ANTIC FORTH#

'Screens Disk one'

```
  oK

SCR # 0
  0 ( SCREEN INDEX REV H   1/1 ) ;S
  1    >> INTERNAL USE ONLY <<
  2  SCREENS 10 TO 24 CONTAIN EDITOR
  3  SCREEN 30 GIVES FULL LOAD
  4 00/00 SCREEN INDEX
  5 01/01 SOURCE CREDITS
  6 02/06 ERROR MESSAGES
  7 07/0A SYSTEM SETUP / BOOTMAKER
  8 0B/0E SUPERCLONE
  9 0F/0F DECOMP DISSEMBLER DATA
  A       THE EDITOR PACKAGE
  B 10/10 EDITOR LOADER
  C 11/14 SCREEN EDITOR
  D 15/17 LINE EDITOR
  E 18/1A EDITOR ADDITIONS
  F 1B/22 DECOMP/STACKDSP/CDMP/PATCH
 10 23/24 COPIES/DUPLICATE
 11 25/26 FIND
 12 27/27 1.4S KERNEL MODS
 13 28/2E ASSEMBLER
 14 2F/2F DECUS MODS
 15 30/30  FULL LOAD
 16 31/35 HARDWARE/GRAPHICS/SOUND
 17 36/36 PON/POFF
 18 37/38 RS232C SUPPORT
 19 39/39 DISPLAY LIST STUFF
 1A 3A/3A PLAYER/MISSILE
 1B 3B/3B LINK/SETPHYS
 1C 3D/3E LPWORDS
 1D 40/44 FORMATTED LIST PROGRAM
 1E 45/4A CHARSETS/CASE
 1F 4B/4F FORTH79/VAR/SETSYS/BDUMP
 oK

SCR # 1
  0 ******  fig-FORTH  MODEL  ******
  1
  2
  3      Through the courtesy of
  4
  5
  6       FORTH INTEREST GROUP
  7          P. O. BOX 1105
  8       SAN CARLOS, CA. 94070
  9
  A        Implemented on the
  B          ATARI  800/400
  C                by
  D           Steve Calfee
  E              1/26/81
  F              4/01/82
 10    PETER LIPSON/ROBIN ZIEGLER
```

```
 11              4/10/82
 12          HARALD  STRIEPE
 13         5/5/82 - 10/16/82
 14  XL Mods - John Stanley 18Jun85
 15      RELEASE 1.4S     REV.H
 16      WITH COMPILER SECURITY
 17      VARIABLE  LENGTH NAMES
 18  SWITCHABLE TOP OF STACK DISPLAY
 19      DECOMPILER/DISSASSEMBLER
 1A   ENHANCED SCREEN EDITOR & FAST
 1B  EDIT WORDS, BASE BORDER DISPLAY
 1C  ENHANCED SYSTEM SET UP/BOOTMKR
 1D        DRIVE 2 LINK/UNLINK
 1E      Further distribution must
 1F      include the above notice.

SCR # 2
  0 BREAK Abort.
  1
  2 IOCB already open.
  3
  4 Non-existant device.
  5
  6 IOCB is write-only.
  7
  8 Invalid command (for this device
  9 )
  A Device or file not open.
  B
  C Bad IOCB #
  D
  E IOCB is read-only
  F
 10 End Of File
 11
 12 Truncated Record
 13
 14 Device Timeout
 15
 16 Device NAK (Negative AcKnowledge
 17 )
 18 Serial Bus input framing error
 19
 1A Cursor out of range
 1B
 1C Serial Bus data-frame overrun
 1D
 1E Serial Bus data-frame checksum e
 1F rror.

SCR # 3
  0 Device-done error
  1
  2 Read-after-write compare error
  3
  4 Function not implemented in hand
  5 ler
  6 Insufficient RAM
  7
```

```
  8
  9
  A
  B
  C
  D
  E
  F
 10
 11
 12
 13
 14
 15
 16
 17
 18
 19
 1A
 1B
 1C
 1D
 1E
 1F

SCR # 4
  0 (  ERROR MESSAGES  )   135  159
  1  9  8  7  10  ;S
  2 empty stack
  3
  4 dictionary full
  5
  6 has incorrect address mode
  7
  8 isn't unique
  9
  A
  B
  C disc range ??
  D
  E full stack !
  F
 10 disc error !
 11
 12
 13
 14
 15
 16
 17
 18
 19
 1A
 1B
 1C THIS IS IT
 1D
 1E HELP ME!
 1F
```

```
SCR # 5
  0 (  ERROR MESSAGES  )
  1
  2 compilation only, use in definit
  3 ion
  4 execution only
  5
  6 conditionals not paired
  7
  8 definition not finished
  9
  A in protected dictionary
  B
  C use only when loading
  D
  E off current editing screen
  F
 10 declare vocabulary
 11
 12 outside allocated file space
 13
 14 writing off current line
 15
 16
 17
 18
 19
 1A string stack empty !!
 1B
 1C
 1D
 1E
 1F

SCR # 6
  0 (  TARGET COMPILER ERROR MESSAGE
  1 S                 WFR-79JUN02 )
  2
  3
  4 below lower bound of virtual mem
  5 ory
  6 disc compiler assembly error in
  7 mode of
  8 can't find in TARGET
  9
  A target redef.
  B
  C T: error, is it paired with T;
  D ?
  E above virtual memory bounds
  F
 10
 11
 12
 13
 14
 15
 16
 17
```

```
 18
 19
 1A
 1B
 1C
 1D
 1E
 1F

SCR # 7
  0  ( SYS/BOOTMKR 82AUG22 1/4 )
  1    FORTH DEFINITIONS   HEX
  2    SAVENFAs
  3  HERE 1C +ORIGIN !  ( FENCE )
  4
  5  HERE 1E +ORIGIN !  ( DP )
  6
  7  HERE DUP FENCE  ! 0 +ORIGIN -
  8
  9  80 / 1+  CONSTANT  #SECT
  A
  B   CODE CALLDK XSAVE STX, E453
  C    JSR, TYA, PHA, ( STATUS )
  D   XSAVE LDX, PUSH JMP, C;
  E
  F
 10 : DKIO 301 ! ( CMD, DRIVE # )
 11   30A ! ( SECT. # ) 304 !
 12   ( RAM BUFFER ) CALLDK ( DKHND)
 13   DUP 0< IF ." ERROR " 0FF AND
 14   BASE @ SWAP DECIMAL
 15   . BASE ! QUIT ENDIF DROP ;
 16 : WTSEC SWAP 304 ! 130 300 !
 17 ( verif $57->) 50 302 C! SECIO ;
 18  : RDSEC SWAP 304 ! 130 300 !
 19          52 302 C! SECIO   ;
 1A : FORMAT ." FORMAT DRIVE " DUP .
 1B   ." -ARE YOU SURE?" 0 PAD ! PAD
 1C  1      EXPECT PAD C@ 59 ( Y) =
 1D   IF 2100 OR PAD 0 ROT DKIO ELSE
 1E         DROP THEN ;
 1F  0 VARIABLE BOOT ( ->CODE ) -->

SCR # 8
  0 ( SYS SET UP/BOOTMKR   2/4 )
  1 : MAKEBOOT FLUSH  EMPTY-BUFFERS
  2  ." INSERT NEW DISK, TYPE Y" CR
  3 0 PAD ! ( DEFAULT ) PAD 3 EXPECT
  4  PAD C@ 59 = IF 1  52  C! CR
  5  ." Writing sectors:" CR CR BOOT
  6  @   1 DUP .  WTSEC #SECT 0 DO I
  7  80 * +ORIGIN I 2 + WTSEC I 2 +
  8  . LOOP 0 52 C! CR ." BOOT COMPL
  9 ETED" CR THEN ; ( BOOT CODE:)
  A  HERE BOOT ! ( PT TO US )
  B  ASSEMBLER 1FF , 480 , ' V1.4S ,
  C  #SECT # LDA, 0= IF, 0 +ORIGIN ,
  D  1 , ENDIF, N STA,    2C8 C@
  E  # LDA, 2C8 STA,  D01A STA,
```

```
  F     2C6  C@   # LDA, 2C6 STA,
 10   D018 STA,
 11  52 # LDA, 302 STA, 48C LDA, 30A
 12  STA, 48D LDA, 30B STA, ( SCT1 )
 13  1 # LDA, 301 STA, 48A LDA, 304
 14 STA, 48B LDA, 305 STA, ( ORIGIN)
 15  BEGIN, 30A INC, 0= IF, 30B INC,
 16        ENDIF, E453 JSR, 303 LDA,
 17  .A ASL, CS IF, RTS, ( FRETURN )
 18        ENDIF, 304 LDA, 80 # EOR,
 19  304 STA, 0< NOT IF, 305 INC,
 1A  ENDIF, ( BUMP PTR.) N DEC, 0=
 1B UNTIL, 48A LDA, 0A STA, 48B LDA,
 1C  0B STA,  E C@ # LDA, 2E7 STA,
 1D  F C@ # LDA, 2E8 STA, CLC, RTS,
 1E    FORTH
 1F   -->

SCR # 9
  0 ( BACKUP HES 82AUG15   3/4 ) (
  1 35F ARRAY BUCD BLK @ BLOCK A0 +
  2 BUCD 35F CMOVE CODE bg E474 JMP,
  3  C; : BACKUP  BUCD 480 35F CMOVE
  4  480 C ! STOF bg ; )       -->
 15 Fig-FORTH 1.4S FAST BACKUP
 14 Vers.1.2   BY H.E.STRIEPE 1982

 15  START   - commence I/O  SELECT
 16   - write with verify    OPTION
 17 - REBOOT               Insert source disk and
 18 press START,   or select OPTION
 19 to REBOOT              Reading SOURCE disk...
 1A Insert destination disk, press
 1B START, or SELECT Writing DESTINA
 1C TION disk...     }****** DUPLICATION
 1D  SUCCESSFUL ****** }***** DISK I
 1E /O ERROR!TRY AGAIN ***** }******
 1F   BREAK KEY INTERRUPT!  ******


SCR # A
  0 ( scr# A   BOOTMKR/SYS 4/4 )
  1
  2  : DoFORget ( forgets below )
  3       ' TEXT NFA      ( FENCE )
  4         FENCE ! 0 FORGET TEXT ;
  5
  6  : SETSYS   ( SETS RESET PARAM )
  7       LMARGN @ DUP ( MARGINS )
  8       LSB ' V1.4S  4 + C!
  9       MSB ' V1.4S  8 + C!
  A       COL1   @ DUP ( COLORS  )
  B       LSB ' V1.4S  C + C!
  C       MSB ' V1.4S 11 + C!
  D       COL4  C@     ( BORDER  )
  E       LSB ' V1.4S 16 + C!   ;
  F
 10
 11  : HOOK  ( hooks your assembly )
```

```
 12        ( routine into WARMSTRT )
 13        ( ->use HOOK word       )
 14     ~[COMPILE] ' ' V1.4S 1+ ! ;
 15
 16  : UNHOOK     ( restore vector )
 17        E4C0 ' V1.4S 1+ ! ;
 18
 19
 1A
 1B   CR ." system words now availab
 1C le" CR
 1D
 1E  ;S
 1F

SCR # B
  0 ( SIO DISK HNDLR 1/2 )
  1   HX   0 DMACTL C!
  2 : CLNDSP  ." }" C 4 POS.
  3       ."  SUPER  CLONE " ;
  4
  5 CODE SIO XSAVE STX, BOT LDA,
  6          E459 JSR, XSAVE LDX,
  7          BOT STY, BOT 1+ STA,
  8      NEXT JMP, C;
  9 : SERR DUP 0< IF 0 100 U/ BASE @
  A        DECIMAL ." ERROR " SWAP
  B        . BASE ! THEN DROP ;
  C  246 CONSTANT DSKTIM
  D : DSIO ( DISK HNDLR VIA SIO )
  E ( BADDR AUXS UNIT-CMD DATFLG )
  F    303 C! ( SET DATA-FLAG )
 10    301 ! ( DUNIT,CMD) 31 300 C!
