DZAP.

DZAP is the result of a much rewritten version of the original TANDOS utility called SECDUMP and was also previously SECFIX.

The transition from SECFIX to DZAP was relatively minor but the differences will be made clear as we go along.

The main change from SECDUMP to DZAP is that the screens (text input and sector output) are much more separated in the way they are controlled. In fact the graphics cursor that can roam about the lower half of the MICROTAN screen, is now controlled in real time and this control has no affect on the upper (text output) part of the display (except in the case of CNTROL O).

First a summary of all commands:-

R (sector track)

- sector read primitive, wil download the specified sector, track into the lower half of the screen. Note that the order of entering the sector,track number has been reversed from SECDMP. The main reason for this is that it is now unnecessary to re-enter the track number if the sector required is on the same track as the last entry. ie. R5 will read sector five from the same track as last time.

W(sector,track)

-sector write primitive. Comments as for sector read in every way except this is obviously the inverse function.

Exxxxxxx

-Enter an ASCII string specified in "xxxxxxx" into the screen buffer (lower). Note that only what is on the screen line can be transferred so if the cursor "falls" off to the next line that text will NOT be transferred and the statement error "?" will be generated. The text will enter from the current buffer cursor position.

H4E,(FF,FF)

- Transfer the Hex bytes specified to the screen buffer. Leading zeroes are ignored and as many bytes as can fit on one screen line may be transferred at once. The bytes will be entered from the current buffer cursor position onwards and its position automatically updated.

L(0-FF)

- Locate the screen buffer cursor at the specified position in the lower half of the screen. Note that the parameter is in Hex and that the display now includes an "index" line immediately above the lower screen in order to help with positioning. This command has largely been made redundant by the change in nature of cursor control which is not now as cumbersome and faster.

CNTRL COMMANDS

- The rest of the commands from SECDMP are now no

longer valid and you do not need to enter MMMM'R etc. for 3 moves right
of the cursor. The graphics cursor in the lower half of the screen is now controlled in real time by our old friends CNTRL R,L,U & D for right, left, up and down respectively. This cursor control can occur at ANY time and will NOT affect in any way what is going on in the upper text part of the screen. For example you may be in the middle of typing in a string of characters after the "E" command ready to transfer when you realise that the buffer cursor is in the wrong position. No problem. Just change the cursor position using the control keys and carry on typing your text. The biggest change from the old SECDMP program is probably in the P command which is no longer valid and has been replaced by CNTRL O(output). As in the cursor control functions this key is also "live" in that it can be used at any time and from any position. On first use of CTRL O, the byte underneath the cursor in the buffer will be output to the upper text screen as a hex number followed by a ",". You have now entered the output "mode" and will remain in it until a CR is issued from the keyboard. Further use of the CTRL O key will result in the next byte being output etc. This means that whole groups of bytes may be output to a single line on the text screen for easier checking when applicable. Note that these CTRL functions "wrap around". i.e. the graphic cursor wil drop onto the next line if using CTRL R or CTRL O etc.

Misc Commands

- F and X are still valid and respectively F(ill) the screen buffer with the specified ASCII character following the F on the screen. Note that F(CR) will clear the screen buffer under the index line. X will invoke a tidy exit from the program DZAP.

General Comments

Note that DZAP now resides in the "transient program area" of the RAM on the DOS card and NOT in normal RAM so in most cases it will not upset or infringe on normal program operation. If the VDU 80/82 board is resident in the sytem and you are running under TUBBUG, then the VDU board will have it's own screen cleared and then further output to it wil be suppressed on start up of DZAP. The X command will reinstate the VDU board. Beware of errors occurring due to the DOS. In most cases, DZAP will trap out any scrolling into the lower part of the screen by virtue of its own screen handling but should a DOS error occur then it is possible for the message that gets output by the DOS to upset the screen formatting. The easiest way around this is to issue a reset and then type in XX 0B960 for an orderly return to DZAP operation.

The differences between SECFIX and DZAP are in the fact that the sector,track parameters in the R & W commands have been reversed for the reasons stated and the CTRL O output format has been extended to tidy the output up.

CREATE:-

This utility is the latest INIT and FORMAT combined, the syntax is as follows:-

CREATE x: where x = Drive Number

TANDOS.NEW:-

This is the object code for Tandos O.S. re-written both for speed and access of Machine code utilities from BASIC. Only the utilities which reside at $8960 > will function with Basic, the syntax is as follows:-

DOSDIR, DOSDZAP, DOSREN 2:xxxx etc. DOS is the command word.

NOTE: When blowing a new eprom containing TANDOS A800 to AFFF MUST BE BLOWN INTO HIGH BLOCK OF 2732, LIKEWISE B000 TO B7FF INTO LOW BLOCK. This is due to an address hardware fault on Tandos PCB.

BACKUP:- A useful utility for single drive owners, will copy whole disk contents without the need to do it file by file. Watch this one as it will write over anything already on the destination disk.

```
                O>IB000
DLOAD ------    B000 4C78B2 JMP $B278
AUTOLOAD ------ B003 4C85B7 JMP $B785
WRITES ------   B006 4CD6B1 JMP $B1D6
READS ------    B009 4CDAB1 JMP $B1DA
RESET ------    B00C 4CBFB1 JMP $B1BF
FDMAIN ------   B00F 4CDCB1 JMP $B1DC
FDSUB ------    B012 4C33B1 JMP $B133
SETINT ------   B015 4C45B1 JMP $B145

INTERRUPT Routine
                B018 AD94BF LDA $BF94
                B01B 6A     ROR A
                B01C 902F   BCC $B04D   -- if b7 = 0 (Data request)
                B01E AD0AB8 LDA $B80A     0110 0001
                B021 293E   AND £$003E    11 111
                B023 8D0AB8 STA $B80A     0111 1111 (7F)

                B026 8D94BF STA $BF94
                B029 A202   LDX £$0002
                B02B BD0FB8 LDA $B80F,X
                B02E 9510   STA $0010,X
                B030 CA     DEX
                B031 10F8   BPL $B02B
                B033 AD0CB8 LDA $B80C
                B036 F005   BEQ $B03D
                B038 A900   LDA £$0000
                B03A 8DFFFF STA $FFFF
                B03D 68     PLA
                B03E 68     PLA
                B03F 68     PLA
                B040 AD90BF LDA $BF90   -- status
                B043 8D09B8 STA $B809
                B046 295D   AND £$005D    0101 1101
                B048 8D08B8 STA $B808
                B04B 58     CLI
                B04C 60     RTS         -- SP F1
                B04D 4C0FB8 JMP $B80F
                B050 A900   LDA £$0000
                B052 AA     TAX
                B053 20BFB1 JSR $B1BF   -- RESET
                B056 4C40B0 JMP $B040
                B059 C020   CPY £$0020
                B05B B02D   BCS $B08A
                B05D C010   CPY £$0010
                B05F 9029   BCC $B08A
                B061 AD01B8 LDA $B801
                B064 8D93BF STA $BF93
                B067 4C8AB0 JMP $B08A
                B06A AD90BF LDA $BF90   -- status
                B06D 6A     ROR A       } -- if b0=1, error 01
                B06E B0D0   BCS $B040   }
                B070 AC07B8 LDY $B807   -- FDMAIN "X" value.
                B073 20E7B0 JSR $B0E7   -> CHK DRV/SIDE
                B076 B0D8   BCS $B050   -- if error
                B078 A920   LDA £$0020
                B07A 2C07B8 BIT $B807   -- if b5=0
                B07D 10DA   BPL $B059   -- if b6=0
                B07F 5013   BVC $B094   -- if b5=1
                B081 D05E   BNE $B0E1
                B083 A910   LDA £$0010
                B085 2C07B8 BIT $B807
```

FF30
FF03
B2DB
B2EE
B4D1
0103
B1E9

Branch depending on B807 (64)

