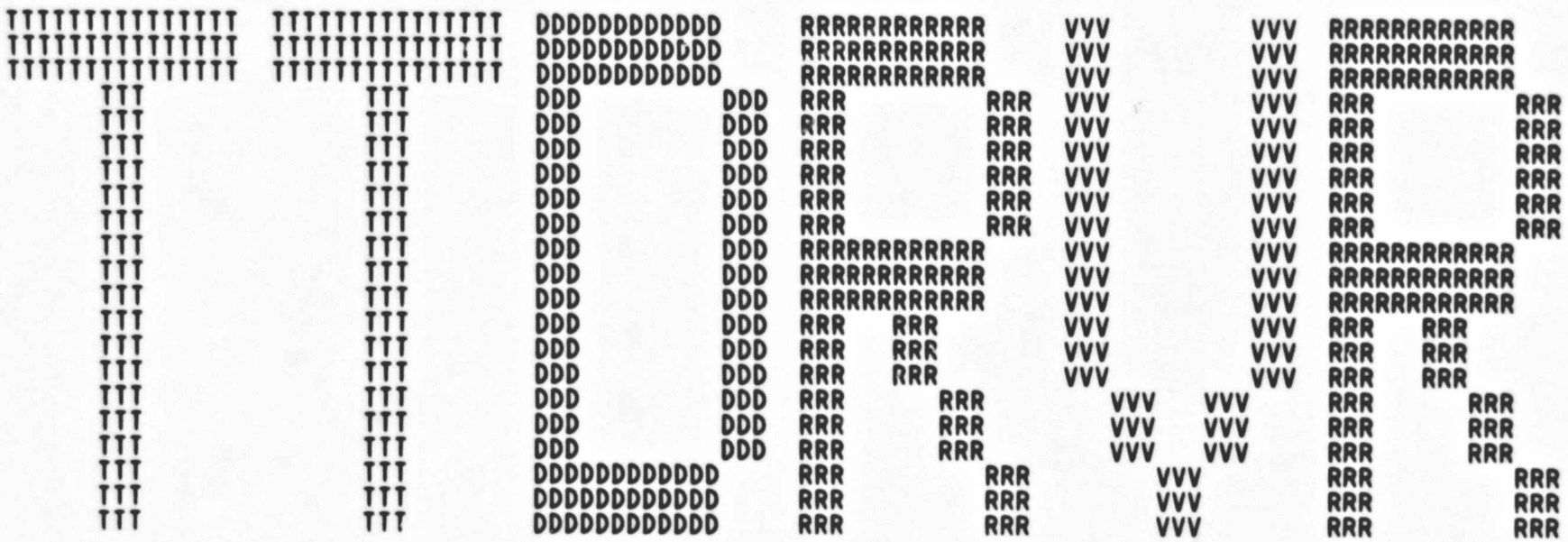
TTTTTTTTTTTTTTT TTTTTTTTTTTTTTT DDDDDDDDDDDD RRRRRRRRRRRR VVV VVV RRRRRRRRRRRR
TTTTTTTTTTTTTTT TTTTTTTTTTTTTTT DDDDDDDDDDDDD RRRRRRRRRRRR VVV VVV RRRRRRRRRRRR
TTTTTTTTTTTTTTT TTTTTTTTTTTTTTT DDDDDDDDDDDDD RRRRRRRRRRRR VVV VVV RRRRRRRRRRRR
TTT TTT DDD DDD RRR RRR VVV VVV RRR RRR
TTT TTT DDD DDD RRR RRR VVV VVV RRR RRR
TTT TTT DDD DDD RRR RRR VVV VVV RRR RRR
TTT TTT DDD DDD RRR RRR VVV VVV RRR RRR
TTT TTT DDD DDD RRR RRR VVV VVV RRR RRR
TTT TTT DDD DDD RRRRRRRRRRRR VVV VVV RRRRRRRRRRRR
TTT TTT DDD DDD RRRRRRRRRRRR VVV VVV RRRRRRRRRRRR
TTT TTT DDD DDD RRRRRRRRRRRR VVV VVV RRRRRRRRRRRR
TTT TTT DDD DDD RRR RRR VVV VVV RRR RRR
TTT TTT DDD DDD RRR RRR VVV VVV RRR RRR
TTT TTT DDD DDD RRR RRR VVV VVV RRR RRR
TTT TTT DDD DDD RRR RRR VVV VVV RRR RRR
TTT TTT DDDDDDDDDDDD RRR RRR VVV RRR RRR
TTT TTT DDDDDDDDDDDD RRR RRR VVV RRR RRR
TTT TTT DDDDDDDDDDDD RRR RRR VVV RRR RRR

```
TTTTTTTTTT  TTTTTTTTTT  YY      YY    CCCCCCCC  HH      HH    AAAAAA    RRRRRRRR      IIIIII
TTTTTTTTTT  TTTTTTTTTT  YY      YY    CCCCCCCC  HH      HH    AAAAAA    RRRRRRRR      IIIIII
    TT          TT      YY      YY  CC          HH      HH  AA      AA  RR      RR      II
    TT          TT      YY      YY  CC          HH      HH  AA      AA  RR      RR      II
    TT          TT        YY  YY    CC          HH      HH  AA      AA  RR      RR      II
    TT          TT        YY  YY    CC          HH      HH  AA      AA  RR      RR      II
    TT          TT          YY      CC          HHHHHHHHHH  AA      AA  RRRRRRRR      II
    TT          TT          YY      CC          HHHHHHHHHH  AA      AA  RRRRRRRR      II
    TT          TT          YY      CC          HH      HH  AAAAAAAAAA  RR  RR        II
    TT          TT          YY      CC          HH      HH  AAAAAAAAAA  RR  RR        II
    TT          TT          YY      CC          HH      HH  AA      AA  RR    RR      II
    TT          TT          YY      CC          HH      HH  AA      AA  RR    RR      II
    TT          TT          YY        CCCCCCCC  HH      HH  AA      AA  RR      RR  IIIIII
    TT          TT          YY        CCCCCCCC  HH      HH  AA      AA  RR      RR  IIIIII

LL              IIIIII      SSSSSSSS
LL              IIIIII      SSSSSSSS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II        SSSSSS
LL                II        SSSSSS
LL                II              SS
LL                II              SS
LL                II              SS
LL                II              SS
LLLLLLLLLL      IIIIII    SSSSSSSS
LLLLLLLLLL      IIIIII    SSSSSSSS
```

