TRSTIMES

Keeping Models 3 & 4 Alive

Volume 1. No. 2. - March 1988 - $3.00

LITTLE ORPHAN EIGHTY

Let's clear up an error that occured in the last issue. In the article "PEEKING & POKING Model 4", it was stated in good faith that Roy Soltoff was the author of both TRSDOS 6.2. and LS-dos 6.3. This, we have since learned, is not correct. Mr. Soltoff wrote neither of these DOS'es. Our apologies goes out to Mr. Soltoff and to the readers for this erroneous information.

Now since Roy Soltoff did not write the DOS'es, then who did?

Our own Roy (Roy Beck) did some detective work and came up with the following:

First, let us introduce the various DOS'es involved in this discussion:

For the Model 4 and 4P:
LSDOS (TRSDOS) 6.0, LSDOS (TRSDOS) 6.1
LSDOS (TRSDOS) 6.2, LSDOS (TRSDOS) 6.3
For the Model III (and 4 in the III mode):
LDOS 5.0, LDOS 5.1, LDOS 5.3
For the Model I:
LDOS 5.0, LDOS 5.1

Next we present the cast of characters, so far as is known to this writer:
Roy Soltoff of MISOSYS
William Schroeder of LSI
Les Mikesell

Now, who wrote which DOS?

LSDOS (TRSDOS) 6.0. was a team effort, with Roy Soltoff leading. Soltoff at that time owned a 1/3 interest in Logical Systems, Inc. (LSI). Other team members included William Schroeder, another 1/3 owner of LSI.

LSDOS (TRSDOS) 6.1.,6.2., and now 6.3. were all by William Schroeder and other LSI people.

LDOS 5.0. was by Les Mikesell, who was an LSI employee at that time. I do not know who did LDOS 5.1. for LSI.

LDOS 5.3. was by Roy Soltoff, but done after he left LSI and after his company (MISOSYS) bought the rights to LDOS 5.1. from LSI. (Note: There is not and will not be a V.5.3. for the Model I. Too small a market to support the effort.)

It is believed that William Schroeder now owns all of LSI.

Both LDOS 5.1. and LSDOS (TRSDOS) 6.2. had the same problem of the date system expiring 12-31-87.

Both Schroeder of LSI and Soltoff of MISOSYS needed to update their respective DOS'es for both the date feature and for other reasons.

Schroeder of LSI updated LSDOS (TRSDOS) 6.2. to LSDOS 6.3., incorporating the much discussed, but little understood, copy-protection feature along with dating into the future and other revisions and corrections.

Soltoff of MISOSYS updated LDOS 5.1. to 5.3. (there is no V.5.2.), incorporating dating to 12-31-99 and adding other corrections and features.

Soltoff did not work on LSDOS 6.3., and Schroeder did not work on LDOS 5.3.

Now, is everybody clear on this?

GOOD NEWS

When 80 Micro deserted us and disaster seemed enevitable, TRSTimes was not the only one to try to pick up the pieces. We have learned that there are at least two other sources offering specific TRS-80 information.

The first is a venture by a group of very dedicated TRS-80 Sysops, headed by Dr. Luis Garcia-Barrio of the 8/N/1 #4 BBS in Philadelphia. It is called TRSLINK and is a monthly electronic magazine packed with interesting and useful items. It features newly developed public domain programs, 'Shareware' programs, nationwide ads, hints and tips, articles and other good 'stuff'.

We have, courtesy of Luis Garcia-Barrio, seen the three issues to date and we heartily recommend that you join up and get yours. The BEST NEWS of this 'GOOD NEWS' section is the unbelievable price of TRSLINK: **IT IS FREE.**

There is absolutely no charge for this dandy collection of TRS-80 information. All you have to do is call your local TRS-80 BBS and download the mag.

Should your BBS not carry TRSLINK, leave a note to the Sysop and tell him to get on the ball. Then call the home of TRSLINK, 8/N/1 #4 and download it. Still no charge. Of course, the phone company will probably charge you for a long distance call unless you call from Philadelphia.

The number is: 215 848-5728

The other is a 'regular' magazine, a monthly newsletter called 'Computer News 80' carrying no advertisements. It will have user or application rather than programmer orientation, and its primary focus will be on the Model 4. The first issue should be available in January. Subscription for one year (12 issues) is $18.00. A sample copy is available for $2.00. Here is the address:

Computer News 80
P.O. Box 680, Casper, WY 82602, 307 265-6483.

We wish both Computer News 80 and TRSLINK the very best of success. Check them out.

And now, on with the show:
Welcome to.........TRSTimes #2

TRSTimes - Volume 1. No. 2. - March 1988

CONTENTS:

TRSTimes is published bi-monthly by TRSTimes publications.
20311 Sherman Way #221. Canoga Park, Ca. 91306.

1988 subscription rates (6 issues):
United States and Canada: $15.00 (U.S.)
All other countries: $20.00 (U.S.)

THE MAIL ROOM

Glad to hear someone's interested in keeping the 3's & 4's going. Hope you will have room for personal classified ads - am personally looking for info on some 3rd party (orphan) hardware and possibly, useful software.

H. May
Hinsdale, IL.

TRSTimes will carry vendor advertising and classified ads on a limited basis as we feel that the readers will want to know what is available for our machines.

Ed.

I would be interested in seeing articles that would help me in converting my Model I assembly language software to function in the Model III mode of the Model 4, and then in the native Model 4 mode. I could also use help in converting PEEK and POKE statements in Basic programs. These are probably fundamental problems to most good TRS-80 users, but I have not progressed that far in programming.

William C Choate
Boring, OR

We are planning to publish memory maps of Model I and III. Also in the works is a large list of PEEKS & POKES for Model III. Model 4 PEEKS & POKES are currently being covered in the series 'Hunting for Buried Treasure'.

Ed.

Just received the first issue of TRSTimes. GREAT! FANTASTIC! Keep up the good work.

Of all the fine articles, "Catting Around With LDOS 5.1.4." was in my opinion the best of the first issue, and, if indicative of the kind of utilities/applications you will publish, success is assured.

Congratulations on a fine piece of work. I will be looking forward to your next issue.

Dick Hollenbeck
El Paso, TX

Check out Scott McBurney's article "Teaching an Old Dos New Tricks". It is of the same mold as Catting Around.

Thank you for the kind words. You just made all the sleepless nights worth it.

Ed.

Here is a program that will address an envelope with your return address and then give you the option of setting a five line outgoing address. The program is written for the Radio Shack DWP 230 and a standard #10 size business envelope. You will have to change to data in lines 210-260 to reflect your name and address. The printer codes in line 200 and 220 may also need to be changed, or eliminated entirely, if you use another printer.

```
10 CLEAR 500:DIM A$(5):CLS
20 FOR X=1 TO 5
30 PRINT"LINE";X;": ";:LINE INPUT A$(X)
40 NEXT
50 CLS
60 PRINT"THE ADDRESS WILL LOOK LIKE THIS:"
70 PRINT:PRINT
80 FOR X=1 TO 5
90 PRINT TAB(10);X;" ";A$(X)
100 NEXT
110 PRINT:PRINT
120 PRINT"PRESS # OF FIELD TO CHANGE. ";
130 PRINT"PRESS <ENTER> TO PRINT";
140 I$=INKEY$:IF I$=CHR$(13) THEN 190
150 I=VAL(I$):IF I<1 OR I>5 THEN 140
160 PRINT:PRINT
170 PRINT"NEW LINE: ":LINE INPUT A$(I)
180 GOTO 50
190 LPRINT" "
200 LPRINT CHR$(27);CHR$(31);
210 LPRINT TAB(4);"JOHN J. DOE";
220 LPRINT CHR$(27);CHR$(32);
230 LPRINT TAB(4);"P.O. BOX 777"
240 LPRINT TAB(4);"LOS ANGELES, CA. 90001"
250 LPRINT TAB(4);"OTHER INFO HERE"
260 LPRINT TAB(4);"OTHER INFO HERE"
270 FOR X=1 TO 7:LPRINT" ":NEXT
280 FOR X=1 TO 5
290 LPRINT TAB(43);A$(X)
300 NEXT
310 FOR X=1 TO 8:LPRINT" ":NEXT
320 CLS
330 PRINT"<A>gain, <N>ew, <Q>uit ";
340 I$=INKEY$:IF I$="A" OR I$="a" THEN 50
350 IF I$="N" OR I$="n" THEN RUN
360 IF I$="Q" OR I$="q" THEN CLS:END
370 GOTO 340
```

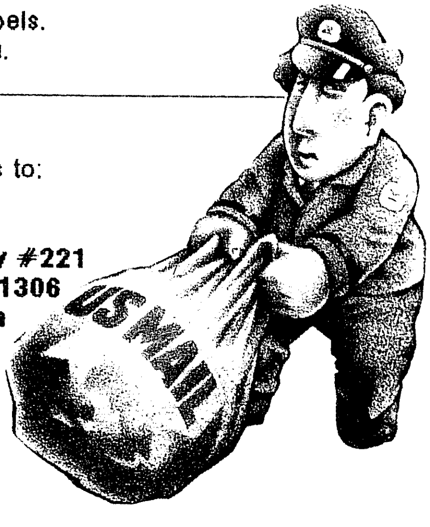
Kenneth Dark
Los Angeles, CA

Sure beats mailing labels.

Ed.

Send your comments,
ideas and suggestions to:

TRSTimes
20311 Sherman Way #221
Canoga Park, Ca. 91306
Attn: The Mail Room



Model 4 - TRSDOS 6.2 & LS-DOS 6.3

Teaching an old DOS new tricks

by Scott McBurney

Frequently, out of habit from the Model I and III days, I type 'KILL filename' to get rid of a disk file and then, when the error occurs, remember that the correct command is 'REMOVE filename'.

TRSDOS 6.x and LS-DOS 6.3 both eliminated 'KILL' from their list of DOS commands. Many people soon discovered that TRSDOS 6.x really did have the KILL command, but that it was disabled and could be brought back to life with one small patch. Unfortunately the command does not exist at all in LS-DOS 6.3.

Maybe you use an MS-DOS machine or even a large mini or mainframe and quite often try to use their commands on your TRS-80 by accident. I do this too. For these reasons, I wrote a small machine language program to help eliminate the confusion.

Program listing 1 is the original program. If you assemble it and call it KILL/CMD, then you can once again type KILL followed by a filename, and the computer will do your bidding.

Now, why stop with just KILL? Many TRS-80 users are also users of other computers, so why not put their commands onto your computer? Program listing 2 is an easy way to do just that. It is a short BASIC program which will ask for the new command name, the DOS command it should be converted to, and then proceed to create the command file for it.

So what commands might you want? Well, if you use an MS-DOS machine, you could now ERASE or DELETE (which is REMOVE on the TRS-80) or you can TYPE files instead of LIST. If you use UNIX, you could LS your directory or RM files. You can create a mix of commands from different machines or you can shorten the names of freqently used commands. For example, D instead of DIR, C instead of CAT, L instead of LIST, etc. The possibilities are virtually endless and, as each new command takes up 1.5K on your disk, the only limitations are your imagination and the available disk space.

Have fun teaching your Model 4 a new and exciting vocabulary.

LISTING 1

```
00100 ; Source for KILL/CMD
00110 ; (c) Copyright 1988
00120 ; by Scott McBurney
00130 ;
00140           ORG     3000H
00150 START     LD      DE,BUFF  ;get buff address
00160 LP1       LD      A,(HL) ;char from DOS buff
00170           LD      (DE),A ;char to new buff
00180           CP      13 ;CR=char move done
00190           JR      Z,EXEC ;exec new cmd
00200           INC     HL ;next buffer location
00210           INC     DE
00220           JR      LP1 ;go get next character
00230 ;
00240 ; Move the command line to our buffer
00250 EXEC      LD      A,24 ;CMDI svc
00260           LD      HL,CMD ;point to REMOVE
00270           RST     28H
00280 ;
00290 ; Pass new command to DOS
00300 CMD       DEFM    'REMOVE '
00310 BUFF      EQU     $
00320           END     START
```

LISTING 2.

```
   10 '(c) Copyright 1988 by Scott McBurney
   20 '
   100 CLS: PRINT "Custom Command Maker - By
Scott McBurney": PRINT
   105 PRINT "Enter the NEW command name: ";:
LINE INPUT NEWCMD$
   110 PRINT "Enter the TRSDOS/LS-DOS com-
mand name: ";: LINE INPUT DOSCMD$
   115 OPEN "O",1,NEWCMD$ + "/CMD"
   120 READ DAT$
   125 WHILE DAT$ < > "ZZ"
   130 IF DAT$="XX" THEN B=LEN(DOSCMD$)+1
   140 IF DAT$="YY" THEN B=LEN(DOSCMD$)+3
   145 IF B > -1 THEN 170
   150 B=ASC(LEFT$(DAT$,1))-48: IF B>9 THEN
B = B - 7
   155 C=ASC(RIGHT$(DAT$,1))-48: IF C>9
THEN C = C - 7
   160 B = B * 16 + C
   165 CHK = CHK + B
   170 PRINT #1, CHR$(B);
   175 READ DAT$
   180 WEND
   185 A$=DOSCMD$+" "+CHR$(2)+CHR$(2)+
CHR$(0) + CHR$(48)
   190 PRINT #1, A$;
   200 CLOSE #1
   200 IF CHK < > 1372 THEN PRINT "Data was
entered wrong - Please check!!"
   1000 DATA 01, 15, 00, 30, 11, XX, 31, 7E, 12,
FE, 0D, 28, 04, 23, 13, 18
   1010 DATA F6, 3E, 18, 21, 00,31, EF, 01, YY, 00,
31,ZZ
```

THE LOST POWERS OF SCRIPSIT

by Eric Bagai

In days of long ago there once did dwell Hidden Figures of strange and wondrous power in the Device of Scripsit. And in those days the people made great harvest and contrived in artful artifice many sentences containing iron brackets and leathern braces, and did so mark some words with underlying lines to make them sound more strongly on the ear. And all was good and words of power were written by all the people.