0X	B08A
1X	B061
2-7X	B08A
8X-BX	B09A
CX	B081
DX	B08A
EX-FX	B0E1

```
B088 F027   BEQ $B0B1   — if B4=0
B08A 2077B1 JSR $B177   — if B4=1
B08D 2045B1 JSR $B145   — set up interrupt vectors.
B090 18     CLC
B091 58     CLI
B092 90FE   BCC $B092
B094 AD91BF LDA $BF91   — Track
B097 CD01B8 CMP $B801
B09A F00F   BEQ $B0AB
B09C A218   LDX £$0018
B09E 20DCB1 JSR $B1DC
B0A1 AD08B8 LDA $B808
B0A4 D09A   BNE $B040
B0A6 A20A   LDX £$000A
B0A8 2033B1 JSR $B133
B0AB AD02B8 LDA $B802   — sector
B0AE 8D92BF STA $BF92   — command sector
B0B1 2077B1 JSR $B177   → set. int. vec.
B0B4 98     TYA         — "x" value.
B0B5 2920   AND £$0020
B0B7 D014   BNE $B0CD
B0B9 2045B1 JSR $B145
B0BC 58     CLI
B0BD AD94BF LDA $BF94
B0C0 0A     ASL A
B0C1 90FA   BCC $B0BD
B0C3 AD93BF LDA $BF93
B0C6 9133   STA ($0033),Y
B0C8 C8     INY
B0C9 D0F2   BNE $B0BD
B0CB F012   BEQ $B0DF
B0CD 2045B1 JSR $B145   SET INT
B0D0 58     CLI
B0D1 AD94BF LDA $BF94   — Board Reg.
B0D4 0A     ASL A
B0D5 90FA   BCC $B0D1   — IRQ stat = 0
B0D7 B133   LDA ($0033),Y
B0D9 8D93BF STA $BF93   — B815
B0DC C8     INY
B0DD D0F2   BNE $B0D1
B0DF F0FE   BEQ $B0DF   — wait for interrupt.
B0E1 2077B1 JSR $B177
B0E4 6C04B8 JMP ($B804)
B0E7 AD00B8 LDA $B800   — unit
B0EA 2907   AND £$0007
B0EC AA     TAX
B0ED BDCBB1 LDA $B1CB,X
B0F0 8D94BF STA $BF94   — store it
B0F3 AA     TAX
B0F4 AD0AB8 LDA $B80A
B0F7 8E0AB8 STX $B80A
B0FA 291C   AND £$001C
B0FC 8533   STA $0033
B0FE 8A     TXA
B0FF 291C   AND £$001C
B101 C533   CMP $0033
B103 F02A   BEQ $B12F
B105 C010   CPY £$0010
B107 9026   BCC $B12F
B109 AD05B8 LDA $B805   — if correct return with C=0
B10C 48     PHA         — B8
```

CHK DRV/SIDE

check drv + side

00 side Drv.
0011 1100

0	20
1	30
2	24
3	34
4	28
5	38
6	2C
7	3C

```
B10D A9FD    LDA £$00FD
B10F 8D05B8  STA $B805
B112 AD0BB8  LDA $B80B   — 07
B115 A2C4    LDX £$00C4
B117 20DCB1  JSR $B1DC   ——→ FDMAIN
B11A 8D0BB8  STA $B80B
B11D 68      PLA
B11E 8D05B8  STA $B805
B121 8C07B8  STY $B807   — was $4
B124 AD08B8  LDA $B808   — 10
B127 D008    BNE $B131
B129 AD92BF  LDA $BF92
B12C 8D91BF  STA $BF91
B12F 18      CLC
B130 60      RTS
B131 38      SEC
B132 60      RTS
```

FDSUBS

Low level disc primitive

```
B133 8A      TXA
B134 48      PHA
B135 A900    LDA £$0000
B137 8D0DB8  STA $B80D
B13A CE0DB8  DEC $B80D
B13D D0FB    BNE $B13A
B13F CA      DEX
B140 D0F3    BNE $B135
B142 68      PLA
B143 AA      TAX
B144 60      RTS
```

SETINT

Start disc motor

```
B145 A90C    LDA £$000C
B147 2C07B8  BIT $B807
B14A 3002    BMI $B14E
B14C F015    BEQ $B163
B14E AD94BF  LDA $BF94   } HLD status
B151 2940    AND £$0040  }
B153 D00E    BNE $B163   — if HLD status is 0
B155 8C90BF  STY $BF90   — command = "X"
B158 A2A7    LDX £$00A7
B15A 2033B1  JSR $B133   } delay.
B15D 2033B1  JSR $B133   }
B160 4C66B1  JMP $B166
B163 8C90BF  STY $BF90   — command = "x"
B166 8C07B8  STY $B807
B169 A000    LDY £$0000          Y = #0
B16B AD0AB8  LDA $B80A   — 20
B16E 0941    ORA £$0041  = 61
B170 8D0AB8  STA $B80A   -
B173 8D94BF  STA $BF94   =
B176 60      RTS
```

set. int. vector

```
B177 78      SEI
B178 A202    LDX £$0002
B17A B510    LDA $0010,X } old interrupt vector   10/11/12
B17C 9D0FB8  STA $B80F,X }
B17F BDD3B1  LDA $B1D3,X } new interrupt vector   4C 18 B0
B182 9510    STA $0010,X }
B184 CA      DEX
B185 10F3    BPL $B17A
B187 AD90BF  LDA $BF90   — status } wait for b7 to go 0
B18A 0A      ASL A
B18B B0FA    BCS $B187
B18D AD06B8  LDA $B806   — page number
B190 F00D    BEQ $B19F
```

```
B192 0A       ASL A
B193 0A       ASL A
B194 0A       ASL A
B195 0A       ASL A
B196 0D06B8   ORA $B806
B199 8DFFFF   STA $FFFF
B19C 8D0CB8   STA $B80C
B19F AD04B8   LDA $B804
B1A2 8533     STA $0033      put DMAA (BB15) in 33|34
B1A4 AD05B8   LDA $B805
B1A7 8534     STA $0034
B1A9 60       RTS
B1AA 204861   JSR $6148      Hard error
B1AD 72       ?
B1AE 64       ?
B1AF 206572   JSR $7265
B1B2 72       ?
B1B3 6F       ?
B1B4 72       ?
B1B5 00       BRK
B1B6 0D524F   ORA $4F52      [CR] RO
B1B9 204469   JSR $6944      Disc
B1BC 73       ?
B1BD 63       ?
B1BE 00       BRK
RESET (Disc restore)
B1BF A900     LDA £$0000
B1C1 8D90BF   STA $BF90
B1C4 2033B1   JSR $B133
B1C7 AD90BF   LDA $BF90
B1CA 60       RTS
B1CB 203024   JSR $2430
B1CE 34       ?
B1CF 28       PLP
B1D0 38       SEC
B1D1 2C3C[illegible] BIT $4C3C
B1D3 [illegible]
B1D5 B0[illegible]
WRITES
B1D6 A4D0     LDY $00D0      A2 A4  LDX #$A4
B1D9 02       ?              D0 02  BNE $B1DC
READS
B1DA A284     LDX £$0084
FDMAIN
B1DC 8E07B8   STX $B807
B1DF 48       PHA
B1E0 98       TYA
B1E1 48       PHA
B1E2 A907     LDA £$0007
B1E4 8D0BB8   STA $B80B
B1E7 206AB0   JSR $B06A      ──→
→ B1EA F01B   BEQ $B207
B1EC A8       TAY  = 10
B1ED 6A       ROR A
B1EE B017     BCS $B207      ── if Y b0 = 1 (2)
B1F0 A920     LDA £$0020                      84
B1F2 2C07B8   BIT $B807      ~~if b6=0~~
B1F5 1014     BPL $B20B      ── if b7 = 0 (3)
B1F7 502B     BVC $B221      ── if b6 = 0 (4)
B1F9 D00C     BNE $B207      ── if b5 = 1 (2)
B1FB A910     LDA £$0010
B1FD 2C07B8   BIT $B807      ~~done~~
B200 F03A     BEQ $B23C      ── if b4 = 0 (5)
(1) → B202 A900   LDA £$0000 ── if b4 = 1 (1)
B204 8D0BB8   STA $B80B
```

```
(2) →  B207 68       PLA
       B208 A8       TAY
       B209 68       PLA
       B20A 60       RTS
(3) →  B20B 98       TYA
       B20C 2918     AND £$0018
       B20E F0F2     BEQ $B202
       B210 C018     CPY £$0018
       B212 F0F3     BEQ $B207
       B214 AD07B8   LDA $B807
       B217 C920     CMP £$0020
       B219 B0EC     BCS $B207
       B21B C910     CMP £$0010
       B21D 901D     BCC $B23C
       B21F B010     BCS $B231
(4) →  B221 98       TYA
       B222 2940     AND £$0040
       B224 D02F     BNE $B255
       B226 C010     CPY £$0010
       B228 9012     BCC $B23C
       B22A AD07B8   LDA $B807
       B22D 2910     AND £$0010
       B22F D0D6     BNE $B207
       B231 AC07B8   LDY $B807
       B234 2009B1   JSR $B109
       B237 8C07B8   STY $B807
       B23A B005     BCS $B241
(5) →  B23C CE0BB8   DEC $B80B
       B23F 10A6     BPL $B1E7
       B241 A900     LDA £$0000
       B243 AA       TAX
       B244 A8       TAY
       B245 20BFB1   JSR $B1BF
       B248 200CF8   JSR $F80C
       B24B AD09B8   LDA $B809
       B24E 201AF8   JSR $F81A
       B251 A200     LDX £$0000
       B253 F002     BEQ $B257
       B255 A20C     LDX £$000C
       B257 BDAAB1   LDA $B1AA,X
       B25A F0AB     BEQ $B207
       B25C 200EF8   JSR $F80E
       B25F E8       INX
       B260 D0F5     BNE $B257
       B262 0D4E6F   ORA $6F4E
       B265 206669   JSR $6966
       B268 6C6500   JMP ($0065)
       B26B 0D4E6F   ORA $6F4E
       B26E 206D65   JSR $656D
       B271 6D6F72   ADC $726F
       B274 792D50   ADC $502D,Y
       B277 00       BRK
DLOAD  B278 2063B7   JSR $B763
ENTRY  B27B 2057B4   JSR $B457
       B27E 2084B6   JSR $B684
       B281 2074B7   JSR $B774
       B284 203DB4   JSR $B43D
       B287 B10A     LDA ($000A),Y
       B289 304C     BMI $B2D7
       B28B C920     CMP £$0020
       B28D D003     BNE $B292
```

