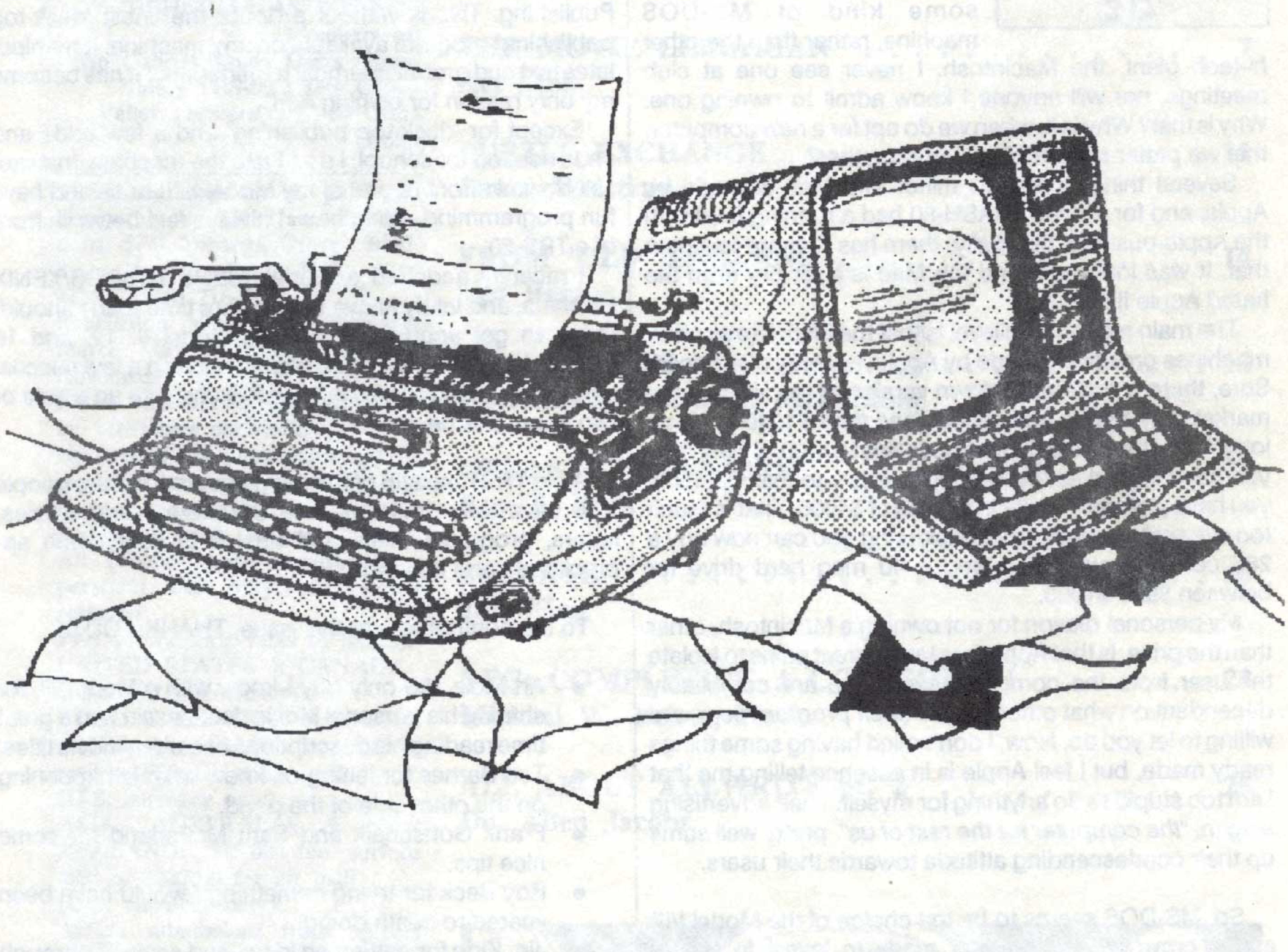
TRSTimes

Volume 4. No. 1. - Jan/Feb 1991 - $4.00



OUR FOURTH YEAR OF COVERING MODEL I/III & 4

LITTLE ORPHAN 80

Though I remain very optimistic about the near future of the TRS-80, the sad fact is that people are leaving our close-knit community every day in pursuit of the pleasures promised by the new generation of hi-tech machines. This is not news, nor should it by any means be unexpected.

What I find interesting about this exodus is that the new equipment is invariably always some kind of MS-DOS machine, rather than the other hi-tech giant, the Macintosh. I never see one at club meetings, nor will anyone I know admit to owning one. Why is that? Why is it, when we do opt for a new computer, that we prefer on machine over the other?

Several things come to mind. The Mac is made by Apple, and for a while TRASH-80 had a running feud with the Apple-pushers; but really, there has to more to it than that. It was long ago, and the Mac is a far cry from the hated Apple II series.

The main reason, I believe, is that the Mac, like all other machines previously made by Apple, is vastly overpriced. Sure, there is a stripped-down version of the Mac on the market now that is advertised '*less than $1000*". Yeah, lotsa luck - make that 5 cents less than $1000. By the time you have stuffed all the needed extras into this machine, you have probably spent in excess of $4000. That is much too expensive, especially considering you can now buy a 286 complete with VGA and a 40 meg hard drive for between $900-$1000.

My personal reason for not owning a Macintosh, other than the price, is that Apple has taken great pains to isolate the user from the computer itself. You are completely dependent on what other people (*their programmers*) are willing to let you do. Now, I don't mind having some things ready made, but I feel Apple is in essence telling me that I am too stupid to do anything for myself. Their advertising slogan, "*the computer for the rest of us*", pretty well sums up their condescending attitude towards their users.

So, MS-DOS seems to be the choice of the Model I/III and 4 crowd when they get ready to invest in a new computer. I consider that logical. After all, our TRS-80s are pure DOS machines and, for the most part, our users have learned the intricacies of the various operating systems available. If you know one DOS on a TRS-80, it doesn't take much new learning to fake it in a different operating system, even if it is MS-DOS. Also, many TRS-80 people are well versed in Basic. All TRS-80 Basics were written by Microsoft, so switching to GW-Basic presents little difficulty. The only real difference is the inclusion of color and graphics commands. However, if you've done any Color Computer or Hi-Res programming, these commands will be very familiar.

Am I writing this to encourage the readers to buy an MS-DOS machine? No, of course not. It is simply a fact that a certain amount of TRS-80 people leave us for a new machine. I have mentioned in previous issues that I also own a PC-clone. It is a very capable machine, and I use it to produce TRSTimes the very best I can. Now, I realize that I could use Allwrite or LeScript to print the pages in two-column format, and that would have been acceptable except that I encountered a PC program called Ventura Publishing. This is without a doubt the finest 'desk-top publishing' program available on any machine. It manipulates text and graphics almost to perfection. It has become my only reason for owning a PC.

Except for 'desk-top publishing' and a few odds and ends needed for school, I don't use the machine. Instead, I sit down in front of one of my Model I, III or 4s and have fun programming. Somehow, I always feel better in front of a TRS-80.

I recently acquired a Model 16B, a TRSDOS/XENIX machine, and lately I have spent more time than I should, trying to get acquainted with its Model II, 12, and 16 modes. I have had some setbacks, but also a few successes. When I learn more I will most likely write an article or two about this new toy.

A big TRSTimes tip of the hat goes out to all the people who have taken the time and trouble to send articles, letters, programs, hints and tips. It is very much appreciated - and needed.

To the contributors to this issue, THANK YOU to

- Art Molz, the only guy I know with a Model IP, for sharing his personal Movie data base. I had a great time reading his descriptions of certain movie titles.
- Ted Barnes for letting us know what is happening on the other side of the pond.
- Frank Gottschalk and Sam McFarland for some nice tips.
- Roy Beck for trying something I would have been scared to death doing.
- Jim King for getting an idea - and seeing it through (and for fixing my garage after my son, 'Mario Andretti Wolstrup' parked my car there at 50 mph.)
- Dr. Ecker for another installment of his fun Recreational & Educational Computing.
- Dr. Jacobs for an illuminating tutorial on Allwrite softkeys.

And now.......Welcome to TRSTimes 4.1

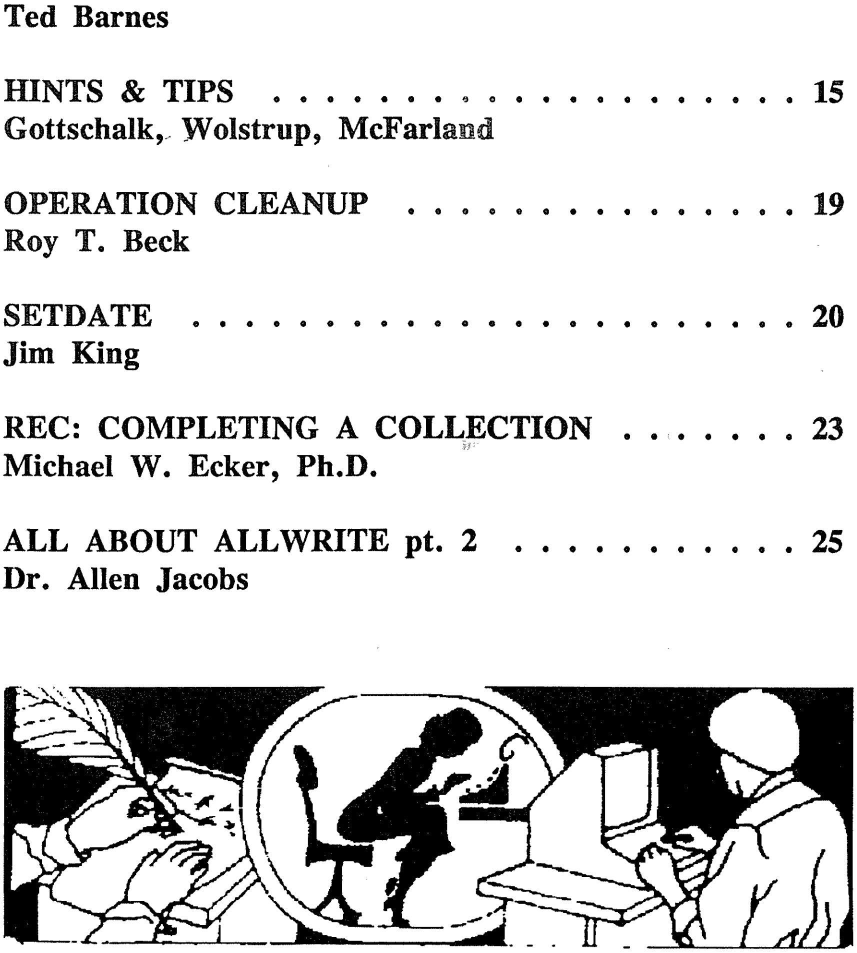
TRSTimes magazine

Volume 4. No. 1. - Jan/Feb 1991

PUBLISHER EDITOR
Lance Wolstrup
CONTRIBUTING EDITORS
Roy T. Beck
Dr. Allen Jacobs
Dr. Michael W. Ecker
TECHNICAL ASSISTANCE
Members of:
San Gabriel Tandy Users Group
Valley TRS-80 Users Group
Valley Hackers' TRS-80 Users Group

TRSTimes magazine is published bi-monthly by TRSTimes Publications. 5721 Topanga Canyon Blvd, Suite 4. Woodland Hills, CA. 91367. (818) 716-7154.
Publication months are January, March, May, July, September and November.

1991 subscription rates (6 issues):
UNITED STATES & CANADA: $18.00 (U.S. currency)
EUROPE, CENTRAL & SOUTH. AMERICA: $23.00 for surface mail or $29.00 for air mail.
(U.S. currency only)
ASIA, AUSTRALIA & NEW ZEALAND: $25.00 for surface mail or $$32.00 for air mail.
(U.S. currency only)
Article submissions from readers are welcomed and encouraged. Anything pertaining to the TRS-80 will be evaluated for possible publication. Please send hardcopy and, if at all possible, a disk with the material saved in ASCII format. Any disk format is acceptable, but please note on label which is used.



THE MAIL ROOM

XLR8er AND MODEL 4 POKES

As a recent subscriber, I want you to know that I am enjoying TRSTimes very much. But I have some problems with two programs you have published recently, and I cannot for the life of me figure out what is wrong. I am using a 128K Model 4 with XLR8er board (384K total RAM) and I simply cannot get your QUEEN4/BAS (TRSTimes 3.4 p. 23-24) and REVERSE4/BAS (TRSTimes 3.6 p. 24-25) to run on my machine. I am using LS-DOS 6.3.1 and Basic version 01.01.02 as enhanced by Roy Soltoff on the DOS 6.3.1 release. I have LISTed both these programs and have triple checked the LISTings against the copy in TRSTimes and I am unable to find any typing errors. (They may be there, but I'm darned if I can find them!)

With both programs, the computer simply locks up after the initial display and text appears on the screen. The cursor stops blinking and the computer will not do anything at all, requiring a reboot. With QUEEN4, I note that the number "3" appears in the graphic blocks between squares 52 and 53, if that is significant. With REVERSE4, I note that a small "u" with an umlaut (dieresis) appears at the lower right-hand corner of the screen and a capital "U" with an umlaut (dieresis) appears as the second "t" in the word "letters" in the line which reads "Arrange the letters in alphabetical order...". I am enclosing a diskette formatted in LS-DOS 6.3.1 containing my version of Basic together with QUEEN4/BAS and REVERSE4/BAS as I have entered them. I am also enclosing hardcopy printouts of the LISTings (from DOS) of both programs. Can you figure out what is wrong? Maybe there is something about the XLR8er board that interferes with the execution of the programs, although I cannot imagine what that might be. I do know that a Model 4 with the XLR8er board and its HD64180 CPU (replacing the Z80A) will not operate in the Model III mode (TRSDOS 5.3 will not run, in other words). I know nothing about programming, and not much about hardware, so I do not know if this makes any difference or not.

In any case, though, you can imagine how frustrating it is to spend the time it takes to input these programs and then not be able to get them to run. I appreciate your assistance.

W.I. Ramsey, Jr.
Gainesville, FL.

Sending a disk and hardcopy of the listings certainly narrowed down the problems, which are quite deceptive. Your version of the programs ran fine on my machine (almost). There were a couple of small errors - I will get to those in a minute.

First, however, your main problem with the programs is that they both disable the screen bank switching, leaving the screen intact starting at F800H. I am guessing that the XLR8er board (I don't have one) plays some kind of games with the memory from F800H and up, thus creating a major conflict with the non-banked screen which causes a crash.

The remedy is easy. In QUEEN4/BAS, do the following:

REMOVE line 3.
CHANGE 260 to read:

```
260 I$=INKEY$:IF I$="" THEN 260 ELSE IF I$="Y" OR
I$="y" THEN CLS:END ELSE PRINT CHR$(15);:
H=52:A$=CHR$(30):FOR V=9 TO 10:GOSUB 23:
NEXT:GOTO 230
```

Now, your errors in QUEEN4/BAS were also deceptive. When you run the game, your version destroys the screen display by putting the word "thinking....." into the chess board. Boy, this took me a while to find. What happened is that you changed A$ in line 340 from "TRS-80's move: " to "Computer's move: ". It is critical for the length of that text to be exactly 16 characters (yours is 17).

Also, in line 350, you have: 350 PRINT"Thinking.....";: etc. It is critical that only 3 dots follow (instead of your 5). Otherwise the text destroys the game board.

REVERSE4/BAS needs the following changes:

REMOVE line 4
CHANGE line 5 to:

```
5 DEFINT C-Z
```

CHANGE line 280 to:

```
280 IF I$="N" OR I$="n" THEN PRINT CHR$(15);:
V=10:H=42: A$=STRING$(3,32):GOSUB 23:
GOTO 170 ELSE IF I$="Q" OR I$="q" THEN CLS:END
```

The only other thing is a cosmetic change to line 370. Put a space before "moves". It looks better.

Hopefully, this takes care of all the problems. Enjoy the games.

Ed.

SUPERSCRIPSIT PRINTER CODES

In the Nov/Dec '90 Mail Room, regarding the question from Walt Danylak about SuperScripsit printer codes. You're right, everything is possible on the TRS-80.

Printer control codes can be sent to the printer from within a SuperScripsit document. Albeit not exactly at the same time that nothing can accept, but in immediate sequence that the printer can understand. The 27 you want to send is the "escape" code that tells the printer to "wake up, I want you to do something".

The 14 (or 15) you want to send, says "now print in 12 pitch (or 10 pitch).

