

```
*****  
;                                                                 *  
;                                                                 *  
; SMITCH-0 LOCAL OPERATING SYSTEM                               *  
;                                                                 *  
;                                                                 *  
; copyright 1978, Tom M. Mitchell and Reid G. Smith             *  
;                                                                 *  
*****
```

; The local operating system provides a number of utility functions

; In addition it enables easy access to the Data Media Emulation
; Software as a "virtual terminal". At the present time, the Local
; Operating System is downloaded from the host, although it could
; be implemented in EPROM.

; The functions provided by the Local Operating System are as follows:

- ; 1. Break the line connection to the host
- ; 2. Set/Reset the 8th bit for characters in video memory.
- ; 3. Format the screen (normal or inverse video, dual intensity, blink).
- ; 4. Logout the job from the host (works at least for TENEX and TOPS-10).
- ; 5. Save and Restore screen.
- ; 6. Transmit screen to host.

; The Local Operating System is called by the two character
; sequence ^U ^V.

```

0040          TERM      EQU  40H      ; addresses of pointers to 2708 Data Media
                                ; Emulator routines
0042          INIT      EQU  42H
0044          DM        EQU  44H
0046          PCHAR     EQU  46H
0048          MULT      EQU  48H
004A          IST       EQU  4AH
004C          LSTAT     EQU  4CH
004E          FULL      EQU  4EH
0050          HALF      EQU  50H
0052          TXCHR     EQU  52H
0054          PQUE      EQU  54H
0056          GQUE      EQU  56H
0058          GQ        EQU  58H
005A          LOCAL     EQU  5AH
005C          LINE      EQU  5CH

; Terminal Flags

1007          BKMOD     EQU  1007H     ; blink mode on/off = 200q/0
1006          IDMOD     EQU  1006H     ; i/d mode on/off = 377q/0
1008          RLMOD     EQU  1008H     ; roll mode on/off = 377q/0
100A          HDMOD     EQU  100AH     ; half duplex mode on/off = 377q/0
1009          LCMOD     EQU  1009H     ; local mode on/off = 377q/0

1002          CURSE     EQU  1002H     ; cursor address location

003F          USTAT     EQU  77Q       ; USART command port
003C          PICU      EQU  74Q       ; local constants
003B          RBELL     EQU  73Q       ; audio bell select
003A          OFCUR     EQU  72Q       ; turn off the cursor
0039          ONCUR     EQU  71Q       ; turn on the cursor
0002          STX       EQU  2         ; ^b
000D          CR        EQU  13        ; <cr>
000A          LF        EQU  10        ; <lf>
0017          EEOL     EQU  23        ; ^w - erase to end of line
0852          DBGIN     EQU  4122Q     ; start address of first line of normal video memory
0FE8          DEND      EQU  7750Q     ; last address of last line of normal video memory
0018          NLINE     EQU  24        ; number of lines per screen (ignoring top line)
0050          NCHAR     EQU  80        ; number of characters displayed per line

```

```

13FD          ORG 13FDH  ; starting address for local operating system

13FD 3E33          MVI A,33H
13FF D33F          OUT USTAT  ; turn off modem receiver

1481 2A4000        LHLD TERM  ; call term to clear the queue
1404 CD9417        CALL ICALL

1407 CD7815        CALL SAVEM  ; save terminal modes then reset
140A 3E0C          MVI A,140  ; enable interrupts
140C D33C          OUT PICU
140E FB           EI
140F D339          OUT ONCUR  ; turn on the cursor
1411 2A0210        LHLD CURSE  ; refresh the cursor position
1414 7E           MOV A,M
1415 D33B          OUT RBELL  ; ring the bell

1417 11F218        MON:   LXI D,PROMP  ; load prompt string address
141A CD0017        CALL INTRP  ; interpret the string
141D 2A5600        LHLD GQUE  ; get keyboard char in A reg
1420 CD9417        CALL ICALL
1423 CDE916        CALL EXEC  ; process/display character
1426 CD7E17        CALL RAISE  ; switch to upper case

;               ; The following code interprets commands and sends control
;               ; to the appropriate routine

1429 FE42          CPI 'B'
142B CA6914        JZ  BREAK  ; B: break the line connection

142E FE44          CPI 'D'
1430 CA7014        JZ  BLNK   ; D: set/reset eighth bit mode

1433 FE45          CPI 'E'
1435 CA7F14        JZ  EXIT   ; E: exit

1438 FE46          CPI 'F'
143A CA9614        JZ  FORMT  ; F: format screen (line status words)

143D FE47          CPI 'G'
143F CAC014        JZ  GO     ; G: go to location (for running programs)
                  ;      follow the command word with a 16 bit
                  ;      octal address

1442 FE4C          CPI 'L'
1444 CADD14        JZ  KILLJ  ; L: logout connected job

1447 FE52          CPI 'R'
1449 CA1E15        JZ  RESTR  ; R: restore screen
    
```