A,X,Y all Ø

Reset

outret

Nexpnt

Print "Hard Error"

exit Print "Ro Disc"

[CR] No file

[CR] No memory — P

set up 40/4

set up error vector

store filename

check it for ? & *

set up store locns

— cursor position

Non ASCII

<spc>

```
B28F C8      INY
B290 10F5    BPL $B287
B292 C94E    CMP £$004E
B294 D005    BNE $B29B
B296 8D4BB9  STA $B94B
B299 10F4    BPL $B28F
B29B C944    CMP £$0044
B29D D005    BNE $B2A4
B29F 8D4CB9  STA $B94C
B2A2 10EB    BPL $B28F
B2A4 C950    CMP £$0050
B2A6 D014    BNE $B2BC
B2A8 2017F8  JSR $F817
B2AB 7003    BVS $B2B0
B2AD 6C4FB9  JMP ($B94F)
B2B0 201CB5  JSR $B51C
B2B3 C9FF    CMP £$00FF
B2B5 F0F6    BEQ $B2AD
B2B7 8D3EB9  STA $B93E
B2BA 10CB    BPL $B287
B2BC C953    CMP £$0053
B2BE D0ED    BNE $B2AD
B2C0 8D4BB9  STA $B94B
B2C3 8D43B9  STA $B943
B2C6 2017F8  JSR $F817
B2C9 50E2    BVC $B2AD
B2CB A514    LDA $0014
B2CD 8D4AB9  STA $B94A
B2D0 A513    LDA $0013
B2D2 8D49B9  STA $B949
B2D5 70B0    BVS $B287
B2D7 98      TYA
B2D8 48      PHA
B2D9 20ECB2  JSR $B2EC
B2DC 68      PLA
B2DD A8      TAY
B2DE AD4BB9  LDA $B94B
B2E1 1006    BPL $B2E9
B2E3 2092B4  JSR $B492
B2E6 6C47B9  JMP ($B947)
B2E9 4C80B4  JMP $B480

LOADFL
B2EC 206AB6  JSR $B66A
B2EF AD2DB9  LDA $B92D
B2F2 8D00B8  STA $B800
B2F5 20EFB4  JSR $B4EF
B2F8 2019B7  JSR $B719
B2FB E000    CPX £$0000
B2FD D00D    BNE $B30C

NFILE
B2FF A962    LDA £$0062
B301 8540    STA $0040
B303 A9B2    LDA £$00B2
B305 8541    STA $0041
B307 200FB5  JSR $B50F
B30A F0A1    BEQ $B2AD
B30C BD31B8  LDA $B831,X
B30F 8D01B8  STA $B801
B312 BD30B8  LDA $B830,X
B315 20E9B4  JSR $B4E9
B318 A200    LDX £$0000
B31A A002    LDY £$0002
B31C 1002    BPL $B320
```

move along to next non-space character.

N — Do run

D — Display start/stop/transfer addresses

P — Page n

Hexpack

ERROR SEQ

check Page ≤ 7

S — Start of hex input

Hexpack

Error SEQ

load start address

OUT RET — return to monitor

LOADFL
Load a file

NFILE
'No File' message.

```
B31E 8A      TXA
B31F A8      TAY
B320 AD3EB9 LDA $B93E
B323 1006   BPL $B32B
B325 B926B8 LDA $B826,Y
B328 8D3EB9 STA $B93E
B32B AD43B9 LDA $B943
B32E D00C   BNE $B33C
B330 B927B8 LDA $B827,Y
B333 8D49B9 STA $B949
B336 B928B8 LDA $B828,Y
B339 8D4AB9 STA $B94A
B33C 38      SEC
B33D AD49B9 LDA $B949
B340 F927B8 SBC $B827,Y
B343 9927B8 STA $B827,Y
B346 AD4AB9 LDA $B94A
B349 F928B8 SBC $B828,Y
B34C 9928B8 STA $B828,Y
B34F 18      CLC
B350 B927B8 LDA $B827,Y
B353 7929B8 ADC $B829,Y
B356 9929B8 STA $B829,Y
B359 B928B8 LDA $B828,Y
B35C 792AB8 ADC $B82A,Y
B35F 992AB8 STA $B82A,Y
B362 E000   CPX £$0000
B364 D016   BNE $B37C
B366 B92CB8 LDA $B82C,Y
B369 8D48B9 STA $B948
B36C B92BB8 LDA $B82B,Y
B36F 8D47B9 STA $B947
B372 D008   BNE $B37C
B374 AD48B9 LDA $B948
B377 D003   BNE $B37C
B379 8D4BB9 STA $B94B
B37C AD4CB9 LDA $B94C
B37F 3039   BMI $B3BA
B381 200CF8 JSR $F80C  — OUTRET
B384 AD4AB9 LDA $B94A
B387 201AF8 JSR $F81A  — HEXPNT
B38A AD49B9 LDA $B949
B38D 201AF8 JSR $F81A  — HEXPNT
B390 A920   LDA £$0020
B392 200EF8 JSR $F80E  — OUTALL
B395 B92AB8 LDA $B82A,Y
B398 201AF8 JSR $F81A  — HEXPNT
B39B B929B8 LDA $B829,Y
B39E 201AF8 JSR $F81A  — HEXPNT
B3A1 AD43B9 LDA $B943
B3A4 D011   BNE $B3B7
B3A6 A920   LDA £$0020
B3A8 200EF8 JSR $F80E  — OUTALL
B3AB B92CB8 LDA $B82C,Y
B3AE 201AF8 JSR $F81A  — HEXPNT
B3B1 B92BB8 LDA $B82B,Y
B3B4 201AF8 JSR $F81A  — HEXPNT
B3B7 2029F8 JSR $F829  — CURSOFF
B3BA AE3EB9 LDX $B93E
B3BD BD1DB8 LDA $B81D,X
B3C0 D92AB8 CMP $B82A,Y
```

```
B3C3 B022     BCS $B3E7
B3C5 AD4AB9   LDA $B94A
B3C8 C9B8     CMP £$00B8
B3CA 9004     BCC $B3D0
B3CC C9C0     CMP £$00C0
B3CE 9017     BCC $B3E7
B3D0 A96B     LDA £$006B
B3D2 8540     STA $0040
B3D4 A9B2     LDA £$00B2
B3D6 8541     STA $0041
B3D8 200FB5   JSR $B50F
B3DB AD3EB9   LDA $B93E
B3DE 18       CLC
B3DF 6930     ADC £$0030
B3E1 200EF8   JSR $F80E   ---- OUTALL
B3E4 4CC7B7   JMP $B7C7   ---- ERROR REQ
B3E7 AD49B9   LDA $B949
B3EA 8540     STA $0040
B3EC AD4AB9   LDA $B94A
B3EF 8541     STA $0041
B3F1 18       CLC
B3F2 98       TYA
B3F3 6908     ADC £$0008
B3F5 AA       TAX
B3F6 F032     BEQ $B42A
B3F8 BD25B8   LDA $B825,X
B3FB F02A     BEQ $B427
B3FD C9FF     CMP £$00FF
B3FF D003     BNE $B404
B401 4C1EB3   JMP $B31E
B404 8D43B9   STA $B943
B407 AD3EB9   LDA $B93E
B40A 0A       ASL A
B40B 0A       ASL A
B40C 0A       ASL A
B40D 0A       ASL A
B40E 6D3EB9   ADC $B93E
B411 8DFFFF   STA $FFFF
B414 A000     LDY £$0000
B416 E8       INX
B417 BD25B8   LDA $B825,X
B41A 9140     STA ($0040),Y
B41C E640     INC $0040
B41E D002     BNE $B422
B420 E641     INC $0041
B422 CE43B9   DEC $B943
B425 D0EF     BNE $B416
B427 E8       INX
B428 D0CE     BNE $B3F8
B42A AD25B8   LDA $B825
B42D 8D01B8   STA $B801
B430 AD26B8   LDA $B826
B433 F007     BEQ $B43C
B435 20E9B4   JSR $B4E9
B438 A202     LDX £$0002
B43A 10BC     BPL $B3F8
B43C 60       RTS
B43D 2029F8   JSR $F829   ---- CURSOR off
B440 A9FF     LDA £$00FF
B442 8D4BB9   STA $B94B
B445 8D4CB9   STA $B94C
```

