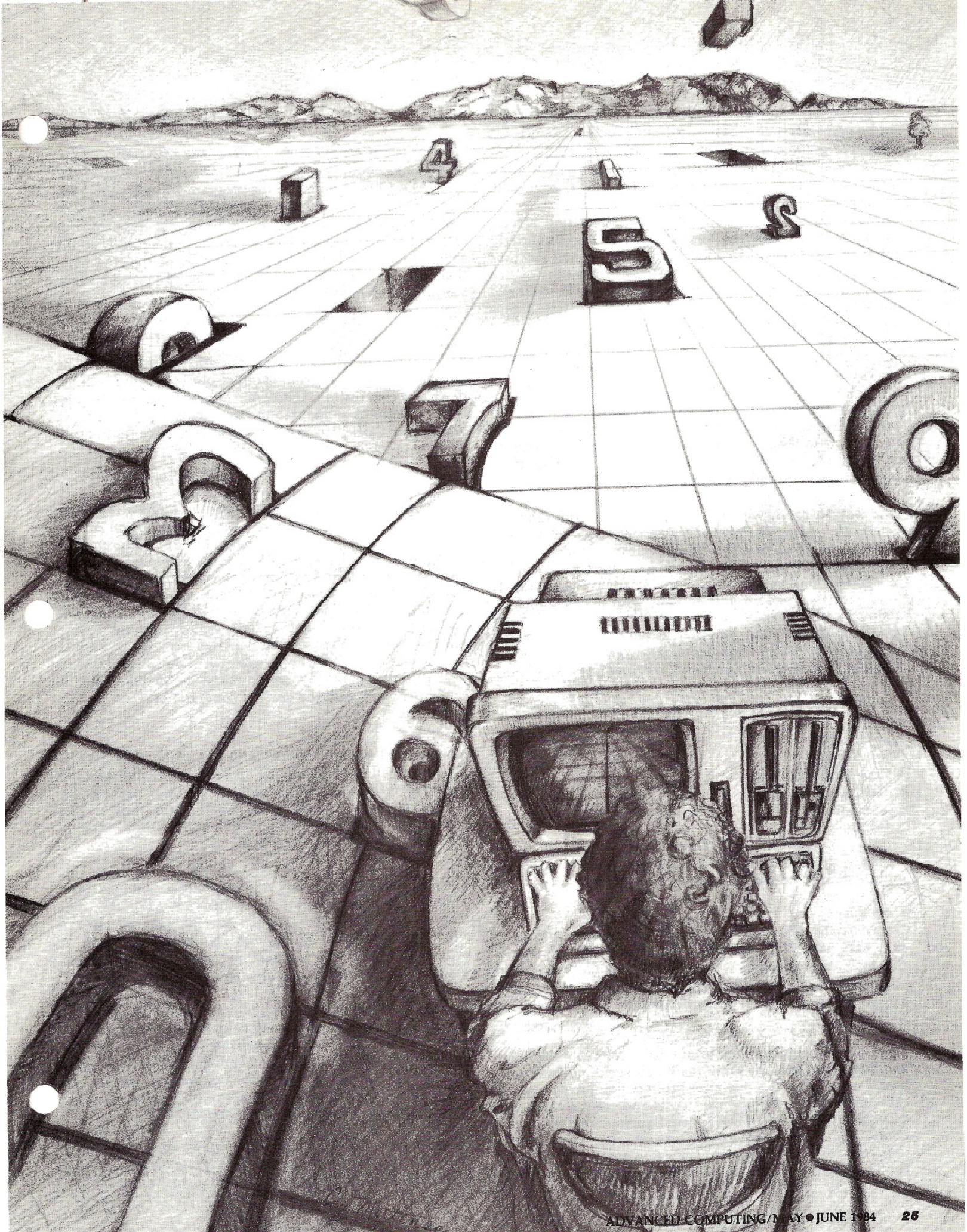
The regression analysis provides regression weights, ANOVA, F- statistics, Durbin-Watson (to check how well a set of independent variables is selected to fit to the data), R-squared (an overall fit of the model to the data), and predicted values. The predicted values can be stored on disk.

Other regression procedures are stepwise accretion, backward elimination, all subsets, a new model, and ridge. Missing data are handled by eliminating offending cases from the computation.

The package has options for 22 data transformations. To most business-users, data transformations appear far removed from what they consider useful. Some transformations are for statistical manipulations to assure that we can use tabled statistical values, but others, such as sum, difference, and quotient of two variables, are useful to everyone.

The package has seven time-series routines:

Moving Averages
Single Exponential Smoothing
Double Exponential Smoothing
Sinusoidal Model
Holt's Two Parameter Linear Exponential Smoothing
Winter's Three Parameter Exponential Smoothing
Adaptive Filtering



These time-series forecasting models are very useful in making business decisions. Each forecasting model produces an attractive graphic display of actual and predicted values, and a companion table printed with the graph shows future forecasted value.

Through experimenting, business-users might find a forecasting model to describe, for example, after-tax quarterly profit of a given group of merchandise in past years, thus enabling them to predict a future trend (by means of this model) to aid in their decision whether or not to continue handling the merchandise for the next fiscal quarters. A simple moving average model and other models described by Daniel B. Nickell in his *Forecasting on Your Microcomputer* (TAB Books, 1983) can be handled by the programs in The Statistician.

When evaluating time-series models, the first question to ask is whether the data show any systematic trend. If no trend exists in the data, no amount of numerical manipulations will provide the means to predict the future behavior of business profit and productivity.

One way to check whether the data have a non-random, systematic and significant trend is to make up a set of completely random data with a similar mean, standard deviation, and distribution form and run it with the time-series model used on the real data. If the results of two analyses (one with the random and another with actual data) are almost indistinguishable, a direct application of a time-series model would not work. The Statistician provides the means to create such random samples under two routines, Random Samples and Generate Random Variables.

The Statistician contains six non-parametric statistics:

Median Test
Man Whitney
Wilcoxon Signed Rank
Kruskal-Wallis Test
Spearman's Rho
Runs test

The raw data are entered into these programs, thus alleviating a need to convert them into rank orders.

Data Base
Data entering, listing, editing, transforming, and sorting are very easy with The Statistician. Data are stored in sequential files, each variable with its own variable name, and entered through a window where all related variables will appear in the window as a row of numbers (i.e., a relational database).

For example, you have ten brand name canned apple juice products of roughly the same quality and price, and for each brand name canned apple juice product you have a record showing weekly advertisement costs and sales revenue for each item. You wish to enter these figures to determine the relationship between advertisement and sales. To create the data base for The Statistician, enter two numbers in a single row for each product, one representing weekly advertisement cost and one representing weekly sales revenue.

After entering all ten rows of data, the program will ask what label should be given to the first number of each row (say, ADCOST/DAT), and the program will store these ten numbers under a disk file name ADCOST/DAT, with one addition. The first number of the file is not the data, but rather shows the number of data points in the file. Then the same is repeated for the second number of the row (say, SALES/DAT). In other words, any ASCII file with a single variable can be used by The Statistician.

Performance
The Statistician works well on all statistical programs. The automatic format for printing out results is designed well. Its descriptive statistics, ANOVAs, and correlations worked well. When the computation takes considerable time, a warning is flashed to prevent panic response, resulting in premature interruption of computation.

I tested a stepwise multiple regression by running examples given in the Draper-Smith book, *Applied Regression Analysis* (2nd Ed., Wiley, 1981. See pages 353-360 for example.) When four variables were entered into the stepwise multiple regression equation, The Statistician and the values in the book matched at least to the third decimal point. F-ratio and R-squared of the book were 9.88 and .767027, while the package's were 9.878 and .767053, amazing accuracy. The sequence of variable selection in the stepwise procedure seems to be accurate. The program, X-Y plot, worked very well, producing a well-balanced graph, printed on a 8.5" X 11" paper right after the graphic computation.

However, when you move the ASCII file containing the X-Y plot information to SCRIPSIT in order to incorporate the figure in the text, you must repair some portion of the graph due to soft spaces created in the ASCII graph image. When you move the ASCII file to WordStar, you will have no problem incorporating the graph.

The Statistician handles any ASCII file as its own data base. Thus, it can handle files generated by PROFILE, i.e. filename/DAn. However, this is true only when you are willing to edit PROFILE files with your word processor, or better still, if you write a BASIC program to rearrange data in PROFILE file so that each file contains one variable.

Manual rearrangement of data with a word processor takes too much time, especially when you have voluminous data. I have a BASIC program to translate PROFILE files to fit the Radio Shack Statistical Package. I anticipate that The Statistician will add a similar translation feature in the future edition so that it can use the files generated by data management programs, such as PROFILE and dBASEII.

Execution speed of The Statistician varies according to the operating system. I have compared Pickles & Trout CP/M 2.2eD and TRSDOS 2.0a in performing simple tasks:

Task Performed	TRSDOS 2.0a	P&T CP/M 2.2eD
Load BASIC, and RUN"CENTRAL/BAS (.BAS)" to get Main Menu	10 seconds	3 seconds
Get to Regression Program Menu from Main Menu	24 seconds	13 seconds
Get to XYPLOT Program Menu from Main Menu	24 seconds	13 seconds

Obviously, the CP/M version is much faster. If you have CP/M, MBASIC, and a word processor to go with CP/M, the choice is clear.

Error Handlings
Errors due to improper selection of parameters are handled by error messages or by repeating the request for parameter entry. Occasionally, some errors due to misunderstanding the program are not detected until the program starts to execute the instructions, and then it encounters BASIC statement(s) which cannot be executed, resulting in an automatic exit from The Statistician.

For example, the utility program for "Multiple Search on Fields with Multiple Conditions" is powerful in creating new files from the existing files. Suppose you have a single PROFILE file containing age (20-55 years old), average weekly sales figure ($500-$2,500), and yearly salary ($25,000-$150,000) for 20 salespersons, and you wish to generate one file each for age, average weekly sales figure, and yearly salary so that you can apply statistics.

With the utility program, you can use only one condition at a time. By specifying a condition of "<60" you can generate a new file for "age." Then you must go back to the PROFILE file to create a yearly pay file by specifying the condition of ">=25,000." When you ask to do more than one "condition", the program does not issue error message, but eventually kicks you out of The Statistician due to errors in BASIC. Reading the instructions carefully will avoid subsequent disappointments.

Documentation
The documentation is demanding, but well organized, and packaged in a three ring binder. The left page of most of the documentation contains textual explanation, with the facing page showing the screen display. Two appendices are provided for retrieving data when EDITDATA.DIR is inadvertently destroyed and interfacing other database files to The Statistician.

A page reference list, "Applied Texts In Statistics," is provided, but it is not explicit enough to help users. The reference list should give the page numbers of the book cited, dealing specifically with a program in The Statistician. Since most users will have less than ideal sophistication on time-series analysis, the reference list should indicate the source of these many techniques, along with general texts. The Statistician is not designed to teach users statistics of time-series analysis, but it could at least provide the mathematical source to which interested users can turn to decide if they understand the basis of techniques.

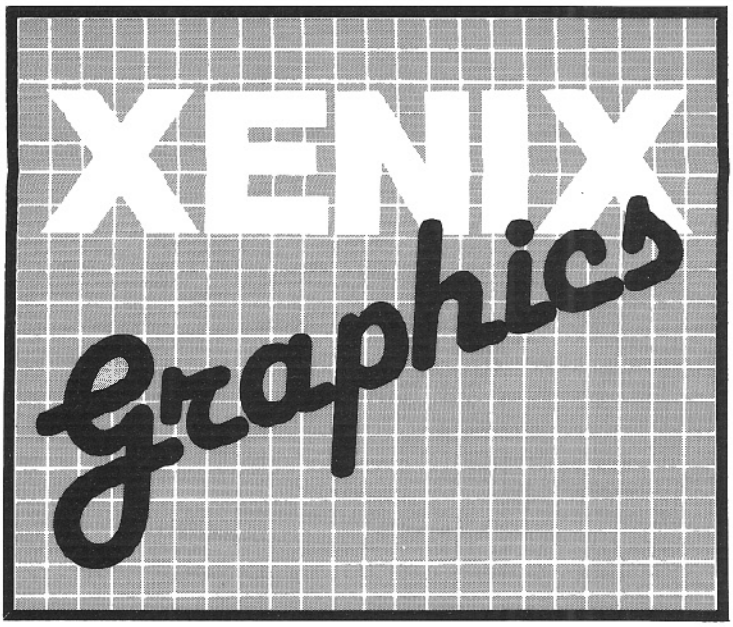
Support
James Hawkes, Ph.D. is the statistician who has developed the Quant Systems. He has been very responsive to feedback and questions from users. The fact that the programs in the Statistician come in unprotected BASIC will be very helpful for more talented users in modifying and customizing them.