144C FE53	CPI 'S'
144E CA4215	JZ SAVES ; S: save screen
1451 FE54	CPI 'T' ; T: transmit screen
1453 CA5415	JZ TRNSMT
1456 FE3F	CPI '?'
1458 CA6F15	JZ HELP ; ?: help - display list of commands
145B FE20	CPI 40Q ; if control char, it has already been
145D DA1714	JC MON ; processed

; Command Error Routine

```
1460 11AF17 ERR: LXI D,ERRM1 ; illegal command
1463 CD0017 CALL INTRP ; print msg and try again
1466 C31714 JMP MON
```

;*****

; Send break character

```
1469 3E08 BREAK: MVI A,100
146B D33F OUT USTAT
146D C31714 JMP MON
```

;*****

; Eighth bit display mode command routine

```
1470 11CD18 BLNK: LXI D,LINK ; output message
1473 CD0017 CALL INTRP
1476 CD0B17 CALL YAN ; get response
1479 D2DC15 JNC RBLNK ; carry = 0 => no
147C C3CF15 JMP SBLNK
```

;*****

; Exit Routine

```
147F 110519 EXIT: LXI D,XIT
1482 CD0017 CALL INTRP
1485 CDAD15 CALL UNSVM ; return from local monitor to terminal
1488 3E37 MVI A,37H ; turn on modem receiver
148A D33F OUT USTAT ; this should really be done in local and
; line routines
148C 3E0F MVI A,170
148E D33C OUT PICU
1490 E1 POP H ; this completes the call from the keyboard
1491 D1 POP D ; interrupt service routine
1492 C1 POP B
1493 F1 POP PSW
1494 FB EI
1495 C9 RET
```

; Screen formatting routine
 ; sets line status of each line to entered digit

```

1496 11E418   FORMT: LXI D,ORMAT ; output message
1499 CD0017   CALL INTRP
149C 11D317   FORM1: LXI D,FPRMP ; prompt
149F CD0017   CALL INTRP
14A2 CD4517   CALL GTOCT ; get line status word
14A5 D2B814   JNC ABRTF ; check for bad data
14A8 0E00     MVI C,0
14AA 45       MOV B,L ; set up arguments
14AB 2A4A00   LHLD IST
14AE 110400   LXI D,4 ; calculate correct address of routine
14B1 19       DAD D ; add 4 to obtain ist1 address -
                                     ; this should be available
14B2 CD9417   CALL ICALL
14B5 C31714   JMP MON
14B8 FE1B     ABRTF: CPI 33Q ; abort on ESC
14BA CA1714   JZ MON
14BD C39C14   JMP FORM1
    
```

; Go Routine - Jump To Octal Address with provision for return

```

14C0 11E517   GO: LXI D,GOM ; display message
14C3 CD0017   CALL INTRP
14C6 11E917   G01: LXI D,GPRMP ; prompt
14C9 CD0017   CALL INTRP
14CC CD4517   CALL GTOCT ; get address
14CF D2D514   JNC ABRTG ; check for bad address
14D2 CD9417   CALL ICALL ; jump to address (with provision on stack for return)
14D5 FE1B     ABRTG: CPI 33Q ; abort on ESC
14D7 CA1714   JZ MON
14DA C3C614   JMP G01
    
```