LDDFLT
Set up default
Dcand parameters

```
B448 8D3EB9 STA $B93E
B44B A900   LDA £$0000
B44D 8D49B9 STA $B949
B450 8D4AB9 STA $B94A
B453 8D43B9 STA $B943
B456 60     RTS
B457 A9AD   LDA £$00AD      B94F = # AD
B459 8D4FB9 STA $B94F
B45C A9B4   LDA £$00B4      B950 = # 64
B45E 8D50B9 STA $B950
B461 60     RTS
B462 A900   LDA £$0000
B464 8D46B9 STA $B946
B467 201DF8 JSR $F81D       Pass 1 Ø.
B46A A501   LDA $0001
B46C C97F   CMP £$007F      cursor?
B46E D008   BNE $B478
B470 200EF8 JSR $F80E       out ACC
B473 CE46B9 DEC $B946
B476 10EF   BPL $B467
B478 C90D   CMP £$000D
B47A F012   BEQ $B48E
B47C C91B   CMP £$001B
B47E D006   BNE $B486
B480 200CF8 JSR $F80C       OUTRET
B483 4CB0B4 JMP $B4B0       controlled return to monitor
B486 200EF8 JSR $F80E       out ACC
B489 EE46B9 INC $B946
B48C D0D9   BNE $B467
B48E AD46B9 LDA $B946
B491 60     RTS
B492 68     PLA             last 2 off the stack into 40/41
B493 8540   STA $0040
B495 68     PLA
B496 8541   STA $0041
B498 AE42B9 LDX $B942       stack pointer + 2
B49B 9A     TXS
B49C A541   LDA $0041       40/41 back onto stack
B49E 48     PHA
B49F A540   LDA $0040
B4A1 48     PHA
B4A2 AD4DB9 LDA $B94D       load original values of 40/41 back into 40/41
B4A5 8540   STA $0040
B4A7 AD4EB9 LDA $B94E
B4AA 8541   STA $0041
B4AC 60     RTS
B4AD 20B6B4 JSR $B4B6
B4B0 2092B4 JSR $B492
B4B3 4C23F8 JMP $F823       Return to monitor
B4B6 A920   LDA £$0020      <spc>
B4B8 200EF8 JSR $F80E       outall
B4BB A93F   LDA £$003F      ?
B4BD 200EF8 JSR $F80E       outall
B4C0 4C0CF8 JMP $F80C       out Ret & Return
B4C3 20D6B1 JSR $B1D6
B4C6 AD08B8 LDA $B808
B4C9 D001   BNE $B4CC
B4CB 60     RTS
B4CC 4CC7B7 JMP $B7C7
B4CF 20DAB1 JSR $B1DA
B4D2 4CC6B4 JMP $B4C6
```

ERRVEC

GETLIN
Get a command line

ERROR SEQ
Restore Zpage & return to TINBUG

QUERY
Print "?"

WRTSCT
write a sector check for errors

RDSCT
Read a sector check for errors

```
GETSSC
Read system
sector into
sector buffer
B4D5 A925     LDA £$0025
B4D7 8D04B8   STA $B804
B4DA A9B8     LDA £$00B8
B4DC 8D05B8   STA $B805
B4DF A900     LDA £$0000
B4E1 8D01B8   STA $B801
B4E4 8D06B8   STA $B806
B4E7 A901     LDA £$0001
B4E9 8D02B8   STA $B802
B4EC 4CCFB4   JMP $B4CF

BADDEV
check unit for legal
drive number
- error message if not.
B4EF AE00B8   LDX $B800
B4F2 BD15B8   LDA $B815,X
B4F5 F001     BEQ $B4F8
B4F7 60       RTS

- error message if not.
B4F8 A905     LDA £$0005
B4FA 8540     STA $0040
B4FC A9B5     LDA £$00B5
B4FE 8541     STA $0041
B500 200FB5   JSR $B50F
B503 F0C7     BEQ $B4CC
B505 0D4E6F   ORA $6F4E
B508 206472   JSR $7264
B50B 6976     ADC £$0076
B50D 6500     ADC $0000

OUTSTR
string output
B50F A000     LDY £$0000
B511 B140     LDA ($0040),Y
B513 F006     BEQ $B51B
B515 200EF8   JSR $F80E   ---- OUT ALL
B518 C8       INY
B519 10F6     BPL $B511
B51B 60       RTS

B51C A514     LDA $0014
B51E D008     BNE $B528
B520 A513     LDA $0013
B522 3004     BMI $B528
B524 C908     CMP £$0008
B526 3002     BMI $B52A
B528 A9FF     LDA £$00FF
B52A 60       RTS

GDSCDT
Get and save
disc system
sector
information.
B52B 20D5B4   JSR $B4D5
B52E A207     LDX £$0007
B530 BD35B8   LDA $B835,X
B533 9D25B9   STA $B925,X
B536 CA       DEX
B537 10F7     BPL $B530
B539 60       RTS

UPDSYS
Update system
sector from
data in memory
B53A 20D5B4   JSR $B4D5
B53D A207     LDX £$0007
B53F BD25B9   LDA $B925,X
B542 9D35B8   STA $B835,X
B545 CA       DEX
B546 10F7     BPL $B53F
B548 4CC3B4   JMP $B4C3

ENDIRB
Enter a file info
block into a dir.
sector
B54B AD40B9   LDA $B940
B54E 8D01B8   STA $B801
B551 AD3FB9   LDA $B93F
B554 20E9B4   JSR $B4E9
B557 A200     LDX £$0000
B559 AC41B9   LDY $B941
B55C BD2EB9   LDA $B92E,X
B55F 9925B8   STA $B825,Y
```

inspect 13,14
if > than #7
return with
A = #FF

```
B562 C8       INY
B563 E8       INX
B564 E010     CPX £$0010
B566 D0F4     BNE $B55C
B568 EE27B8   INC $B827
B56B 4CC3B4   JMP $B4C3
```