Table of contents

```
                        .TITLE  TTYCHARI - terminal input character routines
;MIR0001                .IDENT  'V04-001'
-1
                        ;
                        ;*****************************************************************************
                        ;*                                                                           *
                        ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
                        ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
                        ;*  ALL RIGHTS RESERVED.                                                     *
                        ;*                                                                           *
                        ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED    *
                        ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
                        ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER     *
                        ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY    *
                        ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY    *
                        ;*  TRANSFERRED.                                                             *
                        ;*                                                                           *
                        ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE    *
                        ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT    *
                        ;*  CORPORATION.                                                             *
                        ;*                                                                           *
                        ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS    *
                        ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
                        ;*                                                                           *
                        ;*****************************************************************************
                        ;
                        ;++
                        ; FACILITY:
                        ;
                        ;       VAX/VMS TERMINAL DRIVER
                        ;
                        ; ABSTRACT:
                        ;
                        ;       THIS MODULE CONTAINS ROUTINES NEEDED FOR CHARACTER INPUT.
                        ;
                        ; AUTHOR:
                        ;
                        ;       R.HEINEN  11-AUG-1976
                        ;
                        ; Revision history:
;MIR0001                 .1
;MIR0001                 .2      ;       V04-001 MIR0001         Michael I. Rosenblum            05-Nov-1984
;MIR00C1                 .3      ;               Make F10 honor line editing characteristic
;MIR0001                 .4
                        ;       V03-019 MIR0450         Michael I. Rosenblum            27-Jun-1984
                        ;               Never initiate logins on a terminal marked secure server
                        ;               and noautobaud from unsolicited input.
                        ;
                        ;       V03-018 MIR0370         Michael I. Rosenblum            20-Mar-1984
                        ;               Use TTY$GB_AUTOCHAR to get the character to indicate
                        ;               ready for system password.
                        ;
                        ;       V03-017 MIR0310         Michael I. Rosenblum            07-FEB-1985
                        ;               Fix bug that caused the interrupt key to stop working when
                        ;               in 8 bit mode.  Add code to output a different character
```

```
0000     54 ;               as a ready for system password character.
0000     55 ;               Fix bug in pasthru mode so it will honor control-s and
0000     56 ;               control-q correctly.
0000     57 ;
0000     58 ;       V03-016 MIR0082         Michael I. Rosenblum            19-Aug-1983
0000     59 ;               Make long word reference to a byte field a byte reference.
0000     60 ;
0000     61 ;       V03-015 MIR0080         Michael I. Rosenblum            14-Jun-1983
0000     62 ;               Restructure module.
0000     63 ;
0000     64 ;       V03-014 MIR0051         Michael I. Rosenblum            23-Jun-1983
0000     65 ;               Optomize normal character input code path, make jobcontroler
0000     66 ;               notification logic a subroutine, output bell uppon notification
0000     67 ;               of job controler on login.  Make buffering characters in the
0000     68 ;               typeahead buffer a subroutine.
0000     69 ;
0000     70 ;       V03-013 MIR1050         Michael I. Rosenblum            23-May-1983
0000     71 ;               finish removing code for broadcast.
0000     72 ;
0000     73 ;       V03-012 JLV0255         Jake VanNoy                 23-MAY-1983
0000     74 ;               Add code to allow out-of-band aborts. Set up multi-echo
0000     75 ;               strings to be table driven.
0000     76 ;
0000     77 ;       V03-011 MIR0041         Michael I. Rosenblum            29-Apr-1983
0000     78 ;               Cause autobaud to require a readable CR before initiating
0000     79 ;               loginout.  Clear the passall optomization bit when EOL is
0000     80 ;               set.  Check int before jumping into the getnextchar code path
0000     81 ;               when EOL is set by the passall code path.
0000     82 ;
0000     83 ;       V03-010 MIR0034         Michael I. Rosenblum            07-Apr-1983
0000     84 ;               allow LK201 function key F6 to translate to ^C
0000     85 ;
0000     86 ;       V03-009 MIR0032         Michael I. Rosenblum            05-Apr-1983
0000     87 ;               Allow control-C,Y and O to echo dec crt strings.
0000     88 ;
0000     89 ;       V03-008 RKS0008         RICK SPITZ                      14-MAR-1983
0000     90 ;               ADD SUPPORT FOR LOGICAL UCB
0000     91 ;
0000     92 ;       V03-007 MIR0023         Michael I. Rosenblum            24-Jan-1983
0000     93 ;               Move the location of setting MULTI in the control-C and Y
0000     94 ;               logic to allow MULTI to be cleared during READONE.
0000     95 ;               Clearing MULTI will stop the Read buffer from
0000     96 ;               being modified after readone dealocates it.
0000     97 ;
0000     98 ;       V03-006 MIR0017         Michael I. Rosenblum            05-Jan-1983
0000     99 ;               Change return status of TTY$PUTNEXTCHAR to include a byte value
0000    100 ;               in the UCB, this will move the information from the condition
0000    101 ;               code bits.
0000    102 ;
0000    103 ;       V03-005 MIR0015         Michael I. Rosenblum            20-Dec-1982
0000    104 ;               Change TTY$V_ST_UNSOL and TTY$V_ST_GETAHD to TTY$V_FD_UNSOL
0000    105 ;               and TTY$V_FD_GETAHD, to reflect changes in the fork dispatcher
0000    106 ;               Change calls to port driver to call the class driver jacket
0000    107 ;               routines.
0000    108 ;
0000    109 ;       V03-004 MIR0014         Michael I. Rosenblum            17-Dec-1982
0000    110 ;               Change PORT_XON and PORT_XOFF to CLASS_XON and CLASS_XOFF
```

```
0000   111 ;
0000   112 ;       V03-003 MIR0013         Michael I. Rosenblum            16-Dec-1982
0000   113 ;               Fix up refferences to new ucb structure
0000   114 ;
0000   115 ;       V03-002 MIR0011         Michael I. Rosenblum            18-Nov-1982
0000   116 ;               Change multiecho to always take a length count and address
0000   117 ;               of a string.
0000   118 ;
0000   119 ;       V03-002 MIR0010         Michael I. Rosenblum            09-Nov-1982
0000   120 ;               Move the address of the terminator mask, and the length
0000   121 ;               of the prompt string from the IRP into the terminal read
0000   122 ;               packet.  Also move the count of the characters in the
0000   123 ;               buffer from the UCB into the terminal typeahead buffer packet.
0000   124 ;               Restructured typeahead buffer alarm size calculation slightly
0000   125 ;               to use the count from the typeahead buffer packet.
0000   126 ;
0000   127 ;       V03-001 RKS0001         RICK SPITZ                      23-SEP-1982
0000   128 ;               RESET ALTLEN VALUE WHEN STARTING MULTIECHO STRING, TO INSURE
0000   129 ;               THAT ^Y ECHO DURING READ VERIFY FUNCTIONS PROPERLY.
0000   130 ;
0000   131 ;       V02-026 RKS0026         RICK SPITZ                      25-JAN-1982
0000   132 ;               CHANGE CONTROL O LOGIC TO WORK WITH BURST MODE OUTPUT
0000   133 ;               (PREVENT WRAP IN MID LINE)
0000   134 ;
0000   135 ;       V02-025 RKS0025         RICK SPITZ                      15-DEC-1981
0000   136 ;               CHANGE INSPOST CALL TO WRITE POST IN CTRLC,Y LOGIC.
0000   137 ;
0000   138 ;       V02-024 RKS0024         RICK SPITZ                      20-NOV-1981
0000   139 ;               ADD OUT OF BAND SUPPORT
0000   140 ;
0000   141 ;       V02-023 JLV0099         Jake VanNoy             27-Oct-1981
0000   142 ;               Changed TTYDEFS to $TTYDEFS.
0000   143 ;
0000   144 ;       V02-022 RKS022          RICK SPITZ                      20-AUG-1981
0000   145 ;               ADD ALTERNATE TYPEAHEAD SIZE SUPPORT
0000   146 ;
0000   147 ;       V02-021 RKS021          RICK SPITZ                      12-AUG-1981
0000   148 ;               CORRECT DEFINITION NAMES
0000   149 ;
0000   150 ;       V02-020 JLV0062         Jake VanNoy                     10-Aug-1981
0000   151 ;               Added autobaud code.
0000   152 ;
0000   153 ;       V02-019 RKS019          RICK SPITZ                      27-JULY-1981
0000   154 ;               SEVERAL NEW FEATURES HAVE BEEN ADDED TO SUPPORT THE
0000   155 ;               CLASS/PORT STRUCTURE OF THE TERMINAL SERVICES. THESE
0000   156 ;               INCLUDE ENHANCEMENTS TO SUPPORT QUADWORD STATE AND
0000   157 ;               MOVING MOST XON/XOFF LOGIC TO THE PORT DRIVER. THE
0000   158 ;               PORT FUNCTIONS RESUME AND STOP(2) ARE USED TO HANDLE
0000   159 ;               RECEIVED CONTROL S AND Q. LOGIC TO HANDLE CONTROL S
0000   160 ;               DURING A BROADCAST IS ALSO INCLUDED.
0000   161 ;
0000   162 ;       V02-018 RKS018          RICK SPITZ                      26-FEB-1981
0000   163 ;               REMOVE V2.0 AUDIT TRAILS
0000   164 ;
0000   165 ;--
```

```
                        .SBTTL  Declarations

;
; EXTERNAL SYMBOLS
;
                        $IODEF                          ; DEFINE I/O FUNCTION CODES
                        $IPLDEF                         ; DEFINE IPL'S
                        $IRPDEF                         ; DEFINE IRP
                        $PRDEF                          ; DEFINE PROCESSOR REGISTERS
                        $SSDEF                          ; DEFINE SYSTEM SERVICE STATUS CODES
                        $UCBDEF                         ; DEFINE UCB
                        $TASTDEF                        ; DEFINE OUT OF BAND FLAGS
                        $TTDEF                          ; DEFINE TERMINAL CHARACTERISTICS
                        $TT2DEF                         ; DEFINE TERMINAL CHARACTERISTICS