; Logout connected job

```
14DD 11DE18      KILLJ: LXI D,0GOUT
14E0 CD0017      CALL INTRP
14E3 0603        MVI B,3        ; transmit ^c - need 4 to be sure!?!
14E5 2A5200      LHLD TXCHR
14E8 CD9417      CALL ICALL
14EB 0603        MVI B,3        ; transmit ^c
14ED 2A5200      LHLD TXCHR
14F0 CD9417      CALL ICALL
14F3 0603        MVI B,3        ; transmit ^c
14F5 2A5200      LHLD TXCHR
14F8 CD9417      CALL ICALL
14FB 0603        MVI B,3        ; transmit ^c
14FD 2A5200      LHLD TXCHR
1500 CD9417      CALL ICALL
1503 064B        MVI B,'K'      ; k
1505 2A5200      LHLD TXCHR
1508 CD9417      CALL ICALL
150B 060D        MVI B,CR       ; carriage return
150D 2A5200      LHLD TXCHR
1510 CD9417      CALL ICALL
1513 060D        MVI B,CR       ; carriage return
1515 2A5200      LHLD TXCHR
1518 CD9417      CALL ICALL
151B C31714      JMP MON
```

; Restore screen

```
151E 11F718      RESTR: LXI D,RESTM
1521 CD0017      CALL INTRP
1524 3A0810      LDA RLMOD      ; save roll mode setting
1527 F5          PUSH PSW
1528 3E00        MVI A,0        ; turn off roll mode during unpacking
152A 320810      STA RLMOD
152D 211C19      LXI H,SQUEUE  ; initialize ptr. to beginning of screen
1530 221A19      SHLD SQPTR
1533 CD7116      CALL UNPKS
1536 3E1A        MVI A,26       ; ^z (cursor up)
1538 CDE916      CALL EXEC      ; backup the cursor to reuse prompt line
153B F1          POP PSW        ; restore roll mode
153C 320810      STA RLMOD
153F C31714      JMP MON
```

```
                ;      Save screen

1542 11A417      SAVES: LXI  D,AVEM
1545 CD0017      CALL  INTRP
1548 211C19      LXI  H,SQUEUE ; initialize ptr to beginning of screen
154B 221A19      SHLD SQPTR
154E CD4016      CALL  PCKS
1551 C31714      JMP   MON

                ;*****

                ;      Transmit Screen

1554 119517      TRNSMT: LXI  D,ANSM
1557 CD0017      CALL  INTRP
155A 115208      LXI  D,DBGIN ; initialize display memory pointer
155D 215208      LXI  H,DBGIN ; initialize display memory pointer
1560 0E00        MVI  C,0 ; initialize line counter
1562 CDC416      TRNS0: CALL TRNSL ; transmit a line to the host
1565 0C          INR  C ; bump line counter
1566 79          MOV  A,C ; transmitted 23 lines?
1567 FE17        CPI  NLINE-1
1569 C26215      JNZ  TRNS0 ; no - more lines to transmit
156C C31714      JMP  MON ; yes - return
```

; Help routine

```
156F 11F617 HELP: LXI D,HELPM ; output help message, and return
1572 CD0017 CALL INTRP
1575 C31714 JMP MON
```

; The following routines are utilities used to assist in the
; implementation of some of the functions.

; Save all terminal modes

```
1578 3A0710 SAVEM: LDA BKM0D ; get each mode...
157B 321619 STA BKT ; and save it in temporary location
157E 3A0610 LDA IDMOD
1581 321519 STA IDT
1584 3A0810 LDA RLM0D
1587 321719 STA RLT
158A 3A0910 LDA LCM0D
158D 321819 STA LCT
1590 3A0A10 LDA HDM0D
1593 321919 STA HDT
1596 3E00 MVI A,0
1598 320710 STA BKM0D ; turn blink mode off
159B 320610 STA IDMOD ; and i/d mode
159E 3EFF MVI A,377Q
15A0 320A10 STA HDM0D ; turn on other modes
15A3 320910 STA LCM0D
15A6 320810 STA RLM0D
15A9 CDE715 CALL TOPLN ; refresh top line messages
15AC C9 RET
```