 11    ( DEVICE) 30A ! ( AUXES )
 12    304 ! ( BUFER-ADDR ) 0 SIO ;
 13
 14  : DIO DSKTIM @ 306 C! 80 308 !
 15          ( BUFLEN) DSIO ;
 16  : RDSEC 5200 OR 40 DIO ;
 17  : WTSEC 5700 OR 80 DIO ;
 18  : PTSEC 5000 OR 80 DIO ;
 19  : FORMAT PAD 0 ROT 2100 OR 40
 1A    DIO       ;
 1B  : STATUS 4 308 ! 1 306 ! 2EA 0
 1C    ROT 5300 OR 40 DSIO SERR ;
 1D  : BAILOUT CONSOL C@ 2 AND 0= ;
 1E
 1F -->

SCR # C
  0 ( BADSEC WORDS )
  1 TBL GRBGSECT 38A0 , B9 C, 8C ,
  2  99 C, 180 , 88 C, F710 , 60 C,
  3  20 C, D76 , 390 , 4C C, 924 ,
  4  20 C, E85 , 20 C, F32 , 20 C,
  5  E0A , 20 C, E8A , 20 C, FCD ,
  6  20 C, CCF , 57A9 , 85 , 3FA2 ,
  7  1370 , 2C C, 380 , F910 ,
  8  DAA9 , 385 , CA C, F410 ,
```

```
  9  2FA9 , 85 , 4C C, B7D , 4C C,
  A  B87 ,
  B 40 ARRAY ZERO-SECT ZERO-SECT
  C 80 ERASE
  D  CODE SIO XSAVE STX,
  E  E459 JSR, 0< IF, 1 # LDA,
  F  ELSE, 0 # LDA, ENDIF, PHA,
 10  0 # LDA, XSAVE LDX, PUSH
 11    JMP, C;
 12 : 1ERR? IF ." CAN'T DOWNLOAD"
 13   CR ABORT THEN ;
 14 : 2ERR? IF ." CAN'T WRITE BAD"
 15   ."  SECTOR" CR ABORT THEN ;
 16 : BINIT 2001 301 ! ( DOWNLOAD TO
 17  D1: )   GRBGSECT 304 !
 18  80 DUP 303 C! 308 ! 7 306 C!
 19  31 300 C! SIO 1ERR? ;
 1A   BINIT
 1B : BADSEC 30A ! FF01 301 !
 1C   ZERO-SECT 304 ! 80 DUP 303
 1D   C! 308 ! 7 306 C! 31 300 C!
 1E   SIO 2ERR? ;
 1F  -->

SCR # D
  0 ( SUPER-CLONE WORDS )
  1  180 ARRAY BDSECTS
  2  : WSTRT ." PRESS START " CR
  3   BEGIN CONSOL C@ 1 AND END
  4   BEGIN CONSOL C@ 1 AND 0= END
  5   0 4D C! 5 A  POS. ." DOING IT.
  6 .." ; : GSRC  0 GR. 2 C 4 SE.
  7    10  8 POS. ." READING" ;
  8  : GDST  0 GR. 2 3 4 SE.
  9    10  8 POS. ." WRITING" ;
  A  : RSOME ( START SECT., CNT )
  B    0 DO     I 80 *  PAD + OVER
  C    2 RDSEC FE AND BDSECTS I +
  D    C! BAILOUT IF LEAVE ENDIF
  E    1+ LOOP DROP ;
  F : WSOME ( START SECT., CNT )
 10    0 DO DUP I BDSECTS + C@ IF
 11   BADSEC ELSE I 80 *  PAD + SWAP
 12   1 PTSEC 1 = 0= IF 8 ERROR THEN
 13   THEN 1+ LOOP DROP ;
 14  : COPY-SOME ( START, CNT ) OVER
 15    OVER GSRC RSOME GDST WSOME ;
 16    0 VARIABLE ROOM
 17  : SUPER-CLONE2  ." SOURCE DISK
 18  IN DRIVE #2" CR ." DESTINATION
 19  IN DRIVE #1" CR WSTRT
 1A    HIMEM @  PAD - 80 / ROOM ! 1
 1B    2D0 BEGIN DUP 0 > WHILE OVER
 1C    OVER ROOM @ MIN COPY-SOME
 1D    ROOM @ - SWAP ROOM @ + SWAP
 1E    REPEAT DROP DROP 0 GR. ;
 1F     -->

SCR # E
```

```
 0 ( SUPER-CLONE WORDS    3/3 )
 1
 2      180 ARRAY BDSECTS
 3  : WSTRT ." PRESS RETURN " KEY
 4   CR DROP ." Doing it " ;
 5
 6  : CLNMSG   CR
 7    ." Source disk in Dr 2," CR
 8    ." Destin disk in Dr.1," CR ;
 9
 A  : RSOME ( START SECT., CNT )
 B    0 DO     I 80 *  PAD + OVER
 C    2 RDSEC FE AND BDSECTS I +
 D    C! BAILOUT IF LEAVE ENDIF
 E    1+ LOOP DROP ;
 F : WSOME ( START SECT., CNT )
10    0 DO DUP I BDSECTS + C@ IF
11   BADSEC ELSE I 80 *  PAD + SWAP
12   1 WTSEC 1 = 0= IF 8 ERROR THEN
13   THEN 1+ LOOP DROP ;
14  : COPY-SOME ( START, CNT ) OVER
15    OVER SPACE RSOME SPACE WSOME
16    ;    0 VARIABLE ROOM
17  : CLONE        CLNMSG   WSTRT
18    HIMEM @  PAD - 80 / ROOM ! 1
19    2D0 BEGIN DUP 0 > WHILE OVER
1A    OVER ROOM @ MIN COPY-SOME
1B    ROOM @ - SWAP ROOM @ + SWAP
1C    REPEAT DROP DROP CR BELL ;
1D  : BADM WSTRT
1E       2D1 1 DO I  BADSEC LOOP ;
1F   0 GR. GS DMACTL C! BASE ! ;S

SCR # F
 10 ???ADCANDASLBCCBCSBEQBITBMIBNEBP
 11 LBRKBVCBVSCLCCLDCLICLVCMPCPXCPYD
 12 ECDEXDEYEORINCINXINYJMPJSRLDALDX
 13 LDYLSRNOPORAPHAPHPPLAPLPROLROR
 14 RTIRTSSBCSECSEDSEISTASTXSTYTAXTA
 15 YTSXTXATXSTYA
 16
 17
 18    #X))Y,X,YN).A
 19
 1A
 1B
 1C
 1D >> DECOMP DISASSEMBLER STUFF <<
 1E  DO NOT MOVE FROM THIS SCREEN !
 1F

SCR # 10
  0 ( SCREEN ED. LOAD 1.4S 1/1 )
  1 (              HES  82aug4 )
  2    HEX FORTH DEFINITIONS
  3
  4 VOCABULARY EDITOR IMMEDIATE
  5
  6   ' EDITOR 2 + DUP
```

```
  7     VOC-LINK ! 20 +ORIGIN !
  8    FORTH DEFINITIONS
  9
  A  : EDIT    SCR ! POFF
  B         ~[COMPILE] EDITOR ;
  C  HEX 15 LOAD    ( LINE EDITOR )
  D  HEX 1B LOAD    ( DECOMPILER  )
  E  HEX 20 LOAD    ( STACKDISPLAY)
  F  HEX 11 LOAD    ( SCREEN EDIT )
 10     18 LOAD ( ENHANCEMENTS   )
 11     23 LOAD ( copies/backup  )
 12     25 LOAD ( FIND WORD      )
 13     3D LOAD ( PRINTER WORDS  )
 14     4D LOAD ( SYS WORDS      )
 15     3F LOAD ( PNS STUFF      )
 16   : VERIFY  57 245E C! ;
 17   : NOVERIFY 50 245E C! ;
 18   : SYS 0 DMACTL C! 7 LOAD 22
 19     DMACTL C! ; : CLONE B LOAD ;
 1A  CODE  GOBOOT  E477 JMP, C;
 1B   : GO STACKON GS ;
 1C   : ZV VLIST ;
 1D   : WARNON  1 WARNING ! ;
 1E   : WARNOFF 0 WARNING ! ;
 1F ;S

SCR # 11
  0 ( SCREEN EDITOR DLI    1/4 )
  1 (              HEH  2jul82 )
  2 HEX   EDITOR DEFINITIONS
  3 ( DLI for command window   )
  4      0 VARIABLE COL1T
  5      0 VARIABLE COL2T
  6  CODE EDLI    PHA, TXA, PHA,
  7    COL1T LDA, COLRSH  EOR,
  8    DRKMSK AND, TAX, COL2T LDA,
  9    COLRSH EOR, DRKMSK AND,
  A    WSYNC STA,
  B    D017 STX, D018 STA, PLA,
  C    TAX, PLA, RTI,     C;
  D  : COLSET   COL2 C@ DUP F AND
  E       COL1T C! F0 AND COL1 C@
  F       F AND + COL2T C!    ;
 10 ( Sound words for beeps    )
 11  0 VARIABLE TOPFLAG
 12  1 VAR       L#FLG
 13  0 VAR      SPTCH
 14  1 VAR      SFLG
 15  0 VAR      EDVEC
 16
 17 : SOUNDON  1 TO SFLG ;
 18 : SOUNDOFF 0 TO SFLG ;
 19  FORTH DEFINITIONS
 1A : SNDOFF 0 0 0 SOUND ;
 1B  EDITOR DEFINITIONS
 1C : PN  10 * TO SPTCH SFLG IF
 1D       28 0 DO 0      SPTCH A 8
 1E  SOUND LOOP 0 SNDOFF THEN ;
 1F  -->
```

```
SCR # 12
  0 ( SCREEN ED. DLSETMOD  2/4 )
  1 (               JDS 18JUN85 )
  2    EDITOR DEFINITIONS
  3    28 ARRAY EDBF C ARRAY DLSTMP
  4    TBL EDLST 82 C, 40 C, 202 ,
  5        202 , 02 C, 40 C, 47 C,
  6       EDBF , 41 C, DLST @ ,
  7 0 GR. DLST @ 14 + DLSTMP C CMOVE
  8  : DLSET 200 @ TO EDVEC ' EDLI 2
  9 00 ! EDBF TOPFLAG @ 0= IF " UPP"
  A  ELSE " LOW" THEN SYPE " ER HALF
  B  SCR # " SYPE SCR @ 0 <# #S #>
  C  SYPE   "     " SYPE DROP
  D  DLST @ EDLST C + !
  E  EDLST DLST @ 14 + E CMOVE
  F  FF D40E C!    ." }" ;
 10   208 @  VARIABLE KBDVC
 11  CODE KCHK    54 LDA, 10 # CMP,
 12       0< IF, D209 LDA, C # CMP,
 13       0= IF, PLA, RTI, THEN,
 14       THEN,  KBDVC ) JMP, C;
 15  CODE EDKIS  SEI, 209 LDA, C4 #
 16      CMP, >= IF, KBDVC 1+ STA,
 17      208 LDA, KBDVC STA,   THEN,
 18      ' KCHK LSB  # LDA, 208 STA,
 19      ' KCHK MSB  # LDA, 209 STA,
 1A               CLI, NEXT JMP, C;
 1B  CODE EDKIQ  SEI,
 1C        KBDVC       LDA, 208 STA,
 1D        KBDVC 1 +   LDA, 209 STA,
 1E               CLI, NEXT JMP, C;
 1F    -->

SCR # 13
  0 ( SCREEN ED. MOD 1.4S  3/4 )
  1           ( HES 82AUG18   )
  2  EDITOR DEFINITIONS
  3  : EDCLR   COL1 @ COL3 @ 0 GR.