But they grew slothful in their weavings of the words, and lost the art thereof. And powers of Rapid Scroll and Continuous Insert grew weak until the people used them not. And so there came then upon the minds of the Scribes of Documentation a cloud of forgetfulness. And these powers too were lost.

Time out of mind, the people knew not the Lost Powers and used not the Hidden Figures. But the Powers and the Figures slept inside the Device of Scripsit, and waited for the time when men would need them once more. And they would then come forth.

Folk Legend

Scripsit is the one of the best selling word processors for any computer. It is almost transparent, nearly bug-free, and easy to use. It is most everyone's first "Shack" word processor. The one you recommend to novices; the one your neighbor has; the one you never forget how to use.

If your introduction to Scripsit was the Model 4 version, you got a neatly indexed and completely straightforward booklet, some stickers for your function keys, a summary card, and a disk; all nicely packaged. You also got the result of five years of complaints from users of the Scripsit 1.0, 3.1, and 3.2 documentation.

The early Scripsit documentation was a big step up from the typical author written documentation of the day: it was literate, and you never found yourself wondering how old the author was. On the other hand, it had no index, its summary card was designed for ants, and essential information was presented on audio-tape. Three hours of audio-tape. The dry, New England voice of "Mr. Scripsit" was a blessing to thousands of secretaries, and the bane of at least as many hackers. But still, it was essentially the same great program that it is today.

So now that we know why we use it, here are a few things Tandy never told you about Scripsit.

RAPID HORIZONTAL SCROLL

The cursor can be moved very rapidly and very precisely by using the Repeat Command: <@R>. This is good for getting to specific places on the screen without having to set or reset tabs. For example, if you wanted to get to column twenty on the screen you would press <@R> 20 <ENTER>, and <RIGHT-ARROW>, and the cursor would move exactly twenty spaces to the right. The six keystrokes used to get to column twenty hardly seem worth the effort when you can get there just as well by holding down the arrow key until the cursor gets there. However, this technique becomes particularly handy when you've set the screen width for more than 64 columns and you need to put the cursor exactly on column 102.

Rapid Horizontal Scroll also spans lines, just like regular horizontal scrolling. So, if you wanted to find the 240th character to the right of the cursor you would press <@R> 240 <ENTER>, and <RIGHT-ARROW>.

RAPID VERTICAL SCROLL

Scripsit scrolls up or down the screen at a leisurely seven lines per second. If you use <@R> <ENTER> <DOWN ARROW>, your cursor will move about six times faster. Instead of using the default of 255, you can specify how many lines to move. This is most needed when (for example) you want to find quickly the 113th line of a document: Put the cursor at the top of the document, press <@R> 113 <ENTER>, and <DOWN-ARROW>. Your vision will become blurry and your heart will race (with the cursor) down the screen to line number 113, which just happens to be the start of the third default page.

Want to be sure you are on screen line number one-hundred- thirteen? Just use the Cursor Position command: <BREAK>?C. This command and Rapid Vertical Scroll can get you back and forth in a document very nicely. You can also use the Cursor Position command to tell you where you are after you've made a Find command.

In using the Rapid Vertical Scroll or the Cursor Position Comand to find page-breaks, be careful to account for command lines, double spacing, and extra lines forced between paragraphs.

CONTINUOUS INSERTION

If you've ever used a word processor that toggles the INSERT Command, you have reason to make bad noises when Scripsit requires an insertion command <@S> for each letter or space insertion.

Now you can insert a word, phrase, sentence, or short story in the middle of an already typed line, and do it quickly and easily. This technique involves combining the Repeat Command and the Insert Command. Just move the cursor to the point where you wish to begin the insertion, press <@R> <ENTER> <@S> and begin typing whatever you wish to insert. Everything in front of the cursor will be gently pushed ahead as you type. You can type up to 255 characters before the Insert Command expires. You can also interrupt the continuous insert mode at any time by pressing any of the arrow keys or the <ENTER> or <CLEAR> keys.

This technique does require two more keystrokes than the one mentioned in the Scripsit documentation: <@S> <@X> to insert a blank line, and <CLEAR> to close up afterwards. But somehow it's so much more, well, elegant this way.

CONTINUOUS DELETION

This is a dangerous command because there is no stopping it once it starts. Press <@R> <ENTER> <@D> and <ENTER>. Your words will then crawl back into the cursor, madly wordwrapping themselves into oblivion. This can be disconcerting, especially if you are inclined to panic. Do this too near the end of a file, or in an empty new file, and you may find the place where old hackers go.

ABSOLUTE WIPE-OUT

Like Continuous Deletion, this is one control sequence that you probably do not want to use, and that will get you into trouble if you run across it by accident. It is <@T>. This command will sometimes dump you into DOS, sometimes put you into DEBUG, and sometimes just freeze your keyboard and play games with your screen. Some copies of Scripsit will just give an error message. This seems to be a leftover author's utility; a quick hook into the Scripsit code itself. While you couldn't really call this a bug, neither is it a desirable feature (although assembly programmers may find it useful.) Like other darker powers, this one should be recognized for what it is, and avoided. No other rituals are required.

GLOBAL COMMANDS

There are a number of commands and procedures in the Global Command function that are poorly documented or not mentioned at all in versions 1.x and 3.x. These include:

1. The use of the Global Find Wildcard, which is the question mark. Useful for finding all instances of everything. For example: <@R> <ENTER> <BREAK>F>?m?th<ENTER>, will find every smith, Smith, smyth, and Smyth in your document -- and also every mother, myth, and Samothracian. Other uses might be to find all variations of a control code, a range of dates, or all zip-codes within a Sectional Center Facility.
2. The use of characters peculiar to Global Command procedures, such as the Copy Marker character (^), which you can produce on the command line by pressing <UP-ARROW>; and the first character in Block Markers ([), which you can produce on the command line by pressing <@Q>; and the last character in Block Markers (]), which you can produce on the command line by pressing <@><DOWN-ARROW>.
3. The fact that all boundary markers (line, paragraph, copy, and page) can be Globally replaced, deleted, found, or counted. In case you've forgotten, the ubiquitous line boundary marker (the square thing you get when you press ENTER) can be produced by entering <@X> . . . even on the command line.
4. The fact that all Global Commands can be combined with the Repeat Command to make multiple changes throughout the document at the same time. This is particularly useful for preparing raw ASCII files for transfer between word processors or for modem transmission.

(And for you Model 4 users, too . . . .)
5. The fact that you CAN get a word count.
Begin by Saving the current file. Then:

```
<@R> <ENTER>
<BREAK> <R> (two spaces) >XX <ENTER>
```

Repeat this procedure until you reach the end of the file. Then return to the beginning and:

```
<@R> <ENTER>
<BREAK> F> (one space) <ENTER>
```

Repeat until you reach the end of the document, and keep a tally of how many were found. Then reload your original file. The first series of commands replaces all double blank spaces with strings of Xs. The second series of commands counts the single blank spaces that are left: the ones between words.

BUILDING "MACROS"

Many word processors are very proud of their ability to make MACROs. A MACRO is a function that makes a large group of predetermined characters appear, or an entire command sequence begin, just by entering one or two keystrokes.

Scripsit has always been able to do this.

Scripsit's Macros are, of course, the Block Commands. You can use any character on the keyboard as a MACRO key, except the P, H, F, -, or any lower case letter of the alphabet. Naturally, the arrow, <ENTER>, and <CLEAR> keys are also out. That leaves about 58 characters including numbers and symbols, each of which can be used as a block marker -- and each block marker can contain up to the amount of text that can be put in memory. Keeping a library of 58 Macro commands straight is your problem.

Many writers find it useful to put "boilerplate" (frequently used blocks of text) inside block markers, and save that block, or an indexed group of such blocks, to a special file. Then whenever they make a document needing that "boilerplate" text, they chain the special file to the end of their document and use the Insert Block command to put a particular MACRO wherever they want. (Just remember that whenever you use the Load-Chain Command, the default filename for the file now in memory is changed too.)

Another good use of Scripsit Macros overcomes the complexities of Headers, Footers, and special Print Formats. How many times have you had to dig out the Scripsit manual to figure out how to put centered page numbers at the bottom of each odd-numbered page? What about the format for skipping headers and perforations on continuous letterhead? or keeping the ink off your platen when using a sheet feeder? Why not put each of these headaches in its own special block marker, and then put all the Print Formats in one file, all the Header Formats in another, and the Footers in yet another. Whenever you need one, just chain it onto the current document, use the Insert Block Command to put the Macro of your choice where you want it, and delete the dross. If you need these more than once, you need a set of MACRO files.

An extension of this idea is to develop your own HELP file of useful but little-used Scripsit procedures. After all, your Scripsit system disk is usually a lot closer (and more helpful) than your Scripsit manual.

EDITING BASIC WITH SCRIPSIT

Now you can enjoy the use of Scripsit as a full-screen BASIC language editor. Just follow these simple guidelines:

Keep your program lines less than 240 characters long. (Your Rapid Horizontal Scroll works well here.) It seems that more than 240 characters per line in an ASCII saved BASIC program may rouse the dreaded "Direct Statement in File." Before saving your BASIC program, be sure to remove all Scripsit text markers, and make sure there is a line marker at the end of each program line. Then add at least three extra lines to the end of your program. They should have legitimate line numbers, but contain only filler data. This will pad your file so that the last sector of REAL data will not be lost. Save it with the ,A option, then load and re-save it in BASIC.

REVEALING THE HIDDEN CHARACTERS
(Model 4 users excused)

You can create 10 characters that you probably didn't know Scripsit could produce. These are the right and left braces and brackets, the backslash, the vertical bar, the tilde, the underline, the circumflex, and the plus/minus: {}[]\¦ ^ _ ~‡. These are produced by using the capital or lower case character Y as a control key and pressing two numbers simultaneously. For example:

Y13 = [Y14 = \ Y15 =] Y16 = ^ Y17 = _

y13 = { y14 = ¦ y15 = } y16 = ~ y17 = ‡

When you enter any of these sequences, you also get some garbage along with your new character (an extra "Y" and some numbers). To use your new characters effectively you will have to remember each command (or find it each time by trial and error), and then manually delete the surrounding garbage. . . Or you can do it the easy way and use the Special Character Macro system described below. But first:

Because new functions can be dangerous if not used carefully, it is necessary that you stop here for a machine-language mini-tutorial. When you use any Scripsit command that leaves a text or boundary marker on the screen, the mark that you see there is not really an ASCII character, but is a specially coded symbol that will not print and that performs both screen and printer formatting commands. Your new special characters are not like this at all: they are simply ASCII characters, the same as any letter or number. That is, the End of Block character is not the same as your new right bracket character and they are NOT interchangeable. The same is true of the new backslash, which looks like (but is not the same as) the page boundary marker gotten by pressing <@V>; and of the circumflex, which looks like (but is not the same as) the copy marker gotten by pressing <BREAK> C <ENTER>. This is also true of hyphens produced by Scripsit's automatic hyphenation procedure. The point of this mini-tutorial is that if you don't watch your language, the machine will get you.

USING THE SPECIAL CHARACTERS

This is useful for ALL versions of Scripsit, because three logically mnemonic commands are easier to remember than ten randomly assigned ones.

Starting with an empty file, create the new characters in the following sequence and delete all of the surrounding garbage so that the result looks like this:

```
[{    }]      \¦^_~‡
```

Then put the new characters in block markers, as follows:

```
[B>[{ ]
[E>}] ]
[O>\¦^_~‡ ]
```

Next, save this small file of new characters to your system disk, perhaps using the filename "FIGURE." Now, whenever you want to put a phrase in brackets or braces you can easily do so. Just chain the file FIGURE onto the document you are working on (<BREAK>L,C FIGURE), and then use the INSERT BLOCK command to put the beginning [B> and ending [E> brackets or braces right where you want them. You can use a block containing all the other characters more easily than you can remember which control code makes a tilde.

What are all these goodies good for? Probably not your shopping list. Often, their use depends more on your writing style than on any real need for them. After all, your notes to the milkman don't require diacritical markings. . . unless he forgot the butter AGAIN. If your parenthetical asides need bracing (and every issue [big {especially!} or small] has them), then you're all set. If you write like Vonnegut they make nice paragraph separators.

A FULL-SCREEN EDITOR FOR C, PASCAL, LISP, . . .

Now that you have brackets and braces available in Scripsit, you also have a full-screen editor for several high-level languages besides BASIC. (And while you're at it, make another block marker which contains five blank spaces to use for uniform indentation of program loops, recursions, and comments.)

THE UNDERLINE

The underline character is useful in making printed forms; and, with some effort, it can actually be used to underline things. In either case you will need to add a string of underline characters to your FIGURE file. The easiest way to do this is to get in the CAPs mode, hold down the letter Y, and repeatedly and simultaneously hit the numbers 1 and 7, say about seventy times. Then go back to the start of the line and press <@R> <ENTER>, and then <BREAK> D>7 <ENTER>. All the numeral 7's should have disappeared. Delete the 1's and the Y and all you have left is a continuous string of underline characters. (Model 4 users should not sneer.) You can put a Block Marker around this line and add it to your FIGURE file, so that it looks like this:

```
[B>[{ ]
[E>}] ]
[O>\¦^_~‡]
[L>______________________________________]
```

Now, whenever you wish to put a string of underline characters in a printed form, just chain FIGURE to the end of the document you are using and insert the block of underline characters wherever you want.

You can underline words in much the same way. But first you must set your printer to produce a carriage return without a linefeed. You will have to change the DIP switches inside some machines. Some Radio Shack printers come set this way, and those of you who have them use the LPC/CMD to overcome this "feature." If you are one of these pioneers, just this once don't LPC, see?

Now that your printer is set, begin the document with a format line that includes a command to give you SINGLE-spaced text: LS=2. Then, put boundary markers at the beginning and end of the screen line that contains the word(s) you want to underline. Immediately before this line, insert a format line that includes a command to suppress linefeeds: LS=1. Immediately after the line to be underlined, put in another format line which returns the linefeeds: LS=2. Then insert your block of underline characters, with another boundary marker at the end, and erase the portions you don't need. On the screen it should look something like this:

LS=1
Every other word in this sentence is underlined.
LS=2
_____ __ _______ _________

You may have to experiment to find out what works for you. But hey, that's hackin'!