; Restore all terminal modes

```
15AD 3A1619 UNSVM: LDA BKT
15B0 320710 STA BKM0D
15B3 3A1519 LDA IDT
15B6 320610 STA IDMOD
15B9 3A1719 LDA RLT
15BC 320810 STA RLM0D
15BF 3A1819 LDA LCT
15C2 320910 STA LCM0D
15C5 3A1919 LDA HDT
```

Micro-Symbol LMON.;6 Thursday, October 26, 1978 10:07:39 PAGE 7-2

15C8 320A10
15CB CDE715
15CE C9

STA HDMOD
CALL TOPLN ; refresh top line status information
RET

; Set blink mode

```

15CF 3E0E      SBLNK: MVI A,14      ; ^n
15D1 CDE916      CALL EXEC
15D4 3E80      MVI A,200Q      ; this is 'sticky'. change terminal blink mode too.
15D6 321619      STA BKT
15D9 C31714      JMP MON
    
```

; Reset blink mode

```

15DC 3E00      RBLNK: MVI A,0      ; reset both blink mode flags
15DE 320710      STA BKM0D
15E1 321619      STA BKT
15E4 C31714      JMP MON
    
```

; Refresh top line status information

```

15E7 3A0A10      TOPLN: LDA HDM0D
15EA FE00      CPI 0
15EC CAF515      JZ FD          ; check for full/half duplex
15EF 2A5000      LHLD HALF
15F2 C3F815      JMP C1
15F5 2A4E00      FD:  LHLD FULL
15F8 CD9417      C1:  CALL ICALL
15FB 3A0910      LDA LCM0D      ; check local/on-line mode
15FE FE00      CPI 0
1600 CA0916      JZ LN
1603 2A5A00      LHLD LOCAL
1606 C30C16      JMP C2
1609 2A5C00      LN:  LHLD LINE
160C CD9417      C2:  CALL ICALL
160F 3A0810      LDA RLM0D      ; save 3 modes
1612 F5      PUSH PSW
1613 3A0710      LDA BKM0D
1616 F5      PUSH PSW
1617 3A0610      LDA IDM0D
161A F5      PUSH PSW
161B 3E18      MVI A,24      ; clear these 3 modes
161D CDE916      CALL EXEC
1620 F1      POP PSW      ; retrieve correct idmod
1621 FE00      CPI 0
1623 CA2B16      JZ BK
1626 3E10      MVI A,16      ; and reset mode if needed
1628 CDE916      CALL EXEC
    
```

```

162B F1      BK:    POP PSW      ; do same for blink mode
162C FE00    CPI    0
162E CA3616  JZ    RL
1631 3E0E    MVI   A,14
1633 CDE916  CALL  EXEC
1636 F1      RL:    POP PSW      ; and for roll mode
1637 FE00    CPI    0
1639 C8      RZ
163A 3E1D    MVI   A,29
163C CDE916  CALL  EXEC
163F C9      RET

```

; Save screen on screen queue

```

1640 215208  PCKS: LXI  H,DBGIN ; display mem ptr
1643 115208  LXI  D,DBGIN ; line address
1646 0E00    MVI  C,0
1648 CD9416  PCK0: CALL ELCNT ; determine number of chars in this line
164B FE00    PCK1: CPI    0
164D CA5916  JZ    EOL ; end of line?
1650 46      MOV  B,M ; get next char, and bump ptrs.
1651 23      INX  H
1652 3D      DCR  A
1653 CDB916  CALL  PUT ; save char
1656 C34B16  JMP  PCK1
1659 0617    EOL:  MVI  B,EEOL
165B CDB916  CALL  PUT ; mark end of line with eol
165E 215100  LXI  H,NCHAR+1
1661 19      DAD  D ; update ptrs
1662 54      MOV  D,H
1663 5D      MOV  E,L
1664 0C      INR  C
1665 79      MOV  A,C
1666 FE17    CPI  NLINE-1 ; don't save last line (command line)
1668 C24816  JNZ  PCK0 ; more lines to save
166B 0600    MVI  B,0
166D CDB916  CALL  PUT
1670 C9      RET