  4        COL3 ! COL1 ! CHRST C@
  5        CHBAS C!  ;
  6  : EDLS           EDCLR COLSET
  7   0 DMACTL C! DLSET 22 DMACTL C!
  8           EDKIS 2203 LMARGN !
  9         COL4 C@ DUP F0 AND SWAP
  A        8 +  F AND + COL0 C! ;
  B  : EDLQ   8F D40E C! EDVEC 200 !
  C  EDKIQ ' V1.4S 4 + C@ 52 C! '
  D   V1.4S 8 + C@ 53 C! 0 DMACTL C!
  E    DLSTMP DLST @ 14 + C CMOVE 22
  F    DMACTL C! FF D40E C!  CR ;
 10
 11
 12  : .L# L#FLG IF 10 0 DO DUP 0 I
 13     POS. I + . LOOP THEN DROP 3
 14     12 POS. ;
 15  : ULL      DUP TOPFLAG ! SCR @
 16      EDLS  EDLQ    BLOCK EDLS
```

```
 17                3 0 POS. + 200
 18       1 DUP 2F0 C! 2FE C! TYPE
 19      CR ." DOIT"  AAAA 2B2 ! ;
 1A  : UL   0 ULL 0 .L#
 1B    0 DUP 2FE C! 2F0 C! 6 PN ;
 1C  : LL   200 ULL 10 .L#
 1D    0 DUP 2FE C! 2F0 C! 7 PN ;
 1E        -->
 1F

SCR # 14
  0 ( SCREEN EDITOR 1.4S   4/4 )
  1   EDITOR DEFINITIONS
  2 : DOIT
  3     10 0 DO -1 2B2 ! 3 I POS.
  4     SCR @ BLOCK I 20 * +
  5     TOPFLAG @ + ICBAL ! 20
  6     ICBLL ! GET DROP LOOP
  7          UPDATE
  8     TOPFLAG @ 0= IF UL ELSE LL
  9     ENDIF ;
  A : FORTH   EDLQ ~[COMPILE] FORTH ;
  B
  C   EDITOR DEFINITIONS
  D : COPY    FORTH  COPY  ;
  E : FLUSH   FORTH  FLUSH ;
  F : FH   FLUSH ;
 10
 11 : L#ON 1 TO L#FLG BASE @ > 8 IF
 12             HEX THEN DOIT ;
 13
 14 : L#OFF  0 TO  L#FLG DOIT ;
 15
 16  FORTH DEFINITIONS
 17 ;S
 18
 19
 1A
 1B
 1C
 1D
 1E
 1F

SCR # 15
  0 ( LINE EDITOR          1/3 )
  1 (  TEXT,  LINE, WHERE USED IN
  2  EDITOR    7/7/80-SRC )
  3 FORTH  DEFINITIONS   HEX
  4
  5
  6 : TEXT                   ( ACCEPT
  7  FOLLOWING TEXT TO PAD *)
  8     HERE  C/L  1+   BLANKS  WORD
  9   HERE  PAD  C/L  1+  CMOVE  ;
  A : #OFLINES B/BUF B/SCR * C/L / ;
  B
  C : LINE             ( RELATIVE TO
  D  SCR, LEAVE ADDRESS OF LINE *)
```

```
  E  DUP  #OFLINES            MINUS
  F AND IF ." NOT ON SCREEN" ABORT
 10 ENDIF  ( KEEP ON THIS SCREEN )
 11       SCR  @  (LINE)  DROP  ;
 12
 13
 14 : WHERE                  ( PRINT
 15  SCREEN # AND IMAGE OF ERROR *)
 16     DUP  B/SCR  /  DUP  SCR  !
 17  ." SCR # "   .
 18     SWAP  C/L  /MOD  C/L  *  ROT
 19   BLOCK  +  CR  C/L  TYPE
 1A     CR  HERE  C@  -  SPACES  5E
 1B  EMIT  ~[COMPILE] EDITOR  QUIT  ;
 1C
 1D
 1E -->
 1F

SCR # 16
  0 ( LINE EDITING COMNDS  2/3 )
  1   EDITOR DEFINITIONS
  2 : -MOVE   ( MOVE IN BLOCK BUFFER
  3  ADDR FROM-2,  LINE TO-1 *)
  4      LINE  C/L  CMOVE  UPDATE  ;
  5
  6 : HL                      ( HOLD
  7  NUMBERED LINE AT PAD *)
  8     LINE  PAD  1+  C/L  DUP  PAD
  9   C!  CMOVE  ;
  A : BL                     ( ERASE
  B  LINE-1 WITH BLANKS *)
  C LINE  C/L  BLANKS  UPDATE ;
  D
  E : SL                    ( SPREAD
  F  MAKING LINE # BLANK *)
 10       DUP  1  -  ( LIMIT )
 11 #OFLINES 2 -  ( FIRST TO MOVE )
 12       DO  I  LINE  I  1+  -MOVE
 13   -1  +LOOP  BL ;
 14 : DL            ( DELETE LINE-1,
 15  BUT HOLD IN PAD *)
 16    DUP  HL  #OFLINES   1   -
 17    DUP  ROT
 18       DO  I  1+  LINE  I  -MOVE
 19  LOOP  BL ;
 1A : CL  ( COPY LINE-2 OF SCREEN-1
 1B  TO PAD )
 1C SCR @ >R SCR ! HL R> SCR ! ;
 1D
 1E -->
 1F

SCR # 17
  0 ( LINE EDITING COMNDS  3/3 )
  1 (                 WFR-790105 )
  2 : RL
  3  ( REPLACE ON LINE-1, FROM PAD )
  4       PAD  1+  SWAP  -MOVE  ;
```

```
 5
 6
 7
 8 : $                        ( PUT
 9  FOLLOWING TEXT ON LINE-1 )
 A       1  TEXT  RL  ;
 B
 C
 D
 E : %                ( INSERT TEXT
 F  FOLLOWING AFTER LINE-1 *)
10     1   TEXT  1+  DUP  SL  RL  ;
11
12 : IL       (  INSERT  PAD  AFTER
13  LINE-1 ) 1+  DUP  SL  RL  ;
14
15 : TL   ( TYPE LINE BY #-1,  SAVE
16  ALSO IN PAD *)
17     DUP . ." $ "
18     DUP C/L *  R#  !       HL
19      PAD  1+  C/L  TYPE  CR ;
1A
1B    FORTH DEFINITIONS
1C
1D : COPY SWAP BLOCK SWAP BLOCK 400
1E      CMOVE UPDATE FLUSH ;
1F ;S

SCR # 18
 0 ( VERS 1.4S MODS  HES  1/3 )
 1    FORTH DEFINITIONS    HEX
 2  : HX HEX 93 2C8 C! ;   DECIMAL
 3  : DX DECIMAL  68  712  C! ;
 4  : BX BINARY  248 712 C! ; HEX
 5  : BS  0006 2C5 ! ;
 6  : WS  0A00 2C5 ! ;
 7  : GS  D006 2C5 ! ;
 8  : NS  94CA 2C5 ! ;
 9    EDITOR DEFINITIONS
 A  : LE  EDIT LL ;
 B  : UE  EDIT UL ;
 C  : N 8 PN SCR @ 1+ EDIT UL ;
 D  : P 4 PN SCR @ 1- EDIT UL ;
 E  : L   SCR @     EDIT UL ;
 F : T 4 PN 9 PN ALT @ @ EDIT UL ;
10 : SL DUP SL 10 < IF UL ELSE
11             LL THEN ;
12
13    FORTH DEFINITIONS
14 : EDT ~[COMPILE] EDITOR ;
15 : UE ~[COMPILE] EDITOR
16      EDITOR UE ;
17    FORTH DEFINITIONS
18 : LE ~[COMPILE] EDITOR
19      EDITOR LE ;
1A    FORTH DEFINITIONS
1B : L&  HEX   ( fast load )
1C      0 DUP WARNING ! DMACTL C!
1D      LOAD  22 DMACTL C! ;
```

```
 1E
 1F  -->

SCR # 19
  0 ( FAST EDIT WORDS      2/3 )
  1 ( ZIEGLER/STRIEPE STUFF )
  2   FORTH DEFINITIONS   HEX
  3 : L. LIST ; : L SCR @ LIST ;
  4 : N SCR @ 1+ LIST ;
  5 : P SCR @ 1- LIST ;
  6 : NL EMPTY-BUFFERS LIST ;
  7 : SHOW 1+ SWAP DO I LIST LOOP ;
  8 : LS ~[COMPILE] EDITOR 1 + SWAP
  9  27 53 C! DO I EDITOR TL LOOP ;
  A : SAVE-BUFFERS  FLUSH ;
  B : ERASE-CORE  EMPTY-BUFFERS ;
  C : TRC NFA ID. ;
  D : T  ALT @ @ LIST ;
  E CODE K  XSAVE STX, TSX, 109 ,X
  F   LDA, PHA, 10A ,X LDA, XSAVE
 10   LDX, PUSH JMP,            C;
 11  : EMPTY  0 8 C! COLD ;
 12    D  ARRAY CDAT
 13    22 TEXT ... ......."
 14    PAD 1+ CDAT C  CMOVE
 15  : DATE 1+ SWAP DO I BLOCK
 16      11 + CDAT SWAP  C CMOVE
 17       UPDATE FLUSH  LOOP ;
 18  : NEWDATE CR ." DATE: " CDAT C
 19     TYPE ." " QUIT ;
 1A  : DATE: 9B TEXT PAD 1+ CDAT C
 1B    CMOVE ; EDITOR DEFINITIONS
 1C  : DATE SCR @ BLOCK 11 + CDAT
 1D     SWAP C CMOVE UPDATE UL ;
 1E  : LIST EDLQ LIST ; : L. LIST ;
 1F   -->

SCR # 1A
  0 ( ZIEGLER/STRIEPE V1.4S 3/3 )
  1   FORTH DEFINITIONS
  2 : N->T  SCR C@ S->D
  3    <# #S #>      ;
  4 : ZERO-BLOCK
  5   SCR @ BLOCK DUP DUP
  6   400 20 FILL                "
  7 \ scr#     empty block       1/1
  8  ;S "    ROT SWAP CMOVE
  9   7 + N->T ROT SWAP CMOVE
  A   UPDATE FLUSH ;
  B : LZERO  1+ SWAP DO I SCR !
  C          ZERO-BLOCK LOOP ;
  D   EDITOR DEFINITIONS
  E : ZERO-BLOCK
  F    EDLQ ZERO-BLOCK EDT UL ;
 10 : NUL EMPTY-BUFFERS UL ;
 11 : NLL EMPTY-BUFFERS LL ;
 12 : LOAD  FLUSH DEPTH 0= IF SCR @
 13         THEN LOAD ;
 14 : T. 9 PN 4 PN ALT @ @ EDIT LL ;
```

```
 15  : N. 9 PN SCR @ 1 + EDIT LL ;
 16 : P. 5 PN SCR @ 1 - EDIT LL ;
 17 : WIPE  ZERO-BLOCK ;
 18 : DRAIN EMPTY-BUFFERS ;
 19 : W    ." }RETURN to wipe, N to
 1A  abort "         KEY 4E NOT = IF
 1B              WIPE THEN ;
 1C   FORTH DEFINITIONS
 1D : DRAIN EMPTY-BUFFERS ;
 1E : WIPE  ZERO-BLOCK ;
 1F   ;S

SCR # 1B
  0 ( HIGH-LEVEL DISSASSEMBLER )
  1   FORTH DEFINITIONS
  2                         TASK DOG
  3 ' (;CODE)    CFA CN .;CODE
  4 ' ;S         CFA CN .;S
  5 ' BRANCH     CFA CN .BR
  6 ' 0BRANCH    CFA CN .0BR
  7 ' (DO)       CFA CN .DO
  8 ' (LOOP)     CFA CN .LOOP
  9 ' (+LOOP)    CFA CN .+LOOP
  A ' LIT        CFA CN .LIT
  B   0D6B           CN .CLIT
  C ' (.")       CFA CN .(.")