                        $TTYDEF                         ; DEFINE TERMINAL DRIVER SYMBOLS
                        $TTYMACS                        ; DEFINE TERMINAL MACROS
                        $TTYDEFS                        ; DEFINE TERMINAL DEFINITIONS

00000000                .PSECT  $$$115_DRIVER,LONG      ; DEFINE NON-PAGED PSECT
```

```
                     .SBTTL  TTY$PUTNEXTCHAR - BUFFER CHARACTER

                     ;++
                     ; TTY$PUTNEXTCHAR - BUFFER CHARACTER
                     ;
                     ; FUNCTIONAL DESCRIPTION:
                     ;
                     ; THIS ROUTINE IS CALLED BY PORT DRIVERS TO PASS INPUT CHARACTERS.
                     ;
                     ; CHARACTERS RECEIVED ON NON PASSALL UNITS ARE FILTERED FOR IMMEDIATE
                     ; CONTROL SEQUENCES. THESE SEQUENCES REPRESENT:
                     ;
                     ;       CONTROL Y --    CAUSES THE TYPEAHEAD BUFFER TO BE PURGED, THE
                     ;                       ENABLED PROCESS TO RECEIVE AN AST, A "^Y" TO
                     ;                       BE OUTPUT AND THE CURRENT OPERATION IF ANY TO
                     ;                       BE COMPLETED WITH A ZERO TRANSFER COUNT FOR READ
                     ;                       AND AS IF CONTROL O FOR WRITE.
                     ;
                     ;       CONTROL C --    CAUSES THE RECEIVER OF CONTROL C AST'S OR THE RECEIVER
                     ;                       OF CONTROL Y AST'S TO BE SIGNALLED AS IN CONTROL Y.
                     ;
                     ;       CONTROL X --    CAUSES THE CONTENTS OF THE TYPEAHEAD BUFFER
                     ;                       TO BE PURGED AND A CONTROL U TO BE INSERTED
                     ;                       IN THE INPUT STREAM IF A READ IS IN PROGRESS.
                     ;
                     ;       CONTROL S --    CAUSES ALL OUTPUT ON UNIT TO STOP UNTIL
                     ;                       CONTROL Q,Y OR C.
                     ;
                     ;       CONTROL Q --    RESETS CONTROL S MODE AND STARTS OUTPUT UP.
                     ;
                     ; SLAVE MODE ( NO UNSOLICITED INPUT ) UNITS MUST HAVE OUTSTANDING
                     ; READS OTHERWISE THE CHARACTER, AFTER CONTROL CHARACTER FILTERING
                     ; IS IGNORED.
                     ;
                     ; INPUTS:
                     ;       R3 = CHARACTER TO BUFFER
                     ;       R5 = UCB
                     ;       R1,R2,R4 ARE AVAILABLE FOR USE
                     ;
                     ; OUTPUTS:
                     ;
                     ;       R3 =    0       AND CC = ZERO
                     ;               CHAR    AND CC = PLUS
                     ;               ADDRESS AND CC = NEGATIVE
                     ;
                     ;       R5 = UCB ADDRESS
                     ;--
                     ;
                     ; LONG REACH TABLE
                     ;
014A  31  0000  237  TAB_CHECKPRE:           BRW     CHECKPRE
01B9  31  0003  238  TAB_CONTINT:            BRW     CONTINT
01A2  31  0006  239  TAB_EIGHTBIT:           BRW     EIGHTBIT
020A  31  0009  240  TAB_OUTBAND_CHAR:      BRW     OUTBAND_CHAR
                000C  241  ;
                000C  242  ; MAIN LINE
                000C  243  ;
```

```
                     000C   244 TTY$PUTNEXTCHAR::
52   00B8 C5   9E    000C   245         MOVAB   UCB$Q_TT_STATE(R5),R2   ; ADDRESS STATUS OF UNIT
     8000 8F   B3    0011   246         BITW    #TT$M_EIGHTBIT,UCB$L_DEVDEPEND(R5); 8BIT TERMINAL?
       44 A5         0015
          ED   12    0017   247         BNEQ    TAB_EIGHTBIT            ; IF NEQ THEN YES
 53     80 8F   8A    0019   248         BICB    #^X080,R3               ; STRIP 8TH BIT
                      001D   249         IF_STATE PASALL,TAB_CHECKPRE    ; IF PASSALL THEN OPTOMIZE
                      0021   250
51   0105 C5   9A    0021   251         MOVZBL  UCB$B_TT_INTCNT(R5),R1  ; ARE WE IN THE MIDDLE OF THE
          DB   12    0026   252         BNEQ    TAB_CONTINT             ; THE INTERUPT KEY - YES THEN HANDLE IT
                      0028   253 ;
                      0028   254 ; CHECK FOR OUT OF BAND CHARACTER
                      0028   255 ;
     20   53   91    0028   256         CMPB    R3,#^A/ /               ; CONTROL CHARACTER?
          DC   1F    002B   257         BLSSU   TAB_OUTBAND_CHAR        ; YES THEN PROCESS IT
                      002D   258 ;       BRB     BUFFER_CHAR             ; FALL INTO BUFFER CHARACTER
```

```
                                002D   260          .SBTTL  BUFFER_CHAR - puts character into typeahead buffer
                                002D   261
                                002D   262 ;++
                                002D   263 ; BUFFER_CHAR - INSERT CHARACTER INTO TYPEAHEAD BUFFER
                                002D   264 ;
                                002D   265 ; FUNCTIONAL DESCRIPTION:
                                002D   266 ;
                                002D   267 ; AT THIS POINT WE KNOW THAT THE CHARACTER DOES NOT REPRESENT SOME IMMEDIATE
                                002D   268 ; ACTION.
                                002D   269 ;
                                002D   270 ; INPUTS:
                                002D   271 ;
                                002D   272 ;       R2 = ADDRESS OF THE UNIT STATE VECTOR
                                002D   273 ;       R3 = CHARACTER TO BUFFER
                                002D   274 ;       R5 = UCB ADDRESS
                                002D   275 ;
                                002D   276 ; OUTPUTS:
                                002D   277 ;
                                002D   278 ;       R2,R3,R5 ARE PRESERVED
                                002D   279 ;--
                                002D   280          .ENABLE LSB
                                002D   281 BUFFER_CHAR:                                 ; BUFFER CHARACTER
                                002D   282          IF_NOT_STATE READ,40$               ; IF READ THEN BUFFER CHARACTER
                                0031   283          IF_STATE PRE,30$                    ; A NOFILTER READ THEN OPTOMIZE
                     OF 68 A5   01  E0  0035   284          BBS     #UCB$V_TT_TIMO,UCB$W_DEVSTS(R5),20$; BR IF TIMEOUT
                                003A   285 10$:
                           00B2 30      003A   286          BSBW    BUFFER_INSERT               ; INSERT THE CHARACTER IN THE BUFFER
                                003D   287
                                003D   288 ; BEGIN ECHO ON READ
                                003D   289 ;
                                003D   290 BEGIN_ECHO:
                                003D   291 ;
                                003D   292 ; START UP OUTPUT IF NOT ALREADY STARTED
                                003D   293 ;
                     64 A5      02  B3  003D   294          BITW    #UCB$M_INT,UCB$W_STS(R5); INTERRUPT EXPECTED?
                                03  12  0041   295          BNEQ    16$                 ; IF NEQ THEN YES
                              FFBA' 31  0043   296          BRW     TTY$GETNEXTCHAR             ; FIND THE NEXT CHARACTER FOR THIS UNIT
                              00A1  31  0046   297 16$:     BRW     DISMISS                 ; EXIT
                                    0049   298 ;
                                    0049   299 ; TIMEOUT ACTIVE - RESET THE TIMER THEN FALL INTO THE NORMAL PATH
                                    0049   300 ;
                                    0049   301 20$:
                     54     78 A5  D0  0049   302          MOVL    UCB$L_SVAPTE(R5),R4         ; ADDRESS READ BUFFER BLOCK
                     54     36 A4  3C  004D   303          MOVZWL  TTY$W_RB_TIMOS(R4),R4       ; GET THE NUMBER OF SECONDS
                                E7  13  0051   304          BEQL    10$                 ; BR IF READ WITH ZERO SECOND TIMEOUT.
                                54  C1  0053   305          ADDL3   R4,G^EXE$GL_ABSTIM,UCB$L_TT_RDUE(R5); RESET THE TIME
                     00000000'GF        0055
                           00B0 C5      005A
                                DB  11  005D   306          BRB     10$                 ; RETURN TO THE MAIN LINE
                              00FE  31  005F   307 30$:     BRW     GOPASS
                                    0062   308
                                    0062   309 ;
                                    0062   310 ; NO READ IS CURRENTLY ACTIVE
                                    0062   311 ;
                     1F 44 A5   02  E0  0062   312 40$:     BBS     #TT$V_NOTYPEAHD,UCB$L_DEVDEPEND(R5),5$; IF SLAVED TERMINAL, IGNORE C
                      51    00C0 C5  D0  0067   313          MOVL    UCB$L_TT_LOGUCB(R5),R1      ; GET LOGICAL UCB ADDRESS
                                5C A1  B5  006C   314          TSTW    UCB$W_REFC(R1)              ; UNIT REF COUNT 0?
```

```
                 18   13  006F   315          BEQL    50$                             ; IF EQL THEN JOB CONTROLLER POSSIBILITY
               60 A1   D5  0071   316          TSTL    UCB$L_AMB(R1)                   ; USER ASSOCIATED MAILBOX?
                 C4   13  0074   317          BEQL    10$                             ; IF EQL THEN NO
     BF 44 A5    10   E0  0076   318          BBS     #TT$V_MBXDSABL,UCB$L_DEVDEPEND(R5),10$; BR IF NOT ENABLED
     BA 68 A5    02   E0  007B   319          BBS     #UCB$V_TT_NOTIF,UCB$W_DEVSTS(R5),10$; BR IF ALREADY NOTIFIED
                FF7D'  30  0080   320          BSBW    TTY$NOTIFY
                FFB4   31  0083   321  4$:     BRW     10$
                0061   31  0086   322  5$:     BRW     DISMISS                         ; CONTINUE
                       0089   323
                       0089   324  ; Before checking terminator, check for autobaud detect
                       0089   325
     08 48 A5    01   E1  0089   326  50$:     BBC     #TT2$V_AUTOBAUD,UCB$L_DEVDEPND2(R5),60$ ; Branch if no autobaud
                53   0D  91  008E   327          CMPB    #TTY$C_CR,R3                    ; CR MEANS THAT WE ARE CORRECT
                 08   13  0091   328          BEQL    65$                             ; SO FALL THRU
                FF6A'  30  0093   329          BSBW    TTY$AUTOBAUD                    ; Check for correct baud rate
     EB 48 A5    10   E0  0096   330  60$:     BBS     #TT2$V_SECURE,UCB$L_DEVDEPND2(R5),4$; DON'T EVER INITIATE
                       0098   331                                                  ; HERE IF SECURE SERVER IS SET
      0000'CF    53   E1  0098   332  65$:     BBC     R3,W^TTY$A_STANDARD,4$  ; IF NOT TERMINATOR THEN NOT FOR JOBCTLRLR
                       00A0
       00000068'GF   B5  00A1   333          TSTW    G^SYS$GL_JOBCTLMB+UCB$W_DEVSTS; TERMINALS ENABLED FOR JOBCTLR?
                 DD   15  00A7   334          BLEQ    5$                              ; IF LEQ THEN DISMISS
                FF54'  30  00A9   335  70$:     BSBW    TTY$NOTIFY                      ; NOTIFY THE JOB CONTROLER
                0040   30  00AC   336          BSBW    BUFFER_INSERT                   ; BUFFER THE CHARACTER
     36 64 A5    01   E0  00AF   337          BBS     #UCB$V_INT,UCB$L_STS(R5),DISMISS; DON'T RETURN DATA IF INT EXPECTED
       00000000'GF   9A  00B4   338          MOVZBL  G^TTY$GB_AUTOCHAR,R3    ; AND RETURN THE AUTOBAUDED CHARACTER
                 53                00BA