GFREE
Get a free sector for file use

```
B56E 207CB5   JSR $B57C
B571 F008     BEQ $B57B
B573 EE2BB9   INC $B92B
B576 D003     BNE $B57B
B578 EE2CB9   INC $B92C
B57B 60       RTS
```

GDFR
Get a free sector for directory use

```
B57C AD25B9   LDA $B925
B57F F02B     BEQ $B5AC
B581 8D02B8   STA $B802
B584 AD26B9   LDA $B926
B587 8D01B8   STA $B801
B58A 20CFB4   JSR $B4CF
B58D AD26B8   LDA $B826
B590 8D25B9   STA $B925
B593 AD25B8   LDA $B825
B596 8D26B9   STA $B926
B599 38       SEC
B59A AD29B9   LDA $B929
B59D E901     SBC £$0001
B59F 8D29B9   STA $B929
B5A2 AD2AB9   LDA $B92A
B5A5 E900     SBC £$0000
B5A7 8D2AB9   STA $B92A
B5AA A901     LDA £$0001
B5AC 60       RTS
```

GFRDIR
Find a free directory entry

```
B5AD 20CFB4   JSR $B4CF
B5B0 AD27B8   LDA $B827
B5B3 C90F     CMP £$000F
B5B5 D030     BNE $B5E7
B5B7 AD26B8   LDA $B826
B5BA F00B     BEQ $B5C7
B5BC 8D02B8   STA $B802
B5BF AD25B8   LDA $B825
B5C2 8D01B8   STA $B801
B5C5 10E6     BPL $B5AD
B5C7 AD25B9   LDA $B925
B5CA F039     BEQ $B605
B5CC 8D26B8   STA $B826
B5CF AD26B9   LDA $B926
B5D2 8D25B8   STA $B825
B5D5 20C3B4   JSR $B4C3
B5D8 207CB5   JSR $B57C
B5DB A900     LDA £$0000
B5DD AA       TAX
B5DE 9D25B8   STA $B825,X
B5E1 E8       INX
B5E2 D0FA     BNE $B5DE
B5E4 20C3B4   JSR $B4C3
B5E7 A203     LDX £$0003
B5E9 BD25B8   LDA $B825,X
B5EC F007     BEQ $B5F5
B5EE 8A       TXA
B5EF 18       CLC
B5F0 6910     ADC £$0010
B5F2 AA       TAX
```

```
B5F3 D0F4    BNE $B5E9
B5F5 8A      TXA
B5F6 8D41B9  STA $B941
B5F9 AD01B8  LDA $B801
B5FC 8D40B9  STA $B940
B5FF AD02B8  LDA $B802
B602 8D3FB9  STA $B93F
B605 60      RTS
```

GDIREC
Find a particular directory entry

```
B606 20CFB4  JSR $B4CF
B609 AE41B9  LDX $B941
B60C D035    BNE $B643
B60E 20CFB4  JSR $B4CF
B611 A203    LDX £$0003
B613 A928    LDA £$0028
B615 8540    STA $0040
B617 A9B8    LDA £$00B8
B619 8541    STA $0041
B61B A000    LDY £$0000
B61D B140    LDA ($0040),Y
B61F F022    BEQ $B643
B621 A008    LDY £$0008
B623 B92EB9  LDA $B92E,Y
B626 C92A    CMP £$002A
B628 D002    BNE $B62C
B62A A005    LDY £$0005
B62C B92EB9  LDA $B92E,Y
B62F C92A    CMP £$002A
B631 F00B    BEQ $B63E
B633 C93F    CMP £$003F
B635 F004    BEQ $B63B
B637 D140    CMP ($0040),Y
B639 D008    BNE $B643
B63B 88      DEY
B63C 10EE    BPL $B62C
B63E 8A      TXA
B63F 8D41B9  STA $B941
B642 60      RTS
```

GDI
Continue directory search

```
B643 8A      TXA
B644 18      CLC
B645 6910    ADC £$0010
B647 B00E    BCS $B657
B649 AA      TAX
B64A A540    LDA $0040
B64C 6910    ADC £$0010
B64E 8540    STA $0040
B650 90C9    BCC $B61B
B652 E641    INC $0041
B654 4C1BB6  JMP $B61B
B657 AD26B8  LDA $B826
B65A F00B    BEQ $B667
B65C 8D02B8  STA $B802
B65F AD25B8  LDA $B825
B662 8D01B8  STA $B801
B665 10A7    BPL $B60E
B667 A200    LDX £$0000
B669 60      RTS
```

GETSYS
Get system info from system disc

```
B66A AD15B8  LDA $B815
B66D D0FA    BNE $B669
B66F 8D00B8  STA $B800
B672 A901    LDA £$0001
B674 8D03B8  STA $B803
```

```
B677 A915    LDA £$0015
B679 8D04B8  STA $B804
B67C A9B8    LDA £$00B8
B67E 8D05B8  STA $B805
B681 4CDFB4  JMP $B4DF
```

GETFIL
Get a filename from a command line

```
B684 AD0EB8  LDA $B80E
B687 8D2DB9  STA $B92D
B68A B10A    LDA ($000A),Y   cursor position
B68C C920    CMP £$0020      <SPC>
B68E D003    BNE $B693
B690 C8      INY
B691 10F7    BPL $B68A
```

move up to next space

```
B693 A209    LDX £$0009
B695 A920    LDA £$0020
B697 9D2DB9  STA $B92D,X
B69A CA      DEX
B69B D0FA    BNE $B697
```

put spaces ($20) in B92E - B937

```
B69D C8      INY
B69E B10A    LDA ($000A),Y   cursor position
B6A0 88      DEY
B6A1 C93A    CMP £$003A      :
B6A3 D012    BNE $B6B7 ——→
B6A5 88      DEY
B6A6 2017F8  JSR $F817       Hexpack (13,14)
B6A9 201CB5  JSR $B51C
B6AC C9FF    CMP £$00FF
B6AE D003    BNE $B6B3 ——— if 13,14 ≤ FF
B6B0 4CC7B7  JMP $B7C7 ——— if 13,14 > FF — ERROR SEQ
B6B3 8D2DB9  STA $B92D       store number
B6B6 C8      INY
B6B7 A905    LDA £$0005
B6B9 8D43B9  STA $B943
B6BC A200    LDX £$0000
B6BE 8D44B9  STA $B944
B6C1 8D45B9  STA $B945
B6C4 B10A    LDA ($000A),Y
B6C6 C93F    CMP £$003F      ?
B6C8 F010    BEQ $B6DA ——→
B6CA C92A    CMP £$002A      *
B6CC D005    BNE $B6D3
B6CE 8D45B9  STA $B945
B6D1 1007    BPL $B6DA ——→
B6D3 2006B7  JSR $B706
B6D6 C900    CMP £$0000
B6D8 F00C    BEQ $B6E6 ——— if A is not letter or number
B6DA 9D2EB9  STA $B92E,X
B6DD C8      INY
B6DE CE43B9  DEC $B943
B6E1 3003    BMI $B6E6 ——↘
B6E3 E8      INX
B6E4 10DE    BPL $B6C4
```

Store filename in B92E,X

```
B6E6 AD45B9  LDA $B945
B6E9 C92A    CMP £$002A      *
B6EB D006    BNE $B6F3 ——→
B6ED AE44B9  LDX $B944
B6F0 9D2EB9  STA $B92E,X
B6F3 B10A    LDA ($000A),Y
B6F5 C92E    CMP £$002E      .
B6F7 D00C    BNE $B705 ——— Return
B6F9 A902    LDA £$0002
B6FB 8D43B9  STA $B943
```

finish off with 00 if * in NAM
" " 08 if * in EXT

```
B6FE A206    LDX £$0006
B700 A908    LDA £$0008
B702 C8      INY
B703 10B9    BPL $B6BE
B705 60      RTS
```

