DYNAMIC MEMORIES ™

*™ Tandy Corp./Radio Shack Inc.

JANUARY

I'M WORRIED THAT JOHNY IS TAKING HIS COMPUTER TOO SERIOUSLY

SOUTH BAY TRS-80 USERS GROUP

Page - Contents

SBUG meetings are held the 3rd Tuesday of each month in the
north-east corner of Dysan's building at:
                                   5401 Patrick Henry Drive
     Time - 7:15 to 10:30 PM   Santa Clara, Ca

          January 17, February 21, March 20

Topic of the month:

     TRS-80 Applications and Income Tax Update

     This month our new discussion leader, Casey Parker,
will discuss TRS-80 applications. Also, Ed Hoover plans to
discuss his latest updates on his income tax programs.

Also!!! Don't forget the SIG group that meets at Dysan the
first Wednesday of each month (February 1st). The Model 100
is popular at this meeting. See you there!!!

==

                     MEMBERSHIP

NOTE!!!!!!!!!!! If your dues for 1984 have not been paid,
this will be your last issue of "Dynamic Memories." So
please, won't you make yourself current with SBUG. If you
wish to continue to receive our newsletter   "Dynamic
Memories" then send $18.00 (check/money order) to the
following address:

               South Bay TRS-80 User Group
               P.O. Box 60116
               Sunnyvale, Ca 94088

or come to one of our meetings. If you also wish to
communicate with our bulletin board system (SBUG-80) then
include an additional $25.00 for an account on the system.
You must be a member of SBUG to have an account on SBUG-80.
Please, include your address and phone number. Thanks . . .
==

YOUR STEERING COMMITTEE

Member	SBUG-80 Username	Phone
Discussion Leaders:		
Glenn Vaughn	(GLENN)	
Eric Brewer	(BREWER)	(408) 252-9332
Casey Parker	(CASEY)	(415) 793-2715
Treasurer:		
Larry Gunderson	(TREAS)	(408) 259-5349
Newsletter Editor:		
Joel Lee	(JOELI)	(408) 926-3999
Assistant		Editor:
Chuck Ward	(CHUCK)	(408) 245-7688
Librarians:		
(disc) Bernie Thompson	(BERNIE)	(408) 867-7455
(tape) Gary Dixon	(GDIXON)	(408) 262-6937
(doc.) Bill Jibby	(JIBBYTRM)	(408) 268-0181
SBUG-80 Sysop:		
Robert Byrd	(EDITOR)	(408) 732-6775

OTHER KEY INDIVIDUALS

Cover Artist:		
Dave Fox	(DAVEFOX)	(408) 374-5258
Public Relations:		
Ron Carpenter	(RON)	(415) 726-3487
Host Computer:		
SBUG-80	(EVERYONE)	(408) 737-SBUG

If the need arises, feel free to give one of us a call.

THE EDITOR'S FIRST BYTE(EVERY DOG GETS ONE)
ON ASSUMING PURPLE ROBES

If the above title seems obscure, please be aware that the word is sacred, consequentlly it's keeper,(your editor) is royal. Whether I shall competently fill the shoes(I wear a 9 1/2 EE) or your previous editor, is, as the ancient saying goes, a computer of another color.

I do not shrink, however, from the monumental task before me, because the dprevious editor, Rob Byrd, and my assistant editor, Chuck Ward, support me, and their strength is as the strength of ten. Maybe more. It has to be. Avanti, adelante, forward!! in '84. I promise to do my best.

Joel

MURPHY'S LAW REVISITED
(again and again)

By William Ramsey
BILLRAM @ SBUG-80

One of the first computer journals I read (it seems like eons ago) was one which listed the computer equivalents of Murphy's law (if something can go wrong it will!) I well remember the suggestions: "If a part fails, it will be the most expensive, most difficult to find, and situated in a spot that requires removal of most other components to replace." "If it hasn't failed lately, it soon will." "The failure will always come at the worst time." "If you enjoy cutting traces, learn to enjoy computer repair as well." (The latter is my own addition.)