```

;*****

; Unpack screen

```
1671 3E02 UNPKS: MVI A,STX ; home the cursor
1673 CDE916 CALL EXEC
1676 CDAE16 UNP1: CALL GET
1679 FE00 CPI 0
167B CA9316 JZ UNPRT ; end of screen marked by '0'
167E CDE916 CALL EXEC
1681 FE17 CPI EE0L
1683 C27616 JNZ UNP1
1686 3E0D MVI A,CR
1688 CDE916 CALL EXEC
168B 3E0A MVI A,LF
168D CDE916 CALL EXEC
1690 C37616 JMP UNP1
1693 C9 UNPRT: RET
```

;*****

; Determine column position of last char in line
; D,E reg contain address of first char
; return answer in a reg

```
1694 E5 ELCNT: PUSH H
1695 C5 PUSH B
1696 214F00 LXI H,NCHAR-1
1699 19 DAD D ; H,L now has address of last displayable position
169A 0650 MVI B,NCHAR ; on line of display memory
169C 7E ELC1: MOV A,M
169D FE20 CPI ' '
169F C2AA16 JNZ ELC2
16A2 05 DCR B
16A3 2B DCX H
16A4 78 MOV A,B
16A5 FE00 CPI 0
16A7 C29C16 JNZ ELC1
16AA 78 ELC2: MOV A,B
16AB C1 POP B
16AC E1 POP H
16AD C9 RET
```

;*****

; Get char from screen queue into A reg

```

16AE E5      GET:  PUSH H
16AF 2A1A19  LHL  SQPTR
16B2 7E      MOV  A,M
16B3 23      INX  H
16B4 221A19  SHLD SQPTR
16B7 E1      POP  H
16B8 C9      RET
    
```

;*****

; Put char from B reg. onto screen queue

```

16B9 E5      PUT:  PUSH H
16BA 2A1A19  LHL  SQPTR
16BD 70      MOV  M,B
16BE 23      INX  H
16BF 221A19  SHLD SQPTR
16C2 E1      POP  H
16C3 C9      RET
    
```

;*****

; Transmit one line of display memory to host
; starting address in D, line counter in C

```

16C4 CD9416  TRNSL: CALL ELCNT ; find column of last character in line
16C7 FE00    TR1:   CPI  0      ; end of line?
16C9 CADA16  JZ    TE0L    ; yes - transmit cr
16CC 46      MOV  B,M      ; get the character and transmit it
16CD E5      PUSH H       ; temp store for current character position
16CE 2A5200  LHL  TXCHR
16D1 CD9417  CALL ICALL
16D4 E1      POP  H
16D5 23      INX  H       ; bump pointers for next character
16D6 3D      DCR  A
16D7 C3C716  JMP  TR1     ; do for all characters in the line
16DA 060D    TE0L: MVI  B,CR  ; terminate line with cr
16DC 2A5200  LHL  TXCHR
16DF CD9417  CALL ICALL
16E2 215100  LXI  H,NCHAR+1
16E5 19      DAD  D       ; bump pointer
16E6 54      MOV  D,H
16E7 5D      MOV  E,L
16E8 C9      RET
    
```

;*****

; Execute character in A register using Data Media Emulator
; save machine state

```
16E9 E5      EXEC:  PUSH H
16EA D5      PUSH D
16EB C5      PUSH B
16EC F5      PUSH PSW
16ED D33A    OUT  OFCUR   ; turn off the cursor
16EF 2A4600 LHL D PCHAR
16F2 CD9417 CALL ICALL
16F5 D339    OUT  ONCUR   ; turn on the cursor
16F7 2A0210 LHL D CURSE  ; read from cursor location to set screen cursor
16FA 7E      MOV  A,M
16FB F1      POP  PSW
16FC C1      POP  B
16FD D1      POP  D
16FE E1      POP  H
16FF C9      RET
```

```

; Virtual terminal routine
; D,E contain address of first byte of string of input chars
; pchar is called with each of these input characters
; end of input string marked by a zero