  D ' TASK CFA   @   CN .:
  E ' DOG CFA    @   CN .DOES>
  F ' COMPILE    CFA CN .COMP
 10
 11   0 VARIABLE .IP
 12
 13 ' BLK CFA    @   CN .USR
 14 ' .;S CFA    @   CN .CON
 15 ' .IP CFA    @   CN .VAR
 16    60            CN RTS,
 17    40            CN RTI,
 18
 19  -->
 1A
 1B
 1C
 1D
 1E
 1F

SCR # 1C
  0 ( HIGH-LEVEL DISSASSEMBLER )
  1
  2 : PRNAME 2+ NFA ID. ;
  3
  4 : STRNG ( cfa--cfa   prnt strng)
  5   DUP .(.") = IF PRNAME .IP @
  6   DUP COUNT ROT OVER + 1+ .IP !
  7   TYPE CR R> DROP ENDIF ;
  8
  9 : LIT? ( cfa--cfa    prints lit)
  A   DUP .LIT = IF PRNAME .IP @ @ .
  B   CR 2 .IP +! R> DROP ELSE DUP
```

```
  C   .CLIT = IF ." CLIT " DROP .IP
  D   @ C@ . CR 1 .IP +! R> DROP
  E   ENDIF ENDIF ;
  F
 10 : COMP? DUP .COMP = IF PRNAME
 11   .IP @ @ PRNAME CR 2 .IP +! R>
 12   DROP ENDIF ;
 13
 14 : PROMPT 0 2FE ! ." ok " CR ;
 15
 16 : ENDEF ( cfa--cfa  aborts@end)
 17   DUP .;CODE = OVER .;S = OR IF
 18   PRNAME CR PROMPT QUIT ENDIF ;
 19
 1A : BRNCH ( cfa--cfa   prnts dst)
 1B   DUP .BR = OVER .0BR = OR OVER
 1C   .LOOP = OR OVER .+LOOP = OR
 1D   IF PRNAME ." to " .IP @ DUP @
 1E   + .  CR 2 .IP +! R> DROP ENDIF
 1F  ;    -->

SCR # 1D
  0 ( DECOMP DISSASSEMBLER  PBL 82)
  1
  2      F CN OPTAB  ( STD. $F )
  3    200 CN OPOFF  300 CN MODOFF
  4
  5 : 1OP .IP @ DUP HH ." : " C@ 1
  6  .IP +! DUP CHH  SPACE ; ( --op)
  7
  8 : INDX ( off base--addr) + B/BUF
  9  /MOD ~[ OPTAB B/SCR * ] LITERAL
  A  + BLOCK  + ;
  B
  C : OPLUK  ( op--opind modind #op)
  D  DUP + 0 INDX DUP C@ SWAP 1+ C@
  E  40 /MOD ;
  F
 10 : OPANDP ( #bytes--) DUP -DUP IF
 11  .IP @ C@ CHH  SPACE 1 - IF .IP
 12  @ 1+ C@ CHH  ELSE 2 SPACES
 13  ENDIF ELSE 5 SPACES ENDIF
 14  ."  - " -DUP IF 1 - IF .IP @ @
 15  2 ELSE .IP @ C@ 1 ENDIF .IP +!
 16  HH SPACE ELSE 5 SPACES ENDIF ;
 17
 18 : MODP ( modind--) MODOFF INDX 2
 19  TYPE SPACE ;
 1A
 1B : OPP ( opind--) OPOFF INDX 3
 1C  TYPE ." , " CR ;
 1D
 1E -->
 1F

SCR # 1E
  0 ( DECOMP DISSASSEMBLER  PBL 82)
  1 : BR? ( mode #op--mode) OVER 10
  2  = IF .IP @ DUP C@ CHH
```

```
  3  ."     - " DUP C@ DUP 80 AND IF
  4  FF00 OR ENDIF 1+ + HH   .IP +!
  5  SPACE ELSE OPANDP ENDIF ;
  6 : 1LINE 1OP OPLUK BR? MODP OPP ;
  7
  8 : JMPEX ( --f  test endef jmps)
  9  .IP @ C@ 4C = IF .IP @ 1+ @ DUP
  A  ASSEMBLER NEXT = OVER W 1 - =
  B  OR OVER               POP = OR
  C  OVER PUSH = OR OVER PUT = OR
  D  SWAP POPTWO FORTH = OR DUP IF
  E  ENDIF ELSE 0 ENDIF ;
  F
 10 : ;CEND .IP @ C@ DUP RTS, = SWAP
 11   RTI, = OR JMPEX OR ;
 12
 13 : 1WRD BEGIN 1LINE ;CEND UNTIL
 14   1LINE ; : CSEE 1WRD ;
 15
 16 : DIS .IP ! 1WRD ; : NDIS .IP !
 17   0 DO 1LINE LOOP ;
 18
 19 : ISCODE .IP @ DUP 2 - @ = IF
 1A   ." primitive " CR 1WRD ELSE
 1B   .IP @ CFA @ DUP 2 - @ SWAP
 1C   .IP ! .;CODE = IF
 1D   ." ;CODE word" CR      ELSE
 1E   ." odd entry point" CR ENDIF
 1F   1WRD  ENDIF ; -->

SCR # 1F
  0 ( HIGH-LEVEL DISSASSEMBLER )
  1
  2 : ISCOL ( -- <ff> or <pfa tf> )
  3   .IP @ DUP CFA @ .: - IF DUP
  4   CFA @ DUP .DOES> = IF .IP @ @
  5   .IP ! ." DOES> word" CR DROP 1
  6   ELSE SWAP DROP DUP .CON  = IF
  7   ." CONSTANT : " .IP @ @ HH
  8   CR DROP ELSE DUP .USR = IF
  9   ." USER variable " DROP CR
  A   ELSE .VAR = IF ." VARIABLE : "
  B   .IP @ DUP HH @ ."  = " HH CR
  C   ELSE ISCODE ENDIF ENDIF ENDIF
  D   0 ENDIF ELSE 1 ENDIF ;
  E
  F
 10 : NXTW 2 SPACES .IP @ DUP HH
 11   ."  : " @ 2 .IP +! 2 SPACES
 12   LIT? BRNCH COMP? STRNG ENDEF
 13    PRNAME CR    ;
 14
 15 : FETCHW ~[COMPILE] ' .IP !
 16   ISCOL  IF NFA C@ 40 AND IF
 17   ." immediate" CR ENDIF ELSE
 18            PROMPT QUIT ENDIF ;
 19
 1A : DECOMP  1 2FE C! FETCHW BEGIN
 1B   NXTW ?TERMINAL IF
```

```
 1C   PROMPT QUIT ENDIF AGAIN ;
 1D
 1E : ZZ DECOMP ;
 1F  ;S

SCR # 20
  0 ( CONSTANT INFO DISPLAY 1/3 )
  1   FORTH DEFINITIONS
  2
  3 HEX  4E LOAD
  4
  5 TBL XTRN 40 C, 0 C, 20 C, 60 C,
  6
  7 CODE ASCINT BOT LDA,
  8  .A ROL, .A ROL, .A ROL, .A ROL,
  9  03 # AND, TAY, BOT LDA,
  A  9F # AND, XTRN ,Y ORA,
  B  BOT STA,  NEXT JMP, C;
  C
  D : SYPE ( addr, straddr, cnt )
  E   OVER + SWAP DO I C@
  F   ASCINT OVER C! 1+ LOOP ;
 10
 11
 12
 13
 14 HERE DUP 3F + FFC0 AND
 15 SWAP -  ALLOT
 16
 17 28 ARRAY BUF
 18
 19 TBL DLIST
 1A 5070 , 42 C, BUF , 1 C, 0 ,
 1B
 1C HERE 2 - CN DLPTCH
 1D
 1E -->
 1F

SCR # 21
  0 ( CONSTANT INFO DISPLAY  2/3  )
  1  ' ABORT 6 + @  VARIABLE ABORT1
  2  ' QUIT  A + @  VARIABLE QUIT1
  3  : INIT 0 DMACTL C!
  4   DLST @ DUP C@ 1 -
  5   IF DUP 3 + DLPTCH !
  6   1 OVER C! DLIST SWAP 1+ !
  7   ELSE DROP THEN 22 DMACTL C! ;
  8 : DSPLY BUF " TOS->" SYPE >R
  9  ASSEMBLER UP FORTH @ 6 + @ SP@
  A  10 + MIN SP@ BEGIN 2+ OVER OVER
  B  > WHILE R> OVER @
  C   0 <# # # # # #>
  D  SYPE 1+ >R REPEAT DROP DROP R>
  E  "                      fig-FORT
  F H 1.4S"    SYPE  DROP ;
 10
 11 : SSK    DSPLY INIT CR
 12    BASE @ DUP A = IF 44 2C8 C! E
```

```
 13 LSE DUP 10 = IF 93 2C8 C! ELSE D
 14 UP 2 = IF F8 2C8 C! ELSE DUP 4 2
 15 C8 C! ENDIF ENDIF ENDIF DROP
 16 CHRST C@ CHBAS C! ;
 17
 18 : STACKON   (  HES MOD 12jun82 )
 19   ' SSK      CFA ' ABORT 6 + !
 1A   ' SSK      CFA ' QUIT  A + !
 1B   ' FIX~ CFA '  ~  40 + ! ;
 1C
 1D -->
 1E
 1F

SCR # 22
  0 ( CONST. INFO. / CDUMP 3/3 )
  1 : STACKOFF
  2     ABORT1 @ ' ABORT 6 + !
  3     QUIT1  @ ' QUIT  A + !
  4     2C5 @ 2C8 C@ 0 GR.
  5     2C8 C! 2C5 ! ;
  6
  7 : STON  STACKON ;
  8 : STOF  STACKOFF ;
  9
  A ( HES V.2.0 82SEP9          )
  B :     (  FETCHES LETTERS AND ! )
  C     7E TEXT 10 0 DO DUP
  D     PAD 1+ I + C@ SWAP I + C!