ROMANCING THE LANGUAGE

The tilde(~) and circumflex(^), along with the stress mark (') and the cedilla (,) are useful if you are writing in any of the Romance languages, and can be produced the same way you underline words: by suppressing a linefeed and over printing a line containing only the carefully placed accent mark. Model 4 users also get to have an grave accent (`), which now becomes the only advantage of Model 4 Scripsit over Models 1 and 3.

CONVERTING THE LANGUAGE

When converting Scripsit files to work with other word processors, all of Scripsit's non-ASCII characters (its boundary and copy markers) must be removed or replaced. Many word processing programs use ASCII backslashes, bars, or circumflexes where Scripsit uses its tokenized line, paragraph, and page markers. You can replace all of the Scripsit boundary markers with whatever the other program requires by using the Repeat and Global Change commands. Unless you have a Model 4 version, you will then have to examine each replacement and remove extraneous Y's and numbers. Of course, almost all word processors provide a conversion program specifically for Scripsit files, but some things just need to be said. Besides, some of us have unusual needs.

INVISIBLE CHARACTERS (‡)

This strange little fellow will show up on your screen, but probably cannot be printed. It won't even appear on the paper as a space. If you find a use for it, let me know. If you want to actually print the plus/minus character on paper, you can do it by simply underlining a plus sign (or plusing an underline character.) You can also use this procedure to make new characters and impress small children.

PRINTER DIFFERENCES

If you have a daisy wheel or thimble printer you can use the new Scripsit characters to print the copyright, trademark, registration, dagger, swirl, degree, and paragraph marks, depending on your wheel or thimble. If your dot matrix printer has different fonts, it may find new forms for your new characters.

EXPANDING THE BUFFER

When you type a document in Scripsit you fill up a buffer. When you reach the end of the buffer, that's it. No More Room. Even if you have only a few more words to go, Scripsit won't let you write them.

But wait! You CAN expand the buffer, or at least expand the room inside the buffer. First clear the blanks at the end of the file: <@D> <@F> <CLEAR>. Then reset the screen to the maximum width possible: <BREAK> W=132. Now you just may have a little more room. Scripsit uses one extra character of memory for each screen line, so the fewer screen lines, the more room for your text. This sometimes works for the Disk Full error message, too. However, remember that Scripsit files are most fragile, most easily lost, when they are very long or when your disk is crammed full. Even on a tape-based system, long files ask more of the system than short ones. So, even if Tandy documentation doesn't say so: Short is good.

FINDING NEW POWERS AND FIGURES

The best source of new powers for Scripsit is found in Powerscript, available at (214) 733-4475 from Powersoft for about $40. With Powerscript you can give direct printer commands, chain and embed print queues, and load, chain, or kill files from a sorted directory without leaving Scripsit. It also has on-line help. Powerscript, as much as anything, is responsible for keeping TRS-DOS alive and well in my home.

GLOBAL COMMANDS

Move cursor to the point where the string search is to begin:

REPLACE:<BREAK> R>(old string) >(new string)<ENTER>
DELETE:<BREAK> D>(string)<ENTER>
FIND:<BREAK> F>(string)<ENTER>
NEXT FIND:<BREAK> F<ENTER>

Can be combined with REPEAT commands. REPEAT with FIND gives the total number found. The question mark is a wild-card search mask in FIND.

HEADERS and FOOTERS

Open by typing <@Q>, then type H for Header or F for Footer. Continue with O for Odd, E for Even, or S for all pages. Then type the format instructions and close the line with <ENTER>. Type the actual text lines. The last line must be terminated with <ENTER>. Close the block with <@Q> <@Down-arrow>.

HYPHENATION

1. Set screen width equal to that set in the format line.
2. Position cursor at left edge of line, press <@Q> <->.
3. Move cursor to end of last line, press <@Q> <@Dn-arrow>.
4. Type <BREAK><-><ENTER>.
5. Type a number to define hyphenation range; press <ENTER>.
6. The cursor appears at the first hyphenation location.
7. Move cursor where desired; if no hyphen desired press <ENTER>, otherwise press <->. Repeat as required.
8. To remove hyphens: Type <BREAK><-><ENTER>. Answer message by typing 2. Press <CLEAR> when cursor moves.

INSERTING TEXT

Insert Character	<@S> (character)
Insert Several characters	<@S><@X>(type text) and then <CLEAR>
Insert a Block	<@S><@Q>(block name)
Inserting Text Boundary Markers	
Force End of Line	<ENTER> or <@X>
New Paragraph	<@C>
New Page	<@V>

LOADING TEXT

Load Document:	<BREAK> L(space)(document name)<ENTER>
Reload Document:	<BREAK> L<ENTER>
Load from Tape:	<BREAK> L,T<ENTER>
Load Tape/Slow:	<BREAK> L,S<ENTER>(M3 for M1 tapes)
Load and Chain:	<BREAK> L,C(space)(document name)<ENTER>
Load/Chain Tape:	<BREAK> L,T,C

MACROS

Chain files which contain **your** print formats, **your** embedded pagination in headers and/or in footers, **your** boilerplate, etc.; each within its own labeled block so you can <@S><@Q> it.

MARKERS

Text Boundary Markers

Force End of Line	<ENTER> or <@X>
New Paragraph	<@C>
New Page	<@V>

Copy Markers mark off text for selective printing. Insert a copy marker after a boundary marker by typing: <BREAK> C<ENTER>

PAGE NUMBERING

Usually within a Header or Footer. Open the page block by typing <@Q>, then type P for Page. Continue with up to five pound signs (#) to represent desired digits. Close the block with <@Q> <@Down-arrow>.

PARAGRAPHS

Press <@C> to start a new paragraph. To set the number of indent spaces at the beginning of a paragraph, type: <BREAK> I=(any # from 1 to 133) <ENTER>.

PRINTING TEXT

Basic Print Command	<BREAK> P<ENTER>
To Serial Printer	<BREAK> P,S<ENTER>
Pause Between Pages	<BREAK> P,P<ENTER>
Print Unformated Text	<BREAK> P,I<ENTER>

QUERY DOCUMENT INFORMATION

Video Line Width	<BREAK> ?W<ENTER>
Para.Indent Setting	<BREAK> ?I<ENTER>
Document Line Number	<BREAK> ?C<ENTER>
Document Length (in bytes)	<BREAK> ?L<ENTER>
Available Memory (in bytes)	<BREAK> ?M<ENTER>
Document Name	<BREAK> ?N<ENTER>

REPEAT COMMAND

Commands may be repeated up to 255 times automatically. Type <@R> and then any number from 2 to 255 and <ENTER>. Then type the command and it will be so repeated. Use with Global Commands to replace, find, or delete all instances of a string. Use with Arrow keys go relative to a location.

COPY THIS

There are more copies of Scripsit than Scripsit Manuals, Scripsit Audio Training Tapes, Scripsit Quick Reference Cards, or Scripsit Stick-On Keyboard Labels, Horatio. But truth will out, though the ravages of time make havoc and let slip the fogs of Tandy as a shroud upon a slunk king, yet they shall pass. For the truth is always beautiful: **Eric's Quick Guide To The Guts of Scripsit** is come of hacker sitz fleisch! The well-met word-whizzer perforce must needs meet, make do, and wing it with this small but comely offering the which resideth at your right hand, now and always.

SAVING TEXT

Save Document	<BREAK> S(space)(Filename) <ENTER>
Update Document	<BREAK> S <ENTER>
Save to Tape	<BREAK> S,T <ENTER>
Save in ASCII Format	<BREAK> S,A (space)(Filename) <ENTER>
Verify Tape Save	<BREAK> V <ENTER>

SPECIAL CHARACTERS: Model I/III

Hold down y and	Hold down Y and
press: 13 for {	press: 13 for [
14 for ¦	14 for \
15 for }	15 for]
16 for ~	16 for ^
17 for (unprintable)	17 for _

Press <SHIFT><0> for @

SPECIAL CHARACTERS: Model 4

The special characters are produced by pressing <CLEAR> or <SHIFT-CLEAR> and ,./ or ; to produce [ ]\^ and { }¦~. The underline character is produced by pressing <CLEAR><ENTER>. The Model I/III mode procedures can also be used.

TAB CONTROL

Clear All Tabs	<BREAK> T <ENTER>
Clear One Tab	<BREAK> TC <ENTER>
Set Several Tabs	<BREAK> T=(##,##,##,##,...) <ENTER>
Set One Tab	<BREAK> TS <ENTER>
Move to Tab	<@><RIGHT-ARROW>

UPPER CASE LOCK

In the Model I/III, the upper case or caps lock is toggled by pressing <SHIFT><@>. The Model 4 uses the <CAPS> key.

VIDEO LINE WIDTH

Set the video line width by typing <BREAK> W= followed by any number from 1 to 132 and <ENTER>. Note that this does not change the width of printed copy. See FORMAT.

WINDOW MOTION COMMANDS

Enter <@><W> to move the screen "window" without moving cursor.
Exit Window mode Model I/III = <CLEAR>.
Exit Window mode Model 4 = <SHIFT-CLEAR>.

The Control key is shown as @, even though M4 uses <CLEAR> for Control.

BLOCKS

To open a block, type <@Q>, followed by a one-letter name. The letters P, H, and F are reserved for Page number, Header, and Footer blocks, respectively. To close a block, type <@Q> <@Down-arrow>. See INSERT, DELETE, EXCHANGE.

COMMENT LINE

Create a non-printing line by typing >* at the left margin. End the line with <ENTER>. When printed, the text on this line will not be visible but the page will show a line-feed here.

DELETION

Character	<@D>
Word	<@D><@Z>
Sentence	<@D><@X>
Paragraph	<@D><@C>
Blanks	MI/III=<CLEAR> M4=<SHIFT-CLEAR>
Block	<@D><D>
Unmark block	<@D><U>
To end of text	<@D><@Down-arrow><Y>

ESCAPE

From any command in Model I/III = <CLEAR>; Model 4 = <SHIFT-CLEAR>.
From Scripsit <BREAK> END <ENTER>. From boredom = <@R><ENTER><@D>.

EXCHANGING TEXT

WORDS:	Put cursor on second word and type: <@E><@Z>
PARAGRAPHS:	Put cursor on second paragraph and type: <@E><@C>
BLOCKS:	Put cursor on block marker and type: <@E><@Q>

FORMAT LINES begin with > at left margin and end with <ENTER>

Instruction	Default	Range	Instruction	Default	Range
Page Length	PL=66	1-90	Flush Right	FR=N	Y or N
Left margin	LM=12	0-131	Vertical Ctr.	VC=N	Y or N
Right Margin	RM=72	1-132	Starting Pg #	PN=1	1-65535
Top Margin	TM=6	1-89	Widow Suppress	WS=Y	Y or N
Bottom Margin	BM=60	2-90	Header On	H=Y	Y or N
Line Space	LS=1	1-90	starting page	H=1	1-65535
Par. Space	PF=1	1-90	beg. Odd/Even	H=O	O or E
Justify	J=Y	Y or N	Footer On	F=Y	Y or N
Center	C=N	Y or N	starting page	F=1	1-65535
Print Text	P=Y	Y or N	beg. Odd/Even	F=O	E or O

For another free copy of
Eric's Quick Guide To The Guts Of Scripsit
send a stamped self-addressed envelope to
Box 9747, North Hollywood CA 91609.

HUNTING FOR BURIED TREASURE

More Peeking and Poking Model 4

by Lance Wolstrup

In our last installment we manipulated the FLAG$ table, changed the cursor character, toggled the visibility of the cursor, toggled between space compression and the special character set, learned how to scroll protect the top portion of the screen and more. All of this was done directly from Basic with strategic PEEKs and POKEs to the 'lowcore' portion of TRSDOS 6.2. This chapter will explore scroll protection further, giving Model 4 unheard of capabilities.

TOP SCROLL PROTECTION

First, as discussed in the January 1988 issue, TRSDOS 6.2 does not document that it has a memory location that deals with scroll protecting up to 7 lines from the top of the screen. It was a capability that previously had been available only to the Assembly Language programmer, but now it is easily done from Basic.

The memory location is 0B94H (2964 decimal). The first 3 bits (0-2) indicate the number of lines to be protected from the top of the screen. Since the fourth bit (3) holds information about a different function, we must make sure that this bit is not altered by the POKE value. We can now scroll protect up to 7 lines by issueing the following command:

POKE &HB94, PEEK(&HB94) OR value. Value is the number of lines to protect (1-7). For example POKE &HB94, PEEK(&HB94) OR 3 protects 3 lines. To remove scroll protection issue this command: POKE &HB94, PEEK(&HB94) AND 248.

BOTTOM SCROLL PROTECTION

OK, scroll protection is a handy tool for the Basic programmer, but you are only able to lock out the top of the screen. Wouldn't it be wonderful if we, somehow, would be able to manipulate the bottom of the screen also?

Well, we can and though this trick takes a bit more thought and effort, let's get to work and do it:

TRSDOS 6.2 has a routine to move the cursor forward. Within this routine is a check to see if the cursor has moved beyond the end of the screen. The screen end is defined as FF7FH and is stored at these locations: 0CC5H and 0CC6H.

If you PRINT PEEK(&HCC5) you will get 127, which is decimal for 7FH. PRINT PEEK(&HCC6) will return 255, the decimal value corresponding to FFH.

If the position of the cursor is found to be off the screen, DOS will then move to a routine that issues a scroll. As the routine needs to know the size of the screen, it examines locations 0CD7H and 0CD8H where this information is stored.

PRINT PEEK(&HCD7) shows 128 (80H). PRINT PEEK(&HCD8) reveals 7 (07H).

WHAT DOES IT MEAN?

For those of you who are getting lost at this point, let us take time out to explain just what those numbers mean. Since Model 4 is an 8 bit machine, each memory location is capable of holding only an 8 bit number. That is, a number from 0 to 255.