During the month of October, I experienced a great deal of grief, and am still not fully recovered. I first had one Mod I go down. I did my best to repair it, but I am a real clutz when it comes to this sort of endeavor (although I have happily cut traces, vaporized traces in attempting to repair them, attempted to catch falling knives, etc. I subsequently began to exchange components from my other Mod I, but also found it no longer worked. First I exchanged the "Doubler" board; then the entire interface, then the keyboard unit, then the "pregnant" cable, then the drive cable...etc. I finally called Bruce Carso who told me that he wouldn't attempt to fix my Mod I with cut traces (I already knew what the Shack would say!) Was it time to give the beast a decent burial? Why should I? I have Easter candy that is older than my Mod I! In the process, I also blew two RS 232 boards! and my Mod 100 is still having intermittent Modem problems. I decided that I must try RS and pay whatever they asked to put my systems back in working condition.

To my great surprise, they gladly fixed the Mod I (I had already discovered and fixed the problem with one of them) and they quickly repaired the keyboard unit (in just over one week's time) and at a resonable cost (about $30.00)! It took somewhat longer to fix the Mod 100. When I was in the Mt. View Computer Center, I saw a RS memo which stated that three resistors must be changed if the computer had a low serial number. When I went to pick up the machine, the store wanted to charge me for the service until I quoted the memo. At that point, the manager came over, wrote N/C on the ticket and wished me well. The machine worked for only two calls before going down again. I have not taken it back as yet, for I am still waiting on the repair of the two RS 232 boards (now well over two weeks and still no word except "They are on the bench right now." Those board have "been on the bench" for over two weeks! It must be a huge bench else it would tend to get rather crowded and messy!

JANUARY 1984

While I was computer-less, at least communication-less, I found that I had also "blown" my only disk which contained all my Mod 100 programs! When I examined the disk, I found that all the programs were intact--except sectors 162-179 (track 9), including the directory. Using Superzap (still one of my favorite utilities) I was able to view all the other sectors and hoped to reconstruct the directory. I have gone back to Pennington's "TRS-80 Disk & Other Mysteries" (IJG Inc.) so many times, and each time find it most helpful, but also rather confusing (my mind is too small). Nevertheless, I once again waded through section 6 and at last was able to achieve some understanding. I also wrote the following program to assist me the next time Murphy sneaks up and grabs my directory. I hope that you find it helpful.

1. Make a disk map (by examining each sector of the disk.) Note the sector number which begins the file, the one which ends the file, and the position of the final byte of the file. (I am assuming that you have a contiguous file.) When you have finished this task (prepare to spend some time doing this), write all the information down for use with the BASIC program which follows.

Using Superzap, choose the DNTH function (Hash code) and when prompted for the file name, enter the name you wish to give the file (but no extension at this time). Next it will prompt for the file type (extension). Enter this information also. Superzap will then give you the hash code in hex for that file. Put this information with your disk map information and get ready to resurrect your directory.

I would strongly suggest that you make a copy of your disk and do your resurrection on a disk that is a backup of your "blown" disk. If you ruin it, you can always go back to the original "dud" to make other copies. Be certain that you specify BDU (bypass directory update) when you copy.

If the directory is not readable, you will have to re-format this track. Using NEWDOS/80 2.0 you can format only a section of the disk as follows:

FORMAT n,,PFST=tn, PFTC=tc where n is the disk to format, tn is the track to format (in the case of double density Mod I, it is track 9, and tc is the number of tracks to format (in my case, only 1.) You may want then to copy the directory track of an empty formatted disk to the directory track of your disk.

The second sector of the Directory contains the HIT sector (Hash Index Table). The hash code (which you got from Superzap) should be entered into this sector as follows. If you intend to place the directory information in the 00 position of the 4th sector of the directory

(remember that we start counting from 0--and this is really
the 5th sector but numbered 4), then you would put the hash
code in the 00 row at posotion 4 (counting 0,1,2,3,4.)
Using the + key to advance to the 4th sector, the
information which you have gathered from your mapping
should be entered into the 00 position of the 4th sector.
(See the attached illustrations.)