  E     LOOP DROP PAD FIX~ QUIT ;
  F
 10 : CDUMP   ( adr1 adr2 --- )
 11        1 2FE C! 1+ SWAP DO
 12     I HH ."          " I 10 0 DO
 13     DUP I + C@ EMIT LOOP
 14     DROP SPACE 7E EMIT CR
 15     10 +LOOP 0 2FE C! ;
 16 : PATCH \ new  old --- JAP AUG82
 17     ~[COMPILE] ' CFA ~[COMPILE] '
 18     DUP >R ! ' ;S CFA R> 2+ ! ;
 19 \ Debugging ML rout  HES17OCT82
 1A CODE JMP \ INDIRECT JMP/ML DEBUG
 1B     BOT LDA, N STA, BOT 1+ LDA,
 1C     N 1+ STA,  N ) JMP,  C;
 1D CODE JSR \ INDIRECT JSR/ML DEBUG
 1E     XSAVE STX, ' JMP JSR, XSAVE
 1F     LDX, POP JMP,   C;   ;S

SCR # 23
  0 ( ONE DRIVE.DUPSCR/COPS1/2 )
  1 (   by anonymous/HES 23jun82 )
  2  0 VARIABLE EBLK ( ENDING BLK )
  3  0 VARIABLE SBLK ( START. BLK )
  4  0 VARIABLE PSBLK
  5 : DISP ( -> DEST ADR INFRE RAM )
  6  PSBLK @ B/BUF * HERE 20 + + ;
  7 : GTPAR ( SET UP DO AND PSBLK )
  8 EBLK @ SBLK @ 0 PSBLK ! ;
  9 : MVIN ( MOVE BLOCKS INTO RAM )
```

```
  A GTPAR DO I BLOCK DISP B/BUF
  B CMOVE 1 PSBLK +! LOOP ;
  C : MOVOT ( WRITE RAM TO DISK )
  D GTPAR OFFSET @ + SWAP OFFSET @ +
  E
  F SWAP DO I BUFFER DISP SWAP B/BUF
 10  CMOVE 1 PSBLK +! UPDATE FLUSH
 11 LOOP ;
 12 : DUPLICATE  ( STARTSCR--ENDSCR)
 13  1+ B/SCR * EBLK ! B/SCR *
 14 SBLK ! EBLK @ SBLK @ -
 15  FREE 20 - 400 /
 16     > IF ." TOO MANY " QUIT
 17 ENDIF CR MVIN
 18 ." INSERT DESTINATION DISK " CR
 19  ." RETURN TO CONTINUE " KEY
 1A DROP CR MOVOT ;
 1B
 1C
 1D
 1E
 1F  -->

SCR # 24
  0 ( COPIES          HES  2/2 )
  1 (        82JUN18 / 82AUG14 )
  2   FORTH DEFINITIONS
  3
  4
  5
  6 : CPST     CR ." ? Incorrect scr
  7 een range" CR QUIT ;
  8 : CPNT CR ." scr# " SWAP DUP .
  9      ."  --> " SWAP DUP . ;
  A : CPMP EBLK @ SBLK @ - DUP PSBLK
  B      @ + PSBLK ! 1+ 0  DO
  C      EBLK @ I - PSBLK @ I - CPNT
  D      COPY LOOP ;
  E : CPMD  EBLK @ SBLK @ - 1+ 0 DO
  F      SBLK @ I + PSBLK @ I + CPNT
 10       COPY LOOP  ;
 11 : COPIES PSBLK ! EBLK ! SBLK !
 12       EBLK @ SBLK @ < IF CPST
 13       THEN PSBLK @ SBLK @ > IF
 14       CPMP ELSE CPMD
 15       ENDIF CR  ; IMMEDIATE
 16
 17
 18
 19
 1A
 1B  ;S
 1C
 1D
 1E
 1F

SCR # 25
  0 ( FIND  V.1.1         1/2 )
```

```
 1 ( by R.Mansfield/COMPUTE!  )
 2 ( adapt.&enhanced HES 82aug7 )
 3   FORTH DEFINITIONS   HEX
 4
 5   0 VARIABLE 1STCHAR
 6
 7 : ?CONSOL   -2FE1 C@ 7 XOR ;
 8 : MATCH  ( addr1 addr2 N --- F )
 9  -DUP IF OVER + SWAP
 A        DO DUP C@ I C@ -
 B         IF 0= LEAVE ELSE 1+ THEN
 C        LOOP
 D       ELSE DROP 0= THEN ;
 E : CHECKIT ( addr --- F )
 F        PAD 1+ PAD C@ MATCH ;
10  : HEADER
11       CR ." Searching for "
12       22 EMIT SPACE PAD 1+ PAD
13       C@ TYPE 22 EMIT CR CR
14       ." on scr #"     ;
15 : MARKSTRING
16        ( scr# addr --- scr# )
17      OVER BLOCK - C/L / CR DUP
18      CR CR ." Found on LINE#"
19      CR CR . SPACE OVER .LINE
1A      CR CR CR ." scr#" ;
1B : ?STCK    DEPTH 2 < IF
1C      0 59 PHYSOFF @ -
1D      ENDIF      ;
1E   -->
1F

SCR # 26
 0 ( FIND                 2/2 )
 1
 2  CODE ?CHAR  ( addr --- addr F )
 3    1 # LDA, SETUP JSR,
 4    N )Y LDA, 1STCHAR CMP, 0=
 5    IF, 1 # LDA, PHA, 0 # LDA,
 6    PUSH JMP, THEN,
 7    0 # LDA, PHA, PUSH JMP, C;
 8
 9  : ONEBLK     ( scr# addr --- )
 A     DUP 400 + SWAP
 B       DO I ?CHAR
 C         IF I CHECKIT
 D           IF I MARKSTRING ENDIF
 E         ENDIF
 F       LOOP DROP ;
10
11  : GTWRD  22 WORD HERE DUP C@ 1+
12          PAD SWAP CMOVE ;
13  : FIND ( scr#1 scr#2 text --- )
14      ?STCK GTWRD
15      0 SCR ! PAD 1+ C@ 1STCHAR !
16      HEADER 1+ SWAP
17       DO I DUP DUP SPACE
18          . BLOCK  ONEBLK
19          ?CONSOL IF CR
```

```
 1A          LEAVE ENDIF
 1B       LOOP CR CR
 1C      ." Search ended" CR ;
 1D   ;S
 1E
 1F

SCR # 27
  0 ( VERS1.4S KERNEL ADD  1/1 )
  1 (        REZ / HES 15sep82 )
  2 ( Already in kernel, doc.only)
  3   FORTH DEFINITIONS  HEX
  4   4F LOAD
  5 : NOT 0= ;
  6 : U. 0  D. ;
  7 : CN CONSTANT ;
  8
  9 : (") R> DUP COUNT + >R COUNT ;
  A : " COMPILE (") 22 WORD HERE C@
  B     1+ ALLOT ; IMMEDIATE
  C : DEPTH EA SP@ - 2 / ;
  D : .S CR DEPTH IF EA EA DEPTH 2 -
  E      2* - SWAP DO I ? -2 +LOOP
  F      ELSE 1 MESSAGE
 10     ENDIF ;
 11 : SAVENFAs  #LINKS  0 DO
 12     1CFC 4 + I 4 * + @ 22 I 2*
 13     + +ORIGIN !     LOOP ;
 14  (              HES 82AUG21  )
 15  CODE V1.4S  ( DOSINI VECTOR )
 16     E4C0 JSR, ( APPL.HOOK   )
 17     0 # LDA, 52 STA, ( MARGN )
 18    27 # LDA, 53 STA, (   "   )
 19     6 # LDA, 2C5 STA, D0 # LDA,
 1A     2C6 STA, 93 # LDA, 2C8 STA,
 1B               ( SCREEN COLORS )
 1C                  RTS,  C;
 1D
 1E
 1F    ;S

SCR # 28
  0 (  FORTH-65 ASSEMBLER  1/6 )
  1              ( WFR-79JUN03 )
  2 HEX
  3
  4 VOCABULARY  ASSEMBLER  IMMEDIATE
  5 ' ASSEMBLER 2 + DUP 20 +ORIGIN !
  6                     VOC-LINK !
  7      ASSEMBLER  DEFINITIONS
  8  ( LOCATE EXISTING REGISTERS )
  9
  A FF CONSTANT  XSAVE     0FB  CONS
  B TANT  W    0FD  CONSTANT  UP
  C F8  CONSTANT  IP          F0  CO
  D NSTANT  N
  E
  F
 10 (  LOCATE EXISTING CODE PROCEEDU
```

```
 11 RES )
 12 '  (DO)  0E  +    CONSTANT  POP
 13    ( FROM COMPUTATION STACK *)
 14 '  (DO)  0C  +    CONSTANT  POPT
 15 WO
 16 '  LIT  13  +  CONSTANT  PUT
 17
 18 '  LIT  11  +  CONSTANT  PUSH
 19
 1A '  LIT  18  +  CONSTANT  NEXT
 1B
 1C '  EXECUTE  NFA  11  -  CONSTANT
 1D   SETUP
 1E     -->
 1F

SCR # 29
  0 (  FORTH-65 ASSEMBLER  2/6 )
  1              ( WFR-78OCT03 )
  2    0 VARIABLE INDEX -2 ALLOT
  3
  4 0909 , 1505 , 0115 , 8011 , 8009
  5  , 1D0D , 8019 , 8080 , 0080 , 1
  6 404 , 8014 , 8080 , 8080 , 1C0C
  7 , 801C , 2C80 ,
  8
  9    2 VARIABLE MODE
  A
  B  : .A 0 MODE ! ; : # 1 MODE ! ;
  C  : MEM 2 MODE ! ;
  D  : ,X 3 MODE ! ; : ,Y 4 MODE ! ;
  E  : X) 5 MODE ! ; : )Y 6 MODE ! ;
  F  : )  F MODE ! ;
 10  : BOT   ,X 0 ; ( ADDRESS BOTTOM
 11                    OF STACK    )
 12
 13  : SEC   ,X 2 ; ( ADDRESS SECOND
 14                 ITEM ON STACK  )
 15
 16  : RP)   ,X 101 ;
 17                 ( ADDRESS BOTTOM
 18                OF RETURN STACK )
 19
 1A
 1B
 1C
 1D    -->
 1E
 1F

SCR # 2A
  0 (  UPMODE,  CPU        3/6 )
  1              ( WFR-78OCT23 )
  2
  3
  4 : UPMODE   IF  MODE  C@  8  AND
  5  0=  IF  8  MODE +! ENDIF ENDIF
  6      1 MODE C@  0F AND  -DUP  IF
  7   0  DO  DUP + LOOP  ENDIF
```

```
 8      OVER  1+  @  AND  0=  ;
 9
 A
 B
 C : CPU   <BUILDS  C,  DOES>  C@
 D  C,  MEM  ;
 E      00 CPU BRK,   18 CPU CLC,
 F  D8 CPU CLD,  58 CPU CLI,
10      B8 CPU CLV,   CA CPU DEX,
11  88 CPU DEY,  E8 CPU INX,
12      C8 CPU INY,   EA CPU NOP,
13  48 CPU PHA,  08 CPU PHP,
14      68 CPU PLA,   28 CPU PLP,
15  40 CPU RTI,  60 CPU RTS,
16      38 CPU SEC,   F8 CPU SED,
17  78 CPU SEI,  AA CPU TAX,
18      A8 CPU TAY,   BA CPU TSX,
19  8A CPU TXA,  9A CPU TXS,
1A      98 CPU TYA,
1B
1C -->
1D
1E
1F

SCR # 2B
  0 (  M/CPU,   MULTI-MODE 4/6 )
  1 (     OP-CODES    WFR-79MAR26 )