                 2D   13  00BB   339          BEQL    DISMISS                         ; NO CHARACTER TO RETURN THEN EXIT
                       00BD   340          TIMSET  #1,R1,LOCKOUTPUT                ; LOCK THE OUTPUT STREAM AND TIME OUT THE CH
      0108 C5    01   90  00DE   341          MOVB    #1,UCB$B_TT_OUTYPE(R5)  ;SETUP THAT WE RETURNED DATA
                 05   05  00E3   342          RSB                                     ; THEN RETURN
                       00E4   343  ;
                       00E4   344  ; NO TYPEAHEAD BUFFER - ALLOCATE ONE
                       00E4   345  ;
                       00E4   346  NO_BUFFER:                                      ;
     54    01    D0  00E4   347          MOVL    #TTY$V_FD_GETAHD,R4             ; ASK FOR TYPEAHEAD FORK
        FF16'   30  00E7   348          BSBW    TTY$CRE_FORK                    ; CREATE THE FORK FOR ALLOCATION
                       00EA   349  ;
                       00EA   350  ; DISMISS INTERRUPT - NOTHING TO DO
                       00EA   351  ;
                       00EA   352  DISMISS:
      0108 C5    94   00EA   353          CLRB    UCB$B_TT_OUTYPE(R5)             ; SET NO RETURN CHARACTER
                 05   00EE   354          RSB                                     ; RETURN
                       00EF   355          .DISABLE LSB
```

```
                     00EF   357          .SBTTL  BUFFER_INSERT - INSERT CHARACTERS INTO THE TYPEAHEAD BUFFER
                     00EF   358 ;++
                     00EF   359 ;
                     00EF   360 ; BUFFER_INSERT - BUFFER CHARACTER IN CIRCULAR TYPEAHEAD BUFFER
                     00EF   361 ;
                     00EF   362 ; TEST TO SEE IF THE NUMBER OF CHARACTERS IN THE TYPEAHEAD IS CRITICAL
                     00EF   363 ; THIS TEST DOES NOT WORK FOR TYPEAHEAD BUFFERS BIGGER THAN 32K BYTES.
                     00EF   364 ;
                     00EF   365 ; INPUTS:
                     00EF   366 ;       R3 - CHARACTER TO INSERT
                     00EF   367 ;       R5 - PHYSICAL UCB ADDRESS
                     00EF   368 ;
                     00EF   369 ; OUTPUTS:
                     00EF   370 ;       NONE
                     00EF   371 ;
                     00EF   372 ;       R4,R1 ARE DESTROYED
                     00EF   373 ;
                     00EF   374 ;--
                     00EF   375 BUFFER_INSERT:
54   00E4 C5   D0    00EF   376          MOVL    UCB$L_TT_TYPAHD(R5),R4  ; ADDRESS TYPEAHEAD BUFFER
           EE   13   00F4   377          BEQL    NO_BUFFER               ; IF EQL THEN NONE
           07   E1   00F6   378          BBC     #TT2$V_ALTYPEAHD,-      ; SKIP IF NORMAL TYPE AHEAD SIZE
      1C 48 A5         00F8   379                  UCB$L_DEVDEPND2(R5),55$
                       00FB   380
 00000000'GF   A3    00FB   381          SUBW3   G^TTY$GW_ALTALARM,G^TTY$GW_ALTYPAHD,R1  : WITHIN N CHARS OF TYPAHD F
 00000000'GF         0101
           51        0106
        01 51   3D   0107   382          ACBW    R1,#1,TTY$W_TA_INAHD(R4),60$            ; BRANCH IF NO
      0030 0C A4         010A
 00000000'GF   A0    010E   383          ADDW2   G^TTY$GW_ALTALARM,R1                    ; MAX # TYPEAHEAD CHARS
           51        0114
           12   11   0115   384          BRB     57$                                     ; JOIN COMMON PATH
                     0117   385
           08   A3   0117   386 55$:     SUBW3   #8,G^TTY$GW_TYPAHDSZ,R1         ; WITHIN 8 CHARS OF TYPAHD FULL?
 00000000'GF         0119
           51        011E
        01 51   3D   011F   387          ACBW    R1,#1,TTY$W_TA_INAHD(R4),60$    ; BRANCH IF NO
      0018 0C A4         0122
        51 08   A0   0126   388          ADDW2   #8,R1                           ; MAX # TYPEAHEAD CHARS
         FED4'  30   0129   389 57$:     BSBW    TTY$XOFF                        ; SEND XOFF
                     012C   390          SET_STATE <TYPFUL>                      ; SET UP STOP SEQUENCE
      0C A4 51   B1   0130   391          CMPW    R1,TTY$W_TA_INAHD(R4)           ; TYPEAHEAD BUFFER OVERFLOW?
           08   18   0134   392          BGEQ    60$                             ; IF GEQ THEN NO OVERFLOW - BUFFER CHARACTER
         0C A4  B7   0136   393          DECW    TTY$W_TA_INAHD(R4)              ; RESET COUNT ON OVERFLOW
                     0139   394          SET_STATE <OVRFLO>                      ; INDICATE OVERFLOW CONDITION
                05   013D   395          RSB
                     013E   396 ;
                     013E   397 ; INSERT CHARACTER IN TYPEAHEAD BUFFER
                     013E   398 ;
   00 B4    53   90   013E   399 60$:     MOVB    R3,@TTY$L_TA_PUT(R4)            ; INSERT CHARACTER IN BUFFER
05 64     10 A4   F2   0142   400          AOBLSS  TTY$L_TA_END(R4),TTY$L_TA_PUT(R4),70$; INDEX AND BR IF NOT AROUND
 64     0118 C4   9E   0147   401          MOVAB   TTY$L_TA_DATA(R4),TTY$L_TA_PUT(R4); RESET POINTER
                05   014C   402 70$:     RSB
```

```
                        014D   404          .sbttl  Put next character service routines
                        014D   405 ;
                        014D   406 ; AVOID TYPEAHEAD BUFFER FOR PASSALL MODE
                        014D   407 ;
                        014D   408 .ENABLE LSB
                        014D   409 CHECKPRE:
0A 48 A5    12   E1    014D   410          BBC     #TT2$V_PASTHRU,UCB$L_DEVDEPND2(R5),200$; IS THIS TRUE PASSALL MODE?
                20   93 0152   411          BITB    #TTY$M_CH_CTRL,TTY$A_TYPE[R3]; IS THIS A CONTROL CHAR
00000000'EF43
                1E   12 0154
                        015A   412          BNEQ    250$                    ; YES THEN CHECK FOR CONTORL-S OR Q
                        015C   413 200$:    IF_NOT_STATE PRE,220$           ; NO READ WAITING THEN BUFFER THE CHARACTER
                        0160   414 ;
                        0160   415 ; Pass all optomization verifyed continue
                        0160   416 ;
                        0160   417 GOPASS:
   54   78 A5    D0   0160   418          MOVL    UCB$L_SVAPTE(R5),R4     ; ADDRESS READ BUFFER BLOCK
30 68 A5    01   E0   0164   419          BBS     #UCB$V_TT_TIMO,UCB$W_DEVSTS(R5),270$; BR IF NO TIMEOUT
         FE94'  30   0169   420 210$:    BSBW    PASSALL                 ; OPTOMIZE BY SKIPING THE TYPEAHEAD
                        016C   421          IF_STATE        EOL,230$        ; If read complete then exit
                FF77   31   0170   422          BRW     DISMISS                 ; NORMAL CHARACTER THEN DISMISS THE INTERUPT
                        0173   423 ;
                        0173   424 ; READ TERMINATED
                        0173   425 ;
                        0173   426 230$:    CLR_STATE PRE                   ; EOL THEN CLEAR PRE
                FEC3   31   0177   427          BRW     BEGIN_ECHO              ; AND BEGIN THE EOL SEQUENCE
                        017A   428 ;
                        017A   429 ; Flow control allowed and the character could be a control-s or q
                        017A   430 ;
00000000'EF43   91   017A   431 250$:    CMPB    TTY$A_TYPE[R3],#3!TTY$M_CH_CTRL; IS IT A CONTROL-Q
                23   0181
                D8   1A 0182   432          BGTRU   200$                    ; NO THEN HANDLE NORMALY
                0D   13 0184   433          BEQL    260$                    ; yes then restore flow
00000000'EF43   91   0186   434          CMPB    TTY$A_TYPE[R3],#2!TTY$M_CH_CTRL; IS IT A CONTROL-S
                22   018D
                CC   12 018E   435          BNEQ    200$                    ; NO THEN EXIT
              016A   31 0190   436          BRW     CONTROLQ                ; else stop output
              0176   31 0193   437 260$:    BRW     CONTROLS
                        0196   438 ;
                        0196   439 ; jump to buffer the character
                        0196   440 ;
              FE94   31 0196   441 220$:    BRW     BUFFER_CHAR
                        0199   442 ;
                        0199   443 ; RESET TIMERS AND CONTINUE
                        0199   444 ;
   51   36 A4    3C   0199   445 270$:    MOVZWL  TTY$W_RB_TIMOS(R4),R1   ; GET THE NUMBER OF SECONDS
                CA   13 019D   446          BEQL    210$                    ; BR IF READ WITH ZERO SECOND TIMEOUT.
                51   C1 019F   447          ADDL3   R1,G^EXE$GL_ABSTIM,UCB$L_TT_RDUE(R5); RESET THE TIME
 00000000'GF
      0080 C5
                BE   11 01A9   448          BRB     210$
                        01AB   449          .DISABLE LSB
                        01AB   450          .ENABLE LSB
                        01AB   451
                        01AB   452 ;
                        01AB   453 ; 8 BIT CHARACTER IT MAY BE A CSI
                        01AB   454 ;
                        01AB   455 EIGHTBIT:
```

```
                     01AB   456          IF_STATE PASALL,CHECKPRE          ; IF WE ARE IN PASSALL MODE THEN
                     01AF   457                                            ; DON'T PROCESS CSI
          53   9B 8F   91   01AF   458          CMPB    #TTY$C_CSI,R3      ; NO  THEN IS IT A CSI
                    03   12   01B3   459          BNEQ    305$               ; NO THEN CONTIUE NORMALY
                  0179   31   01B5   460          BRW     CSI                ; ELSE TAKE CSI ACTION.
       51   0105 C5   9A   01B8   461 305$:    MOVZBL  UCB$B_TT_INTCNT(R5),R1 ; ARE WE IN THE MIDDLE OF THE
                    20   13   01BD   462          BEQL    310$               ; THE INTERUPT KEY - YES THEN HANDLE IT
                         01BF   463
                         01BF   464 ;
                         01BF   465 ; CHECK FOR INTERRUPT KEY
                         01BF   466 ;
             0105 C5   94   01BF   467 CONTINT:CLRB    UCB$B_TT_INTCNT(R5)    ; ASUME IT IS THE WRONG KEY
0000'CF41    53   91   01C3   468          CMPB    R3,W^INTERRUPT_KEY[R1] ; IS THIS THE NEXT CHARACTER
                    14   12   01C9   469          BNEQ    310$               ; NO THEN THE STATE IS CORRECT SO EXIT
          51   01   81   01CB   470          ADDB3   #1,R1,UCB$B_TT_INTCNT(R5); INCREMENT THE COUNT
             0105 C5,       01CE
       00'   0105 C5,  91   01D1   471          CMPB    UCB$B_TT_INTCNT(R5),S^#INTERRUPT_KEY_LEN
                    07   12   01D6   472          BNEQ    310$
             0105 C5   94   01D8   473          CLRB    UCB$B_TT_INTCNT(R5)    ; COMPLETE ESCAPE SEQUENCE THEN RESET
                  0062   31   01DC   474          BRW     CONTROLC
          20        53   91   01DF   475 310$:    CMPB    R3,#^A/ /              ; IS THIS CHARACTER OUT OF BAND
                    32   1F   01E2   476          BLSSU   OUTBAND_CHAR           ; YES THEN HANDLE CORRECTLY
                 FE46   31   01E4   477          BRW     BUFFER_CHAR            ; ELSE BUFFER THE CHARACTER
                         01E7   478          .DISABLE LSB
                         01E7   479 ;
                         01E7   480 ; OUT OF BAND CHARACTER PROCESSING
                         01E7   481 ;
                         01E7   482 DELOUTBAND:
          54   009C C1   DE   01E7   483          MOVAL   UCB$L_TL_BANDQUE(R1),R4        ; ADDRESS OF QUEUE
                    56   DD   01EC   484          PUSHL   R6                             ; SAVE R6
          56   00A4 C1   D0   01EE   485          MOVL    UCB$L_TL_CILPID(R1),R6         ; GET THE CONTROLING PID
           00000000'GF   16   01F3   486          JSB     G^COM$DECTRLASTP               ; DELIVER ANY OUTSTANDING ASTS