To simplify this process, I have assumed that the file
will need no protection level when you reconstruct. This
means that the first four bytes will be 1020 the next byte
is not used, the next byte will be the End Of File byte.
This will be followed by a null byte and then the name of
the file (the one you used to get the hash code) in hex,
followed by the extension.

The second line is somewhat more complex, but can be
understood. The first four bytes (two hex numbers make one
byte) are passwords--both of which will be null at this
point 9642 9642. Then follows the length (in hex) of the
file. The next byte is 00 and then the most difficult
bytes--the track count (in hex) and the number of
contiguous sectors (in hex) and their starting point (at
sector 0 or sector 5.) The track count is easily
understood, but the next byte (relative position 7) will be
a number which will begin either with an oddor an even
number depending upon whether the file begins with sector 0
or 5. The second number of the 7th byte will be the number
of contiguous sectors plus one. The following program will
automatically do this all for you and will help take some
of the pain out of the process. I successfully restored a
full disk! If a disk is going to go bad--it will be at the
directory track. USERS BEWARE!!!

. After you have finished reconstructing your disk, you
should then make a copy of the disk using the CBF CFWO
options of NEWDOS/80. This will correct the GAT (Granule
allocation table) without your having to worry about all
those FF FC FE, etc.

Another hint--make a second copy of your directory and
place it somewhere on the disk as if it were another file.
When (not if) things go wrong, all you have to do is copy
that track to your bad track. Remember that the most
frequently used track on a disk is the directory--it will
often be the first track to fail! Like Boy Scouts, BE
PREPARED! We should all do our best to cheat Murphy.