```

```

1700 1A      INTRP: LDAX D      ; get next character of input string
1701 FE00    CPI 0
1703 C8      RZ          ; return if end of input string
1704 CDE916  CALL EXEC    ; execute the character
1707 13      INX D      ; bump command string index
1708 C30017  JMP INTRP   ; and get next input

```

```

; yes/no interrogator
; prompts as follows: "y/n:"
; reprompts if bad response
; sets carry bit if answer is "y", clears if "n"

```

```

170B E5      YAN:  PUSH H
170C D5      PUSH D
170D 110C19  YAN1:  LXI D,YANPR ; prompt
1710 CD0017  CALL INTRP
1713 2A5600  LHL D,GQUE  ; get response
1716 CD9417  CALL ICALL
1719 CDE916  CALL EXEC   ; and display it
171C CD7E17  CALL RAISE  ; switch to upper case
171F FE4E    CPI 'N'
1721 CA3317  JZ NRET    ; if response was "n", return with carry bit = 0
1724 FE59    CPI 'Y'
1726 37      STC      ; if "y" response, return with carry = 1
1727 CA3517  JZ YRET
172A 11C317  LXI D,ERRM2 ; output error msg, and try again
172D CD0017  CALL INTRP
1730 C30D17  JMP YAN1
1733 37      NRET:  STC
1734 3F      CMC
1735 D1      YRET:  POP D
1736 E1      POP H
1737 C9      RET

```


;*****

```

;       ASCII to binary conversion routine
;       input and result in A register
;       input error signaled by cleared carry flag

```

```

1738 D630      ASCBN: SUI  30H      ; subtract ASCII offset
173A DA4017    JC    ERRAS
173D FE08      CPI   8          ; if result between 0 and 8, then return
173F D8        RC
1740 C630      ERRAS: ADI  30H      ; add back offset
1742 37        STC           ; clear carry bit to signal error
1743 3F        CMC
1744 C9        RET

```

;*****

```

;       Get a 16 bit number entered from keyboard in octal
;       input terminated by <cr>
;       value returned - high order in H reg., low order in L reg
;       error signaled by cleared carry bit, bit set if valid data

```

```

1745 210000    GTOCT: LXI  H,0      ; initialize
1748 E5        GTOC1: PUSH H      ; save partial results
1749 2A5600    LHLD GQUE
174C CD9417    CALL ICALL
174F CDE916    CALL EXEC      ; display the new character
1752 FE0D      CPI   CR
1754 CA7B17    JZ    GTRET      ; finished if this is a <cr>
1757 CD3817    CALL ASCBN      ; get one digit
175A DA5F17    JC    GCONT
175D E1        POP  H          ; in case of error, return with clear carry
175E C9        RET           ; and error result in A reg
175F 5F        GCONT: MOV  E,A      ; H,L gets (partial result shifted
1760 E1        POP  H          ; by three bits) + H,L
1761 37        STC           ; this is a 16 bit 3 position left shift
1762 3F        CMC
1763 7D        MOV  A,L
1764 17        RAL
1765 6F        MOV  L,A
1766 7C        MOV  A,H
1767 17        RAL
1768 67        MOV  H,A
1769 7D        MOV  A,L
176A 17        RAL
176B 6F        MOV  L,A
176C 7C        MOV  A,H
176D 17        RAL
176E 67        MOV  H,A
176F 7D        MOV  A,L

```

1770	17	RAL
1771	6F	MOV L,A
1772	7C	MOV A,H
1773	17	RAL
1774	67	MOV H,A

```
1775 1600          MVI D,0
1777 19           DAD D
1778 C34817       JMP GTOC1
177B E1          GTRET: POP H
177C 37          STC          ; set carry to denote valid data
177D C9          RET
```

;*****

; Raise lower case to upper case argument in A register

```
177E FE61       RAISE: CPI 'a'
1780 D8         RC
1781 FE7B       CPI 'z'+1
1783 D0         RNC
1784 D620       SUI 40Q      ; make the upper case conversion
1786 C9         RET
```