In SuperScripsit you can do this with the use of the programed "Code Keys". These are programed with the < S > ystem setup from the main menu, then < C > odes. Actually 20 codes can be stored to include underlining, and bold printing. Select the key you want to program with the arrow key, let's say 2 for 12 pitch. Enter 27 (space) 14, then arrow over to description and enter something like: GO TO 12P if you like. After entering your codes for the number keys, ENTER and you will get sheet two for the SHIFT NUMBER SYMBOLS. Enter more codes if you like and ENTER again. This puts you back into the SYSTEM SETUP menu! BEWARE! It looks like the main menu. BREAK to get back to the MAIN MENU! Otherwise you will probably proceed to change something you don't want to, most likely the Opening Document menu.

These printer commands are then invoked wherever you want them with the CLEAR key followed by the number (or symbol) you want.

Actually I have them all programed to change pitch, print special characters, to a whole series of commands to actually print a vertical +/- sign in a daisywheel printer that does not have a +/- character on it. Hope this helps.

Frank Gottschalk
Fremont, CA

TRS-80, the EIP (everything is possible) machine. Thanks, Frank.

Ed.

REQUESTS

I would like to see a cross-word puzzle maker for the Model 4. I have one for the Color Computer and I think it is the best piece of software that I have seen for teaching slow learners. I do not know the algorithm for making the puzzle generator. I have your word search maker, but a cross-word puzzle maker would add a nice touch to the Model 4 library.

Orest Kowalshyn
West Fernie, B.C. Canada

A cross-word puzzle maker sounds interesting. How about it out there in TRS-land. Would anyone like to tackle this project?

Ed.

MORE RUMMY BUZZARD

The "Rummy Buzzard" seems to have created quite a stir! I wonder that none of the fellows digging around inside the guts of machine code has ever brought it up before. The message is all there, including the "JOE", if you just LIST the HERZ50/BLD file, quite clear if you list in ASCII (which is my default, thanks to Andy's patches). In the hex listing, the repeats and the missing bytes are quite obvious, even to a novice like me.. And it's on a disk I bought in '83. I only got curious when I noticed the different number of records, looking closer at that Feb. 20 disk. Missed the difference completely on my Scripsit disk. Not being in Europe, I had deleted it right away on all my working disks. I think you are dead on in wondering what used to be in these 32 bytes! Could have been quite a bit, and interesting!

Henry Herrdegen
Windsor, Ontario, Canada

Finding obscure messages in the operating systems is pretty useless ---- but so much fun. Found another one, so stay tuned.

Ed.

PATCHES

Many thanks for setting me on the right path to patching both LS-DOS and LDOS. I am sure glad that I have held off buying 6.3.1. With so many patches and typos in some patches it would be a mess.

Since the beginning, I have been a subscriber to TMQ and always thought that the spacing between hex digits was there for readability ONLY. I can not recall ever having read that the spacing is part of the patch syntax. After you pointed it out, I easily found it in the old LDOS manual. What I dislike about /FIX files is that not much is seen on the screen. I will try and use one- or two line /JCL files. At least, if I make a typo in the F-part of a line I can go back to DOS before too much has been "farkled".

I notice that you are a member of a "hackers'" club, for which I envy you. Could you perhaps explain in very general terms how a "dongle" works? I know that they are not a problem in the TRS-80 world, but I wish I had the advertising dollars that Byte magazine rakes in from them. I am wondering how these "marvels" of the computer world operate.

Willi Wald
Hamilton, Ontario, Canada

I must bite the bullet and tell you that the patch typos to SPOOL1/FIX, which I reported in the HINTS & TIPS section in issue 3.6, are not typos at all. I neglected to install the patch from FIX631B/JCL. Installing this patch does bring the bytes in SYS8/SYS to the positions described in Roy Soltoff's SPOOL1/FIX. My fault.

I have compiled all official patches to LS-DOS 6.3.1 and they are listed in the HINTS & TIPS section of this issue. You really should upgrade to 6.3.1. It is now the "standard" DOS for the Model 4. To Roy Soltoff's credit, he fixes things immediately when bugs are reported, not like the old Model I days where Radio Shack let months go by before even admitting something was wrong.

Remember that? LS-DOS 6.3.1 is without doubt the finest, most sophisticated operating system that we have ever had. I have just played with my new Model 16B (which also runs in Model II and 12 mode). There I experienced a whole new generation of TRSDOS'. Not until I got the Model II/12 version of LS-DOS 6.3.1 was the machine worth running. This DOS beat the others hands down. But that's an article in itself, and for another time.

The hackers' club I belong to is the Valley Hackers' TRS-80 Users Group. We are hackers, not the new type who destroy things, but of the old school who enjoy disassembling programs, make hardware modifications, etc. On a given meeting, dignitaries such as Roy Beck, Dr. Allen Jacobs, Eric Bagai, Tim Sewell and Andy Levinson (whenever he is not out of town) are present. Makes for good meetings.

"DONGLES!! Not only can I not explain in general terms what it is, I have never even heard of it. Can you enlighten me?

Ed.

TRS-80 BBS

The AZ TRS-80 Connection BBS.
Runs on Model 4 with a 40 Meg Hard Drive.
Since October '89 with FastPlus
(602) 968-2078
Steve Groce - Sysop

ADDRESS CORRECTION

In your answer to to a letter on page 5 of the Nov/Dec issue you gave an out-of-date address for Mel Patrick. Here is his new address which was on a parcel I received from him in the mail just yesterday.

MEL PATRICK
8056 - 164A STREET
SURREY, BRITISH COLUMBIA
CANADA V3S 7S7

Larry Rossiter
Victoria, BC, Canada

Thanks for the update. Now we all know how to get a hold of the latest version of FAST PLUS and all the other great stuff that Mel has written.

Ed.

HELLO TO ALL!

Let me introduce myself as a TRS-80 diehard. I have a Model I with two 40-, one 80 track DSDD, and a five-meg hard disk, running Newdos 2.5. I also have a Model III with two 40's and a five meg HD, as well as a Model 4 with two drives and a five meg HD, and two LNW-80's with expansions. I'm using them for amateur radio, cad, school homework and other projects. I was an employee of the Tandy Corporation from the late seventies through the early eighties.

I fell in love with the model I at first sight. I have heard a lot of war-stories and experienced a lot of headaches trying to make the model I, III and 4 play, but at this stage of the game I feel that I am the victor.

My machines are now doing all the things Tandy promised could be done, but were never able to deliver. I've spent thousands of hours in and out of the software and hardware to get there, but since I enjoy that sort of thing, it's been a blast!

At this point I would like to say how much TRSTimes has meant to me. I found out quite by accident about the magazine, and now it's time for me to participate.

As an employee of Tandy, I was in the unusual position to be able to order service manuals for much of the Tandy computer products out at the time. I have copies available for FREE to persons wanting to keep their machines going, or to modify them as I have done. I also have a lot of non-RS accessory information. Call or write me regarding those matters.

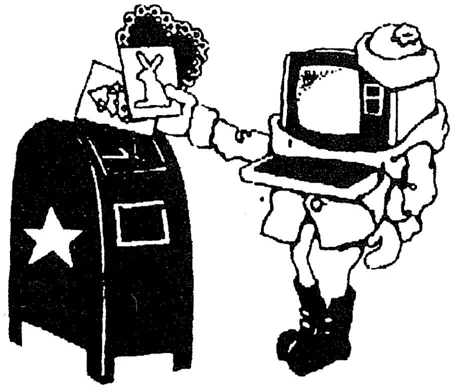
I would like to mention a good friend whose BBS is available at (415) 689-8952. Jim "Strappe" Steuart has a model I, LNW expansion, a 40 track DD, two Eight Inch drives, LNW 5/8 and an AT motherboard as a ramdisk, running the "Yet another BBS". He has a lot of TRS-80 programs which you may download.

Feel free to write or call me about TRS-80 and amateur radio matters. I hope you enjoy my article about the unassigned memory board which I give credit to Jim. It was his idea which I've carried over in hardware to all my machines. If things work out, I'll submit a new project each issue. Write me with your ideas and we'll have some fun!

Skip May
P.O. Box 192
Elmira, Calif. 95625 - 0192
(707) 446-3419

We appreciate Skip's generousity. The 'Unassigned Memory Board' article will appear in the next issue.

Ed.



THE MOVIE LIBRARIAN

Model I/III/4(III) - MULTIDOS

(LDOS, DOSPLUS, TRSDOS 1.3, and NEWDOS/80 v2. with changes)

By Art Molz

I am a movie buff and have over the years accumulated a large collection of my favorites on tape. Keeping track of where to find a particular movie was fast becoming a problem; but, also being a TRS-80 enthusiast, I fired up my Model I and wrote MOVIES/BAS to solve my need.

MOVIES/BAS will handle tapes numbered 000 to 999. If you mark your tapes in sequence, the program will keep track of how many tapes you have, the last tape used, and your total movie records. It will also provide a numeric and/or alphabetically sorted listing of your movies, either to the screen or to the printer.

Since the program is written in Basic and does a lot of string manipulation, it runs best when used in conjunction with TM (TRASH MAN). It also needs a lot of memory, so it is imperative to only allocate 2 buffers when entering Basic to run the program.

As written, MOVIE/BAS will work on Model I and III Multidos only, because the sort employs the special CMD"Q" routine found only in this DOS.

MOVIE/BAS manages three DATA files: MOVIES/NUM, MOVIES/ALP and MOVIES/INS. Before typing in the main program, we need to create these three files.

First we'll create MOVIES/NUM and MOVIES/ALP. Get into Basic, enter the following code and save it to disk as FILCREAT/MOV.

(Note: be sure to add a drive number to MOVIES/NUM and MOVIES/ALP in line 10 and line 70 if you want these files written to a drive other than drive :0).

```
1 'FILCREAT/MOV
2 'creates MOVIES/NUM and MOVIES/ALP
3 'for use with MOVIES/BAS
4 '
10 OPEN"O",1,"MOVIES/NUM"
20 PRINT#1,"1"
30 PRINT#1,CHR$(34)"001 DUMMY"CHR$(34)
40 CLOSE
50 OPEN"O",1,"MOVIES/ALP"
60 PRINT#1,"1"
70 PRINT#1,CHR$(34)"001 DUMMY"CHR$(34)
80 CLOSE
```

Since the program also feature on-line instructions, we need to write the instruction file. This is NOT done in Basic, but rather from a word-processor. I used PSCRIPT (SCRIPSIT), but any other is fine as long as it is capable of writing straight ASCII text to disk.

Load your word-processor and enter the following text. Be sure to press <ENTER> at the end of each video line. After completing the document, save it in ASCII format. In the case of SCRIPSIT, you would press:

<BREAK> **S A MOVIES/INS** <ENTER>

```
 V.C.R.  MOVIE PROGRAM INSTRUCTIONS. 05/10/87
This is a program to keep track of your VCR MOVIES.
It will hold up to 999 records maximum.
To use you should attach a sticker to each of your VCR
tapes
and put a number on them. This number should always
be three
( 3 ) characters. Example.. 001  023  079  189  321, all
are
legal numbers. 2,34,87 are not legal. Use 002,034, or
087.
You should then leave one blank space and then the
name of
the tape, including the main star. EXAMPLE:
023 THE MECHANIC, BRONSON
Maximum allowable length of each record is 57 charac-
ters.
You should use all capital letters so that everything
sorts
correctly.
PRESS any of the letter keys (F,L,D,C,) (after Loading),
and
the computer will prompt you for an input string. You
may
enter anything. To LIST the entire file, press "L" key.
-------------------------------------------------------
FIND: Press the "F" key.
-----------------------
Examples of input strings:
PARTON      will find all your DOLLY PARTON movies.
EASTWOOD    will find all your CLINT EASTWOOD
movies.
BRON        will find all your BRONSON movies.
ART         will also find DOLLY PARTON movies and
others.
-------------------------------------------------------
DELETE:  Press the "D" key.
--------------------------
This will prompt you for an input string to DELETE.
It will then display one record at a time with the search
string in it and let you decide which record you want to
DELETE or Abort the function.
-------------------------------------------------------
CHANGE:  Press the "C" key.
--------------------------
```

```
After the prompt the computer will display one record at
a
time and let you decide which record you want to
CHANGE.
----------------------------------------------------------
Be sure to SAVE your files after any ADD,DELETE, or
CHANGE.
You may print the file without loading it. Just press "P".
The files will be printed side by side. NUMERICAL - AL-
PHABET
on an Epson MX-80 or equivalent.
----------------------------------------------------------
You can print these INSTRUCTIONS by pressing the
"JKL"
(screen print) keys at the same time on each page or
your
Word Processor.
----------------------------------------------------------
Any time a "*" appears next to a command letter, that is
the
default condition. (Y/N*) = The ENTER key is the same
as NO.
----------------------------------------------------------
```

(this space represents 6 empty line - insert the 6 empty lines into the document, but don't type this text)

```
----------------------------------------------------------
This file was created with PSCRIPT (SCRIPSIT) so you
can add
any additional instructions you want. Just be sure to
press
the ENTER key at the end of each video line, SAVE the
file
in ASCII and leave a lower case "end" at the end.

END
end
```

Now, finally, type in the main program, MOVIES/BAS, save it to disk and then RUN"FILCREAT/MOV" to create the two needed data files.

If MOVIES/BAS is to be used with TRASH MAN - no changes are needed. Go back to DOS and type:
TM,BASIC 2,RUN"MOVIES/BAS <ENTER>

If MOVIES/BAS will be used without TRASH MAN you need to load"MOVIES/BAS" back into memory so you can change line 1 to:

```
1 GOTO 10
```

Now SAVE"MOVIES/BAS" back to disk, return to DOS and type:
BASIC 2,RUN"MOVIES/BAS" <ENTER>

You should now see the preliminary menu, telling you that the FILE HAS NOT BEEN ACCESSED YET, and giving you three options. The first, **L** (to load the data file), is the default. You can access it by pressing L or simply pressing <ENTER>. This option will gain entry to the main menu. The **P** option will print your movie listings on the printer. The **I** option will display the instruction file on the screen.

As this is the first time you are using the program, press **L** (or <ENTER>) to get to the main menu.

The MOVIES/NUM file will be loaded into memory, and the main menu appears. It consists of ten choices:

```
<A> Add movies to the file.      <C> Change a record.
<F> Find any movie in the file.  <P> Print all Movies.
<L> To LIST all records in file.   <I> For instructions.
<D> Delete any record in the file. <S> Save all files.
<Q> Quit and return to Multidos. <B> For BASIC.
```

Again, as this is the first time in the program, instead of <**A**> Add movies to the file, choose <**C**> Change a record, to enter your first movie in the file. You will be prompted to:

'Enter search string to CHANGE
<ENTER> to Abort

What we want to do is to replace the "001 DUMMY" record with your real first movie, so type **D** <ENTER>

The program answers by displaying:

Searching for... D
001 DUMMY
CHANGE. (Y/N*) <A> To Abort

Type **Y** and you will be prompted to 'Enter new record'. Type the three-digit number you have assigned to the cassette. Follow the number by a space (this is mandatory) and then type the name of the movie, followed by a comma, a space, and then the name of the star. Now press <ENTER>

The prompt 'CHANGE record (Y/N*)' appears. Type **Y**.

The record is now overwritten and you are again asked to 'Enter search string to CHANGE... <ENTER> to Abort'. This time press <ENTER> and the main menu re-appears.

You are now ready to use the <A> option to add the rest of your movies to the database. Pressing <**A**> displays a line 57 characters wide, with the cursor indicating that you should enter information. Enter all the titles you desire and, when done, just press <ENTER> at the empty line to abort to the main menu.

At this point make sure that you <**S**> Save all files. As a matter of fact, always save the files BEFORE you use the <L> or the <P> options.

Choosing <**L**> allows you to view the movies in either <A>lphabetical or <N>umerical * order. The asterisk indicates that this is the default choice and that <ENTER> may be pressed instead of the letter associated with the choice. Should you chose to view the files in alphabetical order without first having saved the data, all additions, changes, and deletions will be lost. Do NOT forget to save first.

Of course, you can use this little 'gotcha' to your advantage. If you have deleted one or more records in error and not yet saved the files, simply list alphabetically, and the record(s) will be restored.

<P> Print all Movies - prints the numerical and alphabetical listing side by side. Currently the print routine is for an Epson or compatible printer. Line 610 uses LPRINT CHR$(15) to force the Epson into compressed mode. For use with other printers, simply change CHR$(15) to whatever code forces the printer into compressed mode. For example, a Radio Shack DMP series printer should change LPRINT CHR$(15) to:

LPRINT CHR$(27);CHR$(21)

Again, before printing your files, make sure that the files are saved to disk.