My own interaction with Dr. Hawkes has been positive. His response to my report of BASIC program errors in the earlier version of The Statistician was very prompt and thorough.

Recommendation
I recommend The Statistician for users wishing to incorporate statistics in their businesses, teach statistics, and apply statistics to overall data analysis. Because of the very extensive nature and ease of use for regression and time-series programs in the package, I recommend it even to those who already own some other statistical packages. The Statistician offers business-users an application for better business management, and hopefully, a higher profit margin.

The Statistician is available for TRSDOS, CP/M, XENIX, and IBM PC from:

Quant Systems
Box 628
Charleston, SC 29402
800/334-0854
Cost: $299.00



BY JEAN-PIERRE RADLEY

If you've tried some of the TRSDOS BASIC screen commands under XENIX MBASIC, you know that PRINT CHR$(26) does not give you reverse video, and that PRINT CHR$(2) does not turn off the cursor. If any of your old TRSDOS BASIC programs use some of the 32 graphics characters, you may be unhappy about the inability to generate those "pretty screens."

Good news. The capabilities are still there, but they now require different codes to get at them.

The screen control sequences are given in the /etc/termcap file. Look at the "so" (Standout Start) and "se" (Standout End) codes for the trs16 entry of /etc/termcap.

While you're at the XENIX $ prompt, type the command for Standout Start:

<ESC>RD<ENTER>

(Upper/lower case does matter.) Everything you do now will be in reverse video. To get back to normal, you'll need to type the command for Standout End:

<ESC>R@<ENTER>

The manual does not spell out the meaning of all the capability codes, particularly GS, GR, CN, and CF, but I figured out that they mean, respectively, Graphics Start, Graphics End, Cursor On, and Cursor Off.

For some wild effects, type Graphics Start:

<ESC>RG<ENTER>

Try any XENIX command, such as "echo *" or "lc" or "cat.profile". Throw in

<ESC>RD<ENTER>

and try some more commands. Finish up with Graphics End plus Standout Start:

<ESC>Rg<ESC>R@<ENTER>

It appears that Graphics Start filters the screen so that the 32 ASCII values from 95 to 126 map onto the set from 0 to 32 in the TRSDOS Graphics mode, or on to the (same) set from CHR$(128) to CHR$(159) under TRSDOS BASIC. All other characters are not filtered and print normally. The characters you type are correctly interpreted by the shell; only the echo sent back to the screen by XENIX is affected.

You may use "ed" to modify /usr/you/.profile or /etc/motd to include graphics, but don't forget to type in a backslash before the <ESC>. Then when you use "p" under the editor, the desired graphic effects should be there; if not, use "l" instead of "p" to see if you entered the escape sequences correctly. You should see "\33xx" for each sequence, since the ESC key sends back 1B(hexadecimal) = 27(decimal) = 33(octal).

The "screen.bas" program demonstrates these codes, including clear-to-end-of-line and clear-to-end-of-screen. Note that the cursor movement codes neither wrap around the screen nor generate scrolling at the bottom line of the screen. In line 70 there are four functions which you may have seen in Profile but not elsewhere under TRSDOS. One other code: you may clear your screen with CHR$(14) in BASIC, or ^L (same as the <F4 key> on a Model 12) in XENIX or BASIC.

The "more" command in XENIX uses a different method than Standout Start to print reverse video. Load one of the /usr/tsh.help files into ed, and first type "1,$p", then "1,$l". Some tricky coding is used to make a character print in reverse:

_\^H

i.e., underscore, backslash, backspace. This sequence will cause only the next character to be in reverse video, but only when you print with more, not with cat. (The help command of the tsh shell in fact uses the more command of the regular shell.)

Caveat! You can see, by looking at the rest of /etc/termcap, that the effects we've been playing with may either not exist on a different terminal, or at least require totally different escape sequences. When XENIX is supporting several users, it would be worthless to write code which may not function identically on each user's terminal.

```
10  '   "screen.bas" J.P.Radley 320 CPW
    NY 10025, 25Feb84, for Xenix MBASIC
20  DEFSTR A-Z:
    DEFINT X
30  DEF FNSCREEN(Z)=CHR$(27)+Z
40  GRAPHICSTART="RG":
    GRAPHICEND="Rg":
    REVERSEVIDEO="RD":
    NORMALVIDEO="R@"
50  CURSORON="RC"
    CURSOROFF="Rc"
60  CLEARTOENDOFSCREEN="J":
    CLEARTOENDOFLINE="K"
70  INSERTLINE="L":
    DELETELINE="H":
    INSERTCHARACTER="P":
    DELETECHARACTER="Q"
80  UP="A":
    DOWN="B":
    RIGHT="C":
```

```
    LEFT="D":
    HOME="H"
90  PRINT FNSCREEN(CURSOROFF)FNSCREEN
    (HOME):
    X1=-1
100 FOR X=1 TO 6:
    PRINT FNSCREEN(UP);:
    NEXT
110 FOR X=1 TO 50:
    PRINT FNSCREEN(RIGHT);:
    NEXT
120 PRINT "GRAPHICS RUN"X1+2
130 PRINT TAB(15)"Note that the cursor is
    off ...";:
    GOSUB 290
140 FOR X=1 TO 10:
    PRINT FNSCREEN(LEFT);:
    NEXT
150 FOR X=0 TO 10:
    PRINT FNSCREEN(INSERTCHARACTER);:
    NEXT:
    GOSUB 290
160 PRINT"   what?";:
    GOSUB 290
170 FOR X=0 TO 8:
    PRINT FNSCREEN(LEFT);:
    NEXT
180 FOR X=0 TO 10:
    PRINT FNSCREEN(DELETECHARACTER);:
    NEXT:
    GOSUB 290
190 FOR X=0 TO 10:
    PRINT FNSCREEN(INSERTLINE);:
    NEXT:
    GOSUB 290
200 FOR X=1 TO 18:
    PRINT FNSCREEN(RIGHT);:
    NEXT
210 FOR X=0 TO 10:
    PRINT FNSCREEN(DELETELINE);:
    NEXT:
    GOSUB 290
220 PRINT FNSCREEN(CLEARTOENDOFLINE);:
    GOSUB 290
230 PRINT FNSCREEN(DOWN)FNSCREEN(DLEARTO
    ENDOFSCREEN)FNSCREEN(GRAPHICSTART);:
    GOSUB 280
240 PRINT FNSCREEN(REVERSEVIDEO);:
    GOSUB 280
250 IF X1 THEN X1= NOT X1:
    GOTO 100
260 PRINT FNSCREEN(NORMALVIDEO)FNSCREEN
    (GRAPHICEND)FNSCREEN(CURSORON)
270 PRINT TAB(82)"Your mission is to
    figure out why Radley left the second
    run to crash into a    RETURN without
    GOSUB error in 280, and why is run 2
    so different from 1 anyhow?
280 PRINT:
    FOR X=95 TO 126:
    PRINT USING "### !     ";X,CHR$(X);:
    NEXT:
    RETURN
290 FOR X=0 TO 3000:
    NEXT:
    RETURN
```

MODEL 100 & MODEL II/12/16

☆ UPLOADING AND DOWNLOADING INSTRUCTIONS ☆

SETTING PARAMETERS

ON MODEL 100	ON MODEL II/12/16
TELCOM, <ENTER> F3 (STAT)	TERMINAL, <ENTER> S, <ENTER> (get TRSDOS from TERMINAL)
78E1E, <ENTER> (legible) or 88E1E, <ENTER> (faster) or 98E1E, <ENTER> (max speed)	SETCOM A=(4800,8,E,1) (legible speed) or SETCOM A=(9600,8,E,1) (faster) or SETCOM A=(19200,8,E,1) (max speed)
..... F4 (TERM)	R, <ENTER> (open ram buffer) Y, <ENTER> (reset ram buffer) T, <ENTER> (enter TERMINAL mode)

DATA TRANSMISSION

MODEL 100	MODEL II/12/16
F3 (upload) FILENA.DO, <ENTER> <ENTER> (no width default) (UP lable now in reverse video transmission ends when UP is normal) .... F8 (exit TERMINAL mode) Y, <ENTER> (disconnect) F8 (return to MENU)	.... <BREAK> (return to TERMINAL menu) C, <ENTER> (save buffer to diskfile) FILENAME/EXT, <ENTER> (file saved) Q, <ENTER> (exit TERMINAL mode) ....
MODEL II/12/16	**MODEL 100**
.... G, <ENTER> (get file) FILENAME/EXT, <ENTER> X, <ENTER> (xmit ram buffer) .... <BREAK> (return to TERMINAL menu) Q, <ENTER> (exit TERMINAL)	F2 (download) FILENA.DO, <ENTER> .... F8 (exit terminal mode) Y, <ENTER> (disconnect) ....

Compliments of:

T-BUG, May 1983, by Carlos Hidalgo (312) 623-9661

CUSTOM COMPUTING INTRODUCES

FRIENDLY MENU™:

EASIEST WAY TO BREAK THE LANGUAGE BARRIER.

Finally . . .
an effective tool to combat computer language-induced frustration and aggravation. Friendly Menu offers a unique way to translate and display, in your own words; and readily access programs currently available in your system. Basic. Do Files. Machine Language.

CUSTOMIZES

Truly innovative, Friendly Menu allows the computer user to communicate according to his or her normal work style and concept base. Whatever the application, with Friendly Menu, you can translate computerese into familiar, easy to remember, and thus more accessible aliases . . . vernaculars . . . idioms . . . proprietary jargons. The possibilities are limited only by your vocabulary and imagination.

LOOK AT THESE FEATURES

- available for the Model II, XII and XVI
- includes several utilities and samples
- room for up to 66 entries—with three permanent entries: TRSDOS READY, CHANGE DISKS, SHUTDOWN COMPUTER
- provisions for add, delete, edit, change or move names anywhere in the menu
- Friendly Menu for TRSDOS 2.0a or b (floppy version) $69.95; Friendly Menu for TRSDOS 4.2 Hard Disk Version $89.95

MasterCard and Visa welcome.
Certified Check, Money Order or C.O.D.
Add $4.00 for shipping and handling.
NYS purchasers add 5.25% sales tax.

FRIENDLY IS COST EFFECTIVE

The productivity of information management within an organization is often measured by cost-effective utilization of its resources. Thanks to Friendly Menu, computer use has never been easier. And, never so much fun. So, break the language barrier and let's communicate!

If you wish further information, call:

914-471-9318

Custom Computing Company, Inc. 34 Scenic Hills Drive • Poughkeepsie, New York 12603

FRIENDLY MENU™ is a trademark of Custom Computing Company, Inc.

Replacement key caps and graphics make LUCID a user-friendly database management system.

Review Of LUCID

BY ARNOLD H. FISCHTHAL, Ph.D.

Since a great deal of computer usage involves keeping track of large amounts of information, it is not surprising new database management systems continue to be developed, with each new system trying to outdo the others. However, no matter how hard the designers try, no system can be all things to all people.

The database management system LUCID, published by SofterWare, Inc., does not follow in the footsteps of other systems.

Although LUCID is not the most comprehensive database, it has some very distinct and exciting features that make it stand apart from the others. One of these features is its on-line HELP routine. During any phase of data entry, manipulation, report generation and so on, you can press the "HELP" key <F1> and various descriptive messages will appear on the bottom of the screen to guide you through whatever steps are necessary to proceed.

The system comes with replacement key caps for various keys on the keyboard. For example, the <F1> key cap can be replaced with an identical cap that says "HELP" in addition to "F1"; the <F2> cap can be replaced with one that says "START OVER". Several other function keys on the Model 12 and 16B have replacement caps ("DELETE," "ERASE LINE," "BACK WORD," etc.), allowing the user to quickly gain familiarity and confidence in using the system.

A second, and highly enjoyable, feature of LUCID is its exceptional, yet utilitarian, use of graphics, allowing the user to read messages, properly manipulate disks, and watch the screen while the computer is chugging merrily along. For example, when you choose the option of backing up your disk, you will see an excellent (for a Radio Shack computer) reproduction of your computer system. On my disk it showed a two drive Model 12 with drive 1 flashing and a message indicating that the backup disk was to go into that drive. When a sort of the data file was taking place, an hour-glass (no, the sort does not take an hour) was displayed with the sand running down.