  2 : M/CPU   <BUILDS  C,  ,  DOES>
  3
  4        DUP  1+  C@  80  AND  IF
  5  10  MODE  +!  ENDIF  OVER
  6        FF00  AND  UPMODE  UPMODE
  7   IF  MEM  CR  LATEST  ID.
  8        3  ERROR  ENDIF  C@  MODE
  9   C@
  A        INDEX  +  C@  +  C,  MODE
  B   C@  7  AND  IF  MODE  C@
  C        0F  AND  7  <  IF  C,  EL
  D SE  ,  ENDIF  ENDIF  MEM  ;
  E
  F
 10    1C6E 60 M/CPU ADC,  1C6E 20 M
 11 /CPU AND,  1C6E C0 M/CPU CMP,
 12    1C6E 40 M/CPU EOR,  1C6E A0 M
 13 /CPU LDA,  1C6E 00 M/CPU ORA,
 14    1C6E E0 M/CPU SBC,  1C6C 80 M
 15 /CPU STA,  0D0D 01 M/CPU ASL,
 16    0C0C C1 M/CPU DEC,  0C0C E1 M
 17 /CPU INC,  0D0D 41 M/CPU LSR,
 18    0D0D 21 M/CPU ROL,  0D0D 61 M
 19 /CPU ROR,  0414 81 M/CPU STX,
 1A    0486 E0 M/CPU CPX,  0486 C0 M
 1B /CPU CPY,  1496 A2 M/CPU LDX,
 1C    0C8E A0 M/CPU LDY,  048C 80 M
 1D /CPU STY,  0480 14 M/CPU JSR,
 1E    8480 40 M/CPU JMP,  0484 20 M
 1F /CPU BIT,       -->
```

```
SCR # 2C
  0 ( ASSEMBLER CONDITIONALS 5/6)
  1               ( WFR-79MAR26 )
  2 : BEGIN,     HERE  1  ;
  3         IMMEDIATE
  4 : UNTIL,    ?EXEC >R 1 ?PAIRS R>
  5  C, HERE 1+ - C, ; IMMEDIATE
  6 : IF,        C,  HERE  0  C,  2
  7 ;       IMMEDIATE
  8 : ENDIF,   ?EXEC  2  ?PAIRS  HER
  9 E  OVER  C@
  A  IF  SWAP  !  ELSE  OVER  1+  -
  B  SWAP  C!  ENDIF  ; IMMEDIATE
  C : ELSE,    2  ?PAIRS  HERE  1+
  D 1  JMP,
  E       SWAP  HERE  OVER  1+  -  S
  F WAP  C!  2  ;  IMMEDIATE
 10 : NOT    20  +  ;
 11      ( REVERSE ASSEMBLY TEST )
 12 90  CONSTANT  CS               (
 13  ASSEMBLE TEST FOR CARRY SET )
 14 D0  CONSTANT  0=             ( A
 15 SSEMBLER TEST FOR EQUAL ZERO )
 16 10  CONSTANT  0<           ( ASSE
 17 MBLE TEST FOR LESS THAN ZERO )
 18 90  CONSTANT  >=   ( ASSEMBLE TE
 19 ST FOR GREATER OR EQUAL ZERO )
 1A                      ( >= IS ONLY
 1B  CORRECT AFTER SUB, OR CMP,  )
 1C CR   -->
 1D
 1E
 1F

SCR # 2D
  0 (  USE OF ASSEMBLER    6/6 )
  1               ( WFR-79APR28 )
  2 : C;
  3     ( END OF CODE DEFINITION *)
  4    CURRENT  @  CONTEXT  !  ?EXEC
  5   ?CSP  SMUDGE  ; IMMEDIATE
  6
  7
  8 FORTH  DEFINITIONS
  9
  A : CODE      ( CREATE WORD AT ASS
  B EMBLY CODE LEVEL *)
  C       ?EXEC  CREATE  ~[COMPILE]
  D ASSEMBLER
  E       ASSEMBLER  MEM  !CSP  ;
  F    IMMEDIATE
 10 DECIMAL   ;S    ( TILL figFORTH
 11 IS UP )
 12 '  ASSEMBLER  CFA    '  ;CODE  8
 13   +  !  ( OVER-WRITE SMUDGE )
 14 FORTH  DEFINITIONS  DECIMAL
 15 ;S
 16 LATEST  12  +ORIGIN  !  ( TOP NF
 17 A )
```

```
 18 HERE    28  +ORIGIN  !  ( FENCE
 19 )
 1A HERE    30  +ORIGIN  !  ( DP )
 1B
 1C '  ASSEMBLER  6  +    32  +ORIGI
 1D N  !  ( VOC-LINK )
 1E HERE  FENCE  !   ;S
 1F

SCR # 2E
  0 (  compile assembler   1/1 )
  1    and editor  SRC 7/6/80   )
  2 BASE  @  ( PRESERVE THE RADIX )
  3
  4 DECIMAL  31  WIDTH  !
  5
  6 HEX 28 LOAD ( ASSEMBLER )
  7
  8 HEX 2F LOAD ( DECUS FORTH ADDS)
  9
  A HEX 27 LOAD ( VERS 1.4S KERNEL )
  B
  C HEX 30 LOAD   ( EDITOR & OTHER
  D                     WORDS  )
  E FORTH  DEFINITIONS
  F
 10 25 CONSTANT LPWORDS
 11
 12 27 CONSTANT FORMAT DECIMAL
 13
 14 LATEST 12 +ORIGIN ! ( TOP NFA )
 15
 16 HERE  28 +ORIGIN ! ( FENCE )
 17
 18 HERE    30  +ORIGIN  !  ( DP )
 19
 1A HERE  FENCE  !
 1B
 1C 1 WARNING ! ( DISK WARNINGS )
 1D
 1E : TASK  ;  BASE !    ;S
 1F

SCR # 2F
  0 (  DECUS/FORTH MODS    1/1 )
  1
  2 : 1+! 1 SWAP +! ; : 1- 1 - ;
  3
  4 : 0SET 0 SWAP ! ;
  5
  6 : HD DUP 0A < IF 30 ELSE 37
  7   ENDIF + EMIT ;
  8 : CHH DUP 0F0 AND 10 / HD 0F AND
  9   HD ;
  A : CH? C@ CHH ;
  B
  C : HH DUP 0FF00 AND 100 / 0FF AND
  D   CHH  CHH  ;
  E : H? @ HH ;
```

```
  F
 10
 11 : BDUMP 1+ SWAP DO I HH ." : " I
 12   8 0 DO DUP I + CH? SPACE
 13   LOOP  DROP ." \" CR 8 +LOOP ;
 14
 15 : TBL <BUILDS DOES> ;
 16 : ALLOC DUP + ALLOT ; ( FOR RAM
 17 BASED SYSTEMS,)
 18 : ARRAY <BUILDS ALLOC DOES> ;
 19
 1A  ;S
 1B
 1C
 1D
 1E
 1F

SCR # 30
  0 ( FULL UTILITY LOAD REV H HES )
  1     FORTH DEFINITIONS  HEX
  2 ( VLIST patches   HES17OCT82  )
  3   : v1  ( patch beginning )
  4       1 2FE C! ;
  5   : v2  ( patch SPACE after ID.)
  6       55 @ D < IF D 55 ! ELSE 55
  7       @ 1A <  IF 1A 55 ! ELSE CR
  8       THEN THEN ;
  9   : v3  ( patch last CR )
  A       CR 0 2FE C! ;
  B  ' v1  CFA ' VLIST 6 + !
  C  ' DUP CFA ' VLIST 55 + !
  D  ' v2  CFA ' VLIST 95 + !
  E  ' v3  CFA ' VLIST 9B + !
  F  800 ' DR1 2 + ! ( FX DR1 - 810)
 10    HEX 4C LOAD ( VAR/VALUE )
 11    HEX 4A LOAD ( PICK/ROLL )
 12    HEX 45 LOAD ( CASE      )
 13    HEX 46 LOAD ( CHRSET    )
 14    HEX 4B LOAD ( FIG 79    )
 15    HEX 31 LOAD ( CIO/GRAPH )
 16    HEX 36 LOAD ( PON/POFF  )
 17    HEX 37 LOAD ( RS 232C   )
 18    HEX 39 LOAD ( DISPLLST  )
 19    HEX 3B LOAD ( DRIVE LINK)
 1A    HEX 10 LOAD ( EDITOR    )
 1B      FORTH DEFINITIONS
 1C   NOVERIFY GO 1 CHR       ;S
 1D
 1E
 1F

SCR # 31
  0 ( fig-FORTH 1.4S MODS  1/1 )
  1 (             HES 82JUN17  )
  2    FORTH DEFINITIONS  HEX
  3 : BELL C0 0 DO 8 D01F C! 6 0 DO
  4           LOOP 0 D01F C! 6 0 DO
  5           LOOP LOOP ;
```

```
 6   : BINARY  2 BASE ! ;
 7   : BIN   BINARY ;   HEX
 8   : OCTAL 8 BASE ! ;
 9   : OCT OCTAL ;  HEX
 A
 B   : TASK  <BUILDS DOES> ;
 C
 D   : MSBYTE 0 100 U/ SWAP DROP ;
 E   : LSBYTE FF AND ;
 F   : MSB MSBYTE ; : LSB LSBYTE ;
10   : ><    DUP LSBYTE 100 * SWAP
11           MSBYTE + ;
12 CR ." CIO CALLS" CR    32 LOAD
13 CR ." OS/HARDWARE" CR  33 LOAD
14 CR ." GRAPH/SOUND" CR  34 LOAD
15
16   FORTH DEFINITIONS
17 : THERE  MEMTOP @ ;
18 : FREE THERE HERE - ;
19
1A ;S
1B
1C
1D
1E
1F

SCR # 32
 0 ( CIO CALL ROUTINES )
 1
 2 340 VARIABLE IOC 0 VARIABLE IOB
 3
 4 : IOCB 7 MIN 0 MAX 10 * DUP IOB
 5 ! 340 + IOC ! ;
 6 : .IOC <BUILDS , DOES> @ IOC @ +
 7  ;
 8 1 .IOC ICDNO 2 .IOC ICCOM 3 .IOC
 9  ICSTA
 A 4 .IOC ICBAL 6 .IOC ICPTL
 B
 C 8 .IOC ICBLL A .IOC I1CAX B .IOC
 D  I2CAX
 E CODE CIO TXA, PHA, IOB LDX, E456
 F  JSR, PLA, TAX, NEXT JMP, C;
10 CODE Get XSAVE STX, IOB LDX, E45
11 6 JSR,
12 XSAVE LDX, PHA, 0 # LDA, PUSH JM
13 P, C;
14 : GET 7 ICCOM C! Get ;
15
16 : CLOSE 0C ICCOM C! CIO ;
17
18 : OPEN 3 ICCOM C! ICBAL ! I1CAX
19 C! I2CAX C! CIO ;
1A CODE ACIO XSAVE STX, BOT LDA, IO
1B B LDX, E456 JSR,
1C XSAVE LDX, POP JMP, C;
1D
1E ;S
```

```
 1F

SCR # 33
  0 ( OS & HDW CONSTANTS   1/1 )
  1   FORTH DEFINITIONS    HEX
  2 D200 CN F1AUD D201 CN C1AUD
  3
  4 D202 CN F2AUD D203 CN C2AUD
  5
  6 D204 CN F3AUD D205 CN C3AUD
  7
  8 D206 CN F4AUD D207 CN C4AUD
  9
  A D20F CN SKCTL D208 CN AUDCTL
  B
  C  230 CN DLST   22F CN DMACTL
  D
  E   14 CN RTCLK  2F0 CN CRSINH
  F
 10  2F4 CN CHBAS  2C4 CN COL0
 11
 12  2C5 CN COL1   2C6 CN COL2
 13
 14  2C7 CN COL3   2C8 CN COL4
 15
 16 D01F CN CONSOL 2FC CN CH
 17
 18  2BF CN BOTSC   52 CN LMARGN
 19
 1A 2FB CN ATACHR  2E5 CN MEMTOP
 1B
 1C  4D CN ATRACT   4E CN DRKMSK
 1D
 1E  4F CN COLRSH D40A CN WSYNC
 1F   ;S

SCR # 34
  0 ( COLLEEN GRAPHICS     1/2 )
  1
  2 3A53 VARIABLE S: 1 VARIABLE COLO
  3 RC 0 VARIABLE Qbase
  4 : PBASE Qbase @ ;
  5 : GR. 1 IOCB CLOSE 0 ICBLL ! DUP
  6  F AND SWAP
  7 30 AND 10 XOR 0C + S: OPEN MEMTO
  8 P @ 1 + F800 AND 800 -
  9 DUP Qbase ! 17F + MEMTOP ! ; : P
  A OS. 54 C! 55 ! ;
  B 0 GR. : LOC. POS. GET ;
  C : C. DUP COLORC ! ATACHR C! ;
  D : SPB  HIMEM @ F800 AND 800 -
  E    DUP Qbase ! 17F + HIMEM ! ;
  F
 10 : PUT 0B ICCOM C! ACIO ;
 11
 12 : PL. POS. COLORC @ PUT ;
 13   2FD CN FILDAT
 14 : SE. SWAP 10 * + SWAP 2C4 + C!