The program which follows is for a MOD 100 (this IS
supposed to be a Mod 100 column). The reason I spent all
this time in the reconstruction of my disk was to reclaim
all my Mod 100 programs (doesn't you Mod 100 have disk???)
If you are going to run the follow program on a Mod I or
III, you will have to re-write line 250 which uses modulo
arithmatic.

```
1 CLEAR 300:DIM A(50)
10 CLS:INPUT "File Name (without extension)";F1$
20 INPUT "File type (extension)";F2$
25 INPUT "Hash Code (see Superzap)";HC$
30 INPUT "Beginning sector (in decimal)";S1
40 INPUT "Ending sector (in decimal)";S2
50 L1=S2-S1+1:Q=L1:GOSUB 5000:L1$=RIGHT$(HEX$,2)
55 L2=L1+1:Q=L2:GOSUB 5000:L2$=RIGHT$(HEX$,2)
60 INPUT "End of file location (in hex)";B$
70 S1$="102000"+B$+"00"
80 S2$="96429642"+L1$+"00"
90 FOR I=1 TO LEN(F1$):T$=MID$(F1$,I,1)
100 P=ASC(T$):IF P>91 THEN P=P-32
110 Q=P:GOSUB 5000
120 C1$=C1$+RIGHT$(HEX$,2)
130 NEXT I:I=I-1:I=8-I
140 FOR X=1 TO I
150 C1$=C1$+"20":NEXT X
160 S1$=S1$+C1$
170 FOR I=1 TO LEN(F2$): T$=MID$(F2$,I,1)
180 P=ASC(T$): IF P>91 THEN P=P-32
190 Q=P:GOSUB 5000
200 C2$=C2$+RIGHT$(HEX$,2)
210 NEXTI:I=I-1:I=3-I
220 FOR X=1 TO I:C2$=C2$+"20":NEXT X
230 S1$=S1$+C2$
240 T=INT(S1/10):Q=T:GOSUB 5000:S2$=S2$+RIGHT$(HEX$,2)
250 IF S1 MOD 10=0 THEN FR=L2:FR$="1"+RIGHT$(STR$(FR),1):
    FL$=FL$+FR$:S2$=S2$+FL$
259 CLS:PRINT "ENTER THE FOLLOWING VALUES:":PRINT
260 IF S1 MOD 10<>0 THEN FR=L2:FR$="2"+RIGHT$(STR$(FR),1):
    FL$=FL$+FR$:S2$=S2$+FL$
290 IF LEN(S2$)<32 THEN AD$=STRING$(32-LEN(S2$),"FF"):S2$=
    S2$+AD$
300 FOR XX=1 TO 32 STEP 4:PRINT MID$(S1$,XX,4);" ";:NEXT
310 FOR XX=1 TO 32 STEP 4:PRINT MID$(S2$,XX,4);" ";:NEXT
320 PRINT:INPUT "Do you want to write this to file";AN$:
    IF AN$="Y" OR AN$="y"THEN GOTO 400 ELSE GOTO 500
400 OPEN "MAPER" FOR APPEND AS 1
410 PRINT:PRINT #1,F1$;"/";F2$;" - ";HC$:FOR XX=1 TO 32 STEP 4:
    PRINT #1,MID$(S1$,XX,4);" ";:NEXT
420 FOR XX=1 TO 32 STEP 4:PRINT #1,MID$(S2$,XX,4);" ";:NEXT
425 PRINT #1,
430 CLOSE
500 PRINT:INPUT "ANOTHER";AN$:IF AN$="Y" OR AN$="y" THEN RUN
4999 MENU
5000 FOR Z=0 TO 3:A(Z)=0:NEXT:Z=0
5010 A(Z)=((Q/16)-(INT(Q/16)))*16:IF Z=3 THEN GOTO 5030
5020 Z=Z+1:Q=INT(Q/16):GOTO 5010
5030 FOR Z=0 TO 3:IF A(Z)>9 THEN A(Z)=A(Z)+55 ELSE A(Z)=A(Z)+48
5040 H$(Z)=CHR$(A(Z)):NEXT
5050 HEX$=H$(3)+H$(2)+H$(1)+H$(0):RETURN
```

```
DRV 00   A2C4 2C0A 74A9 8E00 0000 0000 0000 0000 ..,.t..........
```

```
2    10   0000 0000 0000 0000 0000 0000 0000 0000  ................
2H   20   1E1E EFDB F430 9000 0000 0000 0000 0000  .....0..........
     30   0000 0000 0000 0000 0000 0000 0000 0000  ................
DRS  40   4AA2 A463 B113 BF00 0000 0000 0000 0000  J..n............
171  50   0000 0000 0000 0000 0000 0000 0000 0000  ................
ABH  60   65B9 F4BA FF3D E300 0000 0000 0000 0000  e....-..........
     70   0000 0000 0000 0000 0000 0000 0000 0000  ................
TRK  80   AE19 AD00 0000 0000 0000 0000 0000 0000  ................
9    90   0000 0000 0000 0000 0000 0000 0000 0000  ................
9H   A0   0000 0000 0000 0000 0000 0000 0000 0000  ................
     B0   0000 0000 0000 0000 0000 0000 0000 0000  ................
TRS  C0   0000 0000 0000 0000 0000 0000 0000 0000  ................
9    D0   0000 0000 0000 0000 0000 0000 0000 0000  ................
9H   E0   0000 0000 0000 0000 0000 0000 0000 0000  ................
 P   F0   0000 0000 0000 0000 0000 0000 0000 0000  ................
```

Figure #1 - Hash Index Table (Disk Sector 171).
This is sector 1 of the DIR/SYS file.
The BYTE under discussion is 04 (a 74 appears there).

```
DRV  00   1020 00E9 004D 5550 4C41 5920 2042 4120  .....MUPLAY..BA.
2    10   9642 9642 1500 1001 1202 FFFF FFFF FFFF  .B.B...........
2H   20   1020 009F 0044 554D 5020 2020 2042 4120  .....DUMP....BA.
     30   9642 9642 0A00 1321 FFFF FFFF FFFF FFFF  .B.B...!........
DRS  40   1020 004D 0052 454E 554D 2020 2042 4120  ...M.RENUM...BA.
176  50   9642 9642 0600 1421 FFFF FFFF FFFF FFFF  .B.B...!........
BOH  60   1020 00CE 0046 5253 4F52 5420 2042 4120  .....FRSORT..BA.
     70   9642 9642 0900 1521 FFFF FFFF FFFF FFFF  .B.B...!........
TRK  80   0000 0000 0000 0000 0000 0000 0000 0000  ................
9    90   0000 0000 0000 0000 0000 0000 0000 0000  ................
9H   A0   0000 0000 0000 0000 0000 0000 0000 0000  ................
     B0   0000 0000 0000 0000 0000 0000 0000 0000  ................
TRS  C0   0000 0000 0000 0000 0000 0000 0000 0000  ................
14   D0   0000 0000 0000 0000 0000 0000 0000 0000  ................
EH   E0   0000 0000 0000 0000 0000 0000 0000 0000  ................
 P   F0   0000 0000 0000 0000 0000 0000 0000 0000  ................
```