<F> Find any Movie in the file - allows you to search the entire file for specific strings. You can be as exact or as vague as you wish. This feature uses the INSTR function so just about anything goes. Play with it. It is pretty powerful.

<D> Delete any record in file - prompts for:
Enter search string to DELETE
<ENTER> to Abort

The delete feature also uses the INSTR function so, as in <F>, you can be as specific or as vague as you wish.

Finally, pressing <Q> exits to Multidos, while pressing <B> exits to Basic.

USING MOVIES/BAS WITH OTHER OPERATING SYSTEMS

To run MOVIE/BAS from LDOS, DOSPLUS, TRSDOS 1.3, or NEWDOS/80 v2, change line 180 to:

180 PRINT" Resorting file....";:CMD"O",RN,A$(1)

Also change 'Multidos' in line 110 to 'LDOS','DOSPLUS', 'TRSDOS', or 'NEWDOS', whichever is appropriate.

After the changes have been made and saved to disk, return to DOS.

If LDOS and TRASH MAN, type: **TM** <ENTER>
BASIC (F=2) RUN"MOVIES/BAS" <ENTER>
Without TRASH MAN, type:
BASIC (F=2) RUN"MOVIES/BAS" <ENTER>

If DOSPLUS and TRASH MAN, type: **TM** <ENTER>
BASIC MOVIES/BAS -F:2 <ENTER>
Without TRASH MAN, type:
BASIC MOVIES/BAS -F:2 <ENTER>

If TRSDOS 1.3 and TRASH MAN, type: **TM** <ENTER>
BASIC <ENTER>
2V <ENTER> (to the HOW MANY FILES prompt)
<ENTER> (to the MEMORY SIZE prompt)
RUN"MOVIES/BAS" <ENTER>
Without TRASH MAN, skip TM <ENTER>, but type the rest of the above.

If NEWDOS/80 v2 and TRASH MAN, type:
TM <ENTER>
BASIC 2,RUN"MOVIES/BAS" <ENTER>
Without TRASH MAN, type:
BASIC 2,RUN"MOVIES/BAS <ENTER>

This should cover most of the available DOS'es, so let me end this article by saying that I have gotten a lot of use from this program and I hope you will find it as worthwhile as I have. Have fun with it.

MOVIES/BAS

```
1 DEFUSR=-578:X=USR(2600):
IFX< >0THENSTOPELSE10
2 I$=INKEY$:IFI$=""THEN2ELSERETURN
4 '**********************************************
5 ' MOVIES/BAS  - last update 11/11/90  by  Art Molz
6 '**********************************************
8 PRINT:PRINTTAB(42)"any key, <A> to ABORT.";:
GOSUB2:CLS:J1=0:RETURN
9 SAVE"MOVIES/BAS":STOP
10 CLEAR28200:DEFINTF-J:DIM I,J,J1,RN,F,K,K1,G
20 DIM I$,Q$,B$,T$,F$,D$,C$,F1$,F2$,FS$,M$,S$,Y$,
A$,A1$,A$(999),M$(4):Q$=CHR$(34):
Y$="(Y/N*)  <A> To Abort."
30 F1$="MOVIES/NUM":F2$="MOVIES/ALP":
F3$="MOVIES/INS":M$(1)="ADD":M$(2)="FIND":
M$(3)="DELETE":M$(4)="CHANGE"
40 M$="Enter search string to ":
S$="Searching for...  ":D$="DELETE.":
C$="CHANGE."
50 '------------- Print Main menu ------------
60 CLS:CLOSE:
PRINT"V C R   M O V I E   R E C O R D S.":
PRINT"by Art Molz"TAB(39)"MAKE SELECTION":PRINT
63 IFK1=1THEN80
66 IFRN< >0THEN90ELSEPRINT"FILE HAS NOT
BEEN ACCESSED YET.";
70 PRINTTAB(38)"*<L>  To LOAD the file.":
PRINTTAB(39)"<P>  To PRINT the files.":
PRINTTAB(39)"<I>  For INSTRUCTIONS.":
GOSUB2:IFI$="P"THEN590ELSEIFI$="I"THEN500
80 PRINT"Loading file...  "F1$" ... with      ( ";:
OPEN"I",1,F1$:INPUT#1,FS$:RN=VAL(FS$):
PRINTFS$" ) records.":FORI=1TORN:
INPUT#1,A$(I):NEXT:K=RN:K1=0:GOTO60
90 PRINT"<A>  Add movies to the file."TAB(40)
"<C>  Change a record.":PRINT"<F>  Find any
Movie in the file."TAB(40)"<P>  Print all Movies."
100 PRINT"<L>  To LIST all records in file."TAB(40)
"<I>  For INSTRUCTIONS"
```

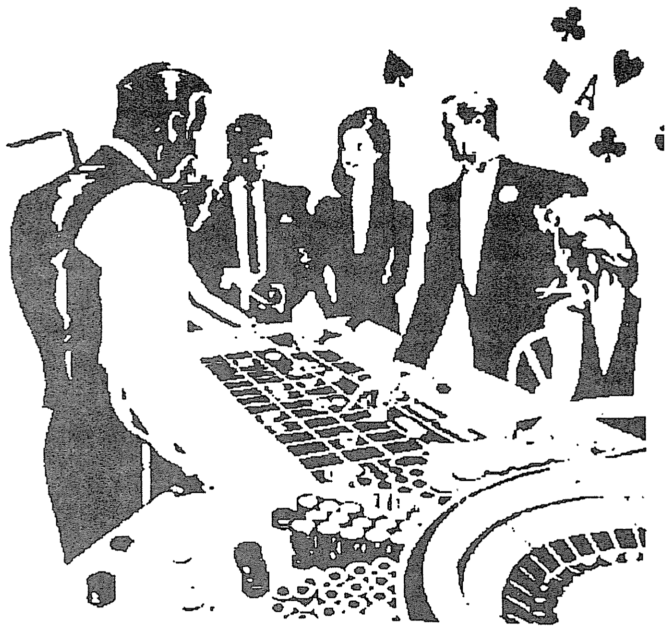
```
110 PRINT"<D>  Delete any record in file."TAB(40)
"<S>  Save all files.":PRINT"<Q>  Quit and return to
MultiDos."TAB(40)"<B>  For BASIC.";
120 GOSUB200
130 GOSUB2
140 ON1+INSTR("AFDQBCSPLI",I$)GOTO130,270,
    320,370,150,170,430,540,590,310,500
150 IFK<>RNTHENPRINT:PRINT"FILE LENGTH HAS
BEEN CHANGED. YOU NEED TO SAVE YOUR DATA.":
GOTO130
160 CMD"S" '------ Return to DOS --------------
170 CLEAR50:END'-- Return to BASIC -------------
180 PRINT"  Resorting file....";:CMD"Q",RN,A$(0)
190 PRINT@64*9,CHR$(31);
200 PRINT@64*9,"No. of records in file ="RN;TAB(40)
"Last tape # =  "LEFT$(A$(RN),3);:RETURN
210 PRINT@64*11,M$M$(F);:GOSUB220:
LINEINPUTT$:RETURN
220 PRINT@64*12,"<ENTER>  to Abort.":RETURN
230 PRINT"Input too long."TAB(55)"any key":
GOSUB2:RETURN
240 PRINT:PRINT"END of list... Any key":GOSUB2:
PRINT@64*12,STRING$(30," ");:RETURN
250 PRINT@64*10,CHR$(255)TAB(255)CHR$(255);:
GOSUB220:PRINT@64*11,STRING$(57,95);:RETURN
260 '-------------------- ADD ------------
270 GOSUB250:PRINT@64*11,"";:LINEINPUTT$:
IFLEN(T$)<5THEN60ELSEIFLEN(T$)>57THEN
GOSUB230:GOTO270ELSEPRINT"Checking for
DUPLICATES...."
280 FORI=1TORN:IFA$(I)<>T$THENNEXT:PRINT:
PRINT"Done... Press ENTER to add to list. <A> to
ABORT":GOSUB2:IFI$="A"THEN60
ELSEA$(RN+1)=T$:T$="":RN=RN+1:GOSUB180:
GOTO270
290 PRINT"DUPLICATE RECORD"TAB(50)"Any key":
GOSUB2:GOTO270
300 '-------------- FIND --------------------
303 PRINT"LOADING ALPHA LIST"
305 PRINT"Loading file...  "F2$" ... with      ( ";:
OPEN"I",1,F2$:INPUT#1,FS$:RN=VAL(FS$):
PRINTFS$" ) records.":FORI=1TORN:INPUT#1,A$(I):
NEXT:K=RN:GOTO330
310 T$=" ":CLS:PRINT"MAKE SELECTION:":PRINT:
PRINTTAB(10)"<A>lphabetical":PRINT:
PRINTTAB(10)"<N>umerical *":GOSUB2:
IFI$="A"THENK1=1:GOTO303ELSE330
320 F=2:GOSUB210:IFLEN(T$)<1THEN60ELSE
PRINT"<P> FOR THE PRINTER, OR <ENTER>.":
GOSUB2:CLS:IFI$="P"THENG=1ELSEG=0
330 J=0:J1=0:FORI=1TORN:
IFINSTR(A$(I),T$)THENPRINTA$(I):J=J+1:
J1=J1+1:IFG=1THENLPRINTA$(I)
340 IFJ1=13THENGOSUB8:IFI$="A"THEN60
ELSECLS
350 NEXT:PRINTTAB(22)"No. of matches ="J:
GOSUB240:GOTO40
360 '-------------- DELETE ------------------
```

```
370 F=3:GOSUB210:IFLEN(T$)<1THEN60ELSE
PRINT@64*12,S$T$;:FORI=1TORN
380
IFINSTR(A$(I),T$)THENPRINT@64*13,STRING$(33," ");:
PRINT@64*13,A$(I);:PRINT@64*14,D$Y$:GOSUB2:
IFI$="A"THEN60ELSEIFI$="Y"THEN400ELSE
PRINT@64*13,STRING$(63," ");:PRINT@64*12,S$;
390 NEXT:GOSUB240:GOTO370
400 IFI=RNTHENA$(I)="":GOTO410ELSE
A$(I)=A$(RN):A$(RN)=""
410 RN=RN-1:PRINT"Record has been DELETED...";:
T$="":GOSUB180:GOTO370
420 '-------------- CHANGE ------------------
430 F=4:GOSUB210:IFLEN(T$)<1THEN60ELSE
PRINT@64*12,S$T$;:FORI=1TORN
440 IFINSTR(A$(I),T$)THENPRINT@64*13,
STRING$(13," ");:PRINT@64*13,A$(I);:
PRINT@64*14,C$Y$:GOSUB2:IFI$="A"THEN60
ELSEIFI$="Y"THEN460ELSE
PRINT@64*13,CHR$(250);:PRINT@64*12,S$;
450 NEXT:GOSUB240:GOTO430
460 GOSUB250:PRINT@64*9,CHR$(255);:
PRINT@64*9,A$(I);:PRINT@64*10,"Enter new record.";
470 PRINT@64*11,STRING$(57,95);:PRINT@64*11,"";:
LINEINPUTT$:IFLEN(T$)<5THEN60ELSE
IFLEN(T$)>57THENGOSUB230:GOTO460
480 PRINT"CHANGE record (Y/N*)":GOSUB2:
IFI$<>"Y"THEN60ELSEA$(I)=T$:T$="":GOSUB180:
GOTO430
490 '------------- INSTRUCTIONS -------------
500 OPEN"I",1,F3$:J1=0:CLS
510 LINEINPUT#1,IN$:IFIN$="end"THENGOSUB8:
CLOSE:GOTO60
520 PRINTIN$:J1=J1+1:IFJ1<>13THEN510ELSE
GOSUB8:IFI$="A"THEN60ELSEGOTO510
530 '---- Saving both Numerical & alphabetical file ----
540 PRINT:GOSUB 180:PRINT:
PRINT"Saving to NUMERICAL file --> "F1$:
OPEN"O",1,F1$:PRINT#1,STR$(RN):FORI=1TORN:
PRINT#1,Q$A$(I)Q$:NEXT:CLOSE
550 PRINT"Rearranging file to ALPHABETIC"
560 FORI=1TORN:A$(I)=RIGHT$(A$(I),LEN(A$(I))-4)
 +LEFT$(A$(I),3):PRINT@754,I;:NEXT:PRINT:GOSUB180
570 PRINT:PRINT"Saving to file -> "F2$:
OPEN"O",1,F2$:PRINT#1,STR$(RN):FORI=1TORN:
PRINT#1,Q$RIGHT$(A$(I),3)" "LEFT$(A$(I),LEN(A$(I))-
3)Q$:NEXT:CLOSE:GOTO80
580 '------------- Lprint routine ------------
590 PRINT:PRINT"Make printer ready,    <A> to
ABORT,    any key.":GOSUB2:IFI$="A"THEN60
600 PRINT"Opening files.....":OPEN"I",1,"MOVIES/NUM":
OPEN"I",2,"MOVIES/ALP":INPUT#1,FS$:INPUT#2,FS$:
PRINT:PRINT"Printing files...."
610 LPRINTCHR$(15)"NUMERICAL"TAB(38)FS$
" VCR MOVIE RECORDS"TAB(67)"ALPHABETICAL"
TAB(122)LEFT$(TIME$,8):LPRINTSTRING$(130,"-")
620 IFEOF(1)THENCLOSE:GOTO60:ELSEINPUT#1,A$:
INPUT#2,B$:LPRINTA$TAB(66)B$:GOTO620
```

PUZZLE EXCHANGE

for Model 4 - Basic

By Lance Wolstrup



Another game! It seems that the only programs I manage to get done on the Model 4 these days are puzzle oriented games. Well, this one I had fun with.

The original version of EXCHANGE is called TWO DIAMONDS. This program, in turn, is an adaptation of a program called LUCAS' PROBLEM. Both were published in the very early editions of Kilobaud Microcomputing (the forerunner of 80 Micro).

The program listing to TWO DIAMONDS certainly had its problems; not only was it riddled with logic errors, but it was also a classic example of 'spaghetti Basic'. Now, don't get me wrong, I am not complaining about 'spaghetti Basic. My outlook on programming is *'if it works - it is good'*, no matter how it is written - and *'if it doesn't work - it is bad'*, no matter how elegant the code.

Unfortunately, this was spaghetti code - and it didn't work, so I spent the better part of an evening unraveling the mystery. Finally, about 3 am, it worked the way, I believe, it was intended to.

The concept of the game was good, but I was not crazy about the way it was implemented. Though written in fairly generic Basic, the program was not intended to run on a TRS-80; most likely it was written for, one or more of the weird CP/M machines of the 1970's. My main objection was, while a board-type game, the board was not clearly marked and, worst of all, after each move the new updated board would scroll onto the screen.

Well, being a sucker for board puzzles, I began to rewrite portions of the code. I still was not happy so, after sleeping for a few hours, I ended up writing a Model 4 version completely from scratch. In the process the name of the program was changed to EXCHANGE/BAS.

The program is a one-player puzzle game played on a board, which is divided into 45 squares (5x9). There are 45 markers (or board pieces), which are divided into 4 categories:

28 dashes, 8 solid white, 8 O-shaped, and 1 blank.

The solid white markers are placed on the left side of the board and the O-shaped markers are placed on the right side of the board. The blank marker is placed at the absolute middle of the board. The 28 dash markers are placed strategically, surrounding the solid white and O-shaped markers.

The object of the puzzle is to move the solid whites to the right side board, while also moving the O-shapes to the left side of the board.

Movement is made by exchanging a solid white's (or O-shape's) board position with that of the blank's board position. The squares containing a dash are 'dead' squares; that is, they cannot be exchanged, and they cannot be moved into. This leaves only vertical movement for the solid white and O-shaped markers. In order for one of these markers to exchange board position with the blank, it must be in the direct next vertical square. The only other way to perform the exchange with the blank is if the marker is in direct vertical position with the blank, but has one marker in between. It may then jump the marker to perform the exchange.

The game is won when all the solid whites are on the right side of the board and all the O-shaped markers are positioned on the left. It is possible to do this in 40-45 moves, which is a very good score. My best effort to date is 56 moves. I consider that decent. When you first attempt this puzzle you will most likely score in the 60-70 range, but don't let that get to you. With a little bit of practice you might just do better than me.

Program notes:

Lines 10-12 set up 3 strings which will, when put together, draw one row of 9 squares. A1$ holds the top of the 9 squares. A2$ holds the center of the squares. A3$ holds the bottom of the squares.