Although LUCID is not the most comprehensive database, it has some very distinct and exciting features that make it stand apart from the others.

When setting up your file structure, you are allowed up to fifty fields, for a grand total of approximately five hundred bytes of data storage per record. After the database has been set up and is in use, you can copy and restructure parts of one database to another to merge databases and perform similar operations between databases.

You are allowed four different sorted indices, which can be used when printing reports. The sort can be performed on any field defined in the record. An added nicety to the sorted files is the sort date. The date on which the sort was performed is displayed, giving you a reference date by which to judge when choosing an updated sort.

There are various choices to make when generating a report. The report itself can be printed or displayed on the screen. It can have a horizontal format, vertical format, or complete format (in which the entire record can be printed). The records to be included can be the entire database, or they can be selected by field, active records, inactive (deleted) records, or both active and inactive.

Compacting a file will permanently delete records that you have flagged as inactive. You have the option of leaving the deleted records blank, to be filled in by subsequent new records, or actually moving the upper part of the database down, by appropriately filling the newly emptied records. However, doing this would change the record numbers of the non-empty records thus moved.

LUCID, as most database management systems, lacks automatic indexing, the continual, automatic updating of an index as records are added to a file. According to Dr. Nathan Relles of SofterWare, automatic indexing will eventually be available on the XENIX version of LUCID.

LUCID is perhaps the most user friendly system I have used. Its on-line help messages, as well as its graphics displays, make day-to-day usage very enjoyable. If you try to enter erroneous data, a message describing the problem and telling you how to remedy the situation will be displayed automatically. If an actual error occurs in disk I/O or BASIC interpretation, an appropriately descriptive message appears, telling the user step-by-step what to do in order to continue.

One possible drawback you might find in this system is its lack of a printed manual. All features are described via the on-line HELP key as well as a menu option, EXPLAIN CHOICES (how to use this system). Although this reviewer is from the school that believes a well-written manual is almost indispensable, Dr. Relles pointed out that he has had no major complaints from LUCID's many users.

What LUCID does, it does very well. A hard disk version is available, as well as a soon-to-be completed XENIX version. Each of these versions has been improved over the floppy version by the inclusion of additional features.

LUCID is available from
SofterWare, Inc.
200 Office Center
Fort Washington, PA 19034
215/628-0400
Cost: Floppy Version $249.00
Hard Disk $349.00
XENIX $499.00

CLASSIFIED

TRS-80 Model 16. 2 Drives. Diablo 630 Printer. Excellent Condition. Perfect Writer, Multiplan, dBase II, dUtil, The Word, Cross Talk, Decision Analyst, Pascal, CPM and more. $5000. Phone: 216/582-2888.

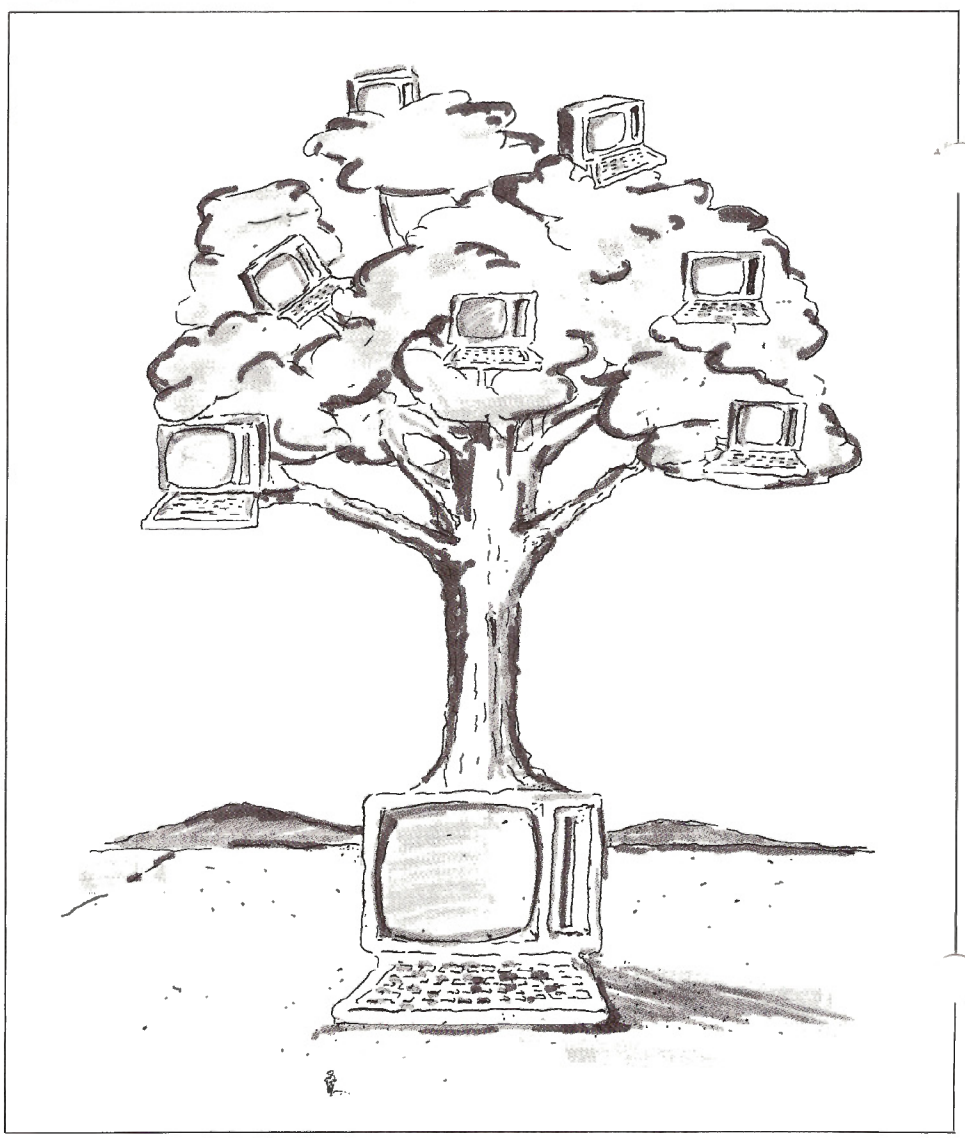
Business Graphs frustrating to create and update? Produce professional bar and lines charts using menu-driven software. For Models II, 12, and 16 with graphics board. Write to: JAF Computing, Box 7657, Nashua, N.H. 03060.

ARCNET
The Sleeping Giant

BY BRO. GARY ECK, S.M.

Since its introduction, very little has been published to describe the use of ARCNET in concrete situations. This article describes the experiences of one ARCNET installation.

A giant of a computer system lies hidden in the Radio Shack product line. Based on the Model II and Model 12 workhorse computers, the ARCNET system allows for the linking of many computers into a powerful, effective local area network system. Actually, ARCNET provides an efficient way of linking many more computers than XENIX.



How ARCNET Was Chosen

Early successful applications of individual TRS-80 Model II computers in a few administrative offices of Chaminade High School, a large Catholic boys' school on Long Island, led to a decision to extend computers to all administrative offices in order to streamline office procedures. One of these early applications was the formation of a comprehensive data base on each student, using Profile Plus (and all its extensions). With an enrollment of approximately 1580, the data base was quite large, requiring four floppy disks to handle all the data, such as names, addresses, birth dates, eighth grade schools, parents' occupations, standardized testing results, and so on. In addition to the necessary summaries, mailing labels, and similar uses, this data base has also allowed the easy preparation of special analyses and detailed data which was previously too demanding to research.

The large number of students generated a constant flow of data changes which were communicated using both the traditional paper memo and frequent copying of all four disks needed for each computer using this data. The need for some type of network linking the several

computers soon became evident. Cost considerations and the favorable experiences of the school personnel using Model IIs gave preference to expanding the TRS- 80s into a network rather than purchasing an entirely new system.

The XENIX operating system was carefully considered. To oversimplify, XENIX uses the Model 16's power and memory to service three users. By partitioning memory and processor activity, XENIX supports three (and more recently six) users simultaneously. The particular needs and administrative responsibilities within the school divided up nicely into four groups of three related users, each which XENIX could have served well, except for the lingering problem of communicating changes and data among the various systems. The chore of transferring updated disks and paper notices still remained if this arrangement were adopted.

The ARCNET system, on the other hand, allows for the linking of more than three computers into a common network. Based on token passing technology licensed from Datapoint Corporation, Tandy's ARCNET allows for an almost instant networking of Model IIs and 12s.

System Components
The system consists of three major parts: applications processors, hubs and cables, and file processors.

Applications Processor
Each user of ARCNET is considered an applications processor. An ARCNET communications board must be installed in each user's computer; when in use, the machine must operate with TRSDOS 4.3 which is very similar to TRSDOS 4.2, with the addition of a few commands to coordinate the ARCNET communications.

After executing a command to activate the ARCNET board, the user must use the MOUNT command which enables the local computer to access the ARCNET files as though they were located on one of the local disk drives. In our application, a brief DO file automatically executes these commands; once this file has executed, the user sees the same Model II or 12 he is accustomed to, with the addition of the contents of an entire hard disk which happens to be located elsewhere in the building.

For example, a commonly used BASIC program looks up the class schedule for a particular student; this program runs as usual at the application processor, but instead of finding the desired files on the local Drive 0 or hard disk, it will find the files on the remote disks provided by ARCNET.

Hubs and Cables
The applications processors are joined to common hubs by ordinary RG-62 coaxial cable. Active hubs have eight ports to link to applications processors, file processors, or other hubs. Passive hubs can be used to more cheaply link three users into one active port (a simple coaxial "Tee" connector is even cheaper when only two users are involved).

Figure 1 shows the arrangement of active hubs and cables connecting the school's ARCNET equipment. In our setting the longest single cable run is 400 feet, although the system can extend up to 2000 feet.

The File Processor
The third major component of an ARCNET system is the file processor. There must be at least one file processor, although more may be added as needed. The file processor is an hard disk based computer which is totally devoted to

serving the system by making its files available to any user who wants to access them. To simplify explanations, we describe these as "public" files which are accessed on ARCNET. When the power is first turned on at the file processor computer, it is an ordinary TRSDOS 4.3 machine and can be used as usual when not activated for ARCNET. The command ARC80FP causes a new operating system to load, overwriting TRSDOS with ARCDOS. After a second entry of date and time, the command START FP is given to complete the transformation of the machine into a file processor for ARCNET. No further attention is needed at the file processor until it is time to turn it off at the end of the day (with the command DO ENDARC). Once ARCDOS is loaded, there is no way back to TRSDOS except the reset switch.

What ARCNET Does

Practically, what the individual user experiences with ARCNET is very simple. Instead of many disks to keep organized and put into various expansion drives, there is only one floppy disk which goes into Drive 0 and does everything. The local computer turns on as usual, except for a couple of brief messages which display as the DO STARTARC file automatically connects to the system after the date and time inquiries. The user runs whatever programs he wants without having to worry about changing floppy disks or having the most recently updated disks at hand. When the work is done the user effectively has a hard disk full of current programs and data, without having the equipment located in the office.

Although the capacity of the hard disk is given through ARCNET, its speed is not; the system runs at floppy speeds in order to accommodate all the necessary communications. Sometimes this can be aggravating as a file intensive program such as Profile Plus repeatedly reads a directory of several hundred items looking for multiple files. Once the user has the files, things do work quite rapidly.