ALPNUM
Check character is alphanumeric

```
B706 C930    CMP £$0030   0
B708 300C    BMI $B716
B70A C95B    CMP £$005B   [
B70C 1008    BPL $B716
B70E C93A    CMP £$003A   :
B710 3006    BMI $B718
B712 C941    CMP £$0041   A
B714 1002    BPL $B718
B716 A900    LDA £$0000
B718 60      RTS
```

Return with A = 0 if A < 30 or > 5A or 3A < A < 41

INIGDR
Find first directory entry for specified file use

```
B719 202BB5  JSR $B52B
B71C AD28B9  LDA $B928
B71F 8D01B8  STA $B801
B722 AD27B9  LDA $B927
B725 8D02B8  STA $B802
B728 A900    LDA £$0000
B72A 8D41B9  STA $B941
B72D 4C06B6  JMP $B606
```

GEDIRB
Save Pointers to a directory entry

```
B730 A209    LDX £$0009
B732 AC41B9  LDY $B941
B735 B92EB8  LDA $B82E,Y
B738 9D2EB9  STA $B92E,X
B73B C8      INY
B73C E8      INX
B73D E010    CPX £$0010
B73F D0F4    BNE $B735
B741 60      RTS
```

PFILNM
Print a file name (from directory file info block)

```
B742 AE41B9  LDX $B941
B745 A006    LDY £$0006
B747 BD25B8  LDA $B825,X
B74A 200EF8  JSR $F80E
B74D E8      INX
B74E 88      DEY
B74F D0F6    BNE $B747
B751 A92E    LDA £$002E
B753 200EF8  JSR $F80E
B756 A003    LDY £$0003
B758 BD25B8  LDA $B825,X
B75B 200EF8  JSR $F80E
B75E E8      INX
B75F 88      DEY
B760 D0F6    BNE $B758
B762 60      RTS
```

ZSAVE
Save 40/41 & SP.

```
B763 A540    LDA $0040
B765 8D4DB9  STA $B94D
B768 A541    LDA $0041
B76A 8D4EB9  STA $B94E
B76D BA      TSX
B76E E8      INX
B76F E8      INX
B770 8E42B9  STX $B942
B773 60      RTS
```

?*CHK

```
B774 A208    LDX £$0008
B776 BD2EB9  LDA $B92E,X
B779 C93F    CMP £$003F   ?
B77B F04A    BEQ $B7C7    ——— ERROR SEQ
```

```
                B77D C92A    CMP £$002A  *
                B77F F046    BEQ $B7C7   ------> ERROR SEQ
                B781 CA      DEX
                B782 10F2    BPL $B776
                B784 60      RTS
                ------------------------------------------------
ALOAD           B785 2063B7  JSR $B763   --- set up 40/41 & save SP
ENTRY           B788 2057B4  JSR $B457   --- set up error vector
                B78B A000    LDY £$0000
                B78D 98      TYA
                B78E 2087B6  JSR $B687   - get filename.
                B791 48      PHA         ---next cursor character.
                B792 AD2EB9  LDA $B92E      first character of Directory Filename
                B795 C920    CMP £$0020
                B797 D006    BNE $B79F
                B799 68      PLA
                B79A 102B    BPL $B7C7
                B79C 4C80B4  JMP $B480   --- return to monitor
                B79F 2074B7  JSR $B774   --- check for ? or *
                B7A2 203DB4  JSR $B43D   --- set up store loc's
                B7A5 4CD7B2  JMP $B2D7   ---------------------------
                ------------------------------------------------  p/o DLOAD> (Y = next char. pos)
                B7A8 FF      ?
                B7A9 FF      ?
                B7AA FF      ?
                B7AB FF      ?
                B7AC FF      ?
                B7AD FF      ?
                B7AE FF      ?
LDDFLT          B7AF 4C3DB4  JMP $B43D
ESCAPE          B7B2 4CB0B4  JMP $B4B0
ZSAVE           B7B5 4C63B7  JMP $B763
LOADFL          B7B8 4CECB2  JMP $B2EC
INIGDR          B7BB 4C19B7  JMP $B719
NFILE           B7BE 4CFFB2  JMP $B2FF
PFILNM          B7C1 4C42B7  JMP $B742
GETLIN          B7C4 4C62B4  JMP $B462
ERRET           B7C7 6C4FB9  JMP ($B94F)  ---> error handler
QUERY           B7CA 4CB6B4  JMP $B4B6
WRTSCT          B7CD 4CC3B4  JMP $B4C3
RDSCT           B7D0 4CCFB4  JMP $B4CF
GETSSC          B7D3 4CD5B4  JMP $B4D5
BADDEV          B7D6 4CEFB4  JMP $B4EF
OUTSTR          B7D9 4C0FB5  JMP $B50F
GDI             B7DC 4C43B6  JMP $B643
GDSCDT          B7DF 4C2BB5  JMP $B52B
UPDSYS          B7E2 4C3AB5  JMP $B53A
ENDIRB          B7E5 4C4BB5  JMP $B54B
GFREE           B7E8 4C6EB5  JMP $B56E
GDFR            B7EB 4C7CB5  JMP $B57C
GFRDIR          B7EE 4CADB5  JMP $B5AD
GDIREC          B7F1 4C06B6  JMP $B606
GETSYS          B7F4 4C6AB6  JMP $B66A
GETFIL          B7F7 4C84B6  JMP $B684
ALPNUM          B7FA 4C06B7  JMP $B706
GEDIRB          B7FD 4C30B7  JMP $B730
                B800 00      BRK
                B801 00      BRK
                B802 00      BRK
                B803 00      BRK
                B804 00      BRK
                B805 00      BRK
                B806 00      BRK
```