Line 13 erases the screen, displays the title and copyright, then jumps to the instructions in line 500. Upon return, the screen is erased from screen line 2 to the end of the display.

Line 14 draws the 5x9 board.

Line 15 prints the horizontal and vertical coordinate numbers on the screen, followed by printing the 'Moves:' message on the extreme right hand side of the screen. All of the above is a one-time shot. It is on the screen where we want it, so Basic no longer needs keep track of it. Thus, to free up the memory it used we CLEAR it out. Next the individual squares of the board is dimensioned into the B array. Finally, M1 is set to 999. M1 is the startup value for best score. It will be altered to reflect the best score you accomplish in a series of games.

Line 16 reads the startup position of the board markers (from the data in lines 17 & 18) into the B array. The board markers are then created and stored in A$, B$, C$ and D$.

Line 19 puts the vertical and horizontal coordinates of the blank marker into Y1 and X1 respectively. Moves are set to 0 (M) and we print that directly under 'Moves:' at the right side of the screen.

Lines 100-150 draws the markers in their correct squares on the board.

Lines 160-210 prompt for the vertical and horizontal coordinate of the piece you wish to exchange. Each prompt is error trapped.

Lines 220-250 checks is the exchange is legal. If not, variable F is set to 1.

Line 260 tests the value of F. If it is equal to 1 the program branches to line 390 where an error message will be displayed.

Line 270 is reached only if the exchange was legal. The number of moves is incremented by one and then displayed.

Lines 280-290 exchanges the board markers on the screen.

Line 300 updates the B array so it always contains the correct sequence of markers as displayed on the screen.

Lines 310-330 check if the markers are in the correct squares to win the game. If just one marker is not placed correctly then, obviously, the game is not won, and we need to exit the loop. This presents a sticky situation: most likely, this will happen before the loop ends naturally. In this case we are inside a nested loop and just jumping out, without finishing, is inviting disaster. Well, here we have no choice: As soon as a non-win marker is found, we have to exit the loops. The solution is to trick Basic into thinking that we have finished the loops naturally. We do this by setting both loop counters to their end values and then letting the NEXT:NEXT instructions do their job.

OK - fine, but once we are out of the loop, how do we know if the game is won or not? Notice in line 320, immediately after the Y and X loop counters were set to their end values, that variable FL was set to 1.

Line 330 exits the loops and then check the value of FL. If FL = 1 then we have exited the loops because of a no-win condition and we branch to line 160. On the other hand, if FL = 0, then all board markers are placed correctly for a win and the program continues by default to the win routine.

Line 340 is the win routine.

Line 350 prompt for another game.

Line 360 checks for the reply to the prompt and ends the program if so desired.

Lines 370-380 are reached if the reply was Y (or y). Here the pointer to the startup board is restored and the current best score is updated and displayed. The program is then sent back to line 16 where we begin a new game.

Lines 390-400 contain the 'Illegal move' error routine.

Lines 500-640 hold the game instructions.

EXCHANGE/BAS

```
0 'EXCHANGE/BAS
1 'another puzzle game for Model 4
2 '(c) copyright 1990 by Lance Wolstrup
3 'all rights reserved
4 '
10 A0$=CHR$(151)+STRING$(5,131)+CHR$(171):
FOR X=1 TO 9:A1$=A1$+A0$:NEXT
11 A0$=CHR$(149)+STRING$(5,32)+CHR$(170):
FOR X=1 TO 9:A2$=A2$+A0$:NEXT
12 A0$=CHR$(181)+STRING$(5,176)+CHR$(186):
FOR X=1 TO 9:A3$=A3$+A0$:NEXT
13 CLS:PRINT CHR$(15);:
PRINT@0,"TRSTimes presents:":
PRINT@36,"EXCHANGE":
PRINT@57,"(c) 1990 Lance Wolstrup";STRING$(80,131):
GOSUB 500:PRINT@(2,0),CHR$(31)
14 V=4:FOR X=1 TO 5:PRINT@(V,8),A1$:
V=V+1:PRINT@(V,8),A2$:
V=V+1:PRINT@(V,8),A3$:
V=V+1:NEXT
15 V=3:H=10:FOR X=1 TO 9:
PRINT@(V,H),X;:H=H+7:NEXT:
V=5:H=5:FOR X=1 TO 5:
PRINT@(V,H),X;:V=V+3:NEXT:
PRINT@(10,74),"Moves:";:
CLEAR:DIM B(5,9):M1=999
16 FOR Y=1 TO 5:FOR X=1 TO 9:
READ B(Y,X):NEXT:NEXT:
A$=STRING$(3,140):
B$=STRING$(3,191):
C$=STRING$(3,32):
D$=CHR$(183)+CHR$(179)+CHR$(187)
17 DATA 99,99,10,99,99,99,100,99,99,
        99,10,99,10,99,100,99,100,99,
        10,99,10,99,0
18 DATA 99,100,99,100,
        99,10,99,10,99,100,99,100,99,
        99,99,10,99,99,99,100,99,99
19 Y1=3:X1=5:M=0:
PRINT@(12,75),USING"###";M;
100 V=5:FOR Y=1 TO 5:H=10:
FOR X=1 TO 9
110 IF B(Y,X)=99 THEN PRINT@(V,H),A$;:
H=H+7
120 IF B(Y,X)=10 THEN PRINT@(V,H),B$;:
H=H+7
```

```
130 IF B(Y,X)=0 THEN PRINT@(V,H),C$;:
H=H+7
140 IF B(Y,X)=100 THEN PRINT@(V,H),D$;:
H=H+7
150 NEXT:V=V+3:NEXT:
PRINT@(3,0),D$;:PRINT@(3,77),B$;
160 PRINT@(20,20),CHR$(31);
"Enter your move - vertical coordinate: ";CHR$(14);
170 I$=INKEY$:IF I$="" THEN 170 ELSE Y=VAL(I$)
180 IF Y<1 OR Y>5 THEN 170
ELSE PRINT CHR$(24);Y;CHR$(15)
190 PRINT@(21,36),"horizontal coordinate: ";CHR$(14);
200 I$=INKEY$:IF I$="" THEN 200 ELSE X=VAL(I$)
210 IF X<1 OR X>9 THEN 200
ELSE PRINT CHR$(24);X;CHR$(15)
220 F=0:IF B(Y,X)=99 THEN F=1
230 IF ABS(X1-X)=0 THEN
IF (ABS(Y1-Y)< >1 OR ABS(Y1-Y)>4) THEN F=1
240 IF ABS(X1-X)=2 THEN IF ABS(Y1-Y)< >2
THEN F=1
250 IF ABS(X1-X)>2 THEN F=1
260 IF F THEN 390
270 M=M+1:PRINT@(12,75),USING"###";M;
280 PRINT@(2+Y*3,3+X*7),C$;
290 IF B(Y,X)=10 THEN
PRINT@(2+Y1*3,3+X1*7),B$;
ELSE PRINT@(2+Y1*3,3+X1*7),D$;
300 B(Y1,X1)=B(Y,X):B(Y,X)=0:X1=X:Y1=Y
310 F=0:FL=0:FOR Y=1 TO 5:FOR X=1 TO 4
320 IF B(Y,X)=10 OR B(3,5)< >0 OR B(Y,X)=0
THEN Y=5:X=4:FL=1
330 NEXT:NEXT:IF FL THEN 160
340 PRINT@(20,20),CHR$(31);" Congratulations - it
took you";M;" moves"
350 PRINT@(22,23),"Would you like another game
(Y/N) ";CHR$(14);
360 I$=INKEY$:IF I$="N" OR I$="n" THEN CLS:
END ELSE IF I$="Y" OR I$="y" THEN 370 ELSE 360
370 PRINT CHR$(15);:RESTORE:
PRINT@(5,75),"Best";:PRINT@(6,74),"score:";:
IF M<M1 THEN M1=M
380 PRINT@(7,75),USING"###";M1;:GOTO 16
390 PRINT@(23,25),"Illegal move - press <ENTER> ";
CHR$(14);
400 I$=INKEY$:IF I$< >CHR$(13) THEN 400
ELSE 160
500 PRINT@(2,0),CHR$(31)
510 PRINT"The game of EXCHANGE is a one-player
puzzle played on a 5x9 board."
520 PRINT:
PRINT"The board is laid out with the following markers:"
530 PRINT"28 dashes, 8 solid white pieces, 8 O-shaped
pieces and 1 empty piece."
540 PRINT:
PRINT"The object of the puzzle is to move the 8 solid
white pieces from the left side"
550 PRINT"to the right side, while moving the 8
O-shaped pieces from the left side to the"
560 PRINT"right side of the board."
570 PRINT:
PRINT"A piece can be moved in one of two ways:"
580 PRINT"If it is diagonally adjacent to the space,
exchange position with the space - or";
590 PRINT"if it is diagonally in line with the space, but
has another piece in the way, it";
600 PRINT"may jump diagonally over that one piece
and exchange positions with the space."
610 PRINT:
PRINT"You cannot move into squares with the dash
markers, and you can only move the"
620 PRINT"pieces diagonally.  Good luck!"
630 PRINT@(23,25),"Press <ENTER> to play
EXCHANGE";
640 IF INKEY$< >CHR$(13) THEN 640
ELSE RETURN
```

FROM 'OLD' ENGLAND

By Ted Barnes

Hello - this is Ted Barnes writing from Greater Manchester UK, with a few notes on what's happening over here with TRS-80.

Around the start of the 1980's, we Brits were beginning to think that you weren't ever going to forgive us for 'past history' but then, out of the blue we are allowed to share your wonderful TRS-80.

In 1980, no other domestic computer was so well supported in our cities and even local High Streets, for Tandy had and still has a network of small stores spanning the country.

It can hardly be said that, here in Cheadle Greater Manchester, I have my finger on the pulse of UK computing; however, I think I can judge the status of TRS-80 fairly accurately using two yardsticks.

The first is our national user group and its magazine NATGUG NEWS, and the second is our local user group, now eleven years old, the North West Computer Users Group - until last year the North West TRS-80 Computer Users Group. The group has its own magazine, the REM-90 (formerly REM-80).

Both user groups have recently shed the TRS-80 element from its title in favour of the wider context, in the hope of containing the shrinking membership. We shall see.

SURVIVAL

A couple of times each year, the national group holds a get-together weekend at Swindon, Wiltshire, the aim of which is to allow members from different parts of the England & Wales to meet, chat, swap etc.

Although I have not yet had the usual magazine account of the latest event, I have recently spoken to a friend who did attend, (Graeme Draper of Salford University - a very knowledgable Model I/III/4 chap), and he said that there was a preponderance of MS-DOS.

Realistically, it seems that the situation at present is that there are two main camps; one contains the very latest in micro technology and seems the largest (by being more vocal), whilst the other contains all the rest - and this is the crux of the matter.

'The rest' will always contain that which is not 'the latest' - and MS-DOS will soon join 'the rest', for it is sure to be replaced with more advanced technology. That is why 'the rest' should not be given up lightly - only when it becomes totally useless, compared with 'the rest'.

Loyalty, sentiment and resistance to change all seem to be playing a part in the survival of TRS-80 machines; in addition, we can not ignore the economic factors; my Models 4, 16 and 100 have all been bought second-hand - I certainly could not have afforded them new, although I did splash out on a new Model I (16k LII tape) when they arrived.

Yes - I do have a PC; an Amstrad PPC640D. It was the first affordable laptop on the market - something I desperately needed for my work as a peripatetic accountant. I would dearly have loved to stick with my familiar Model 4(III) using "Trader", "Time Recording System", "TRS Incomplete Records System", Visicalc, AIDS III, SuperScripsit, etc., but the sheer weight defeated me.

I can well do without an additional operating system, having already to cope with Digital's VAX VMS, and Convergent Technologies' CTOS at the office, and on the leisure side TRSDOS 2.3, 1.3, 2.0, 4.2, LDOS 5.3, 6.3, and CP/M. If the MS-DOS applications don't have the DOS utilities on a menu, then I'm not interested. However, in a hobby you can be labelled 'weird', but you can't afford that tag in business - hence the PC.

Apropos sentiment and loyalty, at our October NORWESCUG (local group) meeting, a young lady present for the first time approached me and said,"I've just got a machine like that (my model 4), and I'me using SuperScripsit, but I've no user guide - can you help me ?"

Naturally, I gave her some quick basic info and offered to lend her the docs. With someone so young, I can't help feeling that the least I SHOULD have done was to point her in the direction of Wordstar on the Model 4 - at least it would have been a bridge to her inevitable transition to a PC. Ah! well, - that's loyalty and sentiment for you.

DISKS ACCROSS THE OCEAN

We were recently able to help with a request from a (USA) TRSTimes reader to locate a particular piece of software. Although it seems unlikely that we should have items over here that you don't, I as an individual am most happy to locate and share software no longer available from suppliers. This would seem to be a most important factor in keeping alive our common interest.

To this end, I will gladly swap a list of my own software with anyone else's if they care to write to me; (Models I,II,III,4,16,100). The offer holds good for docs, mags, books and other info, although mail for such items might be a consideration.

Must go now - got to take 61904 for a walk. '61904'? - here's a clue, "7405 61453 is 2989 for you, especially 48879. You could be 57005 !" (a hex of a problem). Glad to hear from anyone anytime. Cheers.

Ted Barnes is the Model III/4/100/II/12/16 librarian for NORWESCUG. He can be reached at Depleach Road, Cheadle, Cheshire SK8 1DZ, England.

HINTS & TIPS

WILDCARDS & MORE ADDENDUM

By Frank Gottschalk

Just an addendum to WILDCARDS & MORE in Nov/Dec '90 issue.

Being the SLOW typer I am I like to save keystrokes. In LSDOS 6.3, closing) is not needed, nor is SYS, (S will do fine.

Instead of REMOVE SYS11/SYS.SYSTEM6:1

PURGE SYS11 (Q=

will do the same job.

As for backing up ONLY invisible files, this sequence would do it:

BACKUP $:0 :1 (I

PURGE $:1 (Q=

would leave only the invisible files on :1. It however would be slow as it would first copy all visible AND invisible files, then remove only the visible ones, leaving ONLY the invisible files on :1.

As for ONLY the system files

BACKUP /SYS:0 :1 (S

will do that.

An alternative to all this is to use SHELL. Have it display all files, invisible and system, then tag the ones you want and MASS BACKUP them to :1. Or MASS COPY them, but then will stop and ask you for the passwords for the protected files. I have heard of, but not used, programs to strip all passwords that would make this easier.

Yes, everything is possible on the TRS-80.

THE COMPLETE LIST OF OFFICIAL PATCHES TO LS-DOS 6.3.1 (so far)

Compiled by Lance Wolstrup

LS-DOS 6.3.1 is, as of this writing, up to level H. As there have been many requests to list the current patches, below find just that: a complete list of all official patches, as issued by Roy Soltoff in the Misosys Quarterly.