Our System Set-Up

According to the product literature, ARCNET can support up to 255 computers, including one or more file processors. Our application has grown to fourteen users and one file processor. Four users have their own hard disks and rarely connect to the system; these computers are in the administrative offices which maintain the official school records, such as grades, the student data base, schedules, etc. On a regular basis,

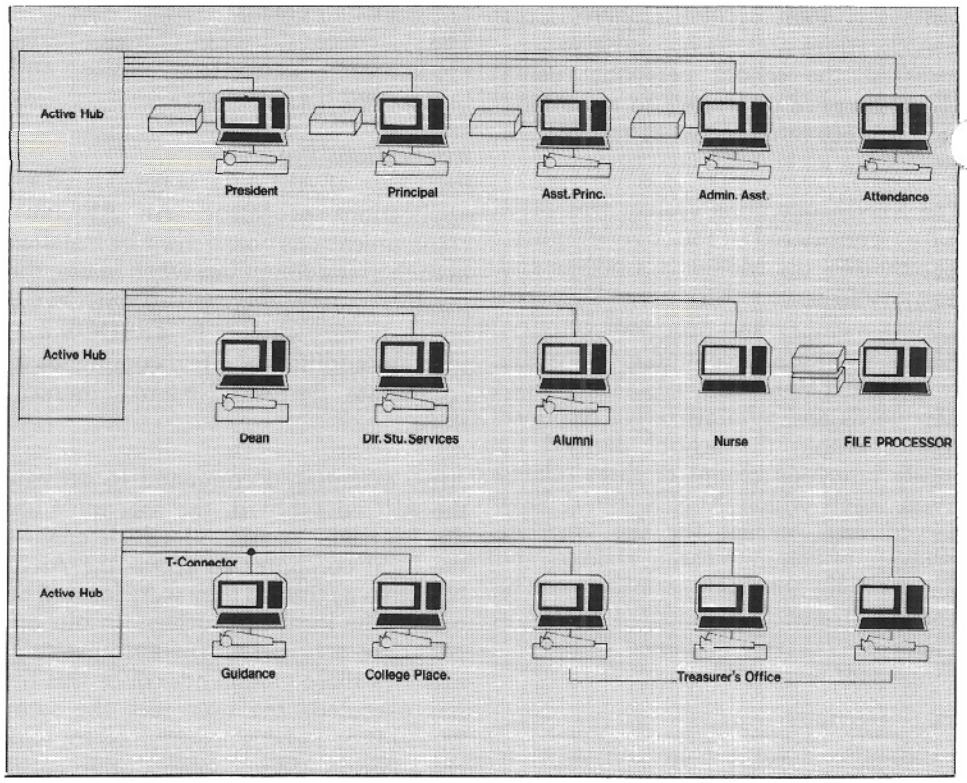
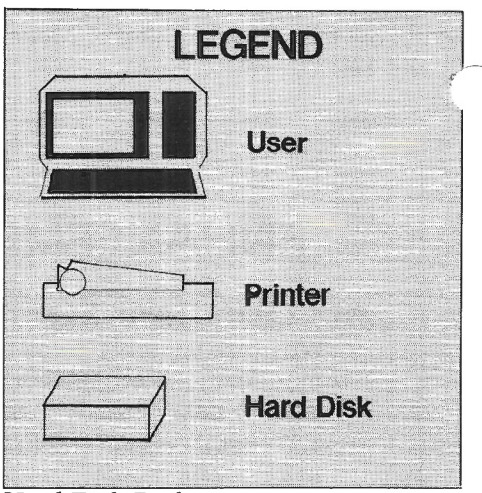


FIGURE 1

these official files are simply copied over to the "public" files, bringing along all updates without requiring paper memos or transferring of disks. The maintenance of the official files distinct from the "public" system prevents any tampering or mischief along the lines of the movie "War Games." The limited use of passwords within Profile Plus and TRSDOS adds additional protection where appropriate.

Eventually twenty-four administrators or secretaries will have access to ARCNET. Each has a Model II or 12 computer, a local printer, and a single disk to access ARCNET. Without ARCNET, a three drive expansion bay or hard disk would be needed at each location. Not only is the financial savings substantial, but there is no need to find room for the additional equipment in each office. With only one disk to gain access to the entire ARCNET public files, there are no longer the problems such as having the right disk in the wrong drive or the disk from one data base being mixed with those from another data base.

ARCNET has proven to be very easy to expand. Adding another user consists of placing the equipment, connecting the single coaxial cable, providing one disk with a few simple files, and updating some identity code files at the file processor.



Hard Disk Backup

In addition to these obvious conveniences and savings, ARCNET has provided a simple, convenient way of doing the necessary tasks of backing-up hard disk files. Instead of saving files onto floppy disks (a tedious task which often involved fifteen or twenty disk swaps), we simply copy critical files onto the "public" ARCNET file and then from the "public" file onto another hard disk. A few DO files further automated this process so that no intervention is needed once the process begins, freeing the user to do other work or to update files on other machines. There have been times when one person was simultaneously backing-up files on four machines without using any floppy disks beyond those needed to access ARCNET.

ARCNET Weaknesses

While ARCNET is very effective in our setting, it is not without its weaknesses. ARCNET is designed to be invisible to the user, which it usually is; when there are problems, however, it is also rather invisible to the person trying to diagnose the problems. Most reference manual suggestions amount to "check the file processor" but give no explanation of what to check for or how to check for it. A good diagnostic manual and software is very much needed.

Another weakness results from the inability of Profile Plus or BASIC to share the same file between two programs. Once a program has claimed a file, even just to read it, no other user can access it. If a second user attempts to open a file which is in use, a "File Busy" error message occurs. Since most of our users are working on different tasks, this is not a serious limitation; when it does occur, a five or ten minute wait usually solves the problem.

A third difficulty arises in that ARCNET is slow in reading the directory for the "public files" at the file processor; the wait can seem rather long if a program needs to read the directory many times. A new, faster version of ARCDOS would be a major improvement in the system.

Conclusions

All in all, ARCNET has been effective in our setting. Already the system has grown far beyond the capabilities of XENIX. With Profile Plus, SCRIPSIT, and BASIC all on ARCNET, we have far outgrown the biggest XENIX system and we have not needed to make any changes to our already existing data bases. By using DO files to automate the connection of a user to ARCNET, only the most minimal training has been necessary for an operator to move from a stand-alone computer to ARCNET.

ARCNET is indeed a powerful system with many capabilities and an enormous capability of expansion. Certainly, not every situation needs to network computers, and even fewer situations need to network as many computers as we have. For places with the larger number of computers to network, ARCNET is effective and workable.

Tandy has developed a sleeping giant of a local area network system. A few improvements may help to awaken the giant to even greater power. ■

If your operating system is CP/M, here's some advice for reducing keystrokes and saving time. Try using single character file names, SUBMIT programs, and automatic start-up utilities.

ONE KEYSTROKE IMPROVES EFFICIENCY

BY WILLIAM JOHN ELLIOTT

While looking over the shoulder of Mr. Robert Yam, a local Model II user known as the "Philippine Phlash," I noticed he was calling up programs and moving files much faster than I was able to. Although not a 90 word per minute typist, I can hold my own with most computer types while typing file names at a rapid rate.

I began to watch my friend's fingers more closely. Sure enough, he simply wasn't hitting as many keys as I did. I asked, "How'd you do that?" I explained to Phlash that he was not following the rules for calling up programs. He responded, "Don't you use single character file names?"

Since I had never heard of using single character file names, Phlash explained the concept, which I present to you here, along with uses for CP/M SUBMIT.COM and ATON's AUTOEXEC.COM.

Single Character File Names

While the concept of a single character file name is easy, its efficient practice is a little more complex. If you have a file name of two or more characters, you may reduce the number of keystrokes required to call that file by renaming the file using a single character file name.

This limits the number of files to the number of characters, greatly increasing the chance of overwriting an existing file. Therefore, the concept must be used sparingly and intelligently to reduce the possibility of error.

I know of no rule book for the application of single character file names, so I developed a few rules of thumb:

Rule 1: If you choose to use a single character file name for a data file, you greatly increase the possibility of overwriting data.

Rule 2: For greatest efficiency, use single character file names for files that are called frequently.

Rule 3: Use single character file names that mean something to you, because they are a customization of your working environment.

Shortened file names, such as WS for WordStar and CS for CalcStar, are well known. The customized file name XD replaces XDIR, an expanded directory program for CP/M. Although XD is not a single character, it illustrates the principle of reduced keystrokes and is definitely a candidate for single character fame due to its frequent use.

To replace XDIR with the single character "X", type

```
A>REN A:X.COM = A:XDIR.COM
```

Another program I use frequently, BID, sets up the Tandy Daisywheel II printer for bi-directional printing and proportional spacing when used with Peachwriter (formerly Magic Wand). Prior to printing a document in Magic Wand I must always run BID. Changing the name to "B" reduced this nuisance by two characters. I will show how to reduce it even more later.

SUBMIT.COM

Shortly after reducing my file names to shadows of their former selves, I discovered another useful tool which has been around forever in CP/M, the SUBMIT.COM program. The occasion again dealt with the BID program used to set up the printer before each print cycle.

Prior to changing the program names, the following sequence was used to print a program (the BID program would take 30 seconds or so to set up the printer):

```
A>BID
A>PRINT (filename)
```

As I stated before, this was a nuisance; two required program calls with a wait after each. Enter the CP/M SUBMIT.COM program. Going into a text editor (in this case Magic Wand), I proceeded to write a submit program named BID.SUB:

```
BID
PRINT $1
```

This program is submitted (run) using SUBMIT.COM in the following manner:

```
A>SUBMIT BID (filename)
```

where the indicated filename is substituted into the position of $1 in the program. When the program is submitted, it automatically runs the BID program, then runs the Magic Wand print program and looks up the file requested by the filename. In this manner, I reduced a double call program to a single submit program.

We're serious about fun!
Software for II/12/16

RIOCHET© This is a fast action game that really tests your reflexes. The object of the game is to get the ball into the goal by deflecting it off the walls before your time is up.
TRSDOS® $25.00

STARTREK™© A large version with lots of graphics of the classic computer game. Takes about two hours to complete. ™ Designates trademark of the PARAMOUNT PICTURES CORPORATION.
TRSDOS® $25.00

STARSHIP/I© (By Richard H. Young. Editor of two /sixteen magazine.) This 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/12/16 and features a constant console display with no scrolling. It is a "strategic" game (Opposed to "tactical" games which involve reflex action but little thought). However, STARSHIP requires quick thinking, as all events are timed. STARSHIP requires quick thinking, as all events are timed. 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).
TRSDOS® $39.00

GAME PACK I© (by ADD, Inc.) Four games that use enhanced graphics not available from Radio Shack. INTERCEPT and REVERSEM for two players; LUNAR LANDAR and SINK THE BATTLESHIP for one player.
920-0001 TRSDOS® $50.00

GAME PACK II© (by ADD, Inc.) Consists of the four games in GAME PACK I© and two additional games. They are EAT'UM where you control a snake moving around the screen trying to catch its prey and REVERSEM II where it's you against the computer.
TRSDOS® $70.00

GAMES/TRS-80© (by Lance Micklus, Inc.) Contains the following six games:

THE MEAN CHECKERS MACHINE© — The game is played just like regulation Checkers. To win, you must eliminate all of your opponent's pieces from the checkerboard.

STARFLIGHT® — This is a Star Trek type game. The object is to collect data about Omega VI, an unexplored area of the galaxy; to locate and destroy the 20 Klingon Invaders.

CONCENTRATION — The computer will make up the game board. You will try to find the matching prizes. After each correct guess, you will have a chance to guess at the secret number. Guess right, and you win all of your prizes.

TREASURE HUNT — The Lumas caves, located in Lumusville, Vermont, are said to have 20 hidden treasures in them. Few explore the caves because it is said that pirates and dragons live there, and there are deep pits which many have fallen into and died. You, a smart and brave human, and I, an alert computer, will explore the caves and try to find the treasure.

BANKO — The Game is similiar to Black Jack. You draw numbers stopping before the total point value exceeds 11. If you go over 11, then your hand will be zeroed. If you draw 5 times without going over 11, then you'll receive bonus points equal to the value of your hand at the time you end your turn.

DOG STAR ADVENTURE® — The evil General Doom and his Roche Soldiers are ready to launch an attack against the forces of freedom lead by Princess Leya. While traveling to her secret command center, Doom attacks Princess Leya's ship. She is now being held prisoner on one of General Doom's battle cruisers. We must try to save the Princess and the treasury of her freedom fighting force.
TRSDOS® $75.00

ADVENTURES 1-12© (by Scott Adams of Adventure International) By definition, an adventure is a dangerous or risky undertaking; a novel, exciting, or otherwise remarkable event or experience. On your personal computer, Adventure is that and much more. In beginning any Adventure, you will find yourself in a specific location: a forest, on board a small spaceship, outside a fun house, in the briefing room of a nuclear plant, in a desert, etc. The top portion of your video display will tell you where you are and what you can see; the bottom section of the display is devoted to inputting commands to your robot computer and receiving messages that may arise as the result of your orders.
The object of a game is to amass treasure for points or accomplish some other goal such as preventing the destruction of the automated nuclear plant in Mission Impossible. Successfully completing a game, however, is far easier to state than achieve. In many cases you will find a treasure but be unable to take it until you are carrying the right combination of objects you find in the various locations.

#1 ADVENTURELAND — Wander through an enchanted realm and try to recover the 13 lost treasures.