;*****

; Debug routine - prints brightened ascending digits at current
; cursor location

```
1787 F5         DEBUG: PUSH PSW
1788 3A1219     LDA SIGNL    ; get signal character
178B CDE916     CALL EXEC    ; display it
178E 3C         INR A
178F 321219     STA SIGNL    ; update signal character
1792 F1         POP PSW
1793 C9         RET
```

;*****

1794 E9 ICALL: PCHL ; routine to make indirect calls to 2708 routines

;*****

; Strings

1795 72616E736D
179A 6974207363
179F 7265656E00

ANSM: DB 'r','a','n','s','m','i','t',' ','s','c','r','e','e','n',0

17A4 6176652073
17A9 637265656E
17AE 00

AVEM: DB 'a','v','e',' ','s','c','r','e','e','n',0

17AF 202D20696C
17B4 6C6567616C
17B9 20636F6D
17BD 6D616E6417
17C2 00

ERRM1: DB ' ','-','i','l','l','e','g','a','l',' ','c','o','m'
DB 'm','a','n','d',EE0L,0

17C3 203F207472
17C8 7920616761
17CD 696E0D0A20
17D2 00

ERRM2: DB ' ','?',' ','t','r','y',' ','a','g','a','i','n',CR,LF,' ',0

17D3 0D0A206C69
17D8 6E65207374
17DD 617475733A
17E2 201700

FPRMP: DB CR,LF,' ','l','i','n','e',' ','s','t','a','t','u','s',' ':'
DB ' ',EE0L,0

17E5 6F201700

GOM: DB 'o',' ',EE0L,0

17E9 0D0A616464
17EE 726573733A
17F3 201700

GPRMP: DB CR,LF,'a','d','d','r','e','s','s',' ':' ',' ',EE0L,0

17F6 0D0A42202D
17FB 2062726561
1880 6B206C696E
1805 65
1806 20636F6E6E
180B 656374696F
1810 6E
1811 0D0A44202D
1816 2073657420
181B 646973706C
1820 6179206D6F
1825 6465

HELPM: DB CR,LF,'B',' ','-',' ','b','r','e','a','k',' ','l','i','n','e'
DB ' ','c','o','n','n','e','c','t','i','o','n'
DB CR,LF,'D',' ','-',' ','s','e','t',' '
DB 'd','i','s','p','l','a','y',' ','m','o','d','e'

1827 0D0A45202D
182C 2065786974 DB CR,LF,'E',' ','-',' ','e','x','i','t'
1831 0D0A46202D
1836 2073657420
183B 7363726565
1840 6E DB CR,LF,'F',' ','-',' ','s','e','t',' ','s','c','r','e','e','n'
1841 20666F726D
1846 6174 DB ' ','f','o','r','m','a','t'
1848 0D0A47202D
184D 20676F2074
1852 6F20 DB CR,LF,'G',' ','-',' ','g','o',' ','t','o',' '
1854 6164647265
1859 737320286F
185E 6374616C29 DB 'a','d','d','r','e','s','s',' ','('','o','c','t','a','l','')'
1863 0D0A4C202D
1868 206C6F676F
186D 7574206A6F
1872 62 DB CR,LF,'L',' ','-',' ','l','o','g','o','u','t',' ','j','o','b'
1873 2066726F6D
1878 20686F7374 DB ' ','f','r','o','m',' ','h','o','s','t'
187D 0D0A52202D
1882 2072657374
1887 6F726520 DB CR,LF,'R',' ','-',' ','r','e','s','t','o','r','e',' '
188B 7363726565
1890 6E DB 's','c','r','e','e','n'
1891 0D0A53202D
1896 2073617665
189B 20 DB CR,LF,'S',' ','-',' ','s','a','v','e',' '
189C 7363726565
18A1 6E DB 's','c','r','e','e','n'
18A2 0D0A54202D
18A7 207472616E
18AC 736D697420 DB CR,LF,'T',' ','-',' ','t','r','a','n','s','m','i','t',' '
18B1 7363726565
18B6 6E20746F20 DB 's','c','r','e','e','n',' ','t','o',' ','h','o','s','t'
18BB 686F7374
18BF 0D0A636E74
18C4 726C206368
18C9 61727300 DB CR,LF,'c','n','t','r','l',' ','c','h','a','r','s',0