Figure #2 - FPDE (File Primary Directory Entry).
This is sector 6 of the DIR/SYS file.
Bytes 00-1F are the bytes under discussion.

MODEM PROTOCOL OVERVIEW

by Glenn Vaughn
GLENN @ SBUG-80

Many people have asked, at one time or another, for information on the XMODEM (DFT) protocol. I found the following information on one of the CP/M BBS's around the country. I don't remember which one or I would give credit to it. I hope this article answers the questions on the protocol, it is written by the author of MODEM7 and is supposed to be the standard.

01/01/82 by Ward Christensen. I will maintain a master

copy of this. Please pass on changes or suggestions via
CBBS/Chicago at (312) 545-8086, or by voice at (312)
849-6279.

NOTE : This does not include things which I am not familiar
       with, such as the CRC option implemented by John
       Mahr.

Last Rev: (none)

     At the request of Rick Mallinak on behalf of the guys
at Standard Oil with IBM P.C.s, as well as several previous
requests, I finally decided to put my modem protocol into
writing. It had been previously formally published only in
the AMRAD newsletter.

--------   Table of Contents

-------- 1. DEFINITIONS.
<soh> 01H
<eot> 04H
<ack> 06H
<nak> 15H
<can> 18H

-------- 2. TRANSMISSION MEDIUM LEVEL PROTOCOL
     Asynchronous, 8 data bits, no parity, one stop bit.

     The protocol imposes no restrictions on the contents of
the data being transmitted. No control characters are
looked for in the 128-byte data messages. Absolutely any
kind of data may be sent - binary, ASCII, etc.   The
protocol has not formally been adopted to a 7-bit
environment for the transmission of ASCII-only (or
unpacked-hex) data , although it could be simply by having
both ends agree to AND the protocol-dependent data with 7F
hex before validating it. I specifically am referring to
the checksum, and the block numbers and their ones-
complement.
     Those wishing to maintain compatibility of the CP/M
file structure, i.e. to allow modemming ASCII files to or
from CP/M systems should follow this data format:
  * ASCII tabs used (09H); tabs set every 8.
  * Lines terminated by CR/LF (0DH 0AH)
  * End-of-file indicated by 'Z, 1AH. (one or more)
  * Data is variable length, i.e. should be considered a
    continuous stream of data bytes, broken into 128-byte

chunks purely for the purpose of transmission.
* A CP/M "peculiarity": If the data ends exactly on a
128-byte boundary, i.e. CR in 127, and LF in 128, a
subsequent sector containing the 'Z EOF character(s)
is optional, but is preferred. Some utilities or
user programs still do not handle EOF without 'Zs.
* The last block sent is no different from others, i.e.
there is no "short block".

-------- 3. MESSAGE BLOCK LEVEL PROTOCOL

 Each block of the transfer looks like:
<SOH><blk #><255-blk #><--128 data bytes--><cksum>
   in which:
<SOH>   = 01 hex
<blk #>   = binary number, starts at 01 increments by 1,
and
                wraps 0FFH to 00H (not to 01)
<255-blk #> = blk # after going thru 8080 "CMA" instr.
              Formally, this is the "ones complement".
<cksum>   = the sum of the data bytes only.   Toss any
carry.