#2 PIRATE ADVENTURE — The lost treasures of Long John Silver lie hidden somewhere — will you be able to recover them?

#3 MISSION IMPOSSIBLE ADVENTURE —In this exciting Adventure, time is of the essence as you race the clock to complete your mission in time —or else the world's first automated nuclear reactor is doomed!

#4 VOODOO CASTLE — The Count has fallen victim to a fiendish curse with you his only possible hope. Will you pull off a rescue, or is he down for the Count for good?

#5 THE COUNT — It begins when you awake in a large brass bed in a castle somewhere in Transylvania. Who are you, what are you doing here and WHY did the postman deliver a bottle of blood?

#6 STRANGE ODYSSEY — At the galaxy's rim, there are rewards aplenty to be harvested from a long-dead alien civilization, Will you be able to recover them and return home?

#7 MYSTERY FUN HOUSE — This adventure puts you into a mystery fun house and challenges you to find your way through and back out of it.

#8 PYRAMID OF DOOM — This is an Adventure that will transport you into a maddening dangerous land of crumbling ruins and trackless desert wastes — into the very PYRAMID OF DOOM!

#9 GHOST TOWN — You must explore a once-thriving mining town in search of the 13 hidden treasures.

#10 SAVAGE ISLAND PART I — A small island in a remote island in a remote ocean holds an awesome secret — will you be able to discover it? This is the beginning of a two-part Adventure, the second half concluding as SAVAGE ISLAND PART 2, Adventure #11.

#11 SAVAGE ISLAND PART II — The suspense begun with Adventure #10 now comes to an explosive conclusion with SAVAGE ISLAND PART II!

#12 GOLDEN VOYAGE — The king lies near death in the royal palace. You have only three days to bring back the elixir needed to rejuvenate him.
TRSDOS® $129.95
CP/M® $129.95

INTERACTIVE FICTION© (by Robert LaFore) The ability to actually interact with characters and situations — to take part in the story-sets INTERACTIVE FICTION™ apart from the competition. the The actions begins when you react to a given situation or event, and then key in your response. You aren't limited to a couple of words; using full sentences is encouraged. You can enter the same situation time and again, and by using a different response, experience new twists of the adventure.

LOCAL CALL FOR DEATH — Meet Detective Sir Colin Drolley, as well as a host of other intriguing rogues, as you set about the task of solving the perfect crime.

TWO HEADS OF THE COIN — You are one of the world's greatest detectives. Mr. Conway's wife, Georgina, has vanished, and now it's up to you to gather the clues and solve the mystery behind her disappearance!

HIS MAJESTY'S SHIP "IMPETUOUS" —You are at the helm of HMS Impetuous. As captain, your decisions could bring either fame or fortune or utter disaster to you and your crew.

SIX MICRO STORIES is an excellent introduction to the world of Interactive Fiction. They are six short, but complete, stories which involve you, the reader, in a variety of fascinating scenarios. They include THE FATAL ADMISSION, EMPTY WORLD, and four other stories.
TRSDOS® $49.95

TYCOON® (By Barry Ansin) This is a monopoly like game that you can play against the computer.
TRSDOS® $35.00

BIORHYTHMS® This will print on 8½ by 11 paper or display biorhythms charts of an individual starting any date and for as many days as desired. Includes date subroutines, which are used for testings, converting, and calculating time between dates.
TRSDOS® $25.00

AMORTIZE© Financial amortization calculator. Prints high quality amortization schedule on 8½ x 11 paper. Calculates one of the missing values, principal, payment, periods, interest rate, or balloon payment. Prints totals for each year. Includes full screen subroutines.
TRSDOS® $25.00

LISTER© Lists BASIC source programs on 8½ x 11 paper with provisions for separate multi-statement lines. Prints program name, date, time and first comment on each page.
TRSDOS® $25.00

DELUXE PERSONAL FINANCE© (by Lance Micklus, Inc.) This is a sophisticated and unique financial analysis package which is expense oriented. It can be readily customized to suit your personal financial situation.
TRSDOS® $79.95

FOREIGN LANGUAGE VOCABULARY BUILDERS© (by ADD, Inc.) Consists of over 1000 words and phrases that are tested in flash card fashion. You may add over 1000 additional words of your own choosing. Languages currently available are: FRENCH, SPANISH and GERMAN.
TRSDOS® (One language per disk) $45.00

HANDICAPPING® For classifying and ranking thoroughbred flats based on scientific methods. Use of this program will give you the edge and build skills resulting in rewards at the track.
TRSDOS® $100.00

Additional Programs Available
Call or Write For Free Brochure
and More Information.

Rizzo Data Systems Corp.

577 Burlington Rd., P.O. Box 458 • Bridgeton, NJ 08302-0356 • 609/451-7979

Shipping Charge $3.00 Per Order . C.O.D.'s Add $2.00 • 30 Day Money Back Guarantee On All Programs
Visa, MasterCard, Personal Checks or Money Orders Accepted

To simplify the process even more, I added the concept of single character filenames. Since I was used to calling this procedure a "BID PRINT" (for BIDirectional PRINT), I decided to use the characters B and P for the filenames. I used the CP/M RENAME utility to change SUBMIT.COM into B.COM (it could have been S.COM), and the program BID.SUB to P.SUB (it could have been B.SUB), which effectively reduced the original two call "BID PRINT" program to a simplified

A>B P (filename)

BID.COM can also be used with WordStar in a bidirectional mode only, so I wrote a submit program to run BID with WordStar. I named this program W.SUB and it looks like this:

BID
WS $1

To activate this program, type:

A>B W (filename)

This automatically calls up the BID program, sets up the Daisywheel printer, and drops you into the WordStar edit mode with the proper file accessed and ready for editing.

Again, the B equals SUBMIT.COM and W is a submit file.

AUTOEXEC.COM

The possibility of entering a program automatically upon boot-up is very useful to users who specifically work on one particular program and do not know an operating system. I know of several cases of temporary employees or employees who use only one program and do not have the time to become CP/M knowledgeable. They can input vast amounts of data in a user friendly program because of built-in instructions, repetitiveness, and simplicity.

To make the program one step easier for the user, I use ATON CP/M's AUTOEXEC.COM program to drop the user into the program when he inserts the disk. He never sees an A>, and all program instructions are in Standard English. AUTOEXEC may also be used with a program such as WordStar, where access to outside programs is available through the WordStar menus or special calls. Once you are in WordStar, all program and file access is accomplished using existing internal program calls, returning you to WordStar when finished.

Of course, Pickles & Trout's new 2.2m version of CP/M also has Auto Execution capability. At **Advanced Computing**, *most all of our floppy disks put the user automatically into WordStar, our most frequently used program. Here's how to do it:*

At the A> prompt, type in MENU. From the first level of menus, select the SC option: "Change System Configuration." You will be presented with another menu, from which you should select the AE option: "Set up AUTO EXECUTE command."

From this point on, answer the questions as presented. You should answer "Do you want to change the command line?" with "Y". After the imperative: "Enter new command," you merely type "WS". Affirm the correctness of your responses throughout the remainder of the question and answer session, and on your next boot, you'll be taken directly into WordStar. Quickly, too, I might add. -Ed.

Keystrokes and time requirements can be reduced by using single character file names, simple SUBMIT programs, and available automatic start up facilities. While these procedures save only seconds, they greatly streamline operations, reduce frustration, and save a large amount of time in the long run. ■

CLASSIFIED ADVERTISING

Advanced Computing Magazine will accept Classified Ads for publication.

Subscribers may place one ad per issue at the rate of $7 per column inch. Ads will be set in the same size type as the text in the magazine.

Ads must be for hardware, software, or services which would be of interest to Model II/12/16 users (no mystical societies, chain letters, book clubs, etc.).

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.

Closing date for classified ads is the 1st day of the month prior to issue date; i.e., ads for the July/August issue must be received by June 1.

For projects involving less than 200 tasks, all with known manpower, duration, and indirect costs, the mainframe computer with its $50K+ CPM package is overkill. For such projects, Milestone and the micro do the job nicely. Very nicely, in fact.

Review Of MILESTONE

BY RICHARD TRAHAN

Milestone is a PERT/CPM project management package written in Sorcim Pascal/M which runs under CP/M for a variety of micros.

Many readers are probably already wondering what PERT/CPM is, so I will attempt a perspicuous propaedeutical of this subject. Only a quickie, I promise.

Critical Path Method, or CPM (not to be confused with the CP/M operating system), is a tool for the planning and management of projects which can be represented as a collection of discrete tasks whose duration, manpower, and indirect cost requirements are fairly well known. The projects must also have known sequential dependencies; that is, each task's prerequisites and successors must be defined.

The network of interconnected tasks will identify one or more paths from project start to completion which has no "slack time"; this is the critical path. Lengthening any task which lies on the critical path will delay the overall project.

A well designed CPM program will allow the project manager not only to identify the critical path, but to experiment with the network interactively to examine the tradeoffs between manpower, cost, and time.

Whereas CPM is a deterministic approach to project management, another method, the Program Evaluation and Review Technique, or PERT, is based on probabilities. In PERT, the duration, manpower, and material requirements of each job are described by probability functions, which in the most common (and most difficult) case will be step functions. For example, building materials may only be shipped in bulk quantities at discrete periods, so inventory levels are not a smooth function of time. This makes PERT network analysis computationally very difficult and beyond the capability of micros.

In the computer world, the differences between CPM and PERT have faded to the point where the terms are used interchangeably (I will use "CPM" to refer collectively to both), and typical programs combine their concepts, like cost reporting, which the original PERT had no formal concern with.

Fortunately, most projects are small and do not involve many advanced concepts. For projects involving less than 200 tasks, all with known manpower, duration, and indirect costs, the mainframe computer with its $50K+ CPM package is overkill.

For such projects, Milestone and the micro do the job nicely. Very nicely, in fact.

Configuration

The process begins with the configuration necessary for most CP/M software. An interactive utility, CONFIG, queries the user about his terminal, printer, data drive selection, and margin preferences for reports. Too bad it doesn't ask about line feeds; more on this later.

For the Model II, Lifeboat CP/M, P&T CP/M, and FMG CP/M are all on a menu of predefined systems, so the tedium of defining all the byte sequences for various screen effects is eliminated.

"Learnhow"

Next the user runs the "learnhow" program, aptly named. Learnhow gives the user practice in reading the standardized display, changing program variables, and responding to prompts.

The display seems to be an original system and is extremely user friendly.

There are some clever innovations in editing field data. For example, when modifying a date, which is always of the form mm/dd/yy, the user can simply type +45d, which will compute the date 45 days from the old date, and construct the proper date string. Other possibilities are of the form +50m, +7w, and -6y, for 50 months hence, 7 weeks hence, and 6 years ago, etc.

A Sample Project

After mastering learnhow, the user experiments with a sample project, the installation of a water main. It makes sense to start with a prepared project, observing the effects of modifications on displays and reports, because then the design of original projects becomes easier.

After invoking the main program, "mileston," the Startup menu appears and offers the opportunity to supply your name and a date which will appear on all reports. Milestone keeps a file of the last used name and date, so you won't have to change these often. Then, on to the main menu, from which you can create a new project or load an existing one.

The Load menu displays the project currently in memory, if any, and allows selection of print, modify, or disk save functions. The "Modify" option will bring up the same menu as creation; this includes ability to modify or define the project name, project leader, time scale, start date, direct cost units, and manpower cost units.

Only one time scale is allowed, with up to 9999 time units per task. The time unit definition affects the project time schedule display. For example, if your units are in hours and you define 8 hours per working day, the time schedule will show 8 columns for each day of the project. If your units are in weeks, the scale will show one column per week, beginning with the starting date defined in the main menu.

There is also an Hours, Days, and Holidays menu. Here you can define the starting and ending hours of the working day, including start and end of lunch, days of the week designated as working days, and specific holidays. These factors will be used in computing calendar elapsed time, and will thereby affect the time schedule display (see below).

Manpower cost units refer to how units are reckoned: dollars, thousands, or millions. On another menu, the user defines how many of these cost units are associated with each person of a particular skill category working for one time unit. The user can define up to nine skill categories, each of which has an associated cost per time unit.

A sample case might be category #1 for laborer at 5 cost units per time unit, category #2 for craftsman at 10 cost units per time unit, etc., where the cost unit is dollars and the time unit is hours. Then, for each task in the project, you enter how many people of each defined category are required, and how long, i.e., how many time units, the task will take. Unfortunately, all persons are assumed to work for the entire task time; if some work less than others, you will have to break the task down into simpler tasks, such that all persons in each subtask work for the duration.