If we use a number larger than 255, we need to use two memory locations to store that number. When the original memory location reaches 256, it reverts to 0 and stores 1 in the next memory location. In other words, a number larger than 255 uses two memory locations. The second location holds a number that indicates how many times the first location has reached 256, and the first location holds the number that has been counted since the last time it was reset to 0. Therefore to find a number that is larger than 255 it becomes neccessary to multiply the number found in the second memory location by 256, and then add the number found in the first memory location.

For example, we found 128 in 0CD7H and 7 in 0CD8H, so 7*256+128=1920. Aha... the Model 4 screen is 80 columns by 24 lines and 80*24=1920. The two memory locations gave us the right information.

Now back to the business at hand. A little further up in DOS is the routine that moves the cursor down one line. Here it again becomes neccessary for DOS to know if the cursor has moved off the screen, so it can take appropriate measures.

The cursor is considered off the screen if it is located at FF80H or higher. The data for this check is stored at locations 0D0BH and 0D0CH (holds 128 and 255 respectively).

Next is the routine to that allows us to erase from the cursor to the end of the screen, CHR$(31). This routine must also be aware of where the end of the screen is. It looks for this data in locations 0D20H (where it finds 127) and 0D21H (it gets 255 here).

Lastly we go to the routine that moves video RAM. It needs to know the screen size. It is 1920, remember?.This information is stored at 0DA8H (holds 128) and 0DA9H (holds 7).

A QUICK RECAP

0CC5H and 0CC6H (holds 127 and 255) = END OF SCREEN

0CD7H and 0CD9H (holds 128 and 7) = SIZE OF SCREEN

0D0BH and 0D0CH (holds 128 and 255) = END OF SCREEN + 1

0D20H and 0D21H (holds 127 and 255) = END OF SCREEN

0DA8H and 0DA9H (holds 128 and 7) = SIZE OF SCREEN

PUTTING IT ALL TOGETHER

This was the work. Now comes the time for logical thinking.

Since the end of the screen is defined as FF7FH, we will be able to fool DOS into thinking that the screen is smaller if we:

1. Find new END OF SCREEN address by subtracting (80 times the number of lines you wish to bottom protect) from FF80H.

2. Find new SIZE OF SCREEN by subtracting (80 times the number of lines you wish to bottom protect) from 1920.

3. Find new END OF SCREEN + 1 by subtracting (80 times the number of lines you wish to bottom protect) from FF81H.

4. POKE these new values into the appropriate addresses.

(for the exact way to do the above, refer to lines 32-42 in POKEDEM2/BAS).

USING BOTTOM SCROLL PROTECTION

Let's be completely honest. You don't really have bottom scroll protection; instead you have two separate screens. The top portion is active while TRSDOS 6.2 does not recognize the bottom part.

To take advantage of this new capability you must first put whatever text or graphics you wish at the bottom of the screen. Do this with the normal PRINT@ command. Then POKE the values for the smaller screen into the appropriate addresses. Now use the top portion of the screen as you would normally. CLS and PRINT CHR$(31) will now erase only to the bottom of the newly defined screen. The locked out bottom part is not affected at all.

Be careful not to attempt to send the cursor into the protected area with a PRINT@ command, as the Model 4 will disagreeably lock up and force you press RESET. Should the need to access the protected area arise, simply restore the screen to normal size by POKEing the regular values back into the above mentioned addresses. Then do whatever needs to be done at the bottom, and lock it out again.

REVERSE VIDEO

Another feature of the Model 4 is its ability to display reverse video. That is, black characters on a white background instead of the normal white characters on a black background.

Reverse video is activated by PRINT CHR$(16) and is turned off by PRINT CHR$(17). This can be used in a variety of ways. For example, type this short program and RUN it:

```
   10 CLS
   20 PRINT@(2,5), CHR$(16); "NAME:"; CHR$(17);
"TRSTimes"
   30 PRINT@(4,5), CHR$(16); "DATE:"; CHR$(17);
"March 1988"
```

You'll see that the headings 'NAME' and 'DATE' are displayed in reverse video, while 'TRSTimes' and 'March 1988' are shown in normal mode on the screen. The opposite is also possible. Type and RUN this program:

```
   10 CLS
   15 PRINT CHR$(16)
   17 FOR X = 1 TO 23: PRINT: NEXT
   20 PRINT@(2,5), CHR$(17); "NAME:"; CHR$(16);
"TRSTimes"
   30 PRINT@(4,5), CHR$(17); "DATE:"; CHR$(16);
"March 1988"
```

This time the entire screen is reverse video. The headings 'NAME' and 'DATE' are displayed in normal video, while 'TRSTimes' and 'March 1988' are shown in reverse video.

Now type CLS and press (ENTER).

What happened? The screen cleared and in the process it switched back to normal mode. We lost the reverse video screen. Not good.

Run the program again. After the program is done, type: NEW

NEW, of course, erased the program from memory, but it also erased the screen back to normal video. It seems that reverse video has some peculiarities.

There are two drawbacks to using the reverse video mode. First, you cannot use CLS or NEW without clearing the screen back to normal video.

Second, the special character set from 128 to 255 is replaced by the alternate character set (128-191) and the space compression codes (192-255). In other words, if your program uses any of the graphic characters, switching to reverse video will drastically change the look of your screen.

Type this program and RUN it:

```
10 CLS
20 FOR X=0 TO 255
30 PRINT CHR$(X);" ";
40 NEXT X
```

You are now looking at the Model 4's normal character set. Keep your eyes glued to the graphic characters and type:

PRINT CHR$(16) and press (ENTER)

All the characters from 128 to 255 changed to the reverse video character set.

Unfortunately, I have no remedy for this, so we will just have to live with not employing graphic characters and reverse video at the same time.

We can, however, take care of the problem with CLS and NEW. First of all let's turn the entire screen into reverse video. This can be done by typing the following:

```
PRINT CHR$(16):FOR X=0 TO 23:PRINT:NEXT
```

This does the trick, but there is a faster way. Type this code:

```
PRINT@(0,0),CHR$(16);CHR$(31)
```

Much better, but since it does not solve our problem with CLS and NEW, it isn't good enough. We will have to modify the routine than enables and disables reverse video. It is so simple that one POKE will do what we want. Location C8DH contains 0 to indicate that reverse video is disabled. CLS and NEW both access this location on their way through the disable routine. If, instead of 0, we store the value 8 there, CLS and NEW will think we want to enable reverse video. We will, in essence trick DOS into doing what we want.

Here is how it is done:

```
POKE &HC8D,8:CLS
```

The entire screen turns to reverse video with lightning speed, and all subsequent CLS or NEW commands keeps the screen in this mode. The only command that will get back normal video is PRINT CHR$(17) and that is only temperary. The next CLS or NEW sets the screen to reverse mode again.

To regain normal mode permanently, we must type:

```
POKE &HC8D,0:CLS (ENTER)
```

The scroll protects and the new video functions are only the tools. The real challenge is for your imagination to go to work and use these new capabilities to make fresh and interesting programs.

I will, as always, enjoy seeing your programming efforts that make use of the things covered in these articles. Send them to me, care of TRSTimes, and I will devote an entire installment of 'Hunting for Buried Treasure' to the best and most imaginiative of them. There is only one rule: It must be written in pure Basic.

In May, we will modify DOS so we can display the clock anywhere we like on the screen. Also, we will force our Model 4 to give us an 80 column screen in Model 3 mode, from BASIC.

Until next issue............Happy Hacking.

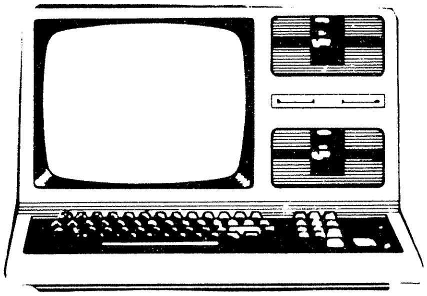
POKEDEMO 2

```
0 '********************************************

1 '* POKEDEM2/BAS - Model 4-TRSDOS 6.x.

2 '* Copyright  1987 Lance Wolstrup

3 '********************************************

10 GOTO 100

14 '***

15 '*** line 20 reads the POKE addresses into
A(0) through A(9)

16 '*** and the POKE values for normal screen
into B(0) through B(9)

17 '***

20 FOR X = 0 TO 9: READ A(X), B(X): NEXT:
DATA &HCC5, &H7F, &HCC6, &HFF, &HCD7,
&H80, &HCD8, &H7, &HD0B, &H80, &HD0C, &HFF,
&HD20, &H7F, &HD21, &HFF, &HDA8, &H80,
&HDA9, &H7

30 A$ = "Copyright 1987 Lance Wolstrup": B$
= "*** P O K E D E M O  2 ***": C$ = "Top and
bottom scroll protection - split screens - all for the
Model 4"
```

```
31 '***

32 '*** Line 40. EN=end address of new screen

33 '*** Line 41. EN2=MSB of new screen address

34 '***          EN3=LSB of first address after new screen end

35 '***          S=new length of screen

36 '***          S2=MSB of new screen length

37 '***          S1=LSB of new screen length

38 '*** Line 42. Pokes above values into addresses from line 20

39 '***

40 EN = &HFF80 - I * 80: IF EN < 0 THEN EN = EN + 65535!

41 EN2 = INT(EN/256): EN1 = EN - EN2 * 256: EN3 = EN1 + 1: S = 1920 - I * 80: S2 = INT(S/256): S1 = S - S2 * 256

42 POKE A(0), EN1: POKE A(1), EN2: POKE A(2), S1: POKE A(3), S2: POKE A(4), EN3: POKE A(5), EN2: POKE A(6), EN1: POKE A(7), EN2: POKE A(8), S1: POKE A(9), S2: RETURN

50 FOR X = 0 TO 9: POKE A(X), B(X): NEXT: POKE &HB94, 0: RETURN

60 POKE &HB97, 1: POKE &HB98, CS: FOR X = 1 TO LEN(L$): PRINT MID$(L$, X, 1);: OUT &H90, 143: FOR Y = 1 TO 60: NEXT: OUT &H90, 142: NEXT: POKE &HB97, 0: PRINT CHR$(32);: RETURN

70 I$ = INKEY$: IF I$ = CHR$(13) THEN RETURN ELSE 70

100 POKE &HB97, 0: CLS: GOSUB 20: PRINT@(21,26), A$;: PRINT@(22,27), B$;: PRINT@(23,5), C$;: I = 4: GOSUB 40

110 POKE &HC8D, 8: CLS '*** reverse video

120 CS = 0: PRINT@82, "";

130 L$ = "This is part 2 of Wolstrup's journey into the deep, dark and perilous jungles": GOSUB 60

131 L$ = " of TRSDOS 6.2's low core memory. Here we will explore something that is not": GOSUB 60

132 L$ = "   normally possible: BOTTOM SCROLL PROTECTION and SPLIT SCREENS.": GOSUB 60

140 FOR Y = 1 TO 1000: NEXT: PRINT: PRINT

141 PRINT " Notice that the screen you are watching now is employing BOTH the reverse"

143 PRINT " video mode (lines 0 to 19) AND the normal video mode (lines 20 to 23). This"

150 PRINT" is possible because the Model 4 has been tricked into thinking that the"

151 PRINT " screen only has twenty lines (0-19). When the text reaches the end of line"

152 PRINT " 19 the scroll function will occur. In other words, the bottom four lines are"

160 PRINT " scroll protected. Even the CLS command will not erase those protected lines.": PRINT

170 L$ = " These new capabilities are not accomplished by some 'fancy schmanzy' machine": GOSUB 60

171 L$ = " language routines. Instead, everything in this program is PURE Basic.": GOSUB 60

180 PRINT: PRINT

190 L$ = " Now sit back, relax and enjoy the demonstration............": GOSUB 60

200 PRINT@(18,33), "";: L$ = "Press (ENTER) ": GOSUB 60: GOSUB 70: PRINT

210 L$ = " Here is what was promised earlier. The bottom four lines are indeed protected,": GOSUB 60

211 L$ = " therefore we scrolled and left the copyright messages completely untouched.": GOSUB 60: PRINT

220 L$ = " We can, however, access the protected lines at any time. For example:": GOSUB 60: PRINT

230 L$ = " Type your name: ": GOSUB 60: POKE &HB97, 1: INPUT N$

240 N1$ = "User of this program is: " + N$: N = INT((80-LEN(N1$))/2)

250 GOSUB 50: PRINT CHR$(17);: PRINT@(20,N), N1$;: GOSUB 40: PRINT CHR$(16);: PRINT@(19,0), "";
```

```
   260 L$ = " O.K. " + N$ + ", look at line 20
and you'll get the idea.": GOSUB 60

   270 PRINT: PRINT

   280 L$ = " Now let us erase the top portion of
the screen...... Press (ENTER)": GOSUB 60:
GOSUB 70

   300 POKE &HC8D, 0: CLS: CS = 191:
PRINT@82, "See, even the CLS command does not
touch the protected lines, which makes"

   305 PRINT" this function much more powerful
than top scroll protection."

   310 PRINT " We will look at top scroll protect in
a minute, but for now let's go ahead"

   315 PRINT " and split the screen in half.";
CHR$(26); " ."; CHR$(26); " ."; CHR$(26); " .";
CHR$(26); "Press (ENTER)": GOSUB 70 :GOSUB
320: GOTO 360

   320 GOSUB 50: CLS

   325 PRINT@(0,26), A$: PRINT@(1,27), B$:
PRINT@(2,5), C$: PRINT@(3,0), STRING$(80,95)

   330 PRINT@(12,0), STRING$(80,95);

   340 L$ = "We are now using the bottom half of
the screen. It has not yet been locked out": GOSUB
60: PRINT

   341 L$ = "so we can use it at will. We can continue to put text here until we issue the": GOSUB 60:
PRINT

   342 L$ = "command to split the screen, and
then the": GOSUB 60

   350 I = 12: GOSUB 40: PRINT: L$ = "next word
switches up to the upper screen, and when the end
of the line has": GOSUB 60: PRINT

   351 L$ = "been reached we scroll instead of
going into the bottom screen.": GOSUB 60: PRINT:
RETURN

   360 L$ = "Notice that the copyright messages
scrolled off the top when the screen was": GOSUB
60: PRINT

   361 L$ = "switched. That can be eliminated by
scroll protecting the top lines.": GOSUB 60: PRINT

   370 L$ = "Let's do it again, but this time we will
protect the top 4 lines.": GOSUB 60: PRINT: PRINT

   380 L$ = "Press (ENTER)": GOSUB 60: GOSUB
70
```

```
   390 GOSUB 50: CLS: POKE &HB94, 4: GOSUB
325

   400 L$ = "At this point we actually have three different screens that can be manipulated": GOSUB
60: PRINT

   401 L$ = "in just about any manner we wish.":
GOSUB 60: PRINT: PRINT

   410 L$ = "We can even change the middle
screen to reverse video while keeping the top":
GOSUB 60: PRINT

   411 L$ = "and bottom screens in normal video.":
GOSUB 60: PRINT: PRINT

   415 L$ = "Press (ENTER)": GOSUB 60: PRINT:
GOSUB 70

   420 PRINT@(13,0), CHR$(16): FOR X = 1 TO 7:
PRINT: NEXT

   430 CS = 0: L$ = " Like this...Just like an Oreo
cookie: reverse video sandwiched in between two":
GOSUB 60

   431 L$ = " layers of normal video.": GOSUB
60: PRINT: PRINT

   440 L$ = " Hopefully, this demonstration will
give you the inspiration to do some of the": GOSUB
60

   441 L$ = " things that people said couldn't be
done from BASIC.": GOSUB 60: PRINT: PRINT

   450 PRINT TAB(31) "";

   460 L$ = "Have fun.....Lance": GOSUB 60:
PRINT: PRINT

   470 PRINT TAB(27)"";

   480 L$ = "Press (ENTER) to end demo":
GOSUB 60: GOSUB 70

   490 GOSUB 50: POKE &HB94, 0: POKE &HB97,
1: POKE &HB98, 95: CLS
```



CP/M
The alternate DOS for Model 4

by Roy T. Beck

Why in the world would anyone be interested in CP/M at this late date in the history of microcomputers, and especially why would a TRS owner be interested? I will try to give you a satisfactory answer to those questions, and at the same time try to convince you to participate in CP/M with me.