-------- 4. FILE LEVEL PROTOCOL

---- 4A. COMMON TO BOTH SENDER AND RECEIVER:

    All errors are retried 10 times. For versions running
with an operator (i.e. NOT with XMODEM), a message is typed
after 10 errors asking the operator whether to "retry or
quit".
    Some versions of the protocol use <can>, ASCII 'X, to
cancel transmission.   This was never adopted as a standard,
as having a single "abort" character makes the transmission
susceptible to false termination due to an <ack> <nak> or
<soh> being corrupted into a <can> and canceling
transmission.
    The protocol may be considered "receiver driven", that
is, the sender need not automatically re-transmit, although
it does in the current implementations.

---- 4B. RECEIVE PROGRAM CONSIDERATIONS:

    The receiver has a 10-second timeout. It sends a <nak>
every time it times out. The receiver's first timeout,
which sends a <nak>, signals the transmitter to start.
Optionally, the receiver could send a <nak> immediately, in
case the sender was ready.   This would save the initial 10
second timeout. However, the receiver MUST continue to
timeout every 10 seconds in case the sender wasn't ready.
    Once into a receiving a block, the receiver goes into a
one-second timeout for each character and the checksum. If
the receiver wishes to <nak> 'a block for any reason
(invalid header, timeout receiving data), it must wait for

the line to clear. See "programming tips" for ideas
    Synchronizing: If a valid block number is received, it
will be: 1) the expected one, in which case everything is
fine; or 2) a repeat of the previously received block.
This should be considered OK, and only indicates that the
receivers   <ack>   got   glitched,   and   the   sender
re-transmitted; 3) any other block number indicates a fatal
loss of synchronization, such as the rare case of the
sender getting a line-glitch that looked like an <ack>.
Abort the transmission, sending a <can>

---- 4C. SENDING PROGRAM CONSIDERATIONS.

    While waiting for transmission to begin, the sender has
only a single very long timeout, say one minute. In the
current protocol, the sender has a 10 second timeout before
retrying.   I suggest NOT doing this, and letting the
protocol   be completely receiver-driven.   This will be
compatible with existing programs.
    When the sender has no more data, it sends an <eot>,
and awaits an <ack>, resending the <eot> if it doesn't get
one. Again, the protocol could be receiver-driven, with the
sender only having the high-level 1-minute timeout to
abort.

-------- 5. DATA FLOW EXAMPLE INCLUDING ERROR RECOVERY

    Here is a sample of the data flow, sending a 3-block
message. It includes the two most common line hits - a
garbaged block, and an <ack> reply getting garbaged. <xx>
represents the checksum byte.

```
SENDER                           RECEIVER
                                 Times    out    after    10
seconds,
                        <---     <nak>
<soh> 01 FE -data- <xx>  --->
                        <---     <ack>
<soh> 02 FD -data- <xx>  --->    (data gets line hit)
                        <---     <nak>
<soh> 02 FD -data- <xx>  --->
                        <---     <ack>
<soh> 03 FC -data- <xx>  --->
   (ack gets garbaged)  <---     <ack>
<soh> 03 FC -data- <xx>  --->
                        <---     <ack>
<eot>                    --->
                        <---     <ack>
```


-------- 6. PROGRAMMING TIPS.

* The character-receive subroutine should be called with a
parameter specifying the number of seconds to wait. The
receiver should first call it with a time of 10, then <nak>

and try again, 10 times.

After receiving the <soh>, the receiver should call the character receive subroutine with a 1-second timeout, for the remainder of the message and the <cksum>. Since they are sent as a continuous stream, timing out of this implies a serious like glitch that caused, say, 127 characters to be seen instead of 128.

* When the receiver wishes to <nak>, it should call a "PURGE" subroutine, to wait for the line to clear. Recall the sender tosses any characters in its UART buffer immediately upon completing sending a block, to ensure no glitches were mis-interpreted.

The most common technique is for "PURGE" to call the character receive subroutine, specifying a 1-second timeout, and looping back to PURGE until a timeout occurs. The <nak> is then sent, ensuring the other end will see it.