 15 ; : DR.  POS. 11 ICCOM C! COLORC
```

```
 16          C@ DUP ATACHR C! FILDAT
 17          C! CIO ;
 18 : GRAPHICS GR. ; : PLOT PL. ;
 19 : LOCATE LOC. ;
 1A : SETCOLOR SE. ; : COLOR C. ;
 1B : POSITION POS. ; : DRAWTO DR. ;
 1C  : CLEAR 0 0 POS. 7D PUT ;
 1D  : XIO18 DUP FILDAT C! ATACHR C!
 1E          12 ICCOM C! CIO ;
 1F  -->

SCR # 35
  0 ( SOUND CONTROL / P/M  2/2 )
  1
  2 : SOUND 3 D20F C! 0 D208 C! SWAP
  3
  4 10 * + 100 * + SWAP 2 * D200 + !
  5  ;
  6 : PADDLE 270 + C@ ;
  7       : PTRIG 27C + C@ ;
  8 : STICK 278 + C@ ;
  9       : STRIG 284 + C@ ;
  A : RND D20A C@ ;
  B    (   22F CONSTANT DMACTL )
  C D01D CONSTANT GRACTL
  D       D407 CONSTANT PMBASE
  E D01B CONSTANT PRIOR
  F       D016 CONSTANT VDELAY
 10  2C0 CONSTANT COLPM
 11        26F CONSTANT GPRIOR
 12 PBASE 1 - HIMEM !
 13
 14 : PLAYER Qbase 1+ C@ PMBASE C! 3
 15         GRACTL C! 2 - IF 1C
 16   ELSE 0C ENDIF DMACTL @ E3 AND
 17 OR      DMACTL C! ;
 18 : HPOS! D000 + C! ;
 19         ( H-posn plyr# -> )
 1A : SIZE! D008 + C! ;
 1B         ( size-code plyr# -> )
 1C : COLPM! COLPM + C! ;
 1D         ( color plyr# -> )
 1E : NOPLY GRACTL 0SET D000 11 0 FI
 1F LL      ; ;S

SCR # 36
  0 ( PON/POFF             1/1 )
  1 (              JDS 18jun85 )
  2   FORTH DEFINITIONS
  3    E406 @ 1+ VARIABLE EOUTC
  4    E436 @ 1+ VARIABLE POUTC
  5         0 VARIABLE ECHR
  6         0 VARIABLE EVTBL F ALLOT
  7
  8 ( routine to send character )
  9 ( to both P: & E:           )
  A  CODE PPUTC POUTC ) JMP, RTS,
  B      C;
  C  CODE EPUTC
```

```
  D    ECHR STA, PHA, TXA, PHA,
  E    ECHR LDA, ' PPUTC JSR, PLA,
  F    TAX, PLA, EOUTC ) JMP, C;
 10  FORTH DEFINITIONS
 11
 12  : PON  E406 @ 1+ EOUTC !
 13         E436 @ 1+ POUTC !
 14       E400 ' EVTBL F CMOVE
 15       ' EPUTC 1- ' EVTBL 6 + !
 16       ' EVTBL 321 ! ;
 17  : POFF  E400 321 ! ;
 18
 19 ;S
 1A
 1B NOTE: the subroutine EPUTC will
 1C       drive decompiler crazy,
 1D       since it cannot find its
 1E       end.
 1F

SCR # 37
  0 ( RS232 SUPPORT        1/2 )
  1
  2 CODE SIO    XSAVE STX, BOT LDA,
  3 E459 JSR, ( SIOV) XSAVE LDX, BOT
  4   STA, BOT 1+ STY, NEXT JMP, C;
  5
  6 : SERR DUP 0< IF 0 100 U/ BASE @
  7         DECIMAL ." SIO ERROR "
  8   . BASE ! QUIT ELSE DROP THEN ;
  9
  A CODE DORL    XSAVE STX, 506 JSR,
  B         HERE 8 + JSR, XSAVE LDX,
  C         NEXT JMP, 0C ) JMP, C;
  D
  E : GETR: HERE 2E7 ! ( SET MEMLO )
  F             FLUSH EMPTY-BUFFERS
 10  150  300 !  ( DDEVIC,DUNIT)
 11  403F 302 !  ( ? CMD,EXPCT DATA)
 12  5    306 C! ( TIMEOUT)
 13  500  304 !  ( BUFFER ADDR)
 14  0C   308 !  ( LENGTH )
 15  0    30A !  ( AUXES )
 16  0 SIO SERR ( ERRORS?)
 17 500 300 0C CMOVE 0 SIO SERR DORL
 18  ( RUN RELOCATOR ) 2E7 @ HERE -
 19  ALLOT HERE FENCE ! ;
 1A
 1B  : R1: " R1: " DROP ;
 1C
 1D  ;S  ( other words not needed )
 1E
 1F   -->

SCR # 38
  0 ( RS232                2/2 )
  1
  2
  3
```

```
 4 : R1OPEN 0 8 R1: OPEN ICSTA CH?
 5 ;
 6 : RYPE -DUP IF 1 IOCB 0B ICCOM C
 7 !       ICBLL ! ICBAL ! CIO
 8   20 ICCOM C! 0 I1CAX ! CIO ELSE
 9         DROP THEN ;
 A : CRR 0A9B SP@ 2 RYPE DROP ;
 B       : REMIT SP@ 1 RYPE DROP ;
 C : SET9600 1 IOCB 0E I1CAX ! 24
 D         ICCOM C! R1: ICBAL !
 E   CIO ICSTA CH? ;
 F
10 : LINER SCR @ (LINE) -TRAILING
11         RYPE ;
12 100 VARIABLE LSPD
13
14 : LISTR DUP SCR ! CRR " SCR#" RY
15 PE      0 <# #S #> RYPE CRR 10 0
16   DO I 0 <# # # #> RYPE I LINER
17         CRR LOOP ;
18  ;S
19
1A
1B
1C
1D
1E
1F

SCR # 39
 0 ( DISPLAY LIST STUFF   1/1 )
 1   HEX
 2 0 VARIABLE 3BYT 0 VARIABLE DLADR
 3
 4 : DINST DLADR @ C@ DUP 0F AND IF
 5
 6 DUP 0F AND 1 = IF 40 AND IF ." J
 7 VB "
 8 ELSE ." JMP " ENDIF DLADR 1+! DL
 9 ADR  @
 A @ DUP DLADR ! HH 3BYT 0SET ELSE
 B DUP 0F AND
 C 8 OVER < IF ." MAP" ELSE ." CHR"
 D
 E ENDIF 7 AND . DUP 10 AND IF ." H
 F "
10 THEN DUP 20 AND IF ." V" THEN DU
11 P
12 80 AND IF ." I" ENDIF DUP 0B0
13
14 AND IF DUP 40 AND IF ." ," ENDIF
15
16 ENDIF 40 AND IF 3 DLADR @ 1+ H?
17 ELSE
18 1 ENDIF 3BYT ! ENDIF ELSE ." BLK
19 "
1A DUP 80 AND IF ." I," ENDIF 70
1B
1C AND 10 / . 1 3BYT ! ENDIF CR
```

```
 1D
 1E 3BYT @ DLADR +! ; ;S
 1F

SCR # 3A
  0 ( PLAYER/MISS.STUFF-RZ 1/1 )
  1 HEX 0 VARIABLE 0VP 64 VARIABLE
  2 0HP 0 VARIABLE 0VPOLD
  3 ( : SPB  HIMEM @ 1+  F800 AND
  4     800 - DUP Qbase ! 17F +
  5               HIMEM ! ; )
  6 : GETPS  0VP  ! ROT BLOCK ROT +
  7     Qbase @ 400 + 0VP  @ + ROT
  8     CMOVE ;
  9 : SPLAY  0 0 HPOS! 7 GR. SPB
  A   Qbase 1+ C@  PMBASE C!
  B   2A 0 COLPM!
  C   0 0 SIZE!   3E D400 C!
  D   3E DMACTL C! 3 GRACTL C!
  E   1C 20 8 64 GETPS  ;
  F  : CLRPM  Qbase @ 800 ERASE ;
 10 : MOVEH   0 STICK F XOR C AND
 11    DUP IF 2 / 3 - ENDIF 0HP @
 12    + DUP 0HP ! 0 HPOS!  ;
 13 : VPOS!  0VPOLD @ 9C00 + DUP
 14   9800 8 CMOVE  8 ERASE  9C00 +
 15   9800 SWAP 8 CMOVE ;
 16 : MOVEV        0 STICK F XOR
 17   3 AND   DUP IF 2 * 3 - ENDIF
 18   -DUP IF 0VP @ DUP 0VPOLD ! +
 19   DUP 0VP !  VPOS! ENDIF ;
 1A : RUNIT BEGIN   MOVEH MOVEV
 1B     2FC C@ FF = NOT END ;
 1C : B/H DUP HEX ." H
 1D EX =" . DECIMAL ." DEC.=" . BIN
 1E QUIT ;          HEX     ;S
 1F

SCR # 3B
  0 \ 3B  DRIVE LINK             1/1
  1  : r/w
  2      301 C@ 1 = IF @ ELSE DROP
  3      0 ENDIF ;
  4
  5  : UNLINK   EMPTY-BUFFERS DR0
  6      ' r/w CFA ' R/W B1 + ! ;
  7
  8  : LINK     EMPTY-BUFFERS DR0
  9      ' @   CFA ' R/W B1 + ! ;
  A   1A VAR  TMPHYS
  B
  C \ SETS BOTH DRIVES
  D
  E : SETPHYS     1FB5 C@ 1FCE C@
  F      100 * + TO TMPHYS DUP
 10   LSB 1FB5 C! MSB 1FCE C! DR0 ;
 11  : RESPHYS     TMPHYS @ DUP LSB
 12      1FB5 C! MSB 1FCE C! DR0 ;
 13 ;S
```

```
 14
 15
 16
 17
 18
 19
 1A
 1B
 1C
 1D
 1E
 1F

SCR # 3C
  0 \ scr# 3C  empty block       1/1
  1  ;S
  2
  3
  4
  5
  6
  7
  8
  9
  A
  B
  C
  D
  E
  F
 10
 11
 12
 13
 14
 15
 16
 17
 18
 19
 1A
 1B
 1C
 1D
 1E
 1F

SCR # 3D
  0 ( LINE PRINTER WORDS   1/2 )
  1 ( 0181 SRC ) 3A50 VARIABLE P:
  2  CODE PCIO XSAVE STX, 70 # LDX,
  3 E456 JSR, XSAVE LDX,  TYA, PHA,
  4 PUSH JMP, C; 0 VARIABLE LPCNT
  5 : PERR? DUP 0< IF FF AND
  6 ." P: ERROR "   ERROR THEN
  7  DROP ;
  8 : LPOPEN 3 3B2 C! P: 3B4 ! 2 3B8
  9  ! 8 3BA ! PCIO PERR? ;
  A : LYP1 3B8 ! 3B4 ! 0B 3B2 C! PCI
```

```
  B O PERR? ; : LPEMIT SP@ 1 LYP1 DR
  C OP ; : LPCR 9B LPEMIT 1 LPCNT +!