I have long known that it is possible to run CP/M on the Models I, II, III, and 4 (with somewhat of a kludge for the Mod I), but only recently have I done any serious 'computering' with it. I won't go into the history of CP/M, at least not now; instead, I will tell you a little about what I have done with it.

One of my Mod I's has had several modifications over the years, and one of these was the Mapper I board from Omikron, long since out of business. Their arrangement was technically clever, although a bit of a kludge. They supplied a small board (Mapper I) with about 1/2 dozen chips on it which plugged into the Z-80 socket. The Z-80 then plugged onto the Mapper I. When you booted the machine, it would ask you if you wanted to run a TRS DOS or CP/M; you replied with T or C accordingly. If you hit T, the machine operated as a normal Mod I, no further CP/M activity took place. If you replied with a C, the machine would then boot the somewhat special Omikron CP/M disk, and you were off and running in CP/M.The scheme worked as advertised, and I did some playing with it. However, that was in the heyday of the Mod I.

When the Mod III and 4 came along, interest focussed on their greater capabilities, and Model 1 activity began to wane. Omikron next offered a version of their system for the Mod III, but they ran out of capital before the system was fully developed and debugged. A friend of mine bought the Mod III version, received the hardware and software along with an apology that part of the software was buggy and the corrected version would be delivered "real soon". But that was Omikron's swan song and Omikron withered away.

Other outfits that offered CP/M included Lifeboat Associates in NY, and Pickles and Trout in Goleta. The Lifeboat version was not standard CP/M in that their programs had to load at 4300H instead of 100H which is standard for CP/M 2.2 This severely impaired interchangeability of Lifeboat's programs, and interchangeability is one of CP/M's major strengths. I never had occasion to get acquainted with Pickles and Trout's version.

CP/M V2.2 was written to run on an 8080 cpu, which of course it still does. But CP/M also runs on the Z-80, which is where it now mostly resides. This has the advantage of allowing all the Z-80 instructions to be used, but prevents any program containing Z-80 instructions from running on an 8080 machine. That's a gotcha to be aware of, although it does not affect the Models 1, 3 and 4.

Another outfit named Montezuma Micro in Texas (with Jesse Bob Overholt of The Alternate Source fame as the honcho) offers CP/M V2.2 for the Model 4. Due to the unique design of the Mod 4, it can be operated as a ROM-less machine. That is, it can have 64K (or 128K) of RAM and no ROM at all. This is exactly what CP/M needs and JBO proceeded to develop a BIOS to mate CP/M to the Mod 4 quite legitimately and respectably.

When the days of the Mod 4 seemed to be numbered, I decided to get into CP/M for the sake of additional software availability, and I bought Montezuma Micro's package. I haven't regretted it.

One of the clever features is JBO's CONFIGuration program which allows the Mod 4 to read and write almost any soft-sectored CP/M format you may run into. I can (and have) read and written Kaypro's formats (these are popular for interchange of programs) and an 80 track DS DD format from a friend's Zenith machine.

What am I doing with CP/M? One serious task is the management of the membership list of the Valley Computer Club, with a roster of over 200 names and addresses. I am using DBase II for this task, which runs very happily on either my Mod 4DD or my Mod 4P, depending upon my mood and which machine is handy.

What, you may ask, is a Mod 4DD?

In my case, it is a former cassette machine which has been slightly enhanced. It now has 2 40 track, DS DD drives, 2 80 track, DS DD drives, and an RS-232-C serial board, along with, of course, 128K.

The "growth" was accomplished by Jack Eich of Orange County in California, who is quite knowledgeable of the internals and capabilities of the Mod 4. He is one of the "good guys" of the TRS world. He quite regularly upgrades Mod III's and 4's to greater capability. He installs disk controllers, towers and drives in cassette machines, and upgrades 2 drive machines to 4 drives. (The 4 drive version requires 2 power supplies, by the way).

Yes, I know, CP/M is regarded by many TRS types as being a rather stupid DOS, because it doesn't have all the bells and whistles we are accustomed to in more elegant DOS' such as NEWDOS-80, LDOS, etc. A common complaint is that CP/M isn't smart enough to look on another disk for a program if it doesn't find it on the one which it is logged onto.

In defense of CP/M, there are reasons for its limitations. A very significant one is that it was written to be SMALL. In fact CP/M is expected to occupy no more than about 6K, including BIOS, BDOS, and CCP. (More on these arcane initials later).

There is also a block of 100H, (256D) bytes at 0000H which contains pointers, etc. When CP/M was layed out for the 8080 CPU the total memory map was 64K, and few people were rich enough to own that much RAM. Consequently, CP/M was written to include only a bare minimum of commands (there are only 6: DIRectory, ERAse, REName, SAVE, TYPE, and USER) and was designed to reside entirely in memory without calling for overlays from the DOS.

"But wait a minute", you may say, "What about Copy, Backup, Format and all those other essential needs?" The answer in CP/M is that these and many other functions are available, but under other names, and they are separate programs instead of being commands within the body of DOS.

Some of the names are intriguing. For COPY, would you believe PIP? That actually stands for Peripheral Interchange Program. It has the kind of functions we would expect from a TRSDOS COPY function, and includes "wild cards" of the sort we now see in PC-DOS. In fact CP/M is the direct ancestor of PC-DOS, but that's a story for another article.

Another feature of CP/M is that one chunk called the Console Command Processor (CCP) can be overwritten by an application program when it is operating, thus freeing up a little more of the precious RAM for use by the application program. Of course the CCP must be restored when it is needed again, and in this respect CCP can be viewed as a form of overlay.

But there is no large group of overlays in CP/M, just CCP, and it is brought in for use only when no application program is in use.

When and for what do we need the CCP?

As the name indicates, the CCP processes commands from the console (reads the keyboard). There is another portion of CP/M called the Basic Input Output Structure (BIOS) which tailors the DOS to the actual hardware of your machine.

But the BIOS only accommodates your machine's peculiarities. It consists mostly of a collection of drivers for such things as the screen memory, the keyboard, and some tables for use with the disks. CCP knows nothing and cares less about the hardware of your machine. It asks the BIOS to fetch input from the keyboard when only the DOS is active. When an application program is operating, the application program works through the BIOS and doesn't need the CCP. This is the trick which allows overwriting of the CCP when an application program is running.

So how do we get the CCP back when we need it? We do a "warm boot". This is accomplished by pressing ^C (Control C).

The Basic Disk Operating System (BDOS) which is the heart of CP/M and the BIOS will not have been disturbed by any well-behaved application program, and a warm boot (^C) brings back the command level with disk parameters, directory flags, etc undisturbed.

Of course if BIOS or BDOS have been damaged by a runaway program, there is no choice but to cold boot the machine and start from scratch. If we need to start completely over in this fashion, we ordinarily can use the RESET key on the Model 4 or 4P to perform a "Cold Boot". This will bring in a fresh copy of the CP/M system, including BIOS, BDOS and CCP, and reset all of the pointers.

But occasionally a TRS machine will crash and be totally unresponsive to the keyboard AND the "RESET" button. What goes on in this case?

Here we must digress. Radio Shack has never (so far as I am aware) provided a genuine RESET switch on any of its machines. The key or pushbutton identified as "RESET" actually initiates a ReStarT (RST) instruction, which operates through software to perform an orderly restart of the machine without wiping everything out of the RAM. RST operates through RAM and the Bootstrap ROM to bring in a new copy of the CP/M system, including BIOS, and restores normal operation of the machine.

What has happened when the machine "locks up" is that the ill-behaved software has destroyed the software linkage between the "Reset" button and the ROM which contains the instructions necessary to recover control of the machine. In these (usually) rare circumstances, there is no choice. Turn off the power, wait a few seconds, and switch the power on again. This is guaranteed to recover control of the machine (and dump whatever file was still in the RAM). The only exception to my "guarantee" would be if there is some electrical or mechanical fault within the machine, and that's a whole nother world, not covered here.

Enough chatter for this issue. Next time I will get into the CONFIG program which allows us to swap programs with many other CP/M machines.

As this will be a regular feature of TRSTimes, I invite comments and questions. I do have several friendly experts to call upon should I get stuck, so I hope to be able to give valid answers to your questions. At any rate let's give it a try.

Until next time....

Roy Beck is an Electrical Engineer in Los Angeles who got into personal computing partly as a hobby and partly to upgrade his technical knowledge. He has Radio Shack Models I, 4D, 4P and 100 along with an IBM clone, all of which are in use.

ITEMS OF INTEREST

USER GROUPS

• California
• Campbell

BYTE BANDITS OF AMERICA TRS-80 COMPUTER CLUB
Contact: R.W. Brown 780 Manx Ave.
Campbell, Ca. 95008

• Los Angeles Area

OCTUG (Orange County TRS-80 Users Group)
Meets the third Sunday of each month at 'The City Shopping Center' in Orange. General meeting starts at 1:30 p.m.
Contact: Franco Collino (714) 750-0887

SAGATUG (San Gabriel Valley TRS-80 User Group)
Meets the second Friday of each month at Arcadia Park Senior Citizens Center. 7:30 p.m.
Contact: Mark Speer (818) 841-2119

VTHG (Valley TRS-80 Hackers' Group)
Meets the first Friday of each month at Valley Plaza Recreation Center. 12240 Archwood St. North Hollywood, Ca. 7:30 p.m.
Contact: Eric Bagai P.O. Box 9747
North Hollywood, Ca. 91609 (818) 982-0467

VTUG (Valley TRS-80 Users Group)
Meets the third Thursday of each month at Glendale Federal Savings. 7129 Topanga Canyon Blvd. Canoga Park, Ca. 7:30 p.m.
Contact: Fred Blechman (818) 346-7024

• Colorado
• Denver

DATA (Denver Area TRS-80/Tandy Association)
Meets the fourth Wednesday of each month (except December) at the Aurora Central Library.
Contact: Norman Rowe 7200 E. Bayaud Ave.
Denver, Co. 80231

• Illinois
• Chicago

T-BUG (Tandy Business Users Group)
Meets the third Wednesday of each month at the OLD WARSAW. Harlem and Lawrence. Chicago
Contact: Linda Hapner, 3329 B. Beacon #50
N. Chicago, IL. 60064

• New Jersey
• Freehold

BUG-80 USERS GROUP (Brookdale User Group)
Meeets the third Friday of each month at Electronics Lab (NAS Building). Brookdale Community College. 7:30 p.m.
Contact: Bug-80 Users Group. 22 Alexander Ave.
Freehold, N.J. 07728
Fred Kagel (201) 577-0606

• Texas
• Arlington

Mid-Cities TRS-80 Users Group
P.O. Box 171566
Arlington, TX. 76003
Meets the second and fourth Tuesday of each month at Arlington Community Center-Garden Room. 2800 S. Center. 7:00 p.m.
Contact: Hal Widdas

• Washington D.C. area

TRS80 "134" Computer User group.Meets the first Wednesday of each month 1-3 p.m. at the Holiday Park Multiservice Center. 3950 Ferrara Drive. Silver Springs, Md.
Contact: Paul Shapiro P.O. Box 2711
White Flint Mall. Kensington, Md. 20895-0824

• Hawaii
• Honolulu

Hawaii Tandy User Group
Contact: Bob White
1320 Thirtieth Ave.
Honolulu, Hi. 96815

• Canada
• British Columbia
• Vancouver Area

SMUG (The Surrey Microcomputer Users Group).
Meets the third Thursday of each month at Bear Creek Pavillion. 88th Ave & King George. 7:30 p.m.
Contact: Karl Mohr 2256 - 153rd Street
Surrey, British Columbia, Canada V4A 4R2

• Ontario
• Ottawa

Micro-80 Computer Club of Ottawa
1551 Riverside Drive, Suite 901
Ottawa, Ontario, Canada K1G 4B5
Contact: Brian Graham

• International
• England

NATGUG (National Amstrad Tandy & General User Group
Newsletter: 'NATGUG News' available at 12 pounds for twelve issues (within the U.K. only - cost to the U.S. is not available at this time.)
Contact: Gordon Collins
11, Elizabeth Road - Sutton Coldfield
West Midlands - B73 5AR - England

BULLETIN BOARDS

• California
• Los Angeles area

COMPUCENTER/LA
(213) 435-3757 & (213) 320-0222
300/1200 baud (24 hrs)
Has TRS-80 section.
Sysop: Rick Gaspa

THE TANDI EXPRESS + BBS
(213) 665-2378)
300/1200 baud. (3 pm - 6 am daily.
24 hrs Saturday & Sunday)
Sysop: Nick Hellens

THE BYTE ZONE BBS
(714) 639-3566
300/1200 baud (24 hrs)
Sysop: Franco Collino

• San Diego

8 BIT TANDY
(619) 571-6366
300/1200 baud - 24 hrs
Sysop: Mike Stark
Model I,III,4,100,102,200,CP/M & OS-9

• Ohio
• Cleveland

RAILNET BBS
(216) 883-6298
300/1200 baud (24 hrs)
Sysop: Rick Demattia

• Pennsylvania
• Philadelphia

8/N/1 #4
(215) 848-5728
300/1200/2400 baud - 24 hrs
Sysop: Luis Garcia-Barrio
Home of TRSLINK

• Canada
• British Columbia
• Vancouver Area

FAST PLUS BBS
(604) 594-7398
300/1200 baud - 24 hrs
Parity 8N1
Sysop: Mel Patrick

The above User Groups and TRS-80 oriented BBS list is far from complete. We would like to include all groups and BBS' that support our machines, but we cannot do it without your help.