"DBASIC"

```
0>IB960
B960 A901    LDA £$0001
B962 850C    STA $000C  = 01
B964 A900    LDA £$0000
B966 8DC2BF  STA $BFC2  = 00
B969 8516    STA $0016  = 00
B96B 8515    STA $0015  = 00
B96D A2FF    LDX £$00FF
B96F 86A9    STX $00A9  = FF
B971 9A      TXS  = FF
B972 A976    LDA £$0076
B974 A0E0    LDY £$00E0
B976 8518    STA $0018  } = E076   Init Basic
B978 8419    STY $0019  }
B97A 851B    STA $001B  }
                        } = E076   Init Basic
B97C 841C    STY $001C  }
B97E A9C1    LDA £$00C1
B980 A0CF    LDY £$00CF
B982 851D    STA $001D  } = CFC1   ?
B984 841E    STY $001E  }
B986 A994    LDA £$0094
B988 A0D1    LDY £$00D1
B98A 851F    STA $001F  } = D194   Fixed to F.P. conversion
B98C 8420    STY $0020  }
B98E A94C    LDA £$004C
B990 8517    STA $0017  = 4C
B992 851A    STA $001A  = 4C
B994 85C3    STA $00C3  = 4C
B996 8521    STA $0021  = 4C
B998 A94A    LDA £$004A
B99A A0D0    LDY £$00D0
B99C 8522    STA $0022  } = D04A   ?
B99E 8423    STY $0023  }
B9A0 A950    LDA £$0050
B9A2 8531    STA $0031  = 50
B9A4 A938    LDA £$0038
B9A6 8532    STA $0032  = 38
B9A8 A21C    LDX £$001C
B9AA BD1BBA  LDA $BA1B,X
B9AD 95E1    STA $00E1,X
B9AF CA      DEX
B9B0 D0F8    BNE $B9AA
B9B2 2006A8  JSR $A806
B9B5 A903    LDA £$0003
B9B7 85C2    STA $00C2  = 03
B9B9 8A      TXA  = 00
B9BA 85D7    STA $00D7  = 00
B9BC 8587    STA $0087  = 00
B9BE 852F    STA $002F  = 00
B9C0 48      PHA
B9C1 852E    STA $002E  = 00
B9C3 20BAC9  JSR $C9BA
B9C6 A288    LDX £$0088
B9C8 8685    STX $0085  = 88
B9CA A9B8    LDA £$00B8
B9CC A0E1    LDY £$00E1
B9CE 2010CA  JSR $CA10
B9D1 20A8CA  JSR $CAA8
B9D4 86E9    STX $00E9
```

```
B9D6 84EA    STY $00EA
B9D8 20E200  JSR $00E2
B9DB A8      TAY
B9DC D00B    BNE $B9E9
B9DE A900    LDA £$0000
B9E0 8DFBBB  STA $BBFB
B9E3 AC1DB8  LDY $B81D
B9E6 C8      INY
B9E7 D010    BNE $B9F9
B9E9 20E800  JSR $00E8
B9EC 20B1C8  JSR $C8B1
B9EF A8      TAY
B9F0 F003    BEQ $B9F5
B9F2 4C92CD  JMP $CD92
B9F5 A900    LDA £$0000
B9F7 A434    LDY $0034
B9F9 85A6    STA $00A6
B9FB 98      TYA
B9FC 38      SEC
B9FD E901    SBC £$0001
B9FF 8DFCBB  STA $BBFC
BA02 E901    SBC £$0001
BA04 8DFEBB  STA $BBFE
BA07 E901    SBC £$0001
BA09 8DFDBB  STA $BBFD
BA0C E901    SBC £$0001
BA0E 8DFFBB  STA $BBFF
BA11 85A7    STA $00A7
BA13 0600    ASL $0000
BA15 38      SEC
BA16 6600    ROR $0000
BA18 4C39BA  JMP $BA39
BA1B 00      BRK
BA1C E6E9    INC $00E9
BA1E D002    BNE $BA22
BA20 E6EA    INC $00EA
BA22 AD60EA  LDA $EA60
BA25 C920    CMP £$0020
BA27 F0F3    BEQ $BA1C
BA29 4C00A8  JMP $A800
BA2C 00      BRK
BA2D 00      BRK
BA2E 00      BRK
BA2F 00      BRK
BA30 00      BRK
BA31 00      BRK
BA32 00      BRK
BA33 00      BRK
BA34 80      ?
BA35 4F      ?
BA36 C7      ?
BA37 52      ?
BA38 B7      ?
BA39 A9C0    LDA £$00C0
BA3B 8514    STA $0014
BA3D A92F    LDA £$002F
BA3F 851F    STA $001F
BA41 A9FF    LDA £$00FF
BA43 851E    STA $001E
BA45 A900    LDA £$0000
BA47 851C    STA $001C
```

```
BA49 851D    STA $001D
BA4B 8513    STA $0013
BA4D 2032F8  JSR $F832
BA50 2000E8  JSR $E800
BA53 4C21E1  JMP $E121 ──→
BA56 9A      TXS
BA57 9D6DB9  STA $B96D,X
BA5A 8A      TXA
BA5B F0F0    BEQ $BA4D
BA5D 2029F8  JSR $F829
BA60 20F4B7  JSR $B7F4
BA63 AD2DB9  LDA $B92D
BA66 8D00B8  STA $B800
BA69 20D6B7  JSR $B7D6
BA6C 20F1BB  JSR $BBF1
BA6F 20E0BB  JSR $BBE0
BA72 F05E    BEQ $BAD2
```

Load basic.

Amend JMP to T/K E0C6, 7, 8 } "EA"

~~Amend~~ E800

JSR E800

)BASIC.ORG

"S"

```
0>LB960,9
B960 00 00 00 00 00 00 00 00
B968 00 FF 00 00 00 00 00 00
B970 00 00 00 00 00 00 00 00
B978 00 00 42 41 4B 0D 4E 53
B980 50 00 0D 44 45 4C 2F 42
B988 41 4B 20 28 59 2C 4E 2F
B990 42 29 20 3F 00 0D 53 59
B998 4E 00 0D 57 52 50 00 7D
B9A0 B9 82 B9 95 B9 9A B9 20
0>
```

```
0 0  B A K CR NSP
P .  CR D E L / B
A K  - ( Y , N /
B )  - ? - CR S Y
N .  CR W R P 0 3
```

NSP DEL/BAK (Y,N/B) SYN WRP

B99F 0 2 4 6

```
O>DLOAD S ND
B960 BBFC B9A7

O>IB9A7
B9A7 20B5B7 JSR $B7B5   ZSAVE
B9AA 20F7B7 JSR $B7F7   GET FILE NAME
B9AD A208   LDX £$0008
B9AF BD2EB9 LDA $B92E,X
B9B2 9D60B9 STA $B960,X
B9B5 C93F   CMP £$003F  — ?.
B9B7 F065   BEQ $BA1E
B9B9 C92A   CMP £$002A  — *
B9BB F061   BEQ $BA1E
B9BD CA     DEX
B9BE 10EF   BPL $B9AF
B9C0 20CEBB JSR $BBCE   — next chr
B9C3 3059   BMI $BA1E   — Non ascii
B9C5 2017F8 JSR $F817   — HEXPACK

B9C8 5054   BVC $BA1E
B9CA A201   LDX £$0001
B9CC B513   LDA $0013,X
B9CE 9D6BB9 STA $B96B,X
B9D1 9D76B9 STA $B976,X
B9D4 CA     DEX
B9D5 10F5   BPL $B9CC
B9D7 20C1BB JSR $BBC1
B9DA 3042   BMI $BA1E
B9DC 2017F8 JSR $F817   HEXPACK
B9DF 503D   BVC $BA1E
B9E1 38     SEC
B9E2 A2FF   LDX £$00FF
B9E4 E8     INX
B9E5 B513   LDA $0013,X
B9E7 9D6DB9 STA $B96D,X
B9EA 9D78B9 STA $B978,X
B9ED FD6BB9 SBC $B96B,X
B9F0 9D74B9 STA $B974,X
B9F3 8A     TXA
B9F4 F0EE   BEQ $B9E4
B9F6 9026   BCC $BA1E
B9F8 20C5BB JSR $BBC5
B9FB 3045   BMI $BA42
B9FD C8     INY
B9FE EE73B9 INC $B973
BA01 48     PHA
BA02 2017F8 JSR $F817   HEXPACK
BA05 68     PLA
BA06 5016   BVC $BA1E
BA08 C950   CMP £$0050  — "P"
BA0A D016   BNE $BA22
BA0C A513   LDA $0013
BA0E 300E   BMI $BA1E
BA10 C908   CMP £$0008
BA12 100A   BPL $BA1E
BA14 8D6AB9 STA $B96A
BA17 8D3EB9 STA $B93E
BA1A A514   LDA $0014
BA1C F0DA   BEQ $B9F8
```