18CD 202D206F6E
18D2 2F6F666628
18D7 792F6E293A LINK: DB ' ','-',' ','o','n',' ','/','o','f','f',' ('','y','/','n',')',' ':'
18DC 2000 DB ' ',0

18DE 6F676F7574
18E3 00 OGOUT: DB 'o','g','o','u','t',0

18E4 6F726D6174
18E9 2073637265

18EE 656E1700 ORMAT: DB 'o','r','m','a','t',' ','s','c','r','e','e','n',EEOL,0

18F2 0D0A3E1700 PROMP: DB CR,LF,'>',EEOL,0

18F7 6573746F72

18FC 6520736372

1901 65656E00

RESTM: DB 'e','s','t','o','r','e',' ','s','c','r','e','e','n',0

1905 7869740D0A

190A 1700

XIT: DB 'x','i','t',CR,LF,EEOL,0

190C 792F6E3A17

1911 00

YANPR: DB 'y','/','n',':',EEOL,0

1912 B1 SIGNL: DB 200Q+'1' ; signal character for debug routine

1913 0108 SIGAD: DW 4001Q ; first displayed location on top line

1915 00 IDT: DB 0 ; temporary storage for terminal modes

1916 00 BKT: DB 0

1917 00 RLT: DB 0

1918 00 LCT: DB 0

1919 00 HDT: DB 0

191A 1C19 SQPTR: DW SQUEUE ; screen queue pointer and queue (must be here)

191C 00 SQUEUE: DB 0

END

NO PROGRAM ERRORS

ABRTF	14B8	ICALL	1794	TERM	0040
ABRTG	14D5	IDMOD	1006	TOPLN	15E7
ANSM	1795	IDT	1915	TR1	16C7
ASCBN	1738	INIT	0042	TRNS0	1562
AVEM	17A4	INTRP	1700	TRNSL	16C4
BK	162B	IST	004A	TRNSMT	1554
BKMOD	1007	KILLJ	14DD	TXCHR	0052
BKT	1916	LCMOD	1009	UNP1	1676
BLNK	1470	LCT	1918	UNPKS	1671
BREAK	1469	LF	000A	UNPRT	1693
C1	15F8	LINE	005C	UNSVN	15AD
C2	160C	LINK	18CD	USTAT	003F
CR	000D	LN	1609	XIT	1905
CURSE	1002	LOCAL	005A	YAN	170B
DBGIN	0852	LSTAT	004C	YAN1	170D
DEBUG	1787	MON	1417	YANPR	190C
DEND	0FE8	MON0	140A	YRET	1735
DM	0044	MULT	0048		
EEOL	0017	NCHAR	0050		
ELC1	169C	NLINE	0018		
ELC2	16AA	NRET	1733		
ELCNT	1694	OFCUR	003A		
EOL	1659	OGOUT	18DE		
ERR	1460	ONCUR	0039		
ERRAS	1740	ORMAT	18E4		
ERRM1	17AF	PCHAR	0046		
ERRM2	17C3	PCK0	1648		
EXEC	16E9	PCK1	164B		
EXIT	147F	PCKS	1640		
FD	15F5	PICU	003C		
FORM1	149C	PQUE	0054		
FORMT	1496	PROMP	18F2		
FPRMP	17D3	PUT	16B9		
FULL	004E	RAISE	177E		
GCONT	175F	RBELL	003B		
GET	16AE	RBLNK	15DC		
GO	14C0	RESTM	18F7		
G01	14C6	RESTR	151E		
GOM	17E5	RL	1636		
GPRMP	17E9	RLMOD	1008		
GQ	0058	RLT	1917		
GQUE	0056	SAVEM	1578		
GTOC1	1748	SAVES	1542		
GTOCT	1745	SBLNK	15CF		
GTRET	177B	SIGAD	1913		
HALF	0050	SIGNL	1912		
HDMOD	100A	SQPTR	191A		
HDT	1919	SQUEUE	191C		
HELP	156F	STX	0002		
HELPM	17F6	TEOL	16DA		