Please send us the particulars about your group or BBS so we can help you get more members and callers. Remember, there are many people out there who are interested, but they can't get to you until they know where you are.

This may be a good time to issue a plea to all TRS-80 User Groups. Get pen and paper, or lift the phone, and contact the other groups. We can no longer (nor could we ever) afford to be 'stuck up' or 'shy'. Make arrangements to send your newsletters and club disks to the other groups.

All of us, whether on the East coast - Midwest - deep South - West coast - Canada or anywhere else, share the same goal, that of promoting the TRS-80, and we are all hungry for new information and material. The short article you wrote or the small program you just put together, believe me, someone, somewhere will 'enjoy the Heck' out of it. So let's get active and do it.

When someone needs help, do take the time and help them, and especially when someone new shows up at your group meeting, talk to them, be interested in them, answer their questions and more often than not, you will make a friend who will participate actively for a long time. I have gone to a few meetings at clubs that did not follow the above etiquette, and you can be sure, I never returned.

TRSTimes will provide space for User Group information of broad interest to the TRS-80 community. If already printed in your local newsletter, please include written permission to reprint.

Send User Group material to:
TRSTIMES
20311 Sherman Way #221
Canoga Park, Ca. 91306
Attn: User Group Info

JUMP 80

A game for Model 4
by Lance Wolstrup

When Model 4 owners indulge in gameplaying, it is almost a sure bet that it is done in Model III mode. While thousands of games exists for Model I & III, next to nothing has been written for the Model 4's native mode. I know of only a handful of public domain games and, to the best of my knowledge, nothing commercial was ever available.

One of the reasons for this 'fun anemia' is the difference between the Model III and 4 graphic character set (CHR$ 128 though 191). At first glance they appear to be identical, however, they are not. The three vertical rows of the Model III graphic characters are all the same size. The Model 4's graphic characters have the top two rows identical, but the bottom row is flattened out, making it smaller. This, along with the fact that the screen cannot be addressed directly, makes graphics and smooth animation difficult.

My main interest in computers is exploration of DOSes and writing utilities, but occasionally I relax and have fun by writing a game or two. JUMP80 is the result of one of those 'breaks'. It is of the solitaire-puzzle genre, employing limited graphics, and it is not difficult to play.

The game consists of nine graphic blocks. Four blocks are checkered, one is blank, and the last four are solid white. The object is to reverse the position of the checkered blocks and the solid white blocks, according to the following two rules:

1. A checkered or solid white block can be moved to the blank block IF it is immediately next to it.

2. A checkered or solid white block can be moved to the blank block IF it can reach it by jumping over ONE block.

These are simple rules for a simple game and most everyone will succeed, but the real challenge is: Do it in 24 moves.

Moving a block is done by pressing the number that is displayed underneath it. Should you attempt an illegal move, no movement will take place, however, you will still be charged for it.

You may quit the game at any time by pressing (Q). Pressing (I) instead of a block number will cause the rules to be displayed.

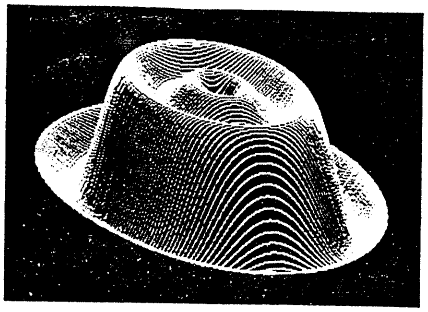
Type in the short program listing and start JUMPing on your TRS-80.

```
0 'JUMP80/BAS - Model 4

5 DEFINT A-Z: DIM BL$(2), BL(9): M = 0: PF = 24:
RT$ = "Puzzle rating: ": PRINT CHR$(15)

10 BL$(0) = STRING$(7,32): BL$(1) =
STRING$(7,153): BL$(2) = STRING$(7,191)

15 GOTO 100

20 H = INT((80 - LEN(L$))/2): PRINT@(V,H), L$;:
RETURN

30 H = I * 9: GOSUB 90: SWAP BL(I), BL(I + 1):
RETURN

40 H = (I + 1) * 9: GOSUB 90: SWAP BL(I), BL(I +
2): RETURN

50 H =(I - 2) * 9: GOSUB 95: SWAP BL(I), BL(I - 1):
RETURN

60 H =(I - 3) * 9: GOSUB 95: SWAP BL(I), BL(I - 2):
RETURN

70 PRINT@(15,0), CHR$(31): PRINT@(16,0), "The ob-
ject of JUMP 80 is to reverse the positions of the check-
ered and white": PRINT "squares.  A square may be
moved if it is immediately next to the empty block or,";

75 PRINT "If the empty block can be reached by jump-
ing over only one square.": PRINT "The puzzle is not dif-
ficult, but it will take skill to get a perfect score.": L$ =
"Press any key to return to the puzzle": V = 22: GOSUB 20

80 I$ = INKEY$: IF I$ = "" THEN 80 ELSE
PRINT@(15,0), CHR$(31): RETURN

90 FOR V = 5 TO 7: PRINT@(V,H), BL$(BL(I));: NEXT:
H = (I - 1) * 9: FOR V = 5 TO 7: PRINT@(V,H),
BL$(0);: NEXT: RETURN

95 FOR V = 5 TO 7: PRINT@(V,H), BL$(BL(I));:NEXT:
H = (I - 1) * 9: FOR V = 5 TO 7: PRINT@(V,H), BL$(0);:
NEXT: RETURN

100 CLS: PRINT@(0,0), "TRSTimes presents  ";
CHR$(21); CHR$(143); CHR$(244); CHR$(245);
CHR$(246); CHR$(21);

110 L$ = "(c) 1988  Lance Wolstrup": PRINT@(0,80-
LEN(L$)),L$;

120 L$ = "J U M P  8 0": V = 0: GOSUB 20

130 PRINT@(1,0), STRING$(80,140)
```

```
200 FOR X = 1 TO 4: BL(X) = 1: NEXT: BL(5)
= 0: FOR X = 6 TO 9: BL(X) = 2: NEXT: FOR V
= 5 TO 7: H = 0: FOR X = 1 TO 4:
PRINT@(V,H), BL$(1);: H = H + 9: NEXT: NEXT

210 FOR V = 5 TO 7: PRINT@(V,H), BL$(0);:
NEXT

220 H = H + 9: I = H: FOR V = 5 TO 7: FOR
X = 1 TO 4: PRINT@(V,H), BL$(2);: H = H + 9:
NEXT: H = I: NEXT

230 V = 9: H = 2: FOR X = 1 TO 9:
PRINT@(V,H), X;: H = H + 9: NEXT

240 PRINT@(10,0), STRING$(80,140)

250 PRINT@(14,35), "Moves: ";:PRINT USING
"###";M

260 L$ = "Press number of the block to move -
(I) for Instructions - or (Q) to quit": V = 22:
GOSUB 20

300 I$ = INKEY$: IF I$ = "" THEN 300 ELSE IF
I$ = "Q" OR I$ = "q" THEN CLS: PRINT CHR$(14):
END ELSE IF I$ = "I" OR I$ = "i" THEN GOSUB
70: GOTO 260 ELSE I = VAL(I$)

310 IF I<1 OR I>9 THEN 300

320 IF BL(I) = 0 THEN 500 ELSE ON I GOTO
410,420,430,430,430,430,430,440,450

410 IF BL(I+1) = 0 THEN GOSUB 30: GOTO
500 ELSE IF BL(I + 2) = 0 THEN GOSUB 40:
GOTO 500 ELSE 500

420 IF BL(I + 1) = 0 THEN GOSUB 30: GOTO
500 ELSE IF BL(I + 2) = 0 THEN GOSUB 40:
GOTO 500 ELSE IF BL(I - 1) = 0 THEN GOSUB
50: GOTO 500 ELSE 500

430 IF BL(I + 1) = 0 THEN GOSUB 30: GOTO
500 ELSE IF BL(I + 2) = 0 THEN GOSUB 40:
GOTO 500 ELSE IF BL(I - 1) = 0 THEN GOSUB
50: GOTO 500 ELSE IF BL(I - 2) = 0 THEN
GOSUB 60: GOTO 500 ELSE 500

440 IF BL(I + 1) = 0 THEN GOSUB 30: GOTO
500 ELSE IF BL(I - 1) = 0 THEN GOSUB 50:
GOTO 500 ELSE IF BL(I - 2) = 0 THEN GOSUB
60: GOTO 500 ELSE 500

450 IF BL(I - 1) = 0 THEN GOSUB 50: GOTO
500 ELSE IF BL(I - 2) = 0 THEN GOSUB 60:
GOTO 500 ELSE 500

500 M = M + 1: PRINT@(14,42), USING
"###"; M: IF M = 999 THEN 595

510 SC = 0: FOR X = 1 TO 4: SC = SC +
BL(X): NEXT: IF SC = 8 THEN 520 ELSE 300

520 SC = 0: FOR X = 6 TO 9: SC = SC +
BL(X): NEXT: IF SC = 4 THEN 530 ELSE 300

530 L$ = STRING$(LEN(L$),32): V = 22:
GOSUB 20

540 L$ = "You solved the puzzle in" + STR$(M)
+ " moves": V = 14:GOSUB 20

550 L$ = "You used" + STR$(M - PF) + " unnecessary moves": V = 16: GOSUB 20

560 IF M = PF THEN L$ = RT$ + "Perfect
score - GENIUS":GOTO 600

570 IF M  PF + 5 THEN L$ = RT$ + "EXCELLENT - but not perfect": GOTO 600

580 IF M  PF + 9 THEN L$ = RT$ +
"AVERAGE - you need practice": GOTO 600

590 L$ = RT$ + "TERRIBLE - hang your head
in shame": GOTO 600

595 L$ = STRING$(LEN(L$),32): V= 22:
GOSUB 20: L$ = "You did not solve the puzzle": V
= 14: GOSUB 20: L$ = "YOU MUST BE VERY
CONFUSED - read the rules"

600 V = 18: GOSUB 20: L$ = "Would you like
to try again (Y/N)": V = 22: GOSUB 20

610 I$ = INKEY$: IF I$ = "" THEN 610 ELSE IF
I$ = "N" OR I$ = "n" THEN CLS: PRINT CHR$(14):
END ELSE IF I$ = "Y" OR I$ = "y" THEN
PRINT@(14,0), CHR$(31): M = 0: GOTO 200
ELSE 610
```

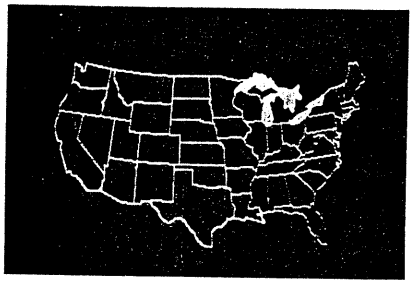
Grafyx Solution™ Save $150.00

Hi-Resolution Graphics for Mod 4/4D/4P/III



Superior Hardware. The Grafyx Solution provides 153,600 pixel elements which are arranged in a 640 × 240 or on the Model III a 512 × 192 matrix. Hundreds of new business, personal, engineering, and educational applications are now possible. The hi-res display can be shown on top of the standard display containing text, special characters, and block graphics. This simplifies program debugging, text labeling, and upgrading current programs to use graphics. The Grafyx Solution fits completely within any tape or disk based Model 4, 4D, 4P, or III. Installation is easy with the plug-in, clip-on Grafyx Solution board.

Superior Basic. Over 20 commands are added to the Basic language. These commands will set, clear or complement points, lines, boxes, circles, ellipses, or arcs. The hi-res screen can be printed on any of 30 popular printers or saved or loaded to disk without leaving Basic. Areas may be filled in with any of 256 patterns. Sections of the screen may be saved and then put back using any of five logical functions. Labels can be printed in any direction. The viewing area can be changed. The entire screen can be complemented or cleared. Graphics Basic provides dot densities of 640 × 240, 320 × 240, 160 × 240, and 160 × 120, all of which can be used in the same display.



Superior Software. The board comes with over 40 programs and files which make it easier to use, serve as practical applications, demonstrate its capabilities, and serve as programming examples. The software works with TRSDOS 1.3, 6.1.2, 6.2, 6.3; Dosplus 3.4, 3.5, 4; LDOS; and Newdos80. The Grafyx Solution is also supported by 30 optional applications programs: Draw, Bizgraph, xT.CAD, 3D-Plot, Slideshow, Mathplot, Surface Plot, Chess, etc.

The Grafyx Solution package is shipped complete for $149.95 (reduced from $299.95). The manual only is $12. Payment may be by check, Visa/MC, or COD. Domestic shipping is free on pre-paid orders. Texas residents add 7% sales tax.