Patch #1 - from level A to level B.

```
.FIX631A/JCL - 03/08/90
.Cause FORMAT/CMD to write all sectors of DIR cyl
.Apply via, DO FIX631A (D=d)
.where "d" is drive to patch
//if -d
//.Must enter drive to patch!
//quit
//end
PATCH BOOT/SYS.SYSTEM6:#d# (D02,1F=42:F02,
1F =41)
PATCH FORMAT/CMD.UTILITY:#d# (D03,7F=21:F03,
7F=32)
//exit
```

Patch #2 - from level B to level C.

```
.FIX631B/jcl - 04/03/90
.add missing code to SETKI
.apply via, DO FIX631B (D=d)
//if -d
//.Must enter drive to patch!
//quit
//end
PATCH BOOT/SYS.SYSTEM6 (d02,1f=43:f02,1f=42)
PATCH SYS8/SYS.SYSTEM6:#d# SETKI1/FIX
//exit
```

```
.SETKI1/FIX - Adds missing code to SETKI
.Use with FIX631B/JCL
LB3
X'2583'=96
X'258E'=3E 60 EF 50 59 79 B7 C9
.Eop
```

Patch #3 - from level C to level D

```
.FIX631C/JCL - 04/18/90
```

```
.Minor correction to DIR
.Apply via, DO FIX631C (D=d)
//if -d
//.Must enter drive to patch!
//quit
//end
PATCH BOOT/SYS.SYSTEM6:#d# (D02,1F=44:F02,1F=43)
PATCH SYS6/SYS.SYSTEM6:#d# DIR1/FIX
//exit
```

```
.DIR1/FIX - 04/18/90
.Patch to LS-DOS 6.3.1 DIR command
.Corrects abort file with ext>4 and (Q=N)
.Apply via, DO FIX631C
D07,D3=DC 04;F07,D3=76 2D
D0B,64=DC 04;F0B,64=76 2D
D0B,9E=FD E5 CD 76 2D;F0B,9E=2B 3E 65 EF 7E
D0B,D1=C3 7C 2D 00;F0B,D1=FD CB 08 66
D10,0C=2B 3E 65 EF 7E C9 FD CB 08 66 FD E1 C3 4F 29;F10,0C="JanFebMarAprMay"
.Eop
```

Patch #4 - from level D to Level E

```
.FIX631D/JCL - 04/30/90
.Minor correction to DIR & MEMDISK/DCT
.Corrects exit code for DIR; BOOT/SYS
.& DIR/SYS passwords in Memdisk
.Apply via, DO FIX631D (D=d)
//if -d
//.Must enter drive to patch!
//quit
//end
PATCH SYS6/SYS.SYSTEM6:#d# DIR2/FIX
PATCH MEMDISK/DCT.UTILITY:#d# (D04,40=F4 71:F04,40=F6 37)
PATCH MEMDISK/DCT.UTILITY:#d# (D04,60=F4 71:F04,60=F6 37)
PATCH BOOT/SYS.SYSTEM6:#d# (D02,1F=45:F02,1F=44)
//exit
```

```
.DIR2/FIX - 04/27/90
.Corrects exit code of DIR in 6.3.1D
.Apply via PATCH SYS6/SYS.SYSTEM6 DIR2
D08,B6=21 00 00 C8 3A 2F 26 0C B9 30 EB
F08,B6=67 6F C8 3A 2F 26 0C B9 D2 32 24
.Eop
```

Patch #5 - from level E to level F

```
.FIX631E/JCL - 04/30/90
.Corrects release of banks  7 in SPOOL
.Apply via, DO FIX631E (D=d)
//if -d
//.Must enter drive to patch!
//quit
//end
PATCH SYS8/SYS.SYSTEM6:#d# SPOOL1/FIX
PATCH BOOT/SYS.SYSTEM6:#d# (D02,1F=46:F02,1F=45)
//exit
```

```
.SPOOL1/FIX - 04/30/90
.Corrects release of banks > 7
.Apply to SYS8/SYS.SYSTEM6
D1F,A5=21;F1F,A5=CA
D20,1E=1F;F20,1E=07
D20,24=CD EF 29;F20,24=32 8C 2A
D21,82=69 26 00 11 8C 2A C5 06 02 3E 5F EF C1 C9
F21,82=53 70 6F 6F 6C 65 72 20 61 6C 72 65 61 64
D22,24=78 20 66 72 65 65 64 20 20
F22,24=20 72 65 6C 65 61 73 65 64
```

Patch #6 - from level F to level G

```
.FIX631F/JCL - 07/16/90
.Corrects minor problem in 6.3.1F
.//KEYIN of JCL now accepts 79 cpl +CR
.instead of 79 cpl including CR
.DIR now displays header of disk formatted
.when DATE was not set
.DO * now finds a SYSTEM/JCL file on a drive
.other than :0
.Apply via, DO FIX631F (D=d)
//if -d
//.Must enter drive to patch!
//quit
//end
PATCH SYS11/SYS.SYSTEM6:#d# (D01,2A=50:F01,2A=4F)
PATCH SYS11/SYS.SYSTEM6:#d# (D01,CE=50:F01,CE=4F)
PATCH SYS6/SYS.SYSTEM6:#d# (D07,F3=7A:F07,F3=3C)
PATCH SYS6/SYS.SYSTEM6:#d# USING DO1/FIX
PATCH BOOT/SYS.SYSTEM6:#d# (D02,1F=47:F02,1F=46)
//exit
```

```
.DO1/FIX - 07/16/90
.PATCH TO LS-DOS 6.3.1 DO command
.Allows DO * to find a SYSTEM/JCL file on a drive
.other than :0
.Apply via, DO FIX631F
D2D,AB=CD AC 2A;F2D,AB=32 EB 29
D33,0B=03;F33,0B=3A
D33,AD="Bad JCL format, process aborted"
```

```
F33,AD="Invalid JCL format, processing "
D33,CC=0D 67 2E 3A 22 EA 29 C9
F33,CC=61 62 6F 72 74 65 64 0D
.Eop
```

Patch #7 - from level G to level H

```
.FIX631G/JCL - 08/27/90
.Corrects MEMORY command display when the
.information uses more than one screen
.Apply via, DO FIX631G (D=d)
//if -d
//.Must enter drive to patch!
//quit
//end
PATCH SYS6/SYS.SYSTEM6:#d# MEMORY1/FIX
PATCH BOOT/SYS.SYSTEM6:#d# (D02,1F=48:F02,
1F=47)
//exit
 .MEMORY1/FIX - 08/27/90
.PATCH TO LS-DOS 6.3.1 SYS6/SYS
.Patch cleans up video screen when data
.goes to 2nd screen
.Apply via, DO FIX631 (D=d)
D04,63=18;F04,63=17
D04,6E=7F;F04,6E=0C
D04,DB=15;F04,DB=0F
D04,F2=CD 9C 27 C3 6A 27;F04,F2="No mem"
D04,F8="No memory"
F04,F8="ory space"
.Eop
```

ANOTHER (SEMI) AUTOMATIC/BAK

By Frank Gottschalk

In response to (SEMI) AUTOMATIC/BAK in the Nov/Dec '90 issue, I do a similar thing, but a little different.

I have an external 80 track drive :3 that I use for archiving backups. I normally run from a 1 MEG RAMDRIVE as :0 for ultra fast speed, but it should work with any drives you use. I have the Key Stroke Multiplier (KSM) set up with two keys as follows. These are invoked with <CLEAR> & M or T.

<CLEAR> & M gives me "Mini-Terminate":
BACKUP $:0 :1 (MOD

This backs up any new or modified files from :0 to my "customer disk" in :1. I do this each time I finish a job for that customer. (The old version, if any is is still on my archive disk in :3.)

<CLEAR> & T gives me "Terminate":
BACKUP $:0 :3

I use this at lunch time (in case of power failure) and at the end of each day before turning off the system. This copies off all the latest versions over any previous versions, and saves any new files to the archive disk in :3.

If I want to save the old version, I copy it to RAMDRIVE :0 to a new name and proceed as usual. I also made all operating files and utilities other than new documents invisible so they don't clutter up the directory displays and won't waste time re-copying them over and over.

Actually, both these keys continue on to do other things like automatically bring up the Invoice program for updating customer invoices after "Mini-Term" and backing up the updated Invoice info to the boot disk so last info is available at next days boot up.

LS-DOS 6.3 TIPS

Model 4 and II/12/16

By Lance Wolstrup

When I purchased a copy of LS-DOS 6.3.1 for my recently obtained Model 16B, Roy Soltoff graciously sent along a copy of the LS-DOS 6.3.1 Update Documentation. It is now in a nice 38 page bound book, rather than the skimpy few pages I received when I upgraded the Model 4 from TRSDOS 6.2.

Towards the end of the booklet are a series of questions of answers, and here I found a very interesting item:

Question: *How can I make a bunch of blank, formatted disks, real quick?*

Answer: Format a blank disk in the normal fashion using the FORMAT command in LS-DOS 6.3. Now simply use DISKCOPY to make copies of this blank disk. DISKCOPY is very smart and only duplicates tracks than contain real data; blank tracks are just formated and verified. Try it, you will be amazed at the speed.

Every so often I will work in Model III mode using NEWDOS/80. Here I have gotten into the habit of using the comma as the delimiter in the DIR command, rather than the normal space. For example: **DIR,1**

Well, I had just finished my NEWDOS session and I switched back to LS-DOS 6.3.1. Forgetting that I was no longer in NEWDOS, I typed: **DIR,1** - AND IT WORKED.

Yes Virginia, you can use the comma as a delimiter in the LS-DOS 6.3 DIR - CAT and FREE commands.

As a matter of fact, you can use each one of the following symbols as substitute for the space between the command and the drive number:

DIR 1	DIR :1	space (normal)
DIR,1	DIR,:1	comma
DIR"1	DIR":1	quote
DIR#1	DIR#:1	number sign
DIR$1	DIR$:1	dollar sign
DIR%1	DIR%:1	percent sign

DIR&1	DIR&:1	ampersand
DIR'1	DIR':1	apostrophe
DIR(1	DIR(:1	left parens
DIR)1	DIR):1	right parens
DIR*1	DIR*:1	asterisk
DIR=1	DIR=:1	equals
DIR-1	DIR-:1	minus
DIR@1	DIR@:1	at sign
DIR‘1	DIR‘:1	shifted at
DIR+1	DIR+:1	plus
DIR;1	DIR;:1	semi colon
DIR<1	DIR<:1	less than
DIR>1	DIR>:1	greater than
DIR?1	DIR?:1	question mark

You can also use combinations of <CLEAR> and the arrow keys, as well as <CLEAR> and the function keys. All will act as a legitimate delimiting character.

The exclamation point, period and slash will not work, nor will combinations of <SHIFT><CLEAR> arrow keys, nor will <SHIFT><CLEAR> function keys.

You now have plenty to choose from, so pick one you like. Somehow, I still use the space. Old habits are hard to overcome.

IT IS COLD OUT THERE

(HOW COLD IS IT?)

Model I/III/4 and II/12/16

By Sam McFarland

Winter is here and it is cold. To find out how cold it REALLY is, knowing the temperature is not enough, we must also take the speed of the wind into consideration. The term 'wind-chill factor' is probably familiar to everyone, so here is a useful little BASIC program that computes the approximate wind chill temperature from a still air temperature and the wind speed.

To keep things simple, the program uses a table of known values for various temperatures and wind speeds. When you enter a temperature or wind speed that falls between entries in the table, the program simply extrapolates the wind chill temperature for the values entered.

```
10 DIM C(8,11)
20 FOR W=0 TO 8:FOR T=0 TO 11
30 READ C(W,T):NEXT T:NEXT W
31 DATA -60,-50,-40,-30,-20,-10,0,10,20,30,40,50
32 DATA -68,-57,-47,-36,-26,-15,-5,6,16,27,37,48
33 DATA -95,-83,-70,-58,-46,-33,-21,-9,4,16,28,40
34 DATA -112,-99,-85,-72,-58,-45,-36,-18,-5,11,22,36
35 DATA -124,-110,-96,-82,-67,-53,-39,-25,-10,3,18,32
36 DATA -133,-118,-104,-88,-74,-59,-44,-29,-15,0,16,30
37 DATA -140,-125,-109,-94,-79,-63,-48,-33,-18,-2,13,28
38 DATA -145,-129,-113,-98,-82,-67,-49,-35,-20,-4,11,27
39 DATA -148,-132,-116,-100,-85,-69,-53,-37,-21,-6,10,26
100 SW=80 ' Model I/III change to SW=64
110 IF SW=80 THEN TB=31 ELSE TB=23
120 CLS:PRINT TAB(TB)"W I N D  C H I L L"
130 PRINT:
PRINT"Temperature (degrees - F. 50 to -60): ";
140 INPUT T:T=INT(T)
150 IF T>50 OR T<-61 THEN PRINT"Temperature is
out of range!":GOTO 130
160 PRINT:PRINT"Wind speed (mph): ";
170 INPUT W
180 IF W<0 THEN PRINT"Wind speed cannot be
negative!":GOTO 160
190 T1=INT((T+60)/10)
200 IF W>40 THEN W=40
210 W1=INT(W/5)
220 A=C(W1,T1)
230 IF T/10=INT(T/10) THEN 360
240 X=C(W1,T1+1)-A
250 D=T/10-INT(T/10)
260 A=A+X*D
270 IF W/5=INT(W/5) THEN 400
280 A1=C(W1+1,T1)
290 X=C(W1+1,T1+1)-A1
300 D=T/10-INT(T/10)
310 A1=A1+X*D
320 D=W/5-INT(W/5)
330 X=A-A1
340 A=A-X*D
350 GOTO 400
360 IF W/5=INT(W/5) THEN 400
370 X=C(W1+1,T1)+A
380 D=W/5-INT(W/5)
390 A=A+X*D
400 A=INT(A)
410 PRINT:PRINT"Approximate wind chill
temperature: ";A;" degrees -F"
420 IF A<-25 THEN PRINT:PRINT"DANGER from
freezing of exposed flesh!"
430 PRINT:PRINT"Another (Y/N) ";
440 I$=INKEY$:IF I$="N" OR I$="n" THEN CLS:END
450 IF I$="Y" OR I$="y" THEN 120 ELSE 440
```

OPERATION CLEANUP

By Roy T. Beck

FOREWORD

This article contains some advertising matter, so don't say I didn't warn you!

AN OPPORTUNITY, (AND A PROBLEM)

Recently I purchased some used hard drives, mostly without bubbles, sight unseen from a fellow in another country. He had purchased them from a government surplus auction, and had removed most of the bubbles for resale to IBM types, leaving the boxes lonely and unwanted. The description made them attractive and the price was right, so I sent the money along. Some time later, the units showed up. Outwardly, they were fairly clean. But when I opened them up.... Talk about dirty!

The hard drives had formerly been used in army vehicle maintenance and repair depots. Having been in the army myself, I have some understanding of how the army operates. These units had obviously been regularly cleaned on the outside, but not internally. (After all, the inspecting officer can't get inside the unit, so why should a soldier clean the inside?) Since each unit has a small fan, any dust in the working area was attracted to the fan. While Radio Shack installed filters on the fans, the filters are not all that effective. Consequently, the units were filthy inside.

Of course, hard drive bubbles are inherently sealed against external contamination, so the units operated successfully even with a layer of dirt all over the inside of the boxes.

THE SOLUTION

Fortunately, Radio Shack mostly assembles their equipment with nuts and bolts, as opposed to rivets or welding, and this permits easy disassembly. Taking advantage of this, I began dismantling the units, and ended up with numerous piles of like pieces stacked around me. Now what? For the metal parts, the answer was easy. Run them through my wife's dishwasher! This removed all the grease and dirt, and air drying in the sunshine completed the task.

Next came the circuit boards. I had heard of circuit boards being washed the same way, so all the boards went into the dishwasher next. I only ran them through the Rinse and Hold cycle, not wanting to abuse them with long duration washing and drying cycles, especially the latter. You never saw such nice, clean boards as came out of the machine! The boards were also allowed to air dry in the sunshine. However, I drew the line at washing the switching power supplies, as I feared water in the transformers might do some permanent damage. Fortunately, the power supplies were not terribly dirty in the first place, so they were ready for reuse.

To reassure any skeptics, (and myself), I ran all the boards through an operational test cycle after their bath and all of them operated correctly, no problems.

After all the above, and inspection of all the parts, I began reassembly of the units. Many washers, nuts and bolts were missing, so I had to obtain replacements for these. Most of them were readily available at the local hardware store. Most of the cabinets were scratched and ugly, and while they were clean, they were not pretty. More scouting, this time for a suitable paint to refinish the boxes. I finally located a paint which was a good match for Radio Shacks original cream color, and this completed the hardware refurbishment task.

WHAT ABOUT BUBBLES?

The next problem was obtaining bubbles for the units, as most of the units were without bubbles when I received them. Fortunately, both new and used full height bubbles are becoming relatively plentiful, and I have been able to obtain some good new and used ones.

SOFTWARE

The next task was software. I purchased hard drivers from MISOSYS for use with LDOS and LS-DOS, and then set about writing JCL files to simplify installation of the drives on a buyer's existing system. Since I format all the drives before shipment, the JCL is tailored to match the partitioning of the drive it goes with. All the buyer has to do is bootup with a fresh copy of his LDOS or LS-DOS (TRSDOS), and call the JCL. The JCL will install the hard drive, backup his DOS to the hard drive, transfer control to the hard drive, and finally SYSGEN the system configuration onto the floppy disk. The floppy is now his bootup disk, and the job is done. Slick, I say!

In the event the buyer prefers different partitioning than the setup I supply, I include a legal copy of RSHARD to allow him to repartition and reformat as he pleases.

CONCLUSION

Considering the labor that goes into this project, I am not getting rich at it. So why do it at all? My reason is that I believe we TRS survivors have to stick together to keep our machines running, and hard drives are really the way to go for modern operation. Since these drives were available, and were in real danger of being junked, I felt it was really a DUTY to salvage them and get them into the hands of those who need them.