Direct cost units follow the same principle. Direct costs are one-time expenses related to each task, and are not affected by duration or manpower. The user can define the number of direct costs for each task in the project.

Sound confusing? That's what the sample project is for. It does take a little experimenting, but after an hour or so, all will become apparent, and defining your own project will be a snap.

The Time Schedule

The "Modify" option of the Load menu gets you into the time schedule, the heart of the whole system. A fairly complex, but friendly, screen display and data input system allows you to define, modify, and delete tasks. Each task has many attributes, which include a unique number (automatic renumbering is possible), brief description, duration, predefined start date if necessary, direct cost, the required number of each people from each skill category, and the prerequisites.

Prerequisites are the tasks that must be completed before the current task can start. For example, if you are adding task #25 to the system, and decide that it cannot begin until tasks #4 and #7 are finished, the prerequisites are listed as "4,7". If task #7 has #4 as a prerequisite, then it is only necessary to list #7. You can insert tasks between others by making use of prerequisites. A task may have more than one prereq, and more than one task can have the same prereq.

As tasks are added, deleted, or modified in the system, Milestone instantly recomputes the "network", which is the array of task dependencies and its associated graph. The graph is very well designed, and shows which tasks are on the critical path, and which contain slack time. Slack time occurs if a task can be late without affecting the project completion date. By definition, tasks on the critical path have no slack.

Since the number of tasks possible (159 on P&T), and the size of the time scheduling graph are usually much larger than the screen can display at one time, Milestone has arranged the graph like a spreadsheet, where the screen is a window and can be moved about with simple keystrokes.

Reports

Milestone produces a number of useful reports, driven by a Reports menu. The reports are Project Description, Job Description, Columnar Job, Time Schedule, and All of the Above (not a different report, but a menu selection that produces everything else).

The three reports above can be sent to either the printer, the disk, or the screen for previewing. The Time Schedule report cannot be previewed.

The Time Schedule report is the image of the spreadsheet type graph discussed earlier. Using information from CONFIG or the Print Options menu, Milestone will print several pages designed to be pasted side by side into one giant graph. You may have to do some experimenting with margins. The result is a professional looking display that conveys important information in a readable form; arrows made from equals signs (=) for tasks on the critical path stand out. Other distinguishing symbols make this report the most attractive of the whole system.

The Occasional Peccadillo

It was a strain to find an occasional peccadillo in Milestone, and I relate these more as suggestions for improvement than as a caveat to the prospective buyer. But make no mistake about it: I unconditionally recommend Milestone to anyone involved in the planning or tracking of small projects. And now for the weak points.

First is the annoying linefeed problem experienced by all Radio Shack LP-series printer owners running CP/M. It's not Milestone's fault, but since there are so many of these printers out there, I don't think the folks at Organic Software would have their integrity compromised if they offered a CONFIG patch to disable that cursed linefeed. I had to write all my reports to disk and use a utility to strip out the LFs, but that requires exiting Milestone each time, which is a nuisance.

The CONFIG file allows the report output device to be defined as a disk file. That means that any report directed to the printer will go to the file instead. The file name may also be defined from within the main program using the Modify Printout option of the Reports menu. However, Milestone

will not allow existing report files to be concatenated, so if you choose the All Reports option on the Reports menu, only the first of the four will be written to disk. The manual claims this is because Pascal will not allow appending to a sequential file. But this assumes that the same file name must be used for all four.

Why not dynamically change the file names by using a numeric filename extension, for example? Or maybe allow the user to define four file names, one for each report. Or maybe just don't close the file if the All Reports option is selected, until the last byte is written. I'm sure there is a straightforward solution to this.

Documentation

The manual, for the most part, is accurate and highly readable, but there are some weak spots. The tutorial is only three pages, and contains some small type in which it is unclear what the user is supposed to do. But you'll figure it out after a bit; striking the wrong keys can't do irretrievable damage, and the menus are self-documenting.

In the project creation section, there are two keystroke sequences which are wrong and will not produce the results shown in the text; but again, if you pay attention to what you are doing, correcting this will be easy.

The only must-fix I found involved previewing reports on the screen. It seems that a formfeed character is sent at the bottom of each page, which has the effect of blanking the screen. Scrolling is done a line at a time in response to any keystroke, so if you're using the Repeat key with some other and happen to hit a formfeed, you lose a screenful of text. This has the subtle effect of creating tension while previewing a document; you just know the screen will suddenly clear, but you don't know exactly when.

Milestone is available from
Digital Marketing Corporaiton
2363 Boulevard Circle
Walnut Creek, CA 94595
800/826-2222
Cost: $295.00
Available for CP/M and CP/M-86

CRITICAL PATH METHOD READING LIST

Critical Path Method has its own jargon derived in part from network and graph theory. We have nodes, master nets, logic dummies, several types of float, simple and complex compression and decompression, optimal and nonoptimal solutions, utility, circle networks, crashed paths, scheduling and resource leveling, Fulkerson algorithms, and much more.

CPM is a very complex field and fairly far along in its development. If you are involved in the planning of large projects, or would like to learn more about it, here's a reading list that contains material at various levels of difficulty.

Antill, James M., and Woodhead, Ronald W. *Critical Path Methods in Construction Practice*, 2nd ed. New York: John Wiley & Sons, 1970.

Bennett, F. Lawrence. *Critical Path Precedence Networks*. New York: Van Nostrand Reinhold, 1977.

Benson, Ben. *Critical Path Methods in Building Construction*. Englewood Cliffs, NJ: Prentice-Hall, 1970.

Faulkner, A. *Project Management with CPM*. Kingston, MA: R.S. Means Co., Inc., 1979.

Kaufmann, A., and Desbazeille, G. *The Critical Path Method*. New York: Gordon and Breach, 1969.

Levy, F.K., Thompson, G.L., and Wiest, J.D. *The ABCs of the Critical Path Method*. Harvard Business Review, September-October 1983, p. 97-108.

Martino, R.L. *Critical Path Networks*. New York: Gordon and Breach, 1967.

Morris, L.N. *Critical Path Construction and Analysis*. New York: Pergamon, 1967.

Packard, Philip C. *Critical Path Analysis for Development Administration*. The Hague, Netherlands: Institute of Social Studies, 1972.

Priluck, Herbert M., and Hourihan, Peter M. *Practical CPM for Construction*. Kingston, MA: R.S. Means Co., Inc., 1968.

Reynaud, C.B. *The Critical Path: Network Analysis & Resource Scheduling*. Aldwych, London: George Godwin, Ltd., 1970.

AUTHORS' INDEX

Richard H. Young, Publisher
Advanced Computing
131 East Orange Street
Lancaster, PA 17602
"Notes & Comment"
Page 2

John J. Esak
2139 P Street, N.W.
Washington, DC 20037
"The Scenery in XENIX Part 2"
Page 4

Ian B. Kazin
153 St. Lawrence Blvd.
Brick Township, NJ 08723
"Review of the RS COBOL Generator and RS COBOL Utilities from David Ray"
Page 12

Richard A. Bilancia
P. O. Box 620127
Littleton, CO 80162
"Creating Inverted Indexes Using TRS-XENIX Utilities"
Page 14

Robert P. VanNatta
222 South First St., Gray Bldg.
St. Helens, OR 97051
"Review of RED and VEDIT"
Page 18

Todd A. C. Heard
50 Skywood Way
Woodside, CA 94062
"A Disk Copy Queue Utility"
Page 22

Paul Naitoh, Ph.D.
Applied Psychophysiology Group
4446 Marseilles Street
San Diego, CA 92107
"The Statistician"
Page 24

Jean Pierre Radley
320 Central Park West
New York, NY 10025
"XENIX Graphics"
Page 28

Carlos Hidalgo
T-BUG
311 Longview Road
Waukegan, IL 60087
"Model 100 & Model II/12/16 Uploading and Downloading Instructions"
Page 30

Arnold H. Fischthal, Ph.D.
Advanced Data Design
184-08 Tudor Road
Jamaica Estates, NY 11432
"Review of LUCID"
Page 32

Bro. Gary B. Eck, S.M.
Chaminade High School
Mineola, NY 11501
"Special User Report: ARCNET, The Sleeping Giant"
Page 34

William John Elliott
ARAMCO Box 4302
Ras Tanura via Dhahran
Saudi Arabia
"One Keystroke Improves Efficiency"
Page 38

Richard Trahan
203 Main Street, Apt. 43
South Bound Brook, NJ 08880
"Review of Milestone"
Page 42

CUMULATIVE INDEX

Indexing conventions used:

1-2:3.4 = Vol-No:Page.Column

Names are last-name-first, except for advertisers, who are listed as given in the ad, but are alphabetized by last name if individuals.