Micro-Labs, Inc. 214-235-0915
902 Pinecrest, Richardson, Texas 75080

Graphics Solutions

High-Resolution Software and Hardware

GBASIC 3.0 - Radio Shack Model 4/4D/4P/III hi-res board owners take note of an enhanced graphics Basic: GBASIC 3.0 It not only provides an equivalent for each of the BASICG commands but adds a number of important new ones while using less memory. Without having to exit Basic, the hi-res screen can be saved to disk, loaded from disk, or printed on any of 30 popular printers: Epson, Star Micronics, Radio Shack, Okidata, C. Itoh, NEC, etc. The software works with TRSDOS 1.3, 6.1.2, 6.2; DOSPLUS 3.4, 3.5, 4; LDOS; and NEWDOS80. The disk contains 40 graphics programs/files. Also included is a detailed manual with assembly language entry addresses. $49.95. (Specify Model 4 or III mode or add $10 for both.)

The following eleven programs run on a Model 4/4D/4P/III equipped with a Radio Shack graphics board and GBASIC 3.0 or a Micro-Labs Grafyx Solution board:

DRAW - A powerful full screen graphics drawing and editing program. $39.95.

BIZGRAPH - Create business graphs from hand-entered or VisiCalc data. $75.00.

xT.CAD - Professional drafting aid which outputs to a printer or plotter. $245.00.

SURFACE PLOT - Plot three-dimensional equations of the form Z=F(x,y). $39.95.

3D-PLOT - View three-dimensional data from any perspective or angle. $39.95.

MATHPLOT - Plot equations of the form Y=F(x) with auto scaling. $39.95.

CHESS - A very powerful program with 10 skill levels, 40 play options. $49.95.

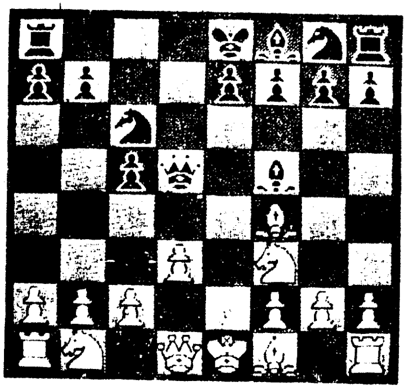
REVERSI - Play Othello with 10 skill levels, 20 execution options. $29.95.

3D Tic-Tac-Toe - Play the computer or a friend on a 4 × 4 × 4 matrix. $19.95.

SLIDESHOW - Create a sequence of hi-resolution picture displays. $19.95.

Biorhythm/USA - Plot your biorhythm or learn the states and capitols. $19.95.

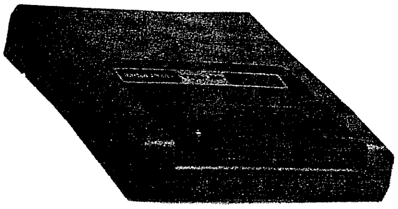
JOY-MOUSE - Allows a Radio Shack Color Computer joystick, mouse, or touch pad to be connected to any Model 4/4D/4P/III. Hardware provides X, Y position values from 0 to 255. $119.95.



GRAFYX SOLUTION - A plug-in, clip-on board enhances any Model 4/4D 4P/III to provide 640 × 240 dot graphics. (512 × 192 on a Model III) The board comes with a 56 page manual and a disk containing both model 3 and 4 mode versions of over 40 programs and files including GBASIC 3.0 which adds over 20 graphics commands to Basic. ~~$199.95~~.
149.95

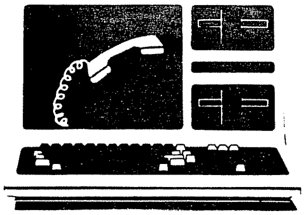
Please specify your exact system configuration when ordering or requesting information. Payment may be by check, Visa, Mastercard, or COD. Domestic shipping is free on pre-paid orders. Texas residents add 5⅛% sales tax.

Micro-Labs, Inc. 214-235-0915
902 Pinecrest, Richardson, Texas 75080



TIM's PD EXPRESS
WHAT IS SHAREWARE?

by Timothy Sewell



"Shareware"
"Freeware"
"Trustware"
"Guiltware"

The names seem to go on endlessly and are appearing more frequently on your favorite BBS system.

But just what are all of these "wares"?

They are all different names for the same thing.

USER SUPPORTED SOFTWARE.

Software authors have a mounting problem to face. Unless you are part of a large company that can afford to take chances, marketing software has become a nightmare and can be very unprofitable for the "little guy".

Sure, you can find a publishing company to handle it for you, but after manufacturing, marketing, advertising, packaging, and distribution fees, the price of the software becomes so inflated that sales are dismal, reaching only a limited amount of "paid" users, and pirating is encouraged. Not a healthy prospect indeed.

The problem facing many software authors is how to get their work exposed to the widest possible audience with the least expense. Many have turned to SHAREWARE.

The concept of Shareware is quite simple. The author allows his program to be distributed over the phone lines through BBS systems, and passed around at computer club meetings. When you obtain a Shareware program, what you usually get is the author's software and a Documentation file explaining what the program does and how it works. In most cases the documentation is minimal and gives you just enough information to get you going so you can get a "feel" of the program and decide if it's something you really want.

If you decide you like the software, you are encouraged to send a small payment or donation to the author and in return you get the author's support for updated versions and in some cases full printed documentation. Donations are very reasonable and average in the $5 to $30 range depending on the particular type of program. In more complex programs such a BBS programs, your support gets you additional utilities and upgrades not made available to the BBS world.

By cutting out the middle man and getting your program directly to the users, you end up saving money on both ends. Everybody comes out ahead. The author has his program in the hands of countless prospective supporters, and you get to decide "hands on" if the program is right for you. You are not at the mercy of a catalog description designed to SELL you the software.

Everybody wins, right? Well almost.

The sucess of Shareware depends on the users. Too many people have a tendency to use this software without donating to the author's time and effort. Rarely is it a case where the user DOESN'T want to support it. Most of the time it's simply a matter of forgetting to sit down for a minute or two and writing out that check. Unless specified for printed manuals and support, donations are what YOU feel the program is worth. The author may suggest a donation amount, but in the end it's up to you.

By donating, you give the author a chance to see how his work is being received, what changes may be needed to the software, and it encourages the author to continue writing programs for your computer. This is VERY important in the case of the TRS-80 Model 3/4 since printed support is at a minimum and there is really no financially practical medium to reach a wide TRS-80 audience.

Most users have turned to the phone lines for support of their computers and more are joining the ranks daily. Others are turning to user groups or individuals who provide this software through mail order. No matter how it is distributed, the important thing is that the software is getting exposure.

If you use a Shareware program, please donate to the author. If you can't donate at least write a letter expressing how much you enjoy the program. Without some sort of feedback, the author sees no return on his time, gets fed up with it all and then tosses his hands up into the air and runs out to buy an IBM clone or a Mac.

So get with it! TRS-80 authors need your support. Without it, our computer will indeed become the dinosaur everybody claims it to be.

The following is a list of some of the more popular Shareware programs available on GEnie or through The File Cabinet:

ARC4 and ARC31 - File compression Archiving utility.

XARC4 - ARC file extraction utility that un-ARC's IBM created Archives.

SHELL - Adds single key commands and file tagging to Trsdos/LS-Dos.

TRSCAT - Model 4 disk catalog program by Mel Patrick.

FASTPLUS - BBS program by Mel Patrick.

SEA PUP - Batch file generator for BBS use.

REMS80 - BBS program by George Lee.

ERACS - BBS program by Macro Computer Solutions.

CREATOR/REPORTER - Program generator.

MACDISP - MacPaint file display/print program for High Resolution.

TRSDRAW - High Resolution drawing program.

These are just a few of the many programs that are available. Though all are certainly worth the search and are worth donating to, I will single one out in particular. It is THE most useful utility found on my operating system disk.
It is SHELL version 2.0 by Stephen Milliken.

SHELL makes your computer easier to use by installing a "shell" over TRSDOS, thus allowing single key execution of programs, enhanced file copy ability, paginated listing of ASCII files, and much more.
Installation is easy. Simply use the INSTALL program included with the main program. It places the SHELL on your system disk as SYS13/SYS and even allows you to configure for a MEMDISK should you choose to use one.
Once installed, all you have to do is type * < ENTER > at the READY prompt and you are on your way into the program.
What you will see is the visible files on drive :0 listed alphabetically along with an option menu displayed at the bottom of the screen. A cursor is flashing at the first file name and SHELL is now waiting for your input.

To see the files on another drive, just type the number of the drive you want and it will be displayed in the same fashon. SHELL supports hard drives and up to 8 directories are allowed.

The most useful feature of SHELL is the ability to "tag" files for copying or deleting. Instead of having to type the COPY command along with the file name, you simply move the cursor to the desired file, type "T", and the name will be displayed in reverse video. This file is now "tagged" and you can copy, delete, make invisible, or make visible a hidden file, all with a single key command. And you are not limited to just one file. You can "tag" as many files as you desire and do mass command execution.

SHELL's ability to add file descriptions is a tremendous aid to anybody who has forgotten what a file name is for. You can add this description to as many files as you have on disk, and you can even print out the file name and descriptions on your printer for reference.
Say you have a disk full of files but you only want to see the files with a /BAS extention. No problem. Define the extention, or part of an extention, that you wish to view and SHELL does the rest. Only your defined extention will be displayed.

Should you wish to run a program, move the cursor to the desired file name and hit < ENTER >. The file must have a /CMD, /BAS, or /JCL extention and BASIC files are prompted for regular BASIC or High Resolution BASICG.

If you are tired of pressing SHIFT @ to pause a scrolling screen, you'll love the extended listing (ELIST) feature of SHELL. Text will be displayed one page at a time and you can review previous pages or advance several ahead.
Just about any DOS function can be performed from SHELL with less keystrokes and saved time. It is truly one of the most useful utilities available today for your TRS-80 Model 4.

SHELL is a SHAREWARE program and a donation is requested by Steve Milliken. This donation will entitle you to further upgrades to SHELL, not otherwise available to the BBS community.

SHELL can be found in ARC format in the Model 4 Down Load section of GEnie (file # 1589) and on some BBS systems. It is also available from The File Cabinet's Public Domain Disk Library.

Stephen Milliken can be reached on GEnie (address is S.MILLIKEN), or you can write to him at:
Stephen Milliken, 10 Cochato Park,
Randolph, Ma. 02368

CATTING around with TRSDOS 1.3

by Lance Wolstrup

Model III
TRSDOS 1.3 - Editor Assembler

CATTING around with LDOS 5.1.4. (January 1988) received a lot of favorable mail. One letter, however, struck a chord when it politely reminded me that not all TRS-80 owners have ventured beyond TRSDOS 1.3.

The letter went on to say that "TRSDOS 1.3 deserves a CAT command just as much as the fancy DOS'es. Do all of us 1.3. diehards a favor and give us one."

That sounded like a reasonable request, so I dusted off my TRSDOS manual and sat down to brush up on the inner workings of our 'parent DOS'.

Almost immediately I found a documented DOS routine that seemed like it ought to do the job. The routine is called $DSPDIR and begins at 4419H.

The manual gave the following conditions:

(X'4271) = ASCII-coded drive number "0","1","2", or "3"

CALL $DSPDIR

All registers are changed on exit.

That seemed easy, so I wrote a short program to do just that. I LoaDed register A with 30H, which is the ASCII code for 0, and then LoaDed memory location 4271H with the contents of the A register. Then I CALLed 4419H. Simple! Nothing to it! Piece of cake, right?

WRONG!! When I ran the program, instead of a nice CATalog display, I got a screen full of garbage. Having written one or two programs in the past that didn't work perfectly the first time, I knew it was DEBUGGING time.

Listing the program revealed nothing. The instructions from the manual had been followed verbatim.

I tried several other ways of coding the instructions and all met with the same result: No CATalog and garbage galore on the screen. This was getting frustrating, but I knew it had to be possible. After all, CMD"D:dn" from Basic produces a CATalog.

Suddenly it dawned on me that maybe the manual was wrong. Did they forget to tell us that another register needed a specific value on entry to $DSPDIR? Well, it was worth a try.

I added a line that LoaDed register B with 0 and ran the program. No garbage; instead drive :0 made one of those long, painful noises that immediately makes one frantically press the orange RESET button. I concluded that register B should definitely NOT contain 0.

Maybe register C would produce better luck, so I changed the line to read LD C,0 and ran the program.

VOILA!! Right in front of my very eyes was the prettiest CATalog ever seen on a Model III running TRSDOS 1.3.

Nothing is as satisfying as success after failure.

CAT/CMD for TRSDOS 1.3 can be used by typing **CAT :drivenumber** from DOS, where drivenumber is valid drive 0 - 3.

CAT/CMD

```
00100           ORG     7000H
00110 START     LD      C,0
00120           LD      A,(HL)
00130           CP      58
00140           JR      NZ,QUIT
00150           INC     HL
00160           LD      A,(HL)
00170           CP      34H
00180           JR      NC,QUIT
00190           LD      (4271H),A
00200           CALL    4419H
00210 QUIT      RET
00220           END     START
```

Hypersoft bridges the TRS-80 - PC Gap.

Programs for PCs and Compatibles.

-=-

PC-Four - A TRS-80 Emulator.

Now you can run your favorite TRS-80 Model 4 programs on your PC with **PC-Four**. Not just BASIC but machine language programs as well!. This is another Hypersoft FIRST!. **PC-Four** is a program that makes your PC or Compatible behave like a TRS-80 Model 4 complete with operating system, Z80 microprocessor and 128K of memory so you can run many of your favorite Model 4 programs such as **ALCOR C, MULTI-BASIC & PASCAL, ALLWRITE, BASCOM, ELECTRIC WEBSTER, PFS FILE, PROFILE, SUPERSCRIPSIT, VISICALC, Model 4 BASIC**, and many more. Recommended by Prosoft for running Allwrite on your PC.

PC-Four even works with assemblers such as **ALDS, EDAS** and **MZAL** and debugger/monitors such as **TASMON** so you can write, assemble, debug and run Z80 machine code programs on your PC. To use it you must transfer your old files to MSDOS disks first. For this we recommend **PCXZ or Hypercross** - see below for details.