                 56 8ED0   01F9   487          POPL    R6                             ; RESTORE R6
                         01FC   488 ;
                         01FC   489 ; CHECK FOR INCLUDE OR ABORT
                         01FC   490 ;
       0F 53    0E   E5   01FC   491          BBCC    #TAST$V_ABO,R3,5$              ; CHECK ABORT FIRST
             0651 8F   3C   0200   492          MOVZWL  #SS$_CONTROLC,UCB$W_BOFF(R5)   ; SET STATUS
                7C A5       0204
             00DC C5   B4   0206   493          CLRW    UCB$W_TT_MULTILEN(R5)          ; CLEAR, NO EHCO TO PREFORM
                FDF3'  30   020A   494          BSBW    TTY$ABORT_IO                   ; ABORT I/O
                    04   11   020D   495          BRB     8$                             ; CONTINUE
       0E 53    0F   E4   020F   496 5$:       BBSC    #TAST$V_INC,R3,CHECASE         ; DONT STRIP CHARACTER
                FED4   31   0213   497 8$:       BRW     DISMISS
                         0216   498 ;
                         0216   499 ; CHECK FOR CONTROL SEQUENCE CHARACTER
                         0216   500 ;
                         0216   501 OUTBAND_CHAR:
                         0216   502
       51   00C0 C5   D0   0216   503          MOVL    UCB$L_TT_LOGUCB(R5),R1         ; GET LOGICAL UCB ADDRESS
0098 C1    53   E0   021B   504          BBS     R3,UCB$L_TL_OUTBAND(R1),DELOUTBAND; NOT IN SUMMARY MASK
                    C6   0220
                         0221   505
                         0221   506 CHECASE:
                         0221   507          CASE    W^TTY$A_TYPE[R3],TYPE=B,LIMIT=#1@TTY$V_CH_CTRL,-
                         0221   508                  <CONTROLC,CONTROLO,CONTROLQ,CONTROLS,CONTROLX,CONTROLY,ESCAPE,CSI>
                FDF2   31   0238   509          BRW     BUFFER_CHAR
```

```
                     023B   511          .SBTTL  PRE-TYPEAHEAD CONTROL CHARACTER ACTION ROUTINES.
                     023B   512 ;
                     023B   513 ; ENTRY FROM CONTROLY AND CONTROLC
                     023B   514 ;
                     023B   515 CANCEL_CTRLS:
FDC2'  30            023B   516          BSBW    TTY$RESUME              ; RESUME ANY PORT OUTPUT
FDFC   31            023E   517          BRW     BEGIN_ECHO              ; BEGIN OUTPUT
```

```
                     0241   519
                     0241   520          .SBTTL  CONTROLC, CONTROLY handlers
                     0241   521
                     0241   522          .ENABL  LSB
                     0241   523
                     0241   524 ;++
                     0241   525 ; CONTROLC - SIGNAL CONTROL C INPUT
                     0241   526 ; CONTROLY - SIGNAL CONTROL Y INPUT
                     0241   527 ;
                     0241   528 ; FUNCTIONAL DESCRIPTION:
                     0241   529 ;
                     0241   530 ; THIS ROUTINE IS ENTERED WHEN A CONTROL C OR Y IS TYPED.
                     0241   531 ; THE ACTION IS TO SIGNAL, VIA AN AST, THE HOLDER OF THE AST ENABLE.
                     0241   532 ; IF NO ENABLE IS PRESENT, AN ATTEMPT IS MADE TO SIGNAL THE JOB CONTROLLER
                     0241   533 ; IF IT HAS NOT BEEN SIGNALED.
                     0241   534 ;
                     0241   535 ; INPUTS:
                     0241   536 ;
                     0241   537 ;       R2 = ADDRESS OF THE UNIT STATE VECTOR
                     0241   538 ;       R5 = UCB ADDRESS
                     0241   539 ;
                     0241   540 ; OUTPUTS:
                     0241   541 ;
                     0241   542 ;       R2 = ADDRESS OF THE UNIT STATE VECTOR
                     0241   543 ;       R5 = UCB ADDRESS
                     0241   544 ;
                     0241   545 ;--
                     0241   546
           00000001  0241   547 M_CTRLC  = 1
           00000002  0241   548 M_REGIS  = 2
           00000004  0241   549 M_DECCRT = 4
                     0241   550
                     0241   551 CONTROLC:                                    ; ENTRY FOR CONTROL C
51   00C0 C5    D0   0241   552          MOVL    UCB$L_TT_LOGUCB(R5),R1      ; GET LOGICAL UCB ADDRESS
54   0094 C1    DE   0246   553          MOVAL   UCB$L_TL_CTRLC(R1),R4       ; GET ADDRESS OF CONTROL C AST LIST
          64    D5   024B   554          TSTL    (R4)                        ; EMPTY?
          10    13   024D   555          BEQL    10$                         ; IF EQL THEN NO CTRLC
     0651 8F    3C   024F   556          MOVZWL  #SS$_CONTROLC,UCB$W_BOFF(R5)    ; SET STATUS AND ZERO COUNT
       7C A5              0253
          53    01   9A   0255   557          MOVZBL  #M_CTRLC,R3                 ; Set "Cancel"
          19    11   0258   558          BRB     15$                         ; Branch
                     025A   559 ;
                     025A   560 ; CONTROL Y PROCESSING
                     025A   561 ;
                     025A   562
                     025A   563 CONTROLY:
51   00C0 C5    D0   025A   564          MOVL    UCB$L_TT_LOGUCB(R5),R1      ; GET LOGICAL UCB ADDRESS
54   0090 C1    DE   025F   565 10$:     MOVAL   UCB$L_TL_CTRLY(R1),R4       ; GET ADDRESS OF CONTROL Y AST LIST
                     0264   566
          64    D5   0264   567          TSTL    (R4)                        ; EMPTY LIST?
          03    12   0266   568          BNEQ    13$                         ; No, branch forward.
        0097    31   0268   569          BRW     TAB_BUFFER                          ; Yes. Don't process it.
                     026B   570 13$:
     0611 8F    3C   026B   571          MOVZWL  #SS$_CONTROLY,UCB$W_BOFF(R5)    ; SET STATUS AND ZERO COUNT
       7C A5              026F
          53    D4   0271   572          CLRL    R3                          ; Set "Interrupt"
                     0273   573          ;
```

```
                     0273   574          ; Common ^C and ^Y code
                     0273   575          ;
                     0273   576 15$:
03 48 A5   1D   E1   0273   577          BBC     #TT2$V_DECCRT,UCB$L_DEVDEPND2(R5),18$; IS THIS A DECCRT?
      53   04   88   0278   578          BISB    #M_DECCRT,R3                ; YES THEN DO A DECCRTECHO
03 48 A5   19   E1   027B   579 18$:     BBC     #TT2$V_REGIS,UCB$L_DEVDEPND2(R5),20$; IS THIS REGIS
      53   02   88   0280   580          BISB    #M_REGIS,R3                 ; YES THEN GET OUT OF REGIS MODE
                     0283   581 20$:
                     0283   582          IF_STATE NINTMULTI,25$              ; DON'T INTERUPT THE NON-INTERUPT MULTIECHOS
51    0000'CF   D0   0287   583          MOVL    W^TTY$A_INTECHO,R1          ; FETCH TABLE POINTER
   53    6143   D0   028C   584          MOVL    (R1)[R3],R3                 ; AND OFFSET
00DC C5    83   9B   0290   585          MOVZBW  (R3)+,UCB$W_TT_MULTILEN(R5); SETUP THE LENGTH OF THE MULTIECHO STRIN
00D8 C5    53   D0   0295   586          MOVL    R3,UCB$L_TT_MULTI(R5)       ; SET UP FOR MULTIECHO
                     029A   587          SET_STATE NINTMULTI                 ; MAKE THIS ONE OF THE NON-INTERUPTABLE
                     029E   588                                              ; MULTIECHO STRING
           56   DD   029E   589 25$:     PUSHL   R6
56    00C0 C5   D0   02A0   590          MOVL    UCB$L_TT_LOGUCB(R5),R6      ; GET THE LOGICAL UCB ADDRESS
56    00A4 C6   D0   02A5   591          MOVL    UCB$L_TL_CTLPID(R6),R6      ; THEN GET THE PID
 00000000'GF   16   02AA   592          JSB     G^COM$DELATTNASTP           ; DELIVER ATTENTION ASTS FOR THIS PID
           56 8ED0   02B0   593          POPL    R6
                     02B3   594
        FD4A'   30   02B3   595          BSBW    TTY$ABORT_IO
        FF82    31   02B6   596          BRW     CANCEL_CTRLS                ; CANCEL CONTROL S AND BEGIN ECHO
```

```
                        02B9    598          .SBTTL  CONTROLO handler
                        02B9    599
                        02B9    600 ;++
                        02B9    601 ; CONTROLO - START OR STOP OUTPUT ON UNIT
                        02B9    602 ;
                        02B9    603 ; FUNCTIONAL DESCRIPTION:
                        02B9    604 ;
                        02B9    605 ; THIS ROUTINE TOGGLES THE OUTPUT ENABLE OF A UNIT.
                        02B9    606 ; OUTPUT IS STOPPED UNTIL THE NEXT READ OPERATION, IO$_WRTCANCTRLO
                        02B9    607 ; OR CONTROL O.
                        02B9    608 ;
                        02B9    609 ; INPUTS:
                        02B9    610 ;
                        02B9    611 ;       R2 = ADDRESS OF THE UNIT STATE VECTOR
                        02B9    612 ;       R5 = UCB ADDRESS
                        02B9    613 ;
                        02B9    614 ; OUTPUTS:
                        02B9    615 ;
                        02B9    616 ;       R2 = ADDRESS OF THE UNIT STATE VECTOR
                        02B9    617 ;       R5 = UCB ADDRESS
                        02B9    618 ;--
                        02B9    619
              00000001  02B9    620 M_CTRLO_DEC = 1
              00000002  02B9    621 M_CTRLO_ON  = 2
                        02B9    622
                        02B9    623 CONTROLO:
                        02B9    624
                        02B9    625          IF_NOT_STATE READ,120$          ; IF NOT READ THEN HONOR
                        02BD    626          IF_NOT_STATE CTRLR,TAB_DISMISS          ; IF CONTROL R THEN IGNORE
04 A2   01    AC        02C1    627 120$:    XORW    #<TTY$M_ST_CTRLO>,4(R2) ; FLOP CONTROL O BIT
                        02C5    628          IF_NOT_STATE CTRLO,OUTPUTON     ; IF NOW CLEAR THEN START OUTPUT
      FD34'   30        02C9    629          BSBW    TTY$ABORT               ; ABORT PORT OUTPUT
        53    D4        02CC    630          CLRL    R3                      ; CLEAR ECHO FLAG
        03    11        02CE    631          BRB     CTRLO_ECHO              ; STARTUP THE OUTPUT
                        02D0    632 ;
                        02D0    633 ; RESTART OUTPUT ON STOPPED UNIT
                        02D0    634 ;
                        02D0    635 OUTPUTON:
      53    02    D0    02D0    636          MOVL    #M_CTRLO_ON,R3          ; SET FLAG
                        02D3    637
                        02D3    638 CTRLO_ECHO:
                        02D3    639          IF_STATE NINTMULTI,TAB_CANCEL_CTRLS; don't bother non-interuptable multiecho
                        02D7    640          SET_STATE <MULTI,NINTMULTI>
        1D    E1        02DF    641          BBC     #TT2$V_DECCRT,-
03 48 A5                02E1    642                  UCB$L_DEVDEPND2(R5),150$ ; BRANCH IF NOT DECCRT
        53    01    A8  02E4    643          BISW    #M_CTRLO_DEC,R3
                        02E7    644 150$:
51    0000'CF   D0      02E7    645          MOVL    W^TTY$A_CTRLOECHO,R1    ; INSERT ADDRESS OF STRING
    53    6143  D0      02EC    646          MOVL    (R1)[R3],R3             ; AND OFFSET
00DC C5   83    9B      02F0    647          MOVZBW  (R3)+,UCB$W_TT_MULTILEN(R5); SETUP THE LENGTH OF THE MULTIECHO STRIN
00D8 C5   53    D0      02F5    648          MOVL    R3,UCB$L_TT_MULTI(R5)   ; SET UP FOR MULTIECHO
                        02FA    649 TAB_CANCEL_CTRLS:
          FF3E  31      02FA    650          BRW     CANCEL_CTRLS            ; RESUME OUTPUT
```

```
                     02FD    652          .SBTTL  CONTROLQ handler
                     02FD    653
                     02FD    654 ;++