Aaron, Robert (author),1-4:10
Abacus Associates (ad),2-2:7
Accounting system, lawyers,2-2:18
Accounting system, XENIX,2-5:36
ACP (Pickles & Trout) review,2-1:47.2
ACP Review, rebuttal to,2-3:32
ADD, Inc (ad),1-2:30.3,1-3:5.3,1-4:33.3,1-5:3.3
Advanced Computing Service,2-5:2
Advanced Computing Service (ad),2-4:C3,2-5:C4
Advanced Computing Service description,2-4:17
Albert, Barbara (author),1-2:14,1-3:19,1-3:26, 1-3:37,1-4:43,1-5:31,1-6:18,2-2:18
Aldevco (ad),1-2:3,1-2:13,1-3:12.3,1-3:21.3, 1-4:5.3,1-4:21.3,1-5:9.2,1-5:21.2
ARCNET user report,3-1:34
Assembler-16,2-1:42
Assembly language course,1-2:4.3
Aton International, Inc. (ad),1-2:C4,2-2:C4
Aton's Micro/Remote 3270,2-3:2.3
Aukland Associates (ad),2-5:42
Authors' Index,2-5:52.3,3-1:46
Back-up, disk,1-2:14
Back-ups, XENIX (article),2-3:42
Ball Technical Services (ad),2-1:19.2,2-2:39.2, 2-3:25.2
Bank switching,2-2:24,2-4:38
Barilleau, Lambert (author),2-3:19.2
BASIC compiler,1-2:3.3,1-2:5.3
BASIC interpreter,1-1:2.3,1-2:2,1-3:33,1-5:24
BASIC interpreter enhancements,1-2:6,1-3:42.2, 1-4:13,1-5:2.2,1-5:30
BASIC Menu system,1-5:4.2
BASIC, PRO-80 review,1-2:11.2
BASIC, PRO-80 review corrected,1-3:32,1-4:22
BASIC Programming Riddle,2-3:33.3
BASIC sorts,1-4:22
BASIC to C language converters,2-1:3.3,2-2:2.3
BASTOC comments,2-5:3
Bates, Bruce V. (author),2-3:12,2-5:4
R. R. Belanger, Ph.D. (ad),1-2:15.3
Bennett, Mark (author),2-4:46
Bilancia, Richard A. (author),2-3:42,2-4:48.2, 2-5:36,3-1:14
Blair, R. Clifford (author),2-5:20
B.O.A. Systems (ad),2-3:27,2-4:15,2-5:41, 3-1:35
Boot error eliminator,1-4:2.2,2-3:1.3
Boyd, Waldo T. (author),2-2:30,2-3:33.3
Bridgeware (ad),1-3:25.3,1-4:3.3,1-5:25.3, 1-6:39.2,2-1:19,2-2:43.2,2-3:31.2,2-4:45.2, 2-5:25
Joseph W. Brooks, Accountant (ad),2-3:43.3, 2-4:12.3
Buffer for printer, review,1-1:15.2
Business programs available,1-5:38.3
Canning, David R. (author),1-3:45
Gary Carpenter (ad),2-5:19.3,3-1:45.2
Case for Model II or 16,1-3:9.2
Chips, computer, discussion,1-6:5
Classified advertising information,2-4:27.2, 2-5:18.3
Clemo, William J. (author),2-4:14,2-5:44
COBOL,1-2:3
COBOL programming service,2-4:3
COBOL, RMCOBOL comments,1-5:18
COBOL Utilities,3-1:12
Code generator,2-2:36
Coffee Break Casino review,1-6:17.2
Color plotter and Tandy Graph review,2-5:20
Communications packages,2-1:3
CompuServe features,2-3:46,2-4:20,2-5:14
Computer chip discussion,1-6:5,2-2:42
Computer evaluation methods,2-2:42
Computer Plus (ad),2-5:11.2
Computer User magazine,2-3:3
Roger Conant (ad),1-3:13
Conant, Roger (author),1-3:47,1-4:36,1-5:12, 1-6:42,2-1:28
Condor 20-3 DBMS review,2-2:4
Condor DBMS,1-2:5.2
CP/M, Aton,2-5:3
CP/M, Pickles & Trout,1-2:7.3,2-2:3,2-4:18, 2-5:3,2-5:3.2
CP/M vs. TRSDOS,1-2:5,1-6:20.2
CP/M-86 review,2-4:42
CP/M 3.0,1-3:1.2,2-1:3
CP/M-68K,2-3:1.3
CP/M-68K, programs for,2-2:3
CP/M-68K review,2-1:4
CP/M '83 EAST Show,2-3:44.3
CP/M - TRSDOS data transfer,1-6:42
CP/M Variants Review, Updated,2-3:3.2
CP/M variants reviewed,1-4:3,1-4:38,1-6:9
Creative Solutions, Inc. (ad),1-1:14,1-2:13.2
Cross reference listing,1-4:18,1-5:2.2
Cursor, underline (patch),1-6:8.3
Custom Computing Company (ad),1-3:18,1-4:21, 1-6:15,2-1:17,2-2:11,2-3:33,2-4:7,2-4:13, 2-5:13,2-5:49,3-1:17,3-1:31
Daisywheel II, Magic Wand for,1-6:31
Data files, converting TRSDOS to XENIX MBASIC,2-3:16
Data Management Systems (ad),1-3:44.3,1-4:27.2, 1-5:9.3,1-6:21.3
Data-MATE computer furniture (ad),2-4:C4, 2-5:47
Data-MATE furniture information,2-5:2.2
DataPro,2-3:2.3
Dataright Software (ad),1-5:17.3,1-6:35.3, 2-1:25.3,2-2:25.3,2-3:9.3,2-4:27.3,2-5:29.3, 3-1:40
DBMS, Condor,1-2:5.2,2-2:4
DDT debugging tool correction,2-5:23
DeBerry, Marshall (author),1-2:18
DEFUSR, BASIC,1-5:24
Dental office mgmnt. review,1-5:40
Directory, file, short-form,1-3:45
Directory of periodicals,1-1:8
Disk back-up,1-2:14
Disk copy queue utility,3-1:22
Disk drives,1-5:13,2-2:21.2
Disk emulator review,1-3:24
Disk to tape conversion,1-4:2.3
Disk usage techniques,1-3:26
DOSFIXES,1-2:26
DOSFIX comments,1-5:18.2
DOSFIX14 alternative,1-3:36
DOSPLUS II review,1-6:34
Draw 5 Poker game review,1-4:10
EASI Software (ad),1-3:43,1-4:9.2,1-5:47.2, 2-1:27.2,2-2:21.2,2-3:7,2-4:11,2-5:11, 3-1:15.2
Eck, Bro. Gary B. (author),3-1:34
Ehlen Enterprises (ad),2-1:33,2-2:7,2-3:7.2, 2-4:39.2,2-5:35.2,3-1:17.2
80 Computer Services (ad),1-4:31.3,1-5:15.3, 1-6:11.3,2-2:41.3,2-3:15.3,2-4:9.3
8086 Dual Processor Option,1-2:9.2,2-2:3.2
Electronic Specialists (ad),1-1:7.3,1-3:43.3, 1-4:38
Elliott, William John (author),1-6:31,2-1:36, 2-5:30,3-1:38
EMACS full screen text editor,2-4:3
Empress Technology (ad),3-1:37.2
Engineering, structural, prog.,2-2:38.2
Epson patches to WordStar,1-1:4
Epson, printing with TRSDOS,1-6:36
Lawrence S. Epstein Assoc. (ad),1-6:41
Equipment failure,1-2:10
Error messages, changing,2-2:17
Esak, John (author),2-1:20,2-2:11.3,2-3:24, 2-5:6,3-1:4
Fancy Font review,2-1:24
Fischthal, Arnold (author),1-2:6,1-3:42.2, 1-4:13,1-5:30,1-6:17.2,2-1:45,2-2:10.2, 2-3:30,2-5:34,3-1:32
Arnold Fischthal, consultant,2-5:2.2
Forge Products (ad),2-3:37.2,2-4:33.2

☞

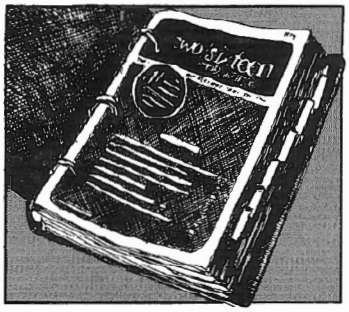
CUMULATIVE INDEX

Formatting discussion,1-6:18
FORMS command,1-3:10.3
Fort Worth Computers (ad),1-3:35,1-4:29,1-5:47, 1-6:29,2-1:19,2-2:35
FORTRAN-77 for CP/M-68K,2-4:3
Friendly Menu review,2-5:4
Furniture for computer use,1-4:2
Future equipment,1-2:3
Games for II/16,1-1:20,1-4:10,1-6:17.2,2-1:45, 2-4:47.2,2-5:9.2
Gandy, Ralph (author),2-1:10,2-2:28
Garbage collection, BASIC,1-5:30
Gash, David A. (author),1-5:2.2,1-6:34,2-2:36
General ledger system,1-5:34
Generic Software (ad),1-6:19.3,2-1:11.3
Glimpse Corporation, The (ad),2-3:45.3,2-4:5.5, 2-5:23.3,3-1:44
Good-Lyddon Data Systems (ad),1-2:23,1-3:7.2, 1-4:5.3,1-5:27,1-6:47.2,2-1:47.2,2-2:17, 2-3:43.2,2-4:12.2,2-5:5.2,3-1:13.2
Goodmans Computerized Business Services (ad),2-4:12.3, 2-5:51.3
Grammatik review,1-3:37
Graphics, high resolution,2-1:36
Graphics, XENIX,3-1:28
Green, Wayne,1-2:17.2,2-2:3.3
Gutkin, Donald (author),1-5:40,2-5:52
H & E Computronics catalog,1-5:38.2
Hard disk,1-2:8,1-5:1.2
Hardware directory,2-3:2.3
Harper McMasters Waterhouse (ad),1-6:19.3,2-1:11.2, 2-2:19.3
Heard, Todd (author),3-1:22
High resolution graphics,2-1:36
IBM 9000 (68000 based micro),1-6:3.3
IMPACC Associates (ad),2-2:19
Ingoglia, Mark A. (author),2-5:16
Intelligent Computing description,2-5:2
International Software Sales (ad),1-2:25.2, 1-3:33.2
Inverted Indices creation in XENIX,3-1:14
Investment Matrix,2-4:3.2
Investment portfolio analysis system,2-4:3.2
ISAM 300 Review,2-3:30
Jenkins, Robert R. (author),2-1:16
Jensen, Richard (author),1-5:48,2-1:46,2-2:21, 2-4:32,2-5:28
JMI Software Consultants (ad),2-5:21,3-1:11
Kaplan, Richard (author),1-6:8
Kazin, Ian B. (author),2-2:4,3-1:12
Keiser, Thomas Evan (author),1-3:24
KeyCon review,2-5:34
Keystroke efficiency,3-1:38
KILL command improvements,1-5:42
KILL prompt removal,1-3:47
Kitchens, Jess (author),2-5:26
Kloosterman, Jim (author),1-4:24,1-5:24,1-6:4, 1-6:24,2-1:12,2-2:42
KWIX 2.0 Review,2-3:34
KWIX (key word index) review,1-6:44
Laderoute, Charles D. (author),2-5:40
Lambe, Dean R., Ph.D. (author),1-6:44,2-2:21.2, 2-2:45,2-3:34
Langer, Robert D., MD (author),1-4:15,1-6:46, 2-2:40
LaPorte, Randy (author),1-2:18
Lawrence, Anthony (author),2-3:16,2-4:34
Lawyers accounting system,2-2:18
Ledger system,1-5:34
Lewis, John A. (author),2-3:14
LEX68 word processing system,2-4:3
Lippey, Gerald (author),2-1:28,2-4:38
Logic Designs, Inc. (ad),3-1:21.3
LUCID review,3-1:32
Lyddon, Joe (author),1-4:18
Lync 3.0 review,2-5:16
M & M Technologies Corporation (ad),2-4:11.2, 2-5:35.2,3-1:27.2
Magic Wand for Daisywheel II,1-6:31
Mailing lists,1-2:1.3
Manufacturers' software,1-5:20.3
Margus, Paul (author),1-6:36
Marymac Industries (ad),2-2:44.3,2-3:35.3, 2-4:9.3,2-5:51.3,3-1:45.3
Maxtek, Inc. (ad),1-5:41.2,1-6:7.2,2-1:7.2
MBASIC spooling in TRS-XENIX,2-4:48
McCurdy, Robert J. (author),1-6:8.3
Medical Micro (column),1-4:15,1-6:46,2-2:40
Memory upgrades,1-2:5
Memory upgrades from Snapp,2-2:3.2
Menu system for BASIC,1-5:4.2
Message to console w/o PAUSE,2-1:28
Micro Business Systems (ad),2-2:27.3
Micro Systems Exchange (ad),1-3:17.3,1-4:47.3, 1-5:37.3,1-6:31.3
Micro Systems Software (ad),1-3:11,1-4:11, 1-5:11,1-6:13
Microlaw of California, Ltd. (ad),2-3:43.3, 2-4:9.3,2-5:21.3,3-1:45.2
MicroLink II review,2-5:16
Micron, Inc. (ad),2-2:9.3,2-2:43.3,2-3:45.2, 2-4:23.3,2-5:42,3-1:41
MicroProgrammers, The (ad),1-1:15.2,1-2:30
MicroShell review,1-3:30
Microstat review,2-1:46
Microstat review correction,2-2:21
MicroTech Exports (ad),1-2:13.3,1-3:44,1-4:27.3, 1-5:15.3,1-6:11.3,2-2:29.3
Milestone review,3-1:42
Model 100 Disk/Video interface features,2-5:3.2
Model 100 review,2-5:30
Model 100 Uploading and Downloading,3-1:30
Model 16 features,1-1:19
Model 16 memory upgrades,2-4:3.2
Model II BASIC conversion to XENIX MBASIC,2-4:34
Model II requests,2-5:2
Modem discussion,1-5:32.2
Montreal TRS-80 Users' Group,2-4:3.2
Muller, Lance (author),2-2:17,2-5:24
Multi-user discussion,1-6:46
Naitoh, Paul, Ph.D. (author),1-5:46,2-3:46, 2-4:20,2-5:14,3-1:24
NewsNet,1-3:3.2
NewsNet (ad),2-4:33.3,2-4:C2,2-5:51.2,2-5:C2
Niessner, A.F. (author),2-4:24
NumCon review,2-5:34
OASIS bulletin board,2-1:3.3
Otten, R. David (author),1-5:13,1-6:9
p-System,2-3:1.3
Pascal 2 Compiler for UNIX,2-3:1.3
Patches, TRSDOS,1-1:6,1-6:8.3
Patches, TRSDOS 4.2,2-5:52
Patches, TRSDOS, Snapp,1-2:26
Patches, WordStar for Epson,1-1:4
Patching techniques,1-5:46
Patching WordStar 3.0,2-4:4
PC Telemart,2-4:3.2
Peek & Poke for Model II,2-3:17.2
Peek & Poke functions,1-6:24
PerfectCalc Spreadsheet review,2-5:28
Performance Learning Systems (ad),3-1:23.2
Periodical directory,1-1:8
Personal Pearl review,2-5:48
Piccoli & Sons, Inc. (ad),2-5:19,3-1:43
Pickles & Trout (ad),1-1:2,1-1:17.3,1-2:12.3, 1-2:15,1-3:9.3,1-3:13.3,1-4:17.3,1-4:25.3, 1-5:5,1-5:21.3,1-6:25.3,1-6:37,2-1:5.2, 2-2:31.2,2-3:17.2,2-4:35.2,2-5:33.2,3-1:7.2
Pickles & Trout ACP review,2-1:47.2
Pickles, Christine (author),2-3:32,2-4:18
Pollack, Bill (author),2-1:42
Power line problems,1-4:45,1-6:21.3
Preventive medicine systems,2-2:40
Printer buffer review,1-1:15.2
Printer comparison (article),1-5:6
Printers, high speed,2-5:3
Pro-80-Systems (ad),1-3:39,1-4:17,1-5:33.3, 2-1:33.3,2-2:7.3,2-3:45,2-5:9,3-1:15.2
Pro-80 BASIC review,1-2:11.2,2-5:24
Pro-80 BASIC review correction,1-3:32,1-4:22
Profile 16 faults,2-5:2.2
Profile 16 Review,2-4:14
Profile 16 user tips,2-5:44
Program generator,2-2:36
Programming services,1-6:3.3,2-1:2.3
Proofreader review,1-3:19