ADVERTISEMENT

The drives are available, in various sizes from 5 to 35 Meg. I would be pleased to make them available to any Model 3/4 owner for reasonable prices. See my ad elsewhere in this issue.

SETDATE

Model I/III & 4 MULTIDOS programming technique - BASIC

by Jim King, MSEE, C10

A few months ago I had the opportunity to get acquainted with a nice TRSDOS 6.2 machine language utility called SETDATE. (I believe this program was written by Jack Decker of 'The Alternate Source' fame.) Boy, this is the way to enter the date, much better than typing it in from the keyboard. In essence, the program presents the previous date entered, which is then brought to the present date by pressing the right arrow.

I thought that this would be a nice feature to have in my favorite operating system (MULTIDOS), so I decided to attempt writing a BASIC version of Mr. Decker's idea. Rather than have the program save the new date to a disk file, I wanted to have the program modify itself to retain the new date in a variable and then write the entire program back to disk, thus saving valuable disk space.

Well, this proved to be somewhat tougher than I anticipated! Let me explain what I found myself up against:

If line 10 is, for example:10 A$="Friday" and then somewhere later in the program A$ is changed to: A$="Saturday", when the program is saved back to disk, line 10 does not change. It remains 10 A$="Friday". This, of course, was the problem. I wanted line 10 to modify itself to read, for example: 10 A$="Saturday".

I knew that by using VARPTR and PEEK to find the address of the variable in question, and then POKEing a new value to it, should modify the code in the line where the variable first appears. However, I just could not get the program to work; no matter where I PEEKed and POKEd, the program either retained the initial value of my variable, or it would simply crash. At this point I decided to get help, so I brought the problem with me to next meeting of VTUG (Valley TRS-80 Users Group) where, after a lively discussion, the gang came up with the solution.

SETDATE/BAS, because it is written in BASIC is slower than the TRSDOS 6.2 machine language version, but it certainly does the job.

The program brings up the title, and the statement that it uses the DEFine FuNctions of Lewis Rosenfelder in his book Basic Faster & Better. These DEF FNs are from line 81 to 88. Line 89 is one of my own that takes a 1 or 2 digit integer and converts it to a 2 digit $tring.

Line 200 displays instructions on how to use it: Right or Left arrows to increase or decrease one day, > and < to increase or decrease 10 days. In addition to the number of the day, it displays the day of the week as a check.

The program works by storing a number that the FNs use to find that date, and the day of the week in A$. This happens in line 80.

The calculation is done in line 210 and 220 to print the day of the week and the date in line 230.

Lines 3 to 9 are subroutines that I use constantly:
3 prints Z centered.
7 changes a lower case string to upper case.
8 is a one character Inkey$, including a call to line 7.
9 causes the cursor to Jump Up 1+JU lines.

Lines 240 through 290 perform the addition to or subtraction from the date number, jumps to line 300 after you press <Enter> for the correct date, or cycle back to line 230 for any other key. If a new date is set by one of the arrows, flag K is set =1, and used later.

In line 300 the new value of the date number (A) is put into $tring AA and used after the time is entered. Time$ is displayed to show what the computer now thinks is the date and time. The time input is then requested in 24 hour clock, 4 digits.

The IF tests in lines 310 and 320 prevent entering illegal times.

The chosen 4 digit time is put into TI$ in line 330.

Line 340 peeks the address of the first date number, A$.

Line 350 peeks the address of the new date AA$.

Line 360 pokes into the address of A$ the number in AA$.

Note that the address of the new value is +1 because in line 300 STR$(A) includes the leading blank.

Line 370 takes the numbers for the month (MO), day (DY), and year (IR) found in line 210 and puts them into DA$ in the proper order. Then IF the date has been changed, and K=1, the whole program is saved, and the date in memory is corrected.

Line 380 saves the time, then again displays Time$ to show if it all took.

I incorporated this routine into a shell program that I use all the time. Now, instead of entering the date and time at boot-up, I set it from within the shell program.

(Editor's note: Jim has kindly submitted his handy shell program. It will be published in an upcoming issue.)

SETDATE/BAS

```
0 CLEAR 99:DEFINT B-J,L-Y:DEFSTR Z:
PRINT LEFT$(TIME$,8)" ";:
Z="SETDATE - Date Setting Using Routines from":
GOSUB 3: Z="BASIC FASTER AND BETTER & OTHER
MYSTERIES, IJG Inc.":GOSUB 3:
Z="Copyright (C) 1981, by Lewis Rosenfelder":
GOSUB 3:GOTO 80'-SetDate
3 PRINTTAB(38-LEN(Z)/2)Z:RETURN
7 FOR H=1 TO LEN(Z):
O=ASC(MID$(Z,H,1)):
IF O>96 THEN MID$(Z,H,1)=CHR$(O-32):
NEXT:RETURN
ELSENEXT:RETURN
ELSERETURN
8 Z=INKEY$:IFZ=""THEN 8
ELSE IF Z=CHR$(31) THEN END
ELSE GOSUB 7:RETURN
9 PRINT CHR$(29)STRING$(JU+1,27)CHR$(31);:
JU=0:RETURN
20 A=FNDN!(YR,MO,DY):ZA=STR$(A):
RETURN'date #
21 PRINT"Day #  ="A:RETURN
22 DA$=FNZ2(MO)+"/"+FNZ2(DY)+"/"+FNZ2(IR):
RETURN
23 TI$=FNN$(HR)+":"+FNN$(MN)+":00":RETURN
24 YR=FNRY%(A):J=FNRJ%(A):
MO=FNRM%(J,YR):DY=FNRD%(YR,MO,J):
RETURN  'not used
25 DA$=FNN$(MO)+"/"+FNN$(DY)+"/"+FNN$(IR):
RETURN  'was line22
30 K=PEEK(VARPTR(A$)+2)*256+
PEEK(VARPTR(A$)+1):IF K>32767 THEN K=K-65536
31 RETURN
80 A$="727179":A=VAL(A$)
81 DEF FNDN!(Y%,M%,D%)=Y%*365+INT((Y%-1)/4)
 +(M%-1)*28
 + VAL(MID$("0003030608111316192124 26",
(M%-1)*2+1,2))-((M%>2)AND((Y%ANDNOT-4)=0))
 +D%'-Day# p110
82 DEF FNRY%(N!)=INT((N!-N!/1461)/365)
' - Year p111
83 DEF FNRJ%(N!)=N!-
(FNRY%(N!)*365+INT((FNRY%(N!)-1)/4))' - p111
84 DEF FNRM%(J%,Y%)=-((Y%ANDNOT-4)<>0)*
(1-(J%>31)-(J%>59)-(J%>90)-(J%>120)-(J%>151)-
(J%>181)-(J%>212)-(J%>243)-(J%>273)-(J%>304)-
(J%>334))-((Y%ANDNOT-4)=0)*(1-(J%>31)-
(J%>60)-(J%>91)-(J%>121)-(J%>152)-(J%>182)-
(J%>213)-(J%>244)-(J%>274)-(J%>305)-(J%>335))
85 DEF FNRD%(Y%,M%,J%)=(J%-((M%-1)*28
 +VAL(MID$("000303060811131619212426",
(M%-1)*2+1,2))))+((M%>2)AND((Y%ANDNOT-4)
 =0))' - p111
86 DEF FNN$(N)=RIGHT$(STR$(N),LEN(STR$(N))-1)
  ' #-->$
87 DEF FNDY$(N!)=MID$("Friday   Saturday Sunday
Monday   Tuesday  WednesdayThursday ",
(N!-INT(N!/7)*7)*9+1,9)  ' - p110
88 'MO=VAL(LEFT$(TIME$,2)):
DY=VAL(MID$(TIME$,4,2)):
YR=VAL(MID$(TIME$,7,2)):
HR=VAL(MID$(TIME$,10,2)):
MN=VAL(MID$(TIME$,13,2)):
YR=YR+1900:
GOSUB 40:GOSUB41
89 DEF FNZ2(N)=RIGHT$(STR$(N),2)
90 'No Error Handling yet
99 MO=0:DY=0:YR=0:
PRINT"PRESS: Right Arrow to Advance 1 Day,  Left
Arrow to Back Up 1 Day,   > or < to increment by 10
days.   <Enter> when Date is Correct."
190 'PRINT"  A","K","K+1","K+2","K+3
200 YR=FNRY%(A):
J=FNRJ%(A):
MO=FNRM%(J,YR):
DY=FNRD%(YR,MO,J):
IR=YR-1900
205 GOSUB 30
210 PRINT FNDY$(FNDN!(YR,MO,DY));'day of week
220 GOSUB 22:PRINT" "DA$
230 GOSUB 8:
IF PEEK(14400)=64 THEN A=A+1:
GOSUB 9:GOTO 200
240 IF PEEK(14400)=32 THEN A=A-1:
GOSUB 9:GOTO 200
250 IF Z=">"THEN A=A+10
260 IF Z="<"THEN A=A-10
270 IF Z=CHR$(13) THEN 285
280 GOSUB 9:GOTO 200
285 AA$=STR$(A):
KK=PEEK(VARPTR(AA$)+2)*256
 +PEEK(VARPTR(AA$)+1):
IF KK>32767 THEN KK=KK-65536
287 GOSUB 30:
FOR X=0 TO 5:
POKE K+X,PEEK(KK+X+1):
NEXT
295 PRINT"A="A
300 'GOSUB 30
310 CMD"DATE "+DA$:
PRINT"Time$  =  "TIME$
400 PRINT"Time: Hour in '24'";:
INPUT HR
410 PRINT TAB(28)CHR$(27);:
INPUT"Minute";MN:
MN=MN+1:GOSUB 23
420 PRINT FNDY$(FNDN!(YR,MO,DY))" "DA$" "TI$
430 YR=YR-1900:
GOSUB 22
440 CMD"DATE "+DA$
450 PRINT"Time$  =  "TIME$:
END
```

ATTENTION TRSDOS 1.3. USERS!

ANNOUNCING "SYSTEM 1.5.", THE MOST COMPREHENSIVE 1.3. UPGRADE EVER OFFERED!

MORE SPEED!! MORE POWER!! MORE PUNCH!!

While maintaining 100% compatibility to TRSDOS 1.3., this DOS upgrade advances TRSDOS 1.3. into the 90's! SYSTEM 1.5. supports 16k-32k bank data storage and 4MGHZ clock speed (4/4P/4D).

DOUBLE SIDED DRIVES ARE NOW 100% UTILIZED! (all models).

CONFIG=Y/N	CREATES CONFIG BOOT UP FILE	DATE=Y/N	DATE BOOT UP PROMPT ON or OFF
TIME=Y/N	TIME BOOT UP PROMPT ON or OFF	CURSOR='XX'	DEFINE BOOT UP CURSOR CHAR
BLINK=Y/N	SET CURSOR BOOT UP DEFAULT	CAPS=Y/N	SET KEY CAPS BOOT UP DEFAULT
LINE='XX'	SET *PR LINES BOOT UP DEFAULT	WP=d.Y/N (WP)	WRITE PROTECT ANY or ALL DRIVES
ALIVE=Y/N	GRAPHIC MONITOR ON or OFF	TRACE=Y/N	TURN SP MONITOR ON or OFF
TRON=Y/N	ADD an IMPROVED TRON	MEMORY=Y/N	BASIC FREE MEMORY DISPLAY MONITOR
TYPE=B/H/Y/N	HIGH/BANK TYPE AHEAD ON or OFF	FAST	4 MGHZ SPEED (MODEL 4'S)
SLOW	2 MGHZ SPEED (MODEL III'S)	BASIC2	ENTER ROM BASIC (NON-DISK)
CPY (parm,parm)	COPY/LIST/CAT LDOS TYPE DISKS	SYSRES=H/B/'XX'	MOVE/SYS OVERLAY(s) TO HI/BANK MEM
SYSRES=Y/N	DISABLE/ENABLE SYSRES OPTION	MACRO	DEFINE ANY KEY TO MACRO
SPOOL=H/B.SIZE	SPOOL is HIGH or BANK MEMORY	SPOOL=D.SIZE='XX'	LINK MEM SPOOLING TO DISK FILE
SPOOL=N	TEMPORARILY DISABLE SPOOLER	SPOOL=Y	REACTIVATE DISABLED SPOOLER
SPOOL=RESET	RESET (NIL) SPOOL BUFFER	SPOOL=OPEN	OPENS, REACTIVATES DISK SPOOLING
SPOOL=CLOSE	CLOSES SPOOL DISK FILE	FILTER *PR.ADLF=Y/N	ADD LINE FEEDS BEFORE PRINTING 0DH
FILTER *PR.IGLF	IGNORES 'EXTRA' LINE FEEDS	FILTER *PR.HARD=Y/N	SEND 0CH to PRINTER (FASTEST TOF)
FILTER *PR.FILTER	ADDS 256 BYTE PRINTER FILTER	FILTER *PR.ORIG	TRANSLATE PRINTER BYTE TO CHNG
FILTER *PR.FIND	TRANSLATE PRINTER BYTE TO CHNG	FILTER *PR.RESET	RESET PRINTER FILTER TABLE
FILTER *PR.LINES	DEFINE NUMBER LINES PER PAGE	FILTER *PR.WIDTH	DEFINE PRINTER LINE WIDTH
FILTER *PR.TMARG	ADDS TOP MARGIN to PRINTOUTS	FILTER *PR.BMARG	ADDS BOTTOM MARGIN to PRINTOUT
FILTER *PR.PAGE	NUMBER PAGES, SET PAGE NUMBER	FILTER *PR.ROUTE	SETS PRINTER ROUTING ON or OFF
FILTER *PR.TOF	MOVES PAPER TO TOP OF FORM	FILTER *PR.NEWPG	SET DCB LINE COUNT TO 1
FILTER *KI.ECHO	ECHO KEYS to the PRINTER	FILTER *KI.MACRO	TURN MACRO KEYS ON or OFF
ATTRIB:d.PASSWORD	CHANGE MASTER PASSWORD	DEVICE	DISPLAYS CURRENT CONFIG INFO

All parms above are installed using the new LIBRARY command SYSTEM (parm,parm). Other new LIB options include DBSIDE (enables double sided drive by treating the "other side" as a new independent drive, drives 0-7 supported) and SWAP (swap drive code table #s). Dump (CONFIG) all current high and/or bank memory data/routines and other current config to a disk data file. If your type ahead is active, you can (optional) store text in the type buffer, which is saved. During a boot, the config file is loaded back into high/bank memory and interrupts are recognized. After executing any active auto command, any stored type ahead data will be output. FANTASTIC! Convert your QWERTY keyboard to a DVORAK! Route printer output to the screen or your RS-232. Macro any key,even F1, F2 or F3. Load *01-*15 overlay(s) into high/bank memory for a memory only DOS! Enter data faster with the 256 byte type ahead option. Run 4MGHZ error free as clock, disk I/O routines are properly corrected! Spool printing to high/bank memory. Link spooling to disk (spooling updates DCB upon entering storage). Install up to 4 different debugging monitors. Print MS-DOS text files, ignoring those unwanted line feeds. Copy, Lprint, List or CATalog DOSPLUS, LS-DOS, LDOS or TRSDOS 6.x.x. files and disks. Add top/bottom margins and/or page numbers to your hard copy. Rename/Redate disks. Use special printer codes eg: LPRINT CHR$(3); toggles printer output to the ROUTE device. Special keyboard codes add even more versatility. This upgrade improves date file stamping MM/DD/YY instead of just MM/YY. Adds optional verify on/off formatting, enables users to examine *01-*15, DIR, and BOOT sectors using DEBUG, and corrects all known TRSDOS 1.3. DOS errors. Upgrade includes LIBDVR, a /CMD driver that enables LIBRARY commands, such as DIR, COPY, DEBUG, FREE, PURGE, or even small /CMD programs to be used within a running Basic program, without variable or data loss.

**By special arrangement with GRL Software,
SYSTEM 1.5. is now distributed exclusively by TRSTimes magazine.**

ORDER YOUR COPY TODAY!

Send $39.95 (U.S. funds) to:

**TRSTimes - SYSTEM 1.5.
5721 Topanga Canyon Blvd., Suite 4
Woodland Hills, CA. 91367**

Recreational & Educational Computing

Completing a Collection

By Michael W. Ecker, Ph.D.

Have you ever noticed how hard it can be to complete a collection? You could be recording winning lottery numbers, collecting baseball cards, or buying valuable old comic books. Somehow, you seem to keep seeing some of the same items before finding those last elusive ones you seek.

To play around with this, let's use our TRS-80s and a BASIC program. The program COLLECT/BAS shown here has the primary virtue of implementing a collection process until we have at least one of each element. The program keeps track of how many tries are made before getting a complete collection.