                     02FD    655 ; CONTROLQ - START OUTPUT ON CONTROL S STOPPED UNIT
                     02FD    656 ;
                     02FD    657 ; FUNCTIONAL DESCRIPTION:
                     02FD    658 ;
                     02FD    659 ; THIS ROUTINE STARTS OUTPUT ON A UNIT WHICH IS STOPPED BECAUSE
                     02FD    660 ; OF CONTROL S.
                     02FD    661 ;
                     02FD    662 ; INPUTS:
                     02FD    663 ;
                     02FD    664 ;       R2 = ADDRESS OF THE UNIT STATE VECTOR
                     02FD    665 ;       R5 = UCB ADDRESS
                     02FD    666 ;
                     02FD    667 ; OUTPUTS:
                     02FD    668 ;
                     02FD    669 ;       R2 = ADDRESS OF THE UNIT STATE VECTOR
                     02FD    670 ;       R5 = UCB ADDRESS
                     02FD    671 ;--
                     02FD    672
                     02FD    673 CONTROLQ:
44 A5    05    E0    02FD    674          BBS     #TT$V_TTSYNC,UCB$L_DEVDEPEND(R5),-
                F8           0301    675                  TAB_CANCEL_CTRLS        ; TERMINAL SYNCH?
                             0302    676 TAB_BUFFER:
                             0302    677          IF_STATE_PRE,TAB_GOPASS         ; PASSALL OPTOMIZATION
                FD24    31   0306    678          BRW     BUFFER_CHAR             ; BUFFER THE CHARACTER NORMALLY
                             0309    679 TAB_GOPASS:
                FE54    31   0309    680          BRW     GOPASS
```

```
                     030C   682          .SBTTL  CONTROLS handler
                     030C   683
                     030C   684 ;++
                     030C   685 ; CONTROLS - STOP OUTPUT TO UNIT
                     030C   686 ;
                     030C   687 ; FUNCTIONAL DESCRIPTION:
                     030C   688 ;
                     030C   689 ; THIS ROUTINE IS ENTERED WHEN THE USER TYPES A CONTROL S.
                     030C   690 ; ON UNITS THAT DO NOT HAVE THE TT$V_TTSYNC CHARACTERISTIC, THE OUTPUT
                     030C   691 ; IS STOPPED BY TURNING THE OUTPUT INTERRUPT ENABLE BIT IN THE UCB
                     030C   692 ; OFF.
                     030C   693 ;
                     030C   694 ; INPUTS:
                     030C   695 ;
                     030C   696 ;       R2 = ADDRESS OF THE UNIT STATE VECTOR
                     030C   697 ;       R5 = UCB ADDDRESS
                     030C   698 ;
                     030C   699 ; OUTPUTS:
                     030C   700 ;
                     030C   701 ;       R2 = ADDRESS OF THE UNIT STATE VECTOR
                     030C   702 ;       R5 = UCB ADDRESS
                     030C   703 ;--
                     030C   704
                     030C   705 CONTROLS:
F1 44 A5    05   E1  030C   706          BBC     #TT$V_TTSYNC,UCB$L_DEVDEPEND(R5),TAB_BUFFER; TERMINAL SYNCH?
            FCEC'    30  0311   707          BSBW    TTY$STOP                    ; STOP PORT OUTPUT
                     0314   708 TAB_DISMISS:
            FDD3     31  0314   709          BRW     DISMISS                     ; AND RETURN TO DRIVER CODE
```

```
                     0317   711          .SBTTL  CONTROLX handler
                     0317   712
                     0317   713 ;++
                     0317   714 ; CONTROLX - PURGE TYPEAHEAD BECAUSE OF CONTROL X INPUT
                     0317   715 ;
                     0317   716 ; FUNCTIONAL DESCRIPTION:
                     0317   717 ;
                     0317   718 ; THIS ROUTINE PURGES THE TYPEAHEAD BUFFER AND IF THE REASON IS THAT
                     0317   719 ; A CONTROL X WAS TYPED DURING A READ THEN A CONTROL U IS INSERTED IN
                     0317   720 ; THE INPUT STREAM TO PURGE THE CURRENT INPUT LINE.
                     0317   721 ;
                     0317   722 ; INPUTS:
                     0317   723 ;
                     0317   724 ;       R2 = ADDRESS OF THE UNIT STATE VECTOR
                     0317   725 ;       R5 = UCB ADDRESS
                     0317   726 ;
                     0317   727 ; OUTPUTS:
                     0317   728 ;
                     0317   729 ;       R2 = ADDRESS OF THE UNIT STATE VECTOR
                     0317   730 ;       R3 = CONTROL U - IF CONTROL X TYPED DURING READ
                     0317   731 ;       R5 = UCB ADDRESS
                     0317   732 ;--
                     0317   733
                     0317   734 CONTROLX:
        FCE6'   30   0317   735          BSBW    TTY$PURGE_AHEAD          ; PURGE TYPEAHEAD
                     031A   736          IF_NOT_STATE READ,TAB_DISMISS    ; DONE IF NO READ IN PROGRESS
     53   15   9A   031E   737          MOVZBL  #TTY$C_CTRLU,R3          ; FORCE A CONTROL U IN INPUT STREAM
        FFDE   31   0321   738          BRW     TAB_BUFFER               ; CONTINUE AND BUFFER CHARACTER
;MIR0001             0324    .1 .disable lsb
```

```
                                0324   740          .SBTTL  ESCAPE, CSI handler
                                0324   741 ;++
                                0324   742 ; Escape - introducer for the cancel/interrupt key
                                0324   743 ;
                                0324   744 ;--
                                0324   745 ESCAPE:
;MIR0001        05 48 A5   0C   E1   0324    .1          BBC     #TT2$V_EDITING,UCB$L_DEVDEPND2(R5),99$; DO NOTHING ON NO LINE
;MIR0001         0105 C5   01   90   0329    .2          MOVB    #1,UCB$B_TT_INTCNT(R5)  ; SET INTERRUPT COUNT
;MIR0001               FCFC 31   032E    .3 99$:        BRW     BUFFER_CHAR
;MIR0001                            0331    .4 ;++
;MIR0001                            0331    .5 ; CSI - CANCEL INTERUPT INTRODUCER
;MIR0001                            0331    .6 ;
;MIR0001                            0331    .7 ;--
;MIR0001                            0331    .8 CSI:
;MIR0001        05 48 A5   0C   E1   0331    .9          BBC     #TT2$V_EDITING,UCB$L_DEVDEPND2(R5),99$; DO NOTHING ON NO LINE
;MIR0001         0105 C5   02   90   0336   .10          MOVB    #2,UCB$B_TT_INTCNT(R5)  ; MOVE BY THE ESC [ IN THE STRING
;MIR0001               FCEF 31   033B   .11 99$:        BRW     BUFFER_CHAR
```

```
033E    756          .SBTTL  End of module
033E    757
033E    758          .END
```

```
BEGIN_ECHO                      0000003D R      02
BUFFER_CHAR                     0000002D R      02
BUFFER_INSERT                   000000EF R      02
CANCEL_CTRLS                    0000023B R      02
CHECASE                         00000221 R      02
CHECKPRE                        0000014D R      02
CNT                           = 00000001
COM$DELATTNASTP                 ********    X   02
COM$DELCTRLASTP                 ********    X   02
CONTINT                         000001BF R      02
CONTROLC                        00000241 R      02
CONTROLO                        000002B9 R      02
CONTROLQ                        000002FD R      02
CONTROLS                        0000030C R      02
CONTROLX                        00000317 R      02
CONTROLY                        0000025A R      02
CSI                             00000331 R      02
CTRLO_ECHO                      000002D3 R      02
DELOUTBAND                      000001E7 R      02
DISMISS                         000000EA R      02
EIGHTBIT                        000001AB R      02
ESCAPE                          00000324 R      02
EXE$GL_ABSTIM                   ********    X   02
F                             = 00000000
GOPASS                          00000160 R      02
INTERRUPT_KEY                   ********    X   02
INTERRUPT_KEY_LEN               ********    X   02
M_CTRLC                       = 00000001
M_CTRLO_DEC                   = 00000001
M_CTRLO_ON                    = 00000002
M_DECCRT                      = 00000004
M_REGIS                       = 00000002
NO_BUFFER                       000000E4 R      02
OUTBAND_CHAR                    00000216 R      02
OUTPUTON                        000002D0 R      02
PASSALL                         ********    X   02
SS$_CONTROLC                  = 00000651
SS$_CONTROLY                  = 00000611
SYS$GL_JOBCTLMB                 ********    X   02
T                             = 0000003B
TAB_BUFFER                      00000302 R      02
TAB_CANCEL_CTRLS                000002FA R      02
TAB_CHECKPRE                    00000000 R      02
TAB_CONTINT                     00000003 R      02
TAB_DISMISS                     00000314 R      02
TAB_EIGHTBIT                    00000006 R      02
TAB_GOPASS                      00000309 R      02
TAB_OUTBAND_CHAR                00000009 R      02
TAST$V_ABO                    = 0000000E
TAST$V_INC                    = 0000000F
TT$M_EIGHTBIT                 = 00008000
TT$V_MBXDSABL                 = 00000010
TT$V_NOTYPEAHD                = 00000002
TT$V_TTSYNC                   = 00000005
TT2$V_ALTYPEAHD               = 00000007
TT2$V_AUTOBAUD                = 00000001
TT2$V_DECCRT                  = 0000001D
TT2$V_EDITING                 = 0000000C
TT2$V_PASTHRU                 = 00000012
TT2$V_REGIS                   = 00000019
TT2$V_SECURE                  = 00000010
TTY$ABORT                       ********    X   02
TTY$ABORT_IO                    ********    X   02
TTY$AUTOBAUD                    ********    X   02
TTY$A_CTRLOECHO                 ********    X   02
TTY$A_INTECHO                   ********    X   02
TTY$A_STANDARD                  ********    X   02
TTY$A_TYPE                      ********    X   02
TTY$CRE_FORK                    ********    X   02
TTY$C_CR                      = 0000000D
TTY$C_CSI                     = 0000009B
TTY$C_CTRLU                   = 00000015
TTY$GB_AUTOCHAR                 ********    X   02
TTY$GETNEXTCHAR                 ********    X   02
TTY$GW_ALTALARM                 ********    X   02
TTY$GW_ALTYPAHD                 ********    X   02
TTY$GW_TYPAHDSZ                 ********    X   02
TTY$L_TA_DATA                 = 00000118
TTY$L_TA_END                  = 00000010
TTY$L_TA_PUT                  = 00000000
TTY$M_CH_CTRL                 = 00000020
TTY$M_ST_CTRLO                = 00000001
TTY$M_ST_MULTI                = 00000040
TTY$M_ST_NINTMULTI            = 08000000
TTY$M_ST_OVRFLO               = 00010000
TTY$M_ST_PRE                  = 04000000
TTY$M_ST_TYPFUL               = 00001000
TTY$NOTIFY                      ********    X   02
TTY$PURGE_AHEAD                 ********    X   02
TTY$PUTNEXTCHAR                 0000000C RG     02
TTY$RESUME                      ********    X   02
TTY$STOP                        ********    X   02
TTY$V_CH_CTRL                 = 00000005
TTY$V_FD_GETAHD               = 00000001
TTY$V_PC_NOTIME               = 00000000
TTY$V_SX_CTRLO                = 00000020
TTY$V_SX_CTRLR                = 00000032
TTY$V_SX_EOL                  = 00000008
TTY$V_SX_MULTI                = 00000006
TTY$V_SX_NINTMULTI            = 0000003B
TTY$V_SX_OVRFLO               = 00000030
TTY$V_SX_PASALL               = 00000022
TTY$V_SX_PRE                  = 0000003A
TTY$V_SX_READ                 = 0000000C
TTY$V_SX_TYPFUL               = 0000002C
TTY$W_RB_TIMOS                = 00000036
TTY$W_TA_INAHD                = 0000000C
TTY$XOFF                        ********    X   02
UCB$B_TT_INTCNT               = 00000105
UCB$B_TT_OUTYPE               = 0000010B
UCB$L_AMB                     = 00000060
UCB$L_DEVDEPEND               = 00000044
UCB$L_DEVDEPND2               = 00000048
UCB$L_DUETIM                  = 0000006C
```