* You may wish to add code recommended by John Mahr to your character receive routine - to set an error flag if the UART shows framing error, or overrun. This will help catch a few more glitches - the most common of which is a hit in the high bits of the byte in two consecutive bytes. The <cksum> comes out OK since counting in 1-byte produces the same result of adding 80H + 80H as with adding 00H + 00H.

■■■

NEWDOS/80 DESCRIPTIVE STARTUP

by Bernie Thompson
BERNIE @ SBUG-80

Many different setups are possible, but the SYSTEM setup is needed first. Refer to page 2-46 of your NEWDOS/80 manual.

AG=Y and AQ=Y are for normal BREAK and CLEAR key operation. Set AL equal to the number of drives in the system. AM may be changed from 10 to a lower number of tries for Model III. I use AM=4. Model I users may desire a higher number. If you normally want to view the directory of a drive other that 0, set AN equal to that drive number. AO is set to the drive number which is the default for new files, but it will not search for a drive less than the drive selected.

AX will need to be set to 127 if you want lower case printed by your printer. Up to 256 is OK.

To allow immediate use of NEWDOS/80 without setting the date and time, set AY=N and AZ=N to retain same date and time after a reset.

BC=Y allows the operator to pause during chaining.
BD=Y allows override of AUTO by holding down ENTER.
BE=Y allows use of R to repeat the previous DOS command.
BH=Y shows the blinking cursor at power on or reset.
BI can be set to the numeric value for different cursors.
BJ is for control of speedup systems, but does not itself
perform any speedup.

BK=Y (and BN=Y for Model I) is required for WRDIRP and
W & C functions of DIRCHECK. This is necessary to
read/write between single density Model I and the double
density Model III. It is not needed for double density
Model I to be compatible with Model III. Note: the Model I
must use BN=N to have NEWDOS/80 created files readable by
the Model I TRSDOS OPERATING SYSTEM.

Next you can work with the PDRIVE specifications for
accessing the drives and compatible files or operating
systems under NEWDOS/80.

To accommodate various types of drives, set TI=? to
the proper letter(s). "A" is standard, "H" is for head
settle delay when a combination of 5 & 8 inch drives is
used. Flag "I" will use sectors starting with 1 (instead
of 0) as in the TRSDOS for Model III. Flag "J" means that
the track numbers start from 1. Flag "K" is for opposite
density track 0.

Flag "L" is two step...allowing a 40 track diskette to
be read on an 80 track drive. Flag "M" is for standard
Model III TRSDOS and allows the copy of files to or from
TRSDOS files. There is a different zap to be used
depending on the TRSDOS version. Use zap 008 for version
1.2 or earlier, which has different end of file and will
not work for version 1.3 or 2.3B files. Use zap 073 for
version 2.7DD. Zap 068 gives examples of copy commands.

The drive specification TD allows for 5 & 8 inch
drives on the same computer, but for double density on the
8 inch drives a double speedup modification is needed on
the CPU.

TC is the track count and can be set in accordance
with the actual drives being used. Note: when using double
sided drives you may have a continuous 36 sector track in
double density, actually 2 cylinders on a 5 inch drive, but
the track count remains at 40 not 80.

SPT is sectors per track, TSR is the track stepping
rate which for older 35 track drives must be set to 3.
Most new 40 track drives work at 0. I use 1 to be safe, I
do not feel it is slow.

GLP or granules per lump is the most confusing for
double density. It maintains the concept of 10 sectors per
lump as was the case of 10 sectors per track in single

density. Any number 2 thru 8 may be used, allowing more directory room for 8 inch disks.

DDSL is the disk directory starting lump for finding or creating the directory.   DDGA is the disk directory granule allocation (default) number from 2 to 6.   Option 2 allows 62 files and option 6 allows for 222 files.

Adding ",A" to the PDRIVE command for drive 0 sets the specifications into memory without doing a reset. With WRDIRP and proper PDRIVE specs, you can have single and double density, and 5/8 inch drive combinations on the same computer.