CUMULATIVE INDEX

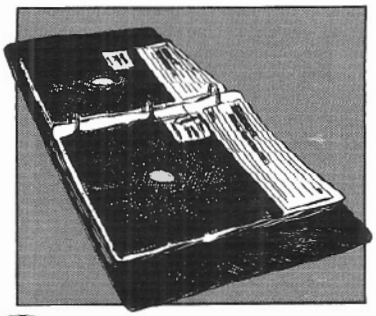
Quant Systems (ad),2-5:17,3-1:9
QWERTY tutorial update,2-4:17.2
Radley, Jean-Pierre (author),2-3:18,2-5:52,
3-1:28
Random House Thesaurus review,1-4:43
David Ray, CPA (ad),1-2:9.3,2-2:5.3
George Reardon (ad),1-5:21,2-1:34.3
RED review,3-1:18
Reprint permission,1-3:17.3
Reprint service,1-3:3.3,1-5:38.2
Resequencing BASIC lines,1-5:3
Rizzo Data Systems Corp. (ad),1-3:2,1-4:48,
1-5:39,1-6:33,2-1:35,2-2:33,2-3:41,2-4:29,
2-5:43,3-1:39
Rogers, Richard T. (author),1-6:28
RSCOBOL Generator and Utilities,3-1:12
Russo, Bradford (author),1-2:11.2
Sanchez, Joe A. (author),2-4:48
SARJ Associates (ad),1-3:25
Screenpix review,2-2:10.2
SCRIPFIX Review,2-3:12
Scripsit comments,2-1:30,2-2:12,2-3:36
Scripsit crashed diskettes,2-4:24
Scripsit crashes,2-5:23.2
Scripsit database,2-5:26
Scripsit query,2-5:35
Scripsit User Defined Seq Keys,1-6:28
Seashore Software (ad),2-3:45.3
See, Inc. (ad),1-5:19.3,1-6:37.3,2-1:15.3,
2-2:9.2,2-3:45,2-4:23.3,3-1:41
SemiDisk disk emulator review,1-3:24
Serial printer use,2-4:48.2
Shuey, Luis S. (author),1-3:36
Single keystrokes,3-1:38
<SIS> Computer Software (ad),2-3:35.3,2-4:17.3,
2-5:15.3,2-5:19,2-5:51.3,3-1:37.3,3-1:44,
3-1:45.3
Skyline Software (ad),1-3:16,1-4:15.3,1-4:37.3,
1-5:35.3,1-5:45.3,1-6:41,2-1:17,2-2:15,
2-3:15,2-4:25
Skylink Software (ad),2-5:27,3-1:29
SL-Micro review,2-4:32
Snapp, Bob (letter re PRO-80),1-4:22
SNAPP, TRSDOS patches,1-2:26
SNAPP-II review,1-2:6
SNAPP-III review,1-3:42.2
SNAPP-IV review,1-4:13
SNAPP-VI review,1-5:30
Snapp's MBASIC for 68000,2-3:1.2
Snappware (ad),1-4:34,1-5:22,1-6:22,2-1:22,
2-2:22,2-3:22,2-4:31
Snappware's SPLITHD Review,2-3:14
Softerware, Inc. (ad),1-4:C2,1-5:C2,1-6:C2
Software directories,1-3:1.2,1-3:14,2-3:2.3
Software distributors,2-1:3.3
SoftWare One (ad),2-4:43.3,2-5:45.3,3-1:41.2
Softworks Limited (ad),1-3:39.3,1-4:47.3,
1-5:5.3
Solutions, Inc. (ad),1-4:7.2,1-6:39.2,2-1:31.2,
2-2:41.2,2-3:19.2,2-4:45.2
Sorting,1-2:4.3
Sorting in BASIC,1-4:22,2-2:11.3
Southern Center for R & I (ad),2-1:9.3,2-2:43,
2-3:47.3,2-4:5.3
Southfork Software (ad),1-1:7.3,1-2:21.3,
1-3:3.2,1-4:33.3,1-5:37.3,1-6:39.3,2-2:13.3,
2-3:49.3,2-4:17.2,2-5:39.3,3-1:21.2
SPELLBINDER review,2-2:45
STARSHIP I (ad),2-5:C3
STARSHIP I game review,2-1:45
Statistical package review,1-5:48,2-1:46,
2-2:21
Statistical programs directory,2-4:2
Statistician review,3-1:24
Steinbach, Robert B. (author),1-6:16.3
Stepping Rate, Read or Set (article),2-3:18
Stockler, Robert C. (author),1-6:19.2
Strating, Wm., CPA (author),1-5:34
Structural engineering progrs.,2-2:36.2
SUPER as a DBMS,2-5:40
Superior Software Systems (ad),2-1:21,2-2:47,
2-3:39,2-4:21,2-5:11,3-1:19
SuperSort,1-2:4.3
SVC calls,1-2:3.2
SVC calls (article),1-4:24,1-4:29,1-4:36,
2-1:12
Switch boxes, printer,1-3:7
T/Maker III addendum,2-5:29.3
T/Maker III review,2-4:10
Tandy 2000 spreadsheet programs,2-5:2.2
Tandy customer support reorganization,2-4:3
Tandy Model 2000, review,2-4:2
Tandy products, new,2-3:2.3
Tandy reorganization,2-2:2.2
Tape to disk conversion,1-4:2.3
Taxbyte, Inc. (ad),2-2:37.3,2-3:29
TCBUG,2-1:20,2-4:9.2
TCBUG conference,2-4:36.3,2-5:3.2
Telexpress, Inc. (ad),3-1:33.2
Thesaurus, Random House review,1-4:43
Thornton, John (author),1-2:10,1-4:45
Trahan, Richard (author),2-4:10,2-5:48,3-1:42
TRICOR (ad),1-2:31
TriSoft (ad),1-5:43,1-6:45,2-1:37,2-2:35,
2-3:5,2-4:41,2-5:31,3-1:9
TRSDOS 1.2,1-2:2.3
TRSDOS-16 assembler,2-1:42
TRSDOS 4.2 patches,2-5:52
TRSDOS - CP/M data transfer,1-6:42,2-4:46
TRSDOS patches,1-1:6
TRSDOS, printing with Epson,1-6:36
TRSDOS to XENIX MBASIC data conversion,2-3:16
TRSDOS vs. other systems,1-2:5,1-4:1,1-6:20.2,
1-6:34
Turkel, Gary (author),1-4:29
II to 16 upgrade,1-2:5
TYPE utility for TRSDOS,2-1:28
Typesetting,1-5:33,2-1:24
Typing practice, program for,2-2:30
UNIQUE (UNIX newsletter),1-6:2.2
United Computer Corp. (ad),1-6:1,2-1:1,2-2:2,
2-3:13,2-4:37
UNIX: BASIC to C converters,2-1:3.3,2-2:2.3
UNIX information sources,2-2:2.2,2-4:2.2
UNIX-like features in CP/M,1-3:30
Users' groups,1-5:2,1-6:3.2,1-6:3.3,2-1:3.2,
2-1:20
USRn, BASIC,1-5:24
Utility programs (article),1-5:12
VanNatta, Robert P. (author),1-4:3,1-5:6,
1-6:5,2-1:4,2-1:47.2,2-3:3.2,2-4:4,2-4:42,
3-1:18
VARPTR, BASIC,1-5:24
VEDIT review,3-1:18
VERSALEDGER II review,1-5:34
VERSALEDGER II review reply,1-6:8
Video controller chip, program,1-2:18
Video tube, replacement,1-4:2.2,2-1:39
VisiCalc printing,2-1:10
VisiCalc tips,1-6:19.2
Viz.A.Con review,2-2:28
Warner, H. B., P. E. (author),2-2:38.2
Waters, James F. (author),2-1:24
Weatherhead, Keith F. (author),1-5:42
Weir, Jim (author),2-1:39
Wilde, Charles A. (author),2-2:24
Word processor (SPELLBINDER),2-2:45
WordStar, patches for Epson,1-1:4
WordStar tips,1-6:16.3
XENIX accounting system,2-5:36
XENIX books,2-5:2.2
XENIX, Discussion of,2-3:24
XENIX features,2-5:6,3-1:4
XENIX graphics,3-1:28
XENIX Impressions (article),2-1:16
XENIX MBASIC, converting data from TRSDOS to,2-3:16
XENIX programs available,2-2:2.2,2-4:2.2
XENIX testing,2-5:2.2
XENIX, TRSDOS conversion to,2-1:16
XKILL program,1-5:42
Young, R. H., background,1-3:18.2
Young, Richard (author),1-1:ALL,1-2:6,1-2:C2.3,
1-3:30,1-3:C2.2,1-4:1.2,1-5:1.2,1-6:2.2,
2-1:2.2,2-2:2.2,2-3:1,2-4:2,2-5:2,3-1:2
Zak Software & Distributing, Inc. (ad),3-1:37
ZIP (code generator) review,2-2:36

ADVANCED COMPUTING™ Service

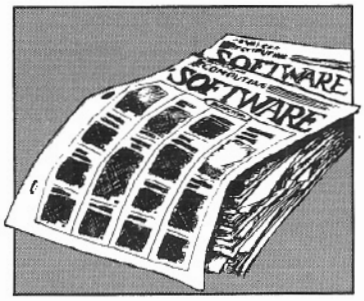


1 Three-ring binder with internal tabs containing all issues of **two/sixteen** magazine from inception through the end of 1983, including a cumulative index, and all 1984 issues of **Advanced Computing** (successor to **two/sixteen**), all in specially printed loose-leaf versions.

2 Newsletter to be published 26 times per year. It will contain product announcements, articles and correspondence outside **Advanced Computing** magazine's scope (because of time-value, large size, or specialized content), and the latest news of interest to serious users (mostly gleaned from dozens of other computer publications, and including abstracts of the more important articles).

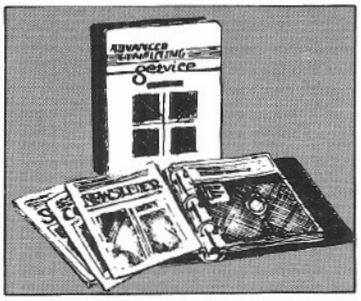


3 Diskettes with documentation (five or six per year), provided in loose-leaf sleeves. Includes all programs published in **Advanced Computing** magazine, all programs and patches that we receive and check out which are available for public distribution (along with our comments and documentation, and source code when available). At least one diskette per year will be devoted to each of the major operating systems (TRSDOS, CP/M, XENIX) used with Radio Shack II/12/16 computers.

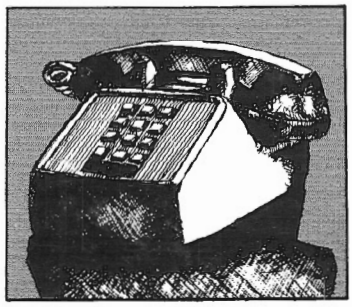


4 Comprehensive software directory of all commercially available programs known to function correctly on II/12/16 computers under one or more of the three major operating systems. Compiling this directory is a major undertaking, but we plan to issue the first one by April, 1984, with updates at least annually.

5 Master cumulative index of all of the above. **Advanced Computing** magazine will include its own cumulative index in every third issue, beginning with Nov-Dec, 1983 (Vol. 2, No. 3). The Advanced Service index will include, in addition to the **Advanced Computing** magazine index, the newsletter, diskettes and documentation, and the software directory, all in a single comprehensive index.



6 Binder for newsletters, diskettes, comprehensive software directory, and master index.



7 Advanced telephone consulting service including follow-up call back on subsequent developments.

☐ Please enroll me for twelve months
Regular Service Fee **$249**

☐ Enclosed is my check for $ _____ or

☐ Please charge my ☐VISA ☐Master Card for $ _____

Card number _____ Expiring _____

Signature _____

Name _____

Address _____

Address _____

City _____

Country (if not U.S.) _____

Please make any address corrections above

ADVANCED COMPUTING SERVICE • P.O. BOX 1216 • LANCASTER • PA • 17603