Symbol table

```
UCB$L_STS                       = 00000064
UCB$L_SVAPTE                    = 00000078
UCB$L_TL_BANDQUE                = 0000009C
UCB$L_TL_CTLPID                 = 000000A4
UCB$L_TL_CTRLC                  = 00000094
UCB$L_TL_CTRLY                  = 00000090
UCB$L_TL_OUTBAND                = 00000098
UCB$L_TT_LOGUCB                 = 000000C0
UCB$L_TT_MULTI                  = 000000D8
UCB$L_TT_RDUE                   = 000000B0
UCB$L_TT_TYPAHD                 = 000000E4
UCB$M_INT                       = 00000002
UCB$M_TIM                       = 00000001
UCB$Q_TT_STATE                  = 000000B8
UCB$V_INT                       = 00000001
UCB$V_TT_NOTIF                  = 00000002
UCB$V_TT_TIMO                   = 00000001
UCB$W_BOFF                      = 0000007C
UCB$W_DEVSTS                    = 00000068
UCB$W_REFC                      = 0000005C
UCB$W_STS                       = 00000064
UCB$W_TT_MULTILEN               = 000000DC
UCB$W_TT_PRTCTL                 = 00000122
W0                              = 00000040
W1                              = 00000008
X                               = 00000000
X0                              = 00000000
X1                              = 00000003
Z0                              = 00000000
Z1                              = 00000003
```