If you are using disks between computers, and any single density ones have been set for interchangeability, you can use the Automatic Density Recognition DDSD/CMD from The Alternate Source. I have had good success with this except for formatting while DDSD is invoked. You can copy to and from, or just use programs or files intermixed quite well so long as you don't try to illegally add Model III TRSDOS.

There isn't a way to go from Model III TRSDOS to Model I TRSDOS directly, but you can convert Model I to Model III. With NEWDOS/80, you can do both directions, in fact, you can use Model I single density as a medium to go to other operating systems which also support the single density media.

Super Utility Plus provides for file transfer between systems as does MULTIDOS.

LDOS is used for the Model IV which allows a convert of Model III TRSDOS and read of single density Model I directly (perhaps the repair alien for readability of the DAM <Directory Address Mark> may be required).

DOSPLUS has a new version 3.5 which I have not used, but it does many things like LDOS such as convert for TRSDOS disks and reads both single/double density automatically. I could not tell if. it can write to a TRSDOS model 3 disk, but it can read the directory and copy files to DOSPLUS disks.   There is provision for changing the DAM's either way using the proper machine on single density diskettes.

All operating systems have provided a master password and file password provisions so you don't accidentally kill files which you wanted protected.   This is important in business or multi-person operations and not always used by hobbyists.
===

MESSAGE FROM RON   AT SBUG-80
RE:MEMORY TEST PGM (TO FIX)
READ COMMAND:.
.I UPLOADED A MEM TEST PGM I FOUND AS FILE MEMTST/BAS. LOOKS
PRIMITIVE. DOES ANYONE HAVE SLIGHTLY FLAKEY RAM WHICH COULD SEE
IF THIS PGM CATCHES ERRORS? I USED TO USE A COMPUTRONICS MEMORY
TEST PGM THAT HAD 3 OPTIONS: QUICK, COMPLETE, LONG, BUT IT FAILED
TO DETECT ONE OF MY RECENT HWD BUGS. ONLY MICROCLINIC(?)'S"WORM
TEST" CAUGHT IT. BUT IT TAKES 15 MINUTES TO GET PAST CLINIC'S
UN-SKIPPABLE FIRST PART OF MEM TEST. ANY COMMENTS? IF YOU FIND
MEMST/BAS OF SIGNIFICANT VALUE, UPLOAD YOUR CLEANED UP VERSION
AND I'LL INDEX IT. (OTHERWISE,ROB,GO AHEAD AND DELETE MEMST/BAS
AFTER AWHILE.

Read command:.
.<Message from RON   AT SBUG-80   12/09/83
RE:Mod 100 game IVADR/BA good!
Read command:.
.For the first time in my life(almost) I've played a computer
game:Invadr(FROM REQ#018 SBUG-80 disk). Took a bit to get edits
described in lines 4-6 in right. Nice sound! P.S. is there a way
to disable typeahead? I over-move the shooter because of
typehead.

Read command :.
.<Message from Kemp   at SBUG-80   12/09/83
RE: NRI Micrcomputer Training
Read comand :.
.NRI gives microcomputer courses that are specific to the R/S
Color & Model 4 computers. Some time ago they gave this same.
course that was specific to the R/S Model I. If any of you folks
either took that course(Model I) or know of someone who did I'd
appreciate a not on it on the Bulletin Board. Thanx, Chet
Kemp(KEMP)

Read command:.
.<Message from JIBBYTRM at SBUG-80 12/12/83
RE:Model 100 Ram
Read command :.
.Hello 100ites

Just received my 8k ram from Purple computing. It cost only
$59.95. It has been installed and is working perfectly. The
DIRMEM.BA which is the program from the board called DIR.ba
requires that the 65336 be changed to 65335 when the ram device
is installed. I was surprised at the price of the RAM being so
low but what the heck. If you are interested you can call them
in Camerillo,Ca. and charge the ram to Mastercharge. The phone #
is (805) 987-4788 if you're interested.

        Your editor humbly begs the pardon of the lords of
computerdom for the lateness of this newsletter.

FIRST CLASS MAIL