Runs on PCs or compatibles with at least 384K of memory. Put it on your lap-top, now you can carry your TRS-80 in your briefcase, wherever you go!. Also runs on IBM PS/2s.

Prices: Order #PC4 $79.95 alone, #PC4H $104.95 with Hypercross SX3PCM4, #PC4Z $119.95 with PCXZ. Send $3 for PC4/PCXZ demo disk - refundable on order. **PC-Four is also available on 3.5" disk format for lap-top machines, Tandy 1000TX, IBM PS/2s etc.**

PCXZ reads TRS80 disks on your PC

PC Cross-Zap **(PCXZ)** is a utility that runs on your PC or PC-compatible. With it you can **copy** files **to** or **from** TRS-80 disks at will. Suitable for all types of files, BASIC, ASCII and Binary. **Converts BASIC and text files automatically** as you copy. You can also format a disk, copy disks, explore, read and write sector data, repair bad directories and much more. Long after your TRS-80 is gone you will still be able to read your old disks. **Formats Supported:** Model I mixed density: DOS+ 3.4, DoubleDOS, LDOS (SOLE), MultiDOS, NEWDOS 80 V2, TRSDOS 2.7/8; Model I/III Double Density: DOS+ 3.5, LDOS 5.x. Model III: DOS+ 3.4, MultiDOS, NewDOS 80, TRSDOS 1.3; Model 4/4P: MultiDOS, DOS+ 4, TRSDOS 6., LSDOS 6.3; Max-80: LDOS 5.1. PCXZ supports single or double sided, 35, 40 and 80 track formats. **Requires:** PC, XT, AT or compatible, Tandy 1000 (1000EX needs DMA), 1200, 3000. **You must have at least one 5-1/4" 360K, 720K or 1.2M drive and 256K memory.** An original program by Hypersoft.
Not for PS/2s: Order # PCXZ $79.95

Also may we recommend for your PC:

XenoCopy II runs on your PC and lets you read, write and format approx. 300 different CP/M, CoCo, P-System and other formats.
Order # Xeno $81.95

Uniform-PC runs on your PC and lets you read, write and format approx. 200 different CP/M and MS-DOS formats. Supports Matchpoint, and Compaticard (see below). **Order # UFPC** $69.95

Matchpoint-PC reads Apple-II Disks on your PC. Includes a half-size card that plugs in your PC plus software. **Reads Apple DOS, PRODOS, SOS, CP/M,** and over 200 CP/M formats including hard sectored types like NorthStar. Includes Uniform-PC. **Order # MPPC** $195.00

Matchmaker - reads MACINTOSH disks! a half size card plugs in your PC and, together with the supplied software, reads, writes and initializes MAC disks. Requires an external MAC drive.
Order # PCMAC $149.00

COMPATICARD disk controller card lets you attach 3.5" (720K or 1.44 Meg), 5.25" (360K, 720K or 1.2 Meg), and 8" disk drives to your PC, XT or AT. Control up to 16 drives with 4 Compaticards. May require Uniform-PC. **Order # CCRD** $175.00

UniDOS Z-80 CP/M card installs in your PC and lets you run CP/M programs on its built in 8 MHz Z80. Includes a free copy of Uniform-PC to transfer your old CP/M programs. **Order # UZ80** $175.00

TRS-80 Corner.

-=-

HyperCross reads CP/M and PC-DOS on TRS-80s

Using **HYPERCROSS** 3.0 you can **COPY** files between TRS-80 disks and those from many **CP/M** and **IBM-PC** type computers on your own TRS-80 Model I, III, 4/4P or Max-80. If you have access to more than one kind of computer, or you are changing to a new machine then you need **HYPERCROSS** to transfer your files. You can FORMAT alien disks, read their directories, copy files to and from them, even copy directly from one alien disk to another.
Formats supported: IBM-PC and MS-DOS including DOS 1.1, 2.0-3.2 Tandy 2000, single and double sided, 3.5 and 5 inch. CP/M from Aardvark to Zorba, including all popular TRS80 CP/M formats such as Holmes, Montezuma, and Omikron. Also supports CoCo format.
HyperCross converts Basic files! HyperCross will automatically convert tokenized Basic file to MSDOS or CP/M as it copies them.
Tried and Tested in 1000s of installations world wide, by Industry, Universities, Government Institutions and nice TRS-80 owners everywhere. Prices include disk and 40 page manual.
HyperCross 2.0 CoCo reads CoCo format (No Basic convert) **Order SX2CCM1, SX2CCM3 or SX2CCM4** $49.95
HyperCross 3.0 PC reads popular MSDOS 1.1-3.2 formats **Order SX3PCM1, SX3PCM3 or SX3PCM4** $49.95
HyperCross XT/3.0 reads 90 different CP/M and PC formats **Order SX3XTM1, SX3XTM3 or SX3XTM4** $89.95
HyperCross XT/3.0-Plus. Reads over 220 formats inc CoCo **Order SX3XTM1+, SX3XTM3+ or SX3XTM4+** $129.95
Specify TRS-80 Model I (needs doubler), III, 4/4P or MAX-80. Dual model versions e.g. Mod 3/4 on one disk add $10 extra.

Amazing HYPERZAP 3.2G Disk Magic!

Do you want to back up your precious copy of Copycat 3, or SU. Do you want to fix or modify a disk - if so then **you need HYPERZAP!**. More than just another disk copying program - it is **the** program for analyzing, copying, repairing, creating floppy disks of all kinds. It works with TRS-80 formats as well as many others such as CP/M, PC, CoCo etc. Designed to handle mixed density sectors on any track in any sequence. Many features for reading, writing, editing track and sector data. **Hyperzap is the tool** that lets you be in charge. **Make your own self booting disks.** Take your own CMD file and turn it into a dual booting Mod I/III/IV disk. **Autopilot mode** learns, saves and repeats procedures. Disk comes with fascinating examples. Use Hyperzap as a learning tool, find out how things are done!.
HYPERZAP 3.2G - nothing else even comes close! **Order # HZ32** - one version runs on all Model I/III/4/4Ps $49.95

VISA

HYPERSOFT
PO Box 51155, Raleigh, NC 27609

MasterCard

Orders: 919 847-4779 8am-6pm, Orders/Tech Support: 919-846-1637 6-11pm EST.
We Accept: MasterCard, Visa, COD, Checks, POs. Shipping: S2, $5 2nd day.

Many of the product names mentioned above are Registered or Copyrighted by Alcor, IBM, Misosys, Prosoft, Tandy and others too numerous to mention.

The Hacker's group is made up of dedicated TRS-80 enthusiasts from the Greater Los Angeles area. They come from varied backgrounds, have different skills, interests and even different ideas about what to do with their machines. But they have one important thing in common: They are willing and eager to share their knowledge.

So if you have a question about Model 3 or 4,

Ask the Hackers.

Q. My current problems are on the subject of telecommunications, specifically errors encountered in downloading programs.

After having downloaded an archived file, I encounter error messages such as 'FAILED CRC CHECK' during the extraction of the files from the archive. If the error was on a /CMD file and I try to run it, I get error messages such as 'LOAD FILE FORMAT ERROR' or RECORD NUMBER OUT OF RANGE'. During the downloading of the file I do not encounter any error messages or any indication that I have had problems during the downloading process. These are rare occurances, maybe 3 times out of 100.

My question is: Where is the problem? Is it on my end (hardware/software), is it with the BBS (GEnie/CIS), or is it with the download source (hardware/software)?

Harvey Hall
Parkton, MD.

A. The problems you are having with your files could stem from one of many that could possibly occur when transfering data through the phone lines.

As you know, XMODEM protocol is the popular method of file transfer in the computer world. Xmodem does not begin the transfer of data until the receiving computer signals that it is ready to receive the data. The receiving computer sends a Negative Acknowledge (NAK) character and this signals the sending computer to start transmitting.

After a NAK is received, the transmitting computer uses a Start of Header (SOH) character and two block numbers to signal the start of a 128-byte block of data. It then sends an error checking value (check sum) that is calculated by adding the ASCII values of each character in the block. This sum is then divided by 255 and the remainder is retained as the check sum. After each block is sent, the receiving computer computes it's own check sum and compares it with what was sent.

If the two values are the same, the receiving computer sends an Acknowledge (ACK) character to let the sending computer know that it's ok to send the next block. If the values are not the same, the receiving computer sends a NAK to signal that the block needs to be sent again.

Xmodem is by no means perfect. Line noise can sometimes fool the receiving computer that the block was sent ok when in fact it isn't. It doesn't happen too often but it can and will.

In the case of a "perfect" download not working, a problem could also happen when a program is being uploaded. If this is a constant problem with the same file, alert the SYSOP about the problem. In the case of GEnie or any other on-line service, alert the person who uploaded the program as well. Without feedback, the powers that be have no way of correcting the problem.

If the problem is occuring exclusively in ARC files (files created with ARC4/CMD) there is a possibility that the ARC was created with version 01.01.00. The original version removed line feeds after each carraige return. This was done because David Huelsmann thought that the program would not be used on anything except ACSII text files. Since line feeds are inherant in MS-DOS files, these were removed to make the file compatible with the TRS-80. The older version of ARC actually added a line feed to make the file compatible with MS-DOS when an ARChive was created (it was then removed when the file was un-ARC'd). Since ARC4 has not been used just for text, this began to cause some problems so the process was reversed in version 01.02.00. The following command line should solve your problems:

ARC4 EN filename:x $/$:x (ENTER).

My other suggestion would be to use XARC4/CMD to do ALL file extractions. The correction is automatically made under XARC4. The following command will do the trick:

XARC4 O:x filename:x (ENTER).

Remember that ARC4 and XARC4 assume /ARC to be the extention. Good luck.

Q. I can't seem to get the COPY function in PFS-file to work. The manual tells me to place the source disk in one drive and the destination disk in another. Since I have only two drives I have to remove the DOS disk, and when I try to copy, the program crashes and my Model 4 hangs up. HELP

R.D. Mitchell
Mobile, AL

A. PFS-file uses DOS SVC's to perform a copy. Unfortunately the DOS copy routine is placed in an overlay on the disk. It must access this code and, since you removed the DOS disk, it cannot.

Immediately before starting PFS type this:

```
SYSTEM (SYSRES = 1) < ENTER >
SYSTEM (SYSRES = 2) < ENTER >
SYSTEM (SYSRES = 3) < ENTER >
SYSTEM (SYSRES = 10) < ENTER >
SYSTEM (SYSRES = 12) < ENTER >
```

The copy function will now do what you want.

CLOSE#2

I've never experienced two months pass as quickly as did January and February this year. It seems as if I just finished with issue 1, and now it is time to mail out #2. Time does fly when you are having fun.

And fun it has been. The letters, phonecalls and the spirit of comraderie that now seem to exist between TRS-80 owners are certainly encouraging signs that our machines will be around for some time to come.

Most of you who communicated with me over the last couple of months indicated either, that you have no intensions of switching over to a PC, or that you own both machines and you prefer the TRS-80. On a personal note, I agree wholeheartedly. I fall in the latter category as, along with Model III 's and 4's, I also own a PC clone. Sure, the PC is a good machine and it can do a lot of 'neat stuff' and I use it; but when I want to work, program, explore, play, or just relax, I turn on my Model 4, and the same feeling I got when I brought home my first Model I returns: EXCITEMENT. The PC is a cold, impersonal machine, the TRS-80 is family.

Issue #2 welcomes the writings of Scott McBurney and Roy Beck. Scott is a very talented college student from Illinois. His article 'Teaching and old DOS new tricks' should be of interest to all Model 4 owners. Try it, you'll like it. Roy Beck is a truly amazing man. When asked to write about CP/M he inquired how long the article should be. I told him that a couple of pages would be great, and within 48 hours he uploaded EXACTLY two pages on CP/M to me. Now if I could, somehow, clone Roy's enthusiasm, knowledge, exactness and punctuality, TRSTimes could be a 'daily'.

Tim continues the PD Express by talking about Shareware, and SHELL in particular. Eric Bagai graces us with a tutorial on 'The lost powers of Scripsit' and includes a handy, dandy quick reference guide. There are more Peeks and Pokes for Model 4, a type-in game and a CAT command for TRSDOS 1.3 from yours truly.

We were contacted by three well-known and respected TRS-80 houses. HYPERSOFT, MICRO-LABS and PROSOFT. They all pointed out that they still provide software and hardware support for our machines. Hyperzap, the Grafyx Solution, Allwrite, need I say more? Their ads appear elsewhere in the issue, so get in touch with them, place your order and tell 'em "you saw it in TRSTimes".

In the struggle to keep issue #2 down to 30 pages we had to make some compromises. Model III did not receive its due. The promised utility to copy to and from Newdos/80 WILL appear in the next issue, along with a program for LDOS that strips all passwords from a disk. Andy Levinson, of 80-Micro 'PATCH' fame, is working on something special for an upcoming issue, and the rest of the gang are at this time consulting with their respective inspirational muses.

We have been innundated with requests for a 'TRSTimes on Disk' and we are seriously considering it, but have no firm plans at this time.

Publishing TRSTimes certainly has had its side benefits. Quite a few of the user groups graciously send copies of their newsletters and most contained a wealth of information. The one that really caught my eye came all the way from England. The national user group there puts out a VERY nice publication called NATGUG NEWS. I enjoyed reading about what is going on overseas and I recommend it. American subscriptions are welcome, and I have just learned that one year costs 15.00 pound sterling, or $22,50. It can be charged to your MasterCard. For further information on this, contact:

NATGUG - c/o Roger Storrs
Oakfield Lodge - Ram Hill
Coalpit Heath
Bristol. BS17 2TY England

The other newsletter that caught my fancy is called The REC Newsletter. REC stands for Recreational and Educational Computing and is published by Dr. Michael W. Ecker. 129 Carol Dr. Clarks Summit, PA. 18411. Annual subscription is $24.00.

It is not a TRS-80 specific publication, but it leans heavily on the mathematical side of computing. If you enjoy math puzzles, as I do, give it a try.

Until May, keep those TRS-80's humming.
Lance W.