?.1

```
BA1E A204    LDX £$0004
                                  } SYN ERROR
BA20 D078    BNE $BA9A
BA22 C954    CMP £$0054  — "T"
BA24 D00C    BNE $BA32
BA26 A201    LDX £$0001
BA28 B513    LDA $0013,X
BA2A 9D6FB9  STA $B96F,X
BA2D CA      DEX
BA2E 10F8    BPL $BA28
BA30 30C6    BMI $B9F8
BA32 C952    CMP £$0052  — "R"
BA34 D0E8    BNE $BA1E
BA36 A201    LDX £$0001
BA38 B513    LDA $0013,X
BA3A 9D71B9  STA $B971,X
BA3D CA      DEX
BA3E 10F8    BPL $BA38
BA40 30B6    BMI $B9F8
-------------------------------------
BA42 AD71B9  LDA $B971
BA45 0D72B9  ORA $B972
BA48 F013    BEQ $BA5D
BA4A 18      CLC
BA4B A2FF    LDX £$00FF
BA4D E8      INX
BA4E BD71B9  LDA $B971,X
BA51 9D6BB9  STA $B96B,X
BA54 7D74B9  ADC $B974,X
BA57 9D6DB9  STA $B96D,X
BA5A 8A      TXA
BA5B F0F0    BEQ $BA4D
BA5D 2029F8  JSR $F829    CURSOR OFF
BA60 20F4B7  JSR $B7F4    GET SYS
BA63 AD2DB9  LDA $B92D
BA66 8D00B8  STA $B800
BA69 20D6B7  JSR $B7D6    BAD DEV
BA6C 20F1BB  JSR $BBF1
BA6F 20E0BB  JSR $BBE0
BA72 F05E    BEQ $BAD2
BA74 A202    LDX £$0002  } → DEL/BAK message
BA76 20B4BB  JSR $BBB4   }
BA79 20C4B7  JSR $B7C4    GET CMD LINE
BA7C 2029F8  JSR $F829    CURSOR OFF
BA7F A011    LDY £$0011
BA81 B10A    LDA ($000A),Y
BA83 C942    CMP £$0042
BA85 F025    BEQ $BAAC
BA87 C94E    CMP £$004E
BA89 F00A    BEQ $BA95
BA8B C959    CMP £$0059
BA8D D0E5    BNE $BA74
BA8F 2003A8  JSR $A803
BA92 4C6CBA  JMP $BA6C
BA95 4CA4BB  JMP $BBA4
BA98 A200    LDX £$0000  }
BA9A 20B4BB  JSR $BBB4   } NSP
BA9D 4CC7B7  JMP $B7C7   } ERROR
BAA0 A003    LDY £$0003
BAA2 B979B9  LDA $B979,Y
BAA5 9933B9  STA $B933,Y
BAA8 88      DEY
BAA9 D0F7    BNE $BAA2
```

```
BAAB 60      RTS



BAAC 20A0BA JSR $BAA0
BAAF 20E0BB JSR $BBE0
BAB2 F003   BEQ $BAB7
BAB4 2003A8 JSR $A803
BAB7 20F1BB JSR $BBF1
BABA 20E0BB JSR $BBE0
BABD 20A0BA JSR $BAA0
BAC0 B92EB9 LDA $B92E,Y
BAC3 9D25B8 STA $B825,X
BAC6 E8     INX
BAC7 C8     INY
BAC8 C009   CPY £$0009
BACA D0F4   BNE $BAC0
BACC 20CDB7 JSR $B7CD   WRITE SECTOR.
BACF 4C6CBA JMP $BA6C
BAD2 AD28B9 LDA $B928
BAD5 8D01B8 STA $B801
BAD8 AD27B9 LDA $B927
BADB 8D02B8 STA $B802
BADE 20EEB7 JSR $B7EE   FIND FREE DIR
BAE1 F0B5   BEQ $BA98 —NSP ERROR
BAE3 20E8B7 JSR $B7E8   GET FREE SECTOR
BAE6 F0B0   BEQ $BA98
BAE8 A900   LDA £$0000
BAEA A201   LDX £$0001
BAEC 8D3DB9 STA $B93D
BAEF 8D38B9 STA $B938
BAF2 8E37B9 STX $B937
BAF5 AD01B8 LDA $B801
BAF8 9D39B9 STA $B939,X
BAFB 9D3BB9 STA $B93B,X
BAFE BD76B9 LDA $B976,X
BB01 9540   STA $0040,X
BB03 AD02B8 LDA $B802
BB06 CA     DEX
BB07 10EF   BPL $BAF8
BB09 A207   LDX £$0007
BB0B BD69B9 LDA $B969,X
BB0E 9D27B8 STA $B827,X
BB11 CA     DEX
BB12 10F7   BPL $BB0B
BB14 A20A   LDX £$000A
BB16 A900   LDA £$0000
BB18 A8     TAY
BB19 8D43B9 STA $B943
BB1C 8E44B9 STX $B944
BB1F E8     INX
BB20 AD3EB9 LDA $B93E
BB23 0A     ASL A
BB24 0A     ASL A
BB25 0A     ASL A
BB26 0A     ASL A
BB27 6D3EB9 ADC $B93E
BB2A 8DFFFF STA $FFFF
BB2D B140   LDA ($0040),Y
BB2F 9D25B8 STA $B825,X
BB32 EE43B9 INC $B943
BB35 AD79B9 LDA $B979
BB38 C541   CMP $0041
BB3A D007   BNE $BB43
```

```
BB3C AD78B9 LDA $B978

BB3F C540   CMP $0040
BB41 F049   BEQ $BB8C
BB43 E640   INC $0040
BB45 D002   BNE $BB49
BB47 E641   INC $0041
BB49 E8     INX
BB4A D0E1   BNE $BB2D
BB4C 2082BB JSR $BB82
BB4F 20CDB7 JSR $B7CD   WRITE SECTOR
BB52 20E8B7 JSR $B7E8   GET FREE SECTOR
BB55 D003   BNE $BB5A
BB57 4C98BA JMP $BA98 ---- NSP ERROR
BB5A AD01B8 LDA $B801
BB5D 8D3CB9 STA $B93C
BB60 AD02B8 LDA $B802
BB63 8D3BB9 STA $B93B
BB66 EE37B9 INC $B937
BB69 D003   BNE $BB6E
BB6B EE38B9 INC $B938
BB6E A202   LDX £$0002
BB70 D0A4   BNE $BB16
BB72 A900   LDA £$0000
BB74 E8     INX
BB75 F005   BEQ $BB7C
BB77 9D25B8 STA $B825,X
BB7A D0F8   BNE $BB74
BB7C 8D25B8 STA $B825
BB7F 8D26B8 STA $B826
BB82 AE44B9 LDX $B944
BB85 AD43B9 LDA $B943
BB88 9D25B8 STA $B825,X
BB8B 60     RTS
BB8C 2072BB JSR $BB72
BB8F 20CDB7 JSR $B7CD   WRITE SECTOR
BB92 AD40B9 LDA $B940
BB95 8D01B8 STA $B801
BB98 AD3FB9 LDA $B93F
BB9B 8D02B8 STA $B802
BB9E 20E5B7 JSR $B7E5   ENDIRB
BBA1 20E2B7 JSR $B7E2   UPDSYS
BBA4 200CF8 JSR $F80C
BBA7 2400   BIT $0000
BBA9 1001   BPL $BBAC
BBAB 60     RTS
BBAC 4CB2B7 JMP $B7B2   ESCAPE
BBAF A206   LDX £$0006 } WRP
BBB1 4C9ABA JMP $BA9A }

ERROR
BBB4 BD9FB9 LDA $B99F,X
BBB7 8540   STA $0040
BBB9 BDA0B9 LDA $B9A0,X
BBBC 8541   STA $0041
BBBE 4CD9B7 JMP $B7D9       OUT STRING

BBC1 98     TYA
BBC2 38     SEC
BBC3 ED73B9 SBC $B973
BBC6 C905   CMP £$0005
BBC8 3004   BMI $BBCE
BBCA A204   LDX £$0004 } SYN ERROR
BBCC D0E3   BNE $BBB1 }

BBCE B10A   LDA ($000A),Y
```

```
BBD0 3007    BMI $BBD9



BBD2 C920    CMP £$0020
BBD4 D003    BNE $BBD9
BBD6 C8      INY
BBD7 10F5    BPL $BBCE
BBD9 08      PHP
BBDA 8C73B9  STY $B973
BBDD 88      DEY
BBDE 28      PLP
BBDF 60      RTS
BBE0 20BBB7  JSR $B7BB    FIND FIRST DIRECTORY
BBE3 E000    CPX £$0000
BBE5 F009    BEQ $BBF0
BBE7 AE41B9  LDX $B941
BBEA BD34B8  LDA $B834,X
BBED 30C0    BMI $BBAF
BBEF 8A      TXA
BBF0 60      RTS
BBF1 A208    LDX £$0008
BBF3 BD60B9  LDA $B960,X
BBF6 9D2EB9  STA $B92E,X
BBF9 CA      DEX
BBFA 10F7    BPL $BBF3
BBFC 60      RTS
BBFD 0C      ?
BBFE BF      ?
BBFF 00      BRK
BC00 FF      ?
BC01 FF      ?
BC02 FF      ?
BC03 FF      ?
BC04 FF      ?
BC05 FF      ?
BC06 FF      ?
BC07 FF      ?
BC08 FF      ?
BC09 FF      ?
BC0A FF      ?
BC0B FF      ?
BC0C FF      ?
BC0D FF      ?
BC0E FF      ?
BC0F FF      ?
BC10 FF      ?
BC11 FF      ?
BC12 FF      ?
BC13 FF      ?
BC14 FF      ?
BC15 FF      ?
BC16 FF      ?
BC17 FF      ?
BC18 FF      ?
BC19 FF      ?
BC1A FF      ?
BC1B FF      ?
BC1C FF      ?
BC1D FF      ?
BC1E FF      ?
BC1F FF      ?
BC20 FF      ?
BC21 FF      ?
```