The program simulates a random collection of 2 to 13 items. When item 5 is chosen, say, an asterisk is placed next to the "5" at the left of the screen. If 5 had been previously chosen, the asterisk appears next to the previous one or more there already. Thus, this creates a histogram. I have deliberately limited this to 13 to take into account the usual screen size limits. There are ways to incorporate collections that allow larger sample sizes, the methods being both pictorial and non-pictorial, but I leave that as an exercise for readers to come up with on their own. In fact, I encourage readers to contact me at the address at the end of the article to share their own ideas, implementations, and improvements.

The Precise Question

Let's paraphrase the problem to read as follows: Suppose we are given a virtually infinite number of each of N types of items, and we choose or draw randomly one item at a time, not knowing which of the N kinds we'll get. About how long, based on this number N, must we expect to have to wait before getting at least one of each of the N different items? In other words, how many draws are needed, on the average, to complete the collection?

The visual display - the histogram - graphically illustrates the counters showing how many there are so far of each item. Each star in a line indicates one more time that the number represented by the line was chosen. Thus, when k is drawn, line k should show an extra asterisk.

To make the program run slower to give you time to see the histogram emerge, I've deliberately included a delay loop -- an empty loop. If the delay is too long, change the value from 500 to a smaller value - say, 200.

Because of display limits, don't use this version of the program for N > 13 or so.

Here's one of the aforementioned programs, the TRS-80 version of COLLECT/BAS:

```
10 CLS
20 PRINT "Program to simulate completing a collection":PRINT "Copyright 1991, Dr. M. Ecker": PRINT
30 INPUT "How many items (2 to 13)"; N
35 DIM C(N): T=0
40 INPUT "How many columns for your TRS-80 (Default=64)"; W
45 IF W=0 THEN W=64
50 FOR J=1 TO N: C(J)=0: NEXT J: CLS
60 FOR J=1 TO N: PRINT@J*W+1, J: NEXT
70 K=INT(RND(N))
80 C(K)=C(K)+1: T=T+1
90 PRINT@ W*K+C(K)+4, "*";
100 FOR DL=1 TO 500: NEXT DL
110 FOR J=1 TO N
120 IF C(J)>0 THEN NEXT: ELSE 70
130 PRINT@ W*(N+2)+1, "Collection of"; N; "items complete in"; T; "drawings."
```

Epilogue

For 13 items, you should expect, on average, to require about 40 drawings. For 100 items the figure grows to 518.

To estimate the number required for N items, I've derived this estimator:

N*(LOG(N)+.577)

Here, the logarithm is the natural log (base e, not base 10), the one used by BASIC when you use LOG. On most calculators it is Ln, not Log. The number .577 is the three-decimal approximation to Euler's constant. It should be easy to add a line at the end of the program to show how many drawings are needed on average to compare it to the result of your Monte Carlo simulation. Bear in mind to expect a degree of variability.

If you wish to contact me, please write to:

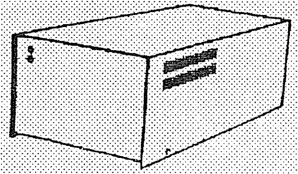
Dr. Michael W. Ecker
Recreational & Educational Computing
909 Violet Terrace
Clarks Summit, PA 18411

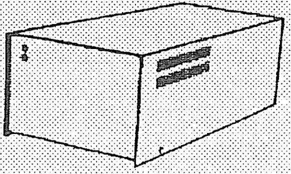
Almost forgot: If you send me a formatted TRS-80 TRSDOS 1.3 disk with a self-addressed mailer containing at least 65 cents postage, I'll send you this program FREE, plus any others I can scrounge up.

Dr. Michael W. Ecker, holder of a Ph.D. in mathematics, is a mathematics professor as well as a computer writer-reviewer and columnist with 300 publication credits. The author of two books, Mike is also Editor/Publisher of **Recreational & Educational Computing** (or **REC**).

Some of you may know Dr. Ecker and his writing as the pseudonymous "David Wade" - TRS-80 columnist for Vulcan's *Computer Monthly.*

REC, from which much of this article has been loosely excerpted, is in its sixth year and is available for $27 per calendar-year of 8 issues, prepaid. It focuses on "mathemagic" and computer recreations, mostly for generic MSDOS, but including TRS-80s. Readers are invited to try a trial subscription of three issues for $10, fully creditable toward a regular subscription. Or, save $3 by ordering the full 1991 year of **REC** for just $24 by mentioning this publication. Please also mention your exact computer brand when you write.





ALL ABOUT ALLWRITE pt. 2

(well - almost)

by Dr. Allen W. Jacobs

Allwrite comes with one of the best soft key systems available, for a 64K machine. The major limitation of Allwrite's soft keys is that they are restricted to 22 keystrokes each, whereas Word Perfect's Macro keys are "unlimited". In practice, this restriction is not nearly as constraining as it sounds. That is because one soft key can call others, and a single key can include commands that execute other soft keys. Effectively, soft keys can be linked together. Also, they can be temporarily reassigned because they are easily reprogrammable. It just never seems to be a problem if you get used to using soft keys creatively. Even if you want to assign larger amounts of text than 22 characters to each key, it can be accomplished by a number of means. However, unless you have the need to do that with soft keys, you may really not want to bother.

However, just to lay that perceived inadequacy to rest, here is a means of including a large block from one file to itself or to another. For simplicity, don't even bother giving the filename an extension. All you have to do is assign a soft key to "PUT" a block of text into a temporary file named, say, "TEXT" and have a soft key assigned to GET the file "TEXT" and insert it as a block. That can insert an entire file into the text in memory to the limit that the combined size of both files is less than the total available text area. Dependent upon the DOS, the system configuration, and any other drivers, buffers, and filters in high memory, that total available text area is between 26K-29K.

Not enough? Name the file "TEXT1" and you can have up to 9999 different blocks of text ("TEXT9999"). This is of course subject to both your memory limits, (ie.: Can you remember what each block contains?), and those limits of your hardware and DOS. It can also be done with labels in a single file. There are many ways. I have never had to use this one, as I have described it, but the keystrokes for it would be:

Position the cursor to start of the block you wish to save and press:

<CLEAR> <B>

Position the cursor to the end of the block and press:

<CLEAR> <P>

Name the block and put a drive number after it (always a good habit). Position the cursor to where you want to insert this or any other block, and press:

<BREAK> <G> <E> <T> <SPACE> [TEXTFILE]
<ENTER>

Remember, since commands only require two letters, the <T> in the GET command above is optional. However, the space is mandatory. I personally never put this activity into a soft key although much of it can be done as described. That is because it is just as easy to type this sequence in, directly, on the occasions you need it.

If you want to handle editing commands rather than text, the 22 keystroke limit is overcome in other ways. Namely, some command sequences can be divided into multiple soft keys and others can "call" other soft keys as functions. If these options are combined with the ability to call other soft key definition files into memory, the true "limitation" of soft keys becomes more complicated.

If you're unaware, the programmable soft keys are:

<CLEAR> <0> through <CLEAR> <9>

<SHIFT> <CLEAR> <1> through
<SHIFT> <CLEAR> <9>

<CLEAR> <X>, <CLEAR> <Y>, and
<CLEAR> <Z>.

They can easily be programmed by typing the key sequence:

<BREAK> <K> <E> <SPACE> <_> <ENTER>.

The <_> is the symbol of the soft key you wish to program, so if your sequence is less than 22 keystrokes long it must be terminated by the keystrokes:

<F1> <Q>

The manual says that actual control key you can use: <CLEAR>, <SHIFT><CLEAR> or <F1> is apparently machine dependent. I have used different notations in the listings - you should experiment and use those you like (as long as they work).

If you think them out, you can create some interesting sequences. Remember, if you find yourself repeatedly going through the same key sequence to do something, consider making that sequence a soft key. Don't be afraid to try them out on a piece of test text. If you quit the editor without purposely saving either the text file or the soft key file, no harm will be done. However, you can optionally save your soft key definitions in a named file, if you desire, as noted later.

My all time favorite has to be what I feel should have been hard programmed into the editor. It is always my <CLEAR><Z> soft key. The sequence is:

<F1><SPACEBAR><DOWN ARROW>
<F1><Q>.

After it has been programmed, use it once on some text you want to delete. If the text is larger than one line, hold the <CLEAR> key down and hit the <Z> key a couple of times. If the cursor is all the way to the left margin (although it doesn't have to be), you can clean out a lot of extraneous text rapidly with it. Don't panic if you go a couple of lines too far. <BREAK><W><ENTER> will restore any text that has not left the screen.

If the cursor is not all the way to the left margin, it will erase to the end of each line upon which it is invoked. This is often desirable. But if you use it on more than one line, you will have to go back and "stabilize" each erased line individually with a carriage return. Otherwise, you will not get a jumble of text when the lines are joined on the screen.

Another favorite is my <CLEAR><X> sequence. It adds a space between two words and, at the option of the user, can be programmed to leave the cursor in overtype or insert mode. This soft key calls two "hard programmed" keys. The sequence is:

<CLEAR><O><SPACEBAR><CLEAR><I>
<F1><Q>

<CLEAR><I> may be excluded to leave you in overtype mode, but I prefer to include it so that I am in insertion mode. That is because I usually use this key when I am editing rather than when entering text.

If the <CLEAR><X> sequence is used alternately as needed in combination with <CLEAR><D> just above it, the words to the right of the cursor can be moved right and left to almost anyplace along the line.

Using <CLEAR><O> insures that the state of the cursor, after it is invoked, is always in overtype mode, no matter what it was before the command was invoked.

<CLEAR><I> just toggles between the insert and overtype modes no matter what mode was in effect previously.

My <CLEAR><Y> sequence works the way <CLEAR><ENTER> should have worked. I use it especially when I paragraph, which I prefer to do after I have written something. Whenever I split a line I always find my self hitting <CLEAR><SPACEBAR><ENTER> to clear up the extra space before the carriage return symbol, on the end of the previous line. That also places me on the next line so that I can insert a paragraph or skip command. Invariably, that is where I want to be. Thus my <CLEAR><Y> sequence is:

<CLEAR><ENTER><CLEAR><SPACEBAR>
<ENTER><F1><Q>.

You,can add <CLEAR><1> to the key sequence above to make an automatic paragraph forming soft key, if you want. However, I prefer to keep these functions separate. That sequence would be:

<CLEAR><ENTER><CLEAR><SPACEBAR>
<ENTER><CLEAR><1><F1><Q>.

The assumption of using <CLEAR><1> in the above sequence is that the paragraph formatter command is placed into text by that soft key. If not, the paragraph formatter command sequence may be placed within the above soft key itself.

A good way to think about soft keys occurred to me in writing the last paragraphs of this article. In order to type the key sequences such as <CLEAR>, <ENTER>, and <SPACEBAR>, above, I became tired of doing it by hand. So, I picked out some soft keys that I believed I temporarily would not need and assigned them as follows: I hit:

<BREAK><K><E><4><ENTER>
<C><L><E><A><R><F1><Q>

<BREAK><K><E><5><ENTER>
<E><N><T><E><R><F1><Q>

<BREAK><K><E><6><ENTER>
<S><P><A><C><E>
<C><L><E><A><R>
<F1><Q>

<BREAK><K><E><7><ENTER>
<B><R><E><A><K><F1><Q>

<BREAK><K><E><8><ENTER>
<CLEAR><O><BACKSPACE><CLEAR><I>

The soft key <8> allows me to type in anything between the "<" and ">" signs. I then put the processor into upper case by hitting the <CAPS> key. Since I will probably not be needing these soft keys after this session, I will not bother to save them. That way, my usual soft keys remain as usual, by reloading my default soft key file which is always to be named: AL/DEF. It is reloaded with the key sequence:

<BREAK><K><E><SPACE><L>
<SPACE>AL/DEF<ENTER>

AL/DEF is Allwrite's name for its default soft key definition file. Therefore, the soft key definition file with this name will be the one that is automatically loaded when the editor is run. With the above sequence, however, you don't have to restart the editor to restore the default soft key definition file.

If I wanted to, I could have saved those temporary soft key definitions above as, say:

<BREAK><K><E><SPACE><S>
<SPACE>TEMPKEY/DEF<ENTER>

You may notice that sometimes I include a space after a command and other times I omit it. I found out through sloppy typing that the editor does not seem to require a space after a two letter command if the third character is a punctuation mark or number. It seems to choke or perform incorrectly, however, on letters in that same space. Thus, <BREAK><K><E><1> will work whereas <BREAK><K><E><X> will not.

Spacing is also critical when using the <BREAK><F><SPACE> command sequence (and the complementary <R> sequence). Actually, the command behaviors are still consistent because those are single letter commands.

Being a basically lazy typist, over time I became impatient with the soft keys provided with Allwrite because it always seemed that they required extra key strokes to "clean up" what they did. Regrettably, there is no way to read the actual key strokes used to make each soft key. However, with experimentation and with some of them being noted in the manual, I was able to duplicate their effects. I then improved on them (at least I think so).

A good way to see what they are doing is to sequentially press each key in a soft key definition from the listing below, while editing a piece of test text. If you don't like what the key sequence does or if you need it to do something else, you can change it. If it is helpful, you can use it elsewhere. This is like stepping through any program when you have the source code. It's a virtual soft key "tracing" exercise and it can be done in a minute or two.

I have also worked out some keys that are helpful to me for special purposes. I will give some of my revised soft key sequences, here:

Soft Key 1:
<CLEAR><ENTER><;><p><p>
<ENTER><F1><Q>

Soft Key 2:
<CLEAR><ENTER><;><p><a>
<ENTER><F1><Q>
Soft Key 3:
<CLEAR><ENTER><;><c><e>
<SPACE><o><F1><Q>

Soft Key 4:
<CLEAR><O><@><CLEAR><O><F1><Q>

Soft Key 5:
<CLEAR><ENTER><;><i><n><SPACE>
<+><5><LEFT ARROW><F1><Q>

Soft Key 6:
<CLEAR><ENTER><;><i><n><SPACE>
<o><f><f><ENTER<F1><Q>

Soft Key 7:
<CLEAR><C><CLEAR><E><CLEAR>
<H><F1><Q>

Soft Key 8: UNCHANGED
Soft Key 9: UNCHANGED

Soft Key 0:
<BREAK><S><R><*><ENTER>
<F1><Q>

Soft Key !:
<CLEAR><ENTER><;><s><k><ENTER>
<F1><Q>

Soft Key ": UNCHANGED

Soft Key #:
<CLEAR><O><;><CLEAR><l><c><m>
<SPACE><CLEAR><l><ENTER><F1><Q>

Soft Key $:
<CLEAR><ENTER><;><o><f><SPACE>
<+><0><ENTER><F1><Q>

Soft Key %:
<CLEAR><ENTER><;><o><f><SPACE>
<+><2><ENTER><F1><Q>

Soft Key &:
<CLEAR><ENTER><;><o><f><SPACE>
<o><f><f><ENTER><F1><Q>

Soft Key ':
<CLEAR><ENTER><;><f><o><SPACE>
<o><F1><Q>

Soft Key (:
< CLEAR > < @ > < CLEAR > < ENTER > < CLEAR >
< SPACEBAR > < CLEAR > < J > < LEFT ARROW >
< DOWN ARROW > < F1 > < Q >

Soft Key):
< CLEAR > < @ > < CLEAR > < ENTER > < CLEAR >
< SPACEBAR > < CLEAR > < J > < LEFT ARROW >
< + > < LEFT ARROW > < DOWN ARROW >
< F1 > < Q >

Since I use different templates of keys at different times, I may assign them to different positions and make up others for immediate use. Those presented here perform two related generic functions. Namely, some construct outlines. Others collapse on-screen formatted text to aid in reformatting with print time tabbing, to form tables (ie. the special "new" feature, first appearing in Word Perfect 5.1, only not automated).

As stated earlier, don't worry about destroying the original definitions of the soft keys. They are stored in the file "AL/DEF" which is always the default soft key definition file loaded with the editor. As the manual says, if you want another soft key file to be the default, simply name it "AL/DEF" after you have named the original file something else. I used "AL0/DEF". It's also on the original disk but I hardly ever use it anymore.