  D  ; : LYPE DUP IF DUP 50 > IF
  E  1 LPCNT +! THEN LYP1 ELSE DROP
  F DROP THEN 20 SP@ 1 LYP1 DROP ;
 10 : CRLP LPCR LPCNT @ 3D > IF
 11 LPCR LPCR LPCR LPCR 0 LPCNT !
 12 THEN ;
 13 : FFLP CRLP BEGIN LPCNT @ WHILE
 14 CRLP REPEAT ;
 15 : SHRINK 1B LPEMIT 14 LPEMIT
 16  CRLP ; : EXPAND 1B LPEMIT 13
 17  LPEMIT CRLP ;
 18 : .CLP 0 <# # # #> LYPE ;
 19 : .LP 0 <# #S #>  LYPE ;
 1A : LINELP DUP .CLP SCR @ (LINE)
 1B  -TRAILING 1 MAX LYPE CRLP   ;
 1C 4353 VARIABLE SCR# 2052 , 2023 ,
 1D  : LISTLP DUP SCR ! SCR# 6 LYPE
 1E .LP LPCR B/SCR B/BUF * C/L /
 1F 0 DO I LINELP LOOP ; -->

SCR # 3E
  0 ( LINE PRINTER WORDS   2/2 )
  1 ( 1/27/81  SRC )
  2 : LPSPC 0 DO 20 LPEMIT LOOP ;
  3 : SHOWLP 1+ SWAP C/L 20 = IF
  4   DO CRLP
  5   SCR# 6 LYPE I .LP
  6   1F LPSPC SCR# 6 LYPE I 1+
  7   .LP CRLP
  8   I 20 0 DO DUP SCR ! I .CLP
  9   I SCR @ (LINE) LYPE
  A   5 LPSPC
  B   DUP 1+ SCR ! I LINELP LOOP
  C   DROP 2 +LOOP
  D   ELSE DO CRLP    I LISTLP LOOP
  E   ENDIF FFLP ;
  F
 10 : LPINDEX 1+ SWAP DO I .LP
 11 0 I (LINE) -TRAILING LYPE LPCR
 12 LOOP ;
 13
 14
 15
 16
 17
 18
 19
 1A
 1B
 1C                              ;S
 1D
 1E
 1F

SCR # 3F
  0 \ pns TRANSLATOR HES 16SEP82 1/1
  1 \ moves screens from drive 2 to
```

```
 2 \ same place on drive 1.
 3   FORTH DEFINITIONS  HEX
 4 \ Expects byte on TOS
 5  : translate    ( n --- n )
 6     DUP  0= IF 20 + \
 7       ELSE
 8     DUP DUP     \ lwr case
 9      60 > SWAP 7B < AND IF
 A         20 - ENDIF ENDIF ;
 B
 C \ Expects buffer address on TOS
 D \
 E  : trnsblk     ( adr1 --- )
 F      3FF 0 DO DUP I + DUP C@
10    translate SWAP C! LOOP DROP ;
11
12 \ Expexts source destin scr TOS
13  : PNSCOPY      ( n1 n2 --- )
14    SWAP BLOCK DUP trnsblk SWAP
15   BLOCK 400 CMOVE UPDATE FLUSH ;
16
17
18  EDITOR DEFINITIONS
19
1A  : PNS      EDLQ SCR @ BLOCK
1B         trnsblk UPDATE EDT  UL ;
1C
1D  FORTH DEFINITIONS
1E  : DR2  800 + ;
1F    ;S

SCR # 40
 0 ( FORMATTED LIST PROG. 1/5 )
 1
 2  VOCABULARY FORMY   IMMEDIATE
 3  FORMY   DEFINITIONS
 4   BASE @ OCTAL 40 CN SPACBYT 54
 5 CN COMCHR : IARRAY 0 VARIABLE -2
 6  ALLOT ; : 0> DUP 0= IF DROP  0
 7 ELSE  0< 0= THEN ;
 8 0 VARIABLE INDENT 106 CN FCONS
 9 111 CN ICONS 0 VARIABLE TLFLG
 A 0 VARIABLE KERKNT 100 CN MAXLIN
 B : NXSPACE >R 1+ >R 0   R> R> DO
 C SPACBYT I C@ = IF DROP I LEAVE
 D THEN LOOP ; : NXNSPACE >R 1+ >R
 E 0 R> R> DO SPACBYT I C@ = 0= IF
 F
10  DROP I LEAVE THEN LOOP ; : GTNX
11 WD DUP IF + OVER SWAP NXSPACE
12 ELSE DROP THEN DUP IF OVER SWAP
13 NXNSPACE DUP IF OVER OVER
14 NXSPACE DUP IF OVER - ELSE DROP
15 OVER OVER - 1+ THEN ELSE DUP
16 THEN ELSE DUP THEN ; : TORLCR TL
17 FLG @ IF CRLP ELSE CR THEN KERKN
18 T 0SET ; : TORLY DUP 1+ KERKNT +
19 ! TLFLG @ IF LYPE ELSE TYPE SPAC
1A E THEN ; : DOIND INDENT @ 0> IF
```

```
 1B INDENT @ 0 DO 0 0 TORLY LOOP THE
 1C N ; : PRWORD DUP 1+ KERKNT @ + M
 1D AXLIN > IF TORLCR THEN KERKNT @
 1E 0= IF DOIND THEN OVER OVER TORLY
 1F  ; : 1SET 1 SWAP ! ;  -->

SCR # 41
  0 ( FORMATTED LIST PROG. 2/5 )
  1 : ( 51 WORD 6 ALLOT ;
  2 : IA IARRAY ; IA L1G 10  , ( :)
  3 ( CODE) ( ,CODE) ( SUBROUTINE)
  4  ( IA) ( IARRAY) ( LABEL) ( TBL)
  5  IA L2G 2 , ( ;) ( C;)
  6  IA L3G 2 , ( NXT,) ( NEXT,) IA
  7 L4G 6 , ( IF) ( DO) ( IF,)
  8   ( CASE) ( BEGIN) ( BEGIN,) IA
  9 L5G 3 , ( ELSE,) ( ELSE)
  A   ( WHILE) IA L6G 16 , ( THEN,)
  B ( THEN) ( END,) ( END) ( SOB,)
  C ( BACK) ( UNTIL) ( AGAIN) ( REPE
  D AT) ( ENDIF,)
  E   ( UNTIL,) ( LOOP) ( +LOOP) ( E
  F NDIF) IA L7G 7 , ( CONSTANT)
 10 ( IR)   ( VARIABLE)       ( CN)
 11 ( ARRAY) ( INTEGER) ( ORCON)
 12    IA L8G 1 , ( () IA L9G 3 , (
 13 LD,) ( ST,) ( LOAD)
 14 IA LAG 1 , ( ;CODE)
 15
 16 : CMPWORD DUP >R C@ OVER = R>
 17 SWAP IF >R OVER
 18   R> SWAP OVER DUP C@ DUP 4 > IF
 19  DROP 4 THEN 0
 1A   DO I OVER + 1+ C@ >R OVER R>
 1B SWAP I + C@
 1C   = 0= IF 0 LEAVE THEN LOOP
 1D
 1E   0= IF DROP DROP 0 THEN ELSE 0
 1F THEN ;   -->

SCR # 42
  0 ( FORMATTED LIST PROG. 3/5 )
  1 : GSCAN DUP @ SWAP 2+ SWAP 0 DO
  2 CMPWORD IF LEAVE
  3   0 ELSE 6 + THEN LOOP IF 0 ELSE
  4  DROP 1 THEN ;
  5 : NEWCR KERKNT @ IF TORLCR THEN
  6 ;
  7 : DUPBC OVER >R >R OVER R> SWAP
  8 R> ;
  9 : FINDCHAR SWAP >R SWAP 1+ R>
  A DO DUP I C@ =
  B   IF DROP I LEAVE 0 THEN LOOP IF
  C  0 THEN ;
  D : PRNEWL PRWORD TORLCR ;
  E : >= OVER OVER = IF DROP DROP
  F 1 ELSE > THEN ;   -->
 10
 11
```

```
 12
 13
 14
 15
 16
 17
 18
 19
 1A
 1B
 1C
 1D
 1E
 1F

SCR # 43
  0 ( FORMATTED LIST PROG. 4/5 )
  1 : EL1G NEWCR INDENT 0SET PRWORD
  2 GTNXWD PRNEWL
  3   10 INDENT ! ;
  4 : EL2G NEWCR PRNEWL INDENT 0SET
  5 ;
  6 : EL3G NEWCR PRNEWL ;
  7
  8 : EL4G NEWCR PRNEWL 2 INDENT +!
  9 ;
  A : EL5G NEWCR -2 INDENT +! PRNEWL
  B  2 INDENT +! ;
  C : EL6G NEWCR -2 INDENT +! PRNEWL
  D  ;
  E : EL7G PRWORD GTNXWD PRNEWL INDE
  F NT 0SET ;
 10 : EL8G DUPBC 51 FINDCHAR DUP
 11
 12   IF SWAP DROP OVER - 1+ PRNEWL
 13 ELSE DROP PRWORD THEN ;
 14 : EL9G PRNEWL ;
 15
 16 : ELAG NEWCR 10 INDENT ! PRNEWL
 17 ;
 18 : ASSWRD DUP 4 >= IF OVER OVER +
 19  1- C@ COMCHR = IF
 1A    OVER DUP C@ ICONS =  SWAP 1+
 1B C@ FCONS = AND
 1C   IF 2 ELSE 1 THEN ELSE 0 THEN E
 1D LSE 0 THEN ;
 1E -->
 1F

SCR # 44
  0 ( FORMATTED LIST PROG. 5/5 )
  1 : PRCWRD L1G GSCAN IF EL1G ELSE
  2 L2G GSCAN IF EL2G ELSE
  3   L3G GSCAN IF EL3G ELSE L4G GSC
  4 AN IF EL4G ELSE L5G GSCAN
  5   IF EL5G ELSE L6G GSCAN IF EL6G
  6  ELSE L7G GSCAN IF EL7G
  7   ELSE L8G GSCAN IF EL8G ELSE L9
  8 G GSCAN IF EL9G ELSE
```

```
  9   LAG GSCAN IF ELAG ELSE ASSWRD
  A IF ASSWRD 2 =
  B   IF EL4G ELSE PRNEWL THEN ELSE
  C PRWORD THEN THEN THEN THEN THEN
  D THEN THEN THEN THEN THEN THEN ;
  E : FORLST TORLCR DUP TLFLG @ IF L
  F ISTLP ELSE
 10  TORLCR LIST THEN TORLCR TORLCR
 11 DUP BLK !
 12   BLOCK DUP 1777 + SWAP KERKNT 0
 13 SET INDENT 0SET 0 BEGIN GTNXWD
 14   DUP IF PRCWRD THEN DUP 0= END
 15 DROP DROP DROP BLK 0SET ;
 16 : ASTER TORLCR 40 0 DO 52 SP@ 1
 17 TORLY DROP LOOP TORLCR ;
 18 : FORSHW 1+ OVER DO ASTER I FORL
 19 ST TORLCR LOOP DROP ;
 1A FORTH DEFINITIONS : FLST FORMY T
 1B LFLG 0SET FORLST ; : FLSTLP FORM
 1C Y TLFLG 1SET FORLST FFLP ; : FSH
 1D W FORMY TLFLG 0SET FORSHW ; : FS
 1E HWLP FORMY  TLFLG 1SET FORSHW
 1F FFLP  ;         ;S

SCR # 45
  0 ( CASE                 1/1 )
  1    FORTH DEFINITIONS   HEX
  2  : CASE  ?COMP CSP @ !CSP 4 ;
  3         IMMEDIATE
  4  : OF   4 ?PAIRS COMPILE OVER
  5     COMPILE = COMPILE 0BRANCH
  6     HERE 0 , COMPILE DROP 5 ;
  7         IMMEDIATE
  8  : ENDOF  5 ?PAIRS COMPILE
  9     BRANCH HERE 0 , SWAP 2
  A     ~[COMPILE] ENDIF 4 ;
  B         IMMEDIATE
  C  : ENDCASE 4 ?PAIRS COMPILE DROP
  D     BEGIN SP@ CSP @ = 0= WHILE
  E     2 ~[COMPILE] ENDIF REPEAT
  F     CSP ! ;   IMMEDIATE
 10
 11   ;S
 12
 13
 14
 15
 16
 17
 18
 19
 1A
 1B
 1C
 1D
 1E
 1F
```