Psect synopsis

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
ABS	00000000 (0.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
$$$115_DRIVER	0000033E (830.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC LONG

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	92	00:00:00.23	00:00:01.09
Command processing	99	00:00:00.56	00:00:01.97
Pass 1	561	00:00:25.63	00:00:40.97
Symbol table sort	0	00:00:03.74	00:00:04.66
Pass 2	148	00:00:04.53	00:00:08.20
Symbol table output	18	00:00:00.17	00:00:00.63
Psect synopsis output	1	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	921	00:00:34.89	00:00:57.54

The working set limit was 1950 pages.
121125 bytes (237 pages) of virtual memory were used to buffer the intermediate code.
There were 120 pages of symbol table space allocated to hold 2253 non-local and 40 local symbols.
765 source lines were read in Pass 1, producing 15 object records in Pass 2.
48 pages of virtual memory were used to define 45 macros.

+----------------------------+
! Macro library statistics !
+----------------------------+

Macro library name	Macros defined
_255DUA18:[SYS.OBJ]LIB.MLB;1	17
_255DUA18:[SYSLIB]STARLET.MLB;3	8
TOTALS (all libraries)	25

2589 GETS were required to define 25 macros.

There were no errors, warnings or information messages.

```
MACRO/LIS=LIS$:TTYCHARI/OBJ=OBJ$:TTYCHARI MSRC$:TTYCHARI/UPDATE=(BUG$:TTYCHARI)+EXECML$/LIB
```

0448 AH-EF71A-SE
VAX/VMS V4.1 SRC LST MCRF UPD

CSP
LIS

TTYCHARI
LIS

TTDRVR

OPDRVWS
LIS

TTDRIVER
MAP

YCDRIVER
MAP

CSPQUORUM
LIS

TTYCHARO
LIS