One problem with multiple key definition files like this is that it becomes difficult to remember what each key does. I don't have very good handwriting and so I do not fill in the cardboard templates that came with Allwrite. Instead, I have devised a template that I can edit to print out the way I have programmed a given soft key file. It will only work with a printer capable of printing eight lines to the inch instead of the normal six. This is necessary because the keys on a TRS-80 are three fourths of an inch apart. Thus, I print on every sixth line out of eight. Including a title for the template page, it JUST fits on an 8 1/2" x 11" piece of paper. After printing, just lay the paper on its side and fold or cut it appropriately. You may find it narrower than you prefer. You can make it wider but it will start to get in the way of your screen, especially on a desktop model. If you like, you can print the key titles, using a smaller pitch.

I have seen keyboard templates for "Word Perfect" that sell for between $18.00 to $24.00. In combination with the template card that came with Allwrite they don't have much more information on them than the template below. Additionally, you can make a new one and alter it to your preference whenever you want to; and the price is right.

I have created a template to complement the above soft key file. It is titled for its primary purpose, however, I use it almost all of the time. Some of the soft keys I have eliminated I simply type in, as they are usually used few times during an editing session. If you need more than one soft key definition file, they can be swapped by loading in the desired one. Just make a template for each. Remember, these key titles describe the function of the soft keys. They are not the actual key strokes. Those are listed above. Also, the purpose of some of the soft keys may not be immediately clear. I will cover them soon. While not the last word, the "outlining" soft key definition file template I have thus far evolved is as follows:

```
;cm Start the paper at the very top of the page, due to
the title.
;pl 80
;li 8
;ls 6
;pi 13
;fo off
1 = ;pp             ! = ;sk
2 = ;pa             " = RU ALF/CMD
3 = ;ce o           # = ;cm
4 = @_              $ = ;of +0
5 = ;in +5          % = ;of +2
6 = ;in off         & = ;of off
7 = list2end        ' = ;fo o
8 = VA              ( = del spc
9 = STATUS          ) = del spc +
0 = SR*               = capslock
: = chr nmbr        * = print scrn
- = ____            = = hard space
x = ins spc         y = split line
z = del line
```

If you are familiar with the commands, following the keystrokes in the file above, then reading the corresponding soft key "title" may help explain what they are doing.

Soft key 1 and 2 do what they originally did but without placing an extra carriage return into the text. They also now insert to the line above rather than the line below the current position. They work "neatly" when inserted to existing text. Try the original soft key first, then temporarily reprogram the same key as noted above. I think you will like the difference.

Soft key 3 allows any number of lines to be centered, rather than just one. It does this by positioning the cursor where centering can be turned oN or oFF, followed by an < ENTER >. It uses the same general insertion scheme that I used for soft keys 1 and 2, and I now use for all formatter commands.

Soft key 4 now allows the placement of all emphasis marks and frees soft key 5 for other uses. The downside of this softkey is that you have to know the emphasis marks you want to use.

Soft key 5 and 6 are used for outlining in conjunction with soft keys $, %, and &. They will be explained later.

Soft key 7 is useful for entering lists or tables of largely duplicate entries with the same format but slightly different information. You know the kind. It also will be explained later.

Soft key 0 is most useful when doing global editing. It is also the first key I usually program on the fly if needed because it is such a short sequence. I don't mind losing it for a while.

I put ;sk into soft key ! because I found the original command almost indistinguishable from soft key ". Also, it is convenient to have the choice of indenting a paragraph with soft key 1 or not indenting it with soft key !.

Soft key # is most useful for eliminating lines from being printed such as telephone numbers to the addressee, in a letter. It works best on existing lines but can also be used before they are written.

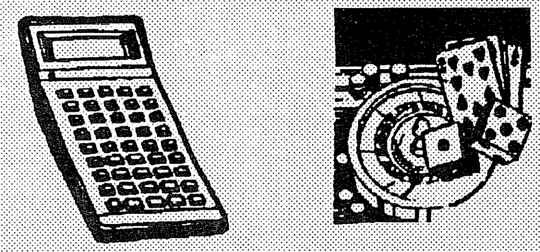
Soft key ', like soft key 3, allows any number of lines to be printed with formatting enabled or disabled. It does this by positioning the cursor where formatting can be turned oN or oFF, followed by an <ENTER>.

I must credit the basis for soft keys (and) to our fearless editor, Lance Wolstrup. I previously used two soft keys, requiring two separate passes, to accomplish space elimination and table collapsing for placement of print time tabbing control characters. Lance, who doesn't even use Allwrite, used a totally different scheme to combine the two soft key functions into one. That permitted me to use the space for a similar complementary function. He did it by seeing a demonstration of my soft key Y and the <CLEAR> <J> command JUST ONCE. Not bad. Think of what he could do if he actually WORKED with the program.

Soft key (is used to eliminate all the spaces between two words except for one. It is very useful to clean up text that has been heavily edited or when you find yourself deleting text with the space bar. It also works with blocks that have been moved around and edited. Its main advantage is that it is fast, for what it does. When used on unclean text files that have been pulled in from a DOC file or from another program, it is almost indispensable. In summary, this soft key almost, but not quite, duplicates the "delete spaces" command in plain vanilla Scripsit. (Are you listening, Eric?)

Soft key) functions exactly as soft key (except that it places a + or other tab character of your choice, between the two collapsed words. The table is then re-expanded at print time by a tabbing formatter command placed ahead of the text. This technique is especially useful because it is often easier to conceptualize a table if it is printed to the screen "properly". With this command, hours of work and "K's" of space can be saved by taking output files from other programs and BBS DOC files and reformatting it. You will not use it all the time; but then again you might, depending upon what you are doing.

Next issue I will continue with the last of this three-part article. I hope by then there will some feedback from you fellow Allwrite users - maybe even an article.



MISOSYS 20 or 40 MB Hard Drives for your Model III or 4

Our 15.5" x 7" x 5.25" (LWH) beige drive case has space for two half-height drives, 115V/230V 60 watt power supply and fan, hard disk controller (HDC), host adaptor, and a 50-pin SCSI female connector for the host interface. Our host adaptor sports a hardware real time clock. With its internal battery lifetime in excess of 10 years, never enter date and time again. A joystick port option adds a Kraft MAZEMASTER joystick with a port interface identical to the old Alpha Products joystick; thus, any software which operated from that joystick will operate from this one. Software provided with the host adaptor includes: a low level formatter; an installation utility and driver; a high level formatter used to add DOS directory information; a sub-disk partitioning utility; Utilities to archive/restore the hard disk files onto/from floppy diskettes; a utility to park the drive's read/write head; a utility to set or read the hardware clock; a keyboard filter which allows the optional joystick to generate five keycodes; and a utility to change the joystick filter's generated "keystroke" values after installation. Optional LDOS 5.3 software is available. 20MB drive packages are currently built with a Kalok 3.5" drive; 40MB packages use a Seagate ST251-1 28 millisecond drive. Drive packages are offered as '*pre-assembled kits*', assembled to order and fully tested; all you will need to do is plug it in and install the software. 50-pin host interface cable included.

1991 Prices currently in effect:

Complete Drive Kits:	
20 Megabyte kit:	$475
40 Megabyte kit:	$595
Joystick option	$20
LDOS software interface	$30
Piece Parts:	
20 Meg drive (KL320)	$200
40 Meg drive (ST-251-1)	$320
Case & Power Supply	$125
H/A with software	$75
Xebec 1421 HDC	$95
Adaptec 4010 HDC	$95
Drive power Y cable	$5
XT drive cable set	$5

Note: freight charges are additional.
Prices subject to change without notice

PRO-WAM™ Version 2

Window & Application Manager

Our applications turn your 128K Model 4 into a sophisticated business or personal machine because easily installed PRO-WAM comes with many useful and powerful menu-driven time savers and work organizers. PRO-WAM is accessed with a single keystroke; its export and import functions allow you to move data across windows between programs.

- Address CARDS, LABELS, and HEAD display & export
- BRINGUP tickler file; new PRINTING and sorting
- CALENDAR flags BRINGUP items visually on screen
- Ten 3 x 5 CARD files with FORMS and FIELDS
- Virtual PHRASE access for export
- New TODO list manager with "who does it"
- Plus many other vital applications!

PRO-WAM [M-51-025] $74.95 + $5S&H

LB Data Manager

A flexible data manager

LB is easily used by anyone for managing their data. It's menu driven for ease of use; absolutely no programming needed. Requires a Model 4 with 128K or a hard drive. LB86, an MS-DOS version is also available. Now activate PRO-WAM from newly compiled LB beta release (hardware restrictions apply)

- Store up to 65534 records per data base
- Up to 1024 characters (64 fields) per record
- Nine field types for flexibility
- Select and sort on up to 8 fields (multiple indexes)
- 10 input/update screens per data base
- 10 printout formats per data base
- Extensive on-line help available

LB [L-50-510] $74.95 + $5S&H

All Model I/III/4 Language software 30% off until March 31st

BSORT: high speed array sort for BASIC	$10.47 + $2S&H
DSM: high speed disk virtual sorting utility	$41.97 + $2S&H
DSMBLR: Disassemble CMD files to disk	$17.47 + $2S&H
EDAS: Macro assembler and full screen editor	$31.47 + $4S&H
EnhComp: BASIC compiler with in-line assembler	$41.97 + $4S&H
HartFORTH: 79Standard FORTH runs under DOS	$34.97 + $3S&H
MC: A complete K&R C compiler (requires MRAS)	$55.97 + $5S&H
MRAS: Advanced relocatable object code assembler	$41.97 + $4S&H
RATFOR (M4 only): Converts RATFOR to FORTRAN	$41.97 + $4S&H
UNREL: convert /REL object modules to /ASM	$20.97 + $2S&H

Specify Model I/III or Model 4

Ribbon Cable Assemblies

We use a Cirris Systems tester for 100% test of shorts and opens on custom manufactured ribbon cables. Using switchable test assemblies, different kinds of cables can be assembled and tested for you. We can custom fabricate a cable according to your specifications or provide standard replacement cables; these all use unshielded ribbon cable. Cables can use: DB-9 and DB-25 M/F; 20, 34, 40, or 50-pin header; 34 or 50-pin edgecard M/F; 36-pin printer; 50-pin SCSI M/F.

We make replacement FDC-Floppy cables @ $15 for all TRS-80 Model 4's (4, 4D, 4P); printer cables @ $15 for Model III/4 or II/12/16/6000; RSHD primary-secondary interconnect cables @ $15/set; RSHD host cable @ $15. Need a replacement RS232 or printer cable? Probably about $10-$20 depending on the connectors used. We also stock KelAM 34-pin M/F edgecard connectors ($13/pair).

MISOSYS, Inc.
P. O. Box 239
Sterling, VA 22170-0239
703-450-4181
orders to 800-MISOSYS (647-6797)

VISA and MasterCARD accepted
Virginia residents please include 4.5% State sales tax

Note: Our 800 ORDER LINE can accept calls from all 50 United States and CANADA.

Floppy Drives and Accessories

5.25" 360K 1/2-height	$75+$4 S&H
3.5" 720K in 5.25" 1/2-height frame	$85+$3 S&H
2SV5 drive case & P/S	$60+$5 S&H
Single drive host cable *	$10
Dual floppy extender cable **	$15

Our Model 2SV5 dual vertical external floppy disk drive case will hold two 5.25" half-height disk drives or one standard height drive. * needed for one or two drives; ** needed for two drives. S&H prices are UPS ground to continental U.S.

TRS-80 PUBLIC DOMAIN SOFTWARE BONANZA

We have bought collections of software from people leaving the TRS-80 world. As fast as we can, we are weeding out the good Public Domain and Shareware from the Commercial programs and the junk. So far, we have come up with 18 disks for the Model I & III, and 3 disks for the Model 4.

Model I & III

PD#1: binclock/cmd, binclock/doc, checker/bas, checker/doc, chomper/bas, cls/cmd, dduty3/cmd, driver/cmd, driver/doc, drivtime/cmd, mazeswp/bas, minibase/bas, minitest/dat, mx/cmd, piazza/bas, spdup/cmd, spdwn/cmd, vici/bas, vid80/cmd, words/dic.

PD#2: creator/bas, editor/cmd, maze3d/cmd, miner/cmd, note/cmd, poker/bas, psycho/cmd, supdraw/cmd, vader/cmd

PD#3: d/cmd, trsvoice/cmd, xmodem/cmd, xt3/cmd, xt3/txt, xthelp/dat

PD#4: cobra/cmd, disklog/cmd, flight/bas, flight/doc, narzabur/bas, narzabur/dat, narzabur/his, narzabur/txt, othello/bas, vid80x24/cmd, vid80x24/txt

PD#5: eliza/cmd, lu31/cmd, sq31/cmd, usq31/cmd

PD#6: clawdos/cmd, clawdos/doc, cocoxf40/cmd, dskrnam/bas, menu/cmd, ripper3/bas, sky2/bas, sky2/his, space/cmd, stocks/bas, trs13pat/bas, vidsheet/bas

PD#7: cards/bas, cities/bas, coder/bas, eye/bas, heataudt/bas, hicalc/bas, life/bas, moustrap/bas, ohare/bas, slots/bas, stars/cmd, tapedit/bas

PD#8: craps/bas, fighter/bas, float/bas, hangman/bas, jewels/cmd, lifespan/bas, varidump/bas, xindex/bas, xor/bas

PD#9: bublsort/bas, chess/bas, finratio/bas, homebudg/bas, inflat/bas, mathdril/bas, midway/bas, nitefly/bas, pokrpete/bas, teaser/bas

PD#10: ltc21/bas, ltc21/ins, lynched/bas, match/bas, math/bas, message/bas, message/ins, portfol/bas, portfol/ins, spellegg/bas, storybld/bas

PD#11: alpha/bas, caterpil/cmd, cointoss/bas, crolon/bas, cube/cmd, dragon/cmd, fastgraf/bas, fastgraf/ins, lunarexp/bas, music/bas, music/ins, planets/bas, volcano/cmd

PD#12: baccarat/bas, backpack/bas, backpack/ins, doodle/bas, dragons/bas, dragons/ins, king/bas, sinewave/bas, snoopy/bas, wallst/bas, wallst/ins

PD#13: atomtabl/bas, boa/bas, chekbook/bas, conquer/cmd, dominos/bas, morse/bas, mountain/bas, quiz/bas, signbord/bas, sketcher/bas

PD#14: autoscan/bas, checkers/bas, craps/bas, ducks/bas, isleadv/bas, nim/bas, rtriangl/bas, sammy/cmd, typing/bas, wordpuzl/bas

PD#15: budget/bas, corp/bas, corp/ins, fourcolr/bas, fullback/bas, grapher/bas, illusion/bas, jukebox/bas, ledger/bas, maze/cmd, reactest/bas, shpspree/bas, states/bas, tapecntr/bas, tiar/bas, tiar/ins

PD#16: amchase/bas, constell/bas, filemastr/bas, foneword/bas, geometry/bas, heartalk/bas, hidnumbr/bas, lgame/bas, marvello/bas, powers/bas, scramble/bas, speed/bas, subs/bas

PD#17: conundrm/bas, eclipse/bas, esp/bas, esp/ins, hustle/bas, jacklant/bas, mindblow/bas, othello/bas, pleng/bas, rubik/bas, trend/bas, ufo/bas, veggies/bas

PD#18: backgam/bas, chess/cmd, cosmip/cmd, distance/bas, hexpawn/bas, music/cmd, stokpage/bas, texted/bas, texted/ins, trex/bas, twodates/bas, wanderer/bas

Model 4

M4GOODIES#1: day/cmd, day/txt, gomuku/cmd, llife/cmd, llife/doc, writer/cmd, writer/doc, writer/hlp, yahtzee/bas

M4GOODIES#2: arc4/cmd, arc4/doc, cia/bas, etimer/cmd, index/cmd, index/dat, mail/bas, mail/txt, trscat/cmd, trscat/txt, util4/cmd, xt4/cmd, xt4/dat, xt4hlp/dat

M4GOODIES#3: convbase/bas, dates/bas, dctdsp/cmd, dmu/cmd, dmu/doc, dskcat5/cmd, dskcat5/doc, editor/cmd, editor/doc, fedit/cmd, fkey/asm, fkey/cmd, fkey/doc, hangman/cmd, m/cmd, m/src, membrane/bas, miniop2/cmd, miniop2/src, move/cmd, move/doc, othello4/bas, scroll4/cmd, scroll4/src, setdate6/cmd, setdate6/doc, setdate6/fix, spaceadv/bas, taxman/bas, utilbill/bas, utilbill/doc

Each disk is $5.00 (U.S.)
or get any 3 disks for $12.00 (U.S.)
please specify the exact disks wanted.

TRSTimes PD-DISKS
5721 Topanga Canyon Blvd., Suite 4
Woodland Hills, CA. 91367