```
BBBBBBBBBBBB        000000000        000000000     TTTTTTTTTTTTTTT     SSSSSSSSSSSS
BBBBBBBBBBBB        000000000        000000000     TTTTTTTTTTTTTTT     SSSSSSSSSSSS
BBBBBBBBBBBB        000000000        000000000     TTTTTTTTTTTTTTT     SSSSSSSSSSSS
BBB         BBB  000         000  000         000        TTT        SSS
BBB         BBB  000         000  000         000        TTT        SSS
BBB         BBB  000         000  000         000        TTT        SSS
BBB         BBB  000         000  000         000        TTT        SSS
BBB         BBB  000         000  000         000        TTT        SSS
BBB         BBB  000         000  000         000        TTT        SSS
BBBBBBBBBBBB     000         000  000         000        TTT           SSSSSSSSS
BBBBBBBBBBBB     000         000  000         000        TTT           SSSSSSSSS
BBBBBBBBBBBB     000         000  000         000        TTT           SSSSSSSSS
BBB         BBB  000         000  000         C00        TTT                    SSS
BBB         BBB  000         000  000         000        TTT                    SSS
BBB         BBB  000         000  000         000        TTT                    SSS
BBB         BBB  000         000  000         000        TTT                    SSS
BBB         BBB  000         000  000         000        TTT                    SSS
BBB         BBB  000         000  000         000        TTT                    SSS
BBBBBBBBBBBB        000000000        000000000           TTT        SSSSSSSSSSSS
BBBBBBBBBBBB        000000000        000000000           TTT        SSSSSSSSSSSS
BBBBBBBBBBBB        000000000        000000000           TTT        SSSSSSSSSSSS
```

```
**FILE**ID**READPRMPT                                        N 9




RRRRRRRR    EEEEEEEEEE    AAAAAA    DDDDDDDD    PPPPPPPP    RRRRRRRR    MM      MM  PPPPPPPP    TTTTTTTTTT
RRRRRRRR    EEEEEEEEEE    AAAAAA    DDDDDDDD    PPPPPPPP    RRRRRRRR    MM      MM  PPPPPPPP    TTTTTTTTTT
RR      RR  EE          AA      AA  DD      DD  PP      PP  RR      RR  MMMM  MMMM  PP      PP      TT
RR      RR  EE          AA      AA  DD      DD  PP      PP  RR      RR  MMMM  MMMM  PP      PP      TT
RR      RR  EE          AA      AA  DD      DD  PP      PP  RR      RR  MM  MM  MM  PP      PP      TT
RR      RR  EE          AA      AA  DD      DD  PP      PP  RR      RR  MM  MM  MM  PP      PP      TT
RRRRRRRR    EEEEEEEE    AA      AA  DD      DD  PPPPPPPP    RRRRRRRR    MM      MM  PPPPPPPP        TT
RRRRRRRR    EEEEEEEE    AA      AA  DD      DD  PPPPPPPP    RRRRRRRR    MM      MM  PPPPPPPP        TT
RR  RR      EE          AAAAAAAAAA  DD      DD  PP          RR  RR      MM      MM  PP              TT
RR  RR      EE          AAAAAAAAAA  DD      DD  PP          RR  RR      MM      MM  PP              TT
RR    RR    EE          AA      AA  DD      DD  PP          RR    RR    MM      MM  PP              TT          ....
RR    RR    EE          AA      AA  DD      DD  PP          RR    RR    MM      MM  PP              TT          ....
RR      RR  EEEEEEEEEE  AA      AA  DDDDDDDD    PP          RR      RR  MM      MM  PP              TT          ....
RR      RR  EEEEEEEEEE  AA      AA  DDDDDDDD    PP          RR      RR  MM      MM  PP              TT          ....


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
```

```
0000            .TITLE  READPRMPT - READ AND PROMPT ROUTINE
0000            .IDENT  'V04-000'
0000
0000    ;
0000    ;****************************************************************************
0000    ;*                                                                          *
0000    ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0000    ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
0000    ;*  ALL RIGHTS RESERVED.                                                    *
0000    ;*                                                                          *
0000    ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
0000    ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000    ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
0000    ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
0000    ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
0000    ;*  TRANSFERRED.                                                            *
0000    ;*                                                                          *
0000    ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
0000    ,*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000    ;*  CORPORATION.                                                            *
0000    ;*                                                                          *
0000    ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
0000    ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
0000    ;*                                                                          *
0000    ;*                                                                          *
0000    ;****************************************************************************
0000    ;
0000
0000    ;++
0000    ; FACILITY:
0000    ;
0000    ; ABSTRACT:
0000    ;       This module contains a routine (BOO$READPROMPT) which writes a
0000    ;       prompt line and reads a line of input from the console terminal
0000    ;       using QIOs.  Either writing the prompt line or reading the input line
0000    ;       may be bypassed.
0000    ;
0000    ; ENVIRONMENT:  User mode
0000    ;
0000    ; AUTHOR:  STEVE BECKHARDT,     CREATION DATE:  27-Sep-1979
0000    ;
0000    ; MODIFIED BY:
0000    ;
0000    ;       V03-002 KDM0090         Kathleen D. Morse       01-Dec-1983
0000    ;               Make psect word aligned.
0000    ;
0000    ;       V03-001 JLV0134         Jake VanNoy             31-Dec-1981
0000    ;               Add routine RIO$OUTPUT_LINE.
0000    ;
0000    ;--
```

```
                                 0000             .SBTTL  DECLARATIONS
                                 0000    ;
                                 0000    ; INCLUDE FILES:
                                 0000    ;
                                 0000
                                 0000    ;
                                 0000    ; MACROS:
                                 0000    ;
                                 0000
                                 0000    ;
                                 0000    ; EQUATED SYMBOLS:
                                 0030    ;
                                 0000
                                 0000    ;
                                 0000    ; OWN STORAGE:
                                 0000    ;
                                 0000
                             00000000             .PSECT  BOO$SYSGEN,WRT,WORD
                                 0000
                                 0000    IOSTBLK:                                  ; I/O status block
                      00000008   0000             .BLKQ   1
                                 0008
                                 0008    CHANNEL:                                  ; Channel
                          0000   0008             .WORD   0
                                 000A
                                 000A    DEVNAM_DSC:                               ; Device name descriptor
 30 41 50 4F 5F 00000012'010E0000' 000A           .ASCID  /_OPA0/
                                 0017
                          0000   0017    RIO$GW_OUTLEN:: .WORD   0
                                 0019    RIO$AB_OUTBUF::
                      00000100   0019             .LONG   256                      ; Descriptor
                      00000021'  001D             .LONG   RIO$AB_BUFFER            ; Buffer pointer
                                 0021    RIO$AB_BUFFER::
                      00000121   0021             .BLKB   256                      ; Buffer
                                 0121
                             00000000             .PSECT  BOO$READPROMPT,RD,NOWRT,EXE
                          0000          .SBTTL  BOO$READPROMPT - Prompt and read input string
                          0000  ;++
                          0000  ; Functional Description:
                          0000  ;       BOO$READPROMPT outputs the specified ASCIZ prompt string on the
                          0000  ;       console terminal then checks the count of characters to be read.
                          0000  ;       If zero it exits, otherwise it reads the console terminal until
                          0000  ;       either a carriage return is encountered or the character count
                          0000  ;       is satisfied.  The specified buffer is filled with an ASCIC
                          0000  ;       string containing the characters read but not including the
                          0000  ;       terminating carriage return.
                          0000  ;
                          0000  ; Calling Sequence:
                          0000  ;       CALLG   ARGLIST,BOO$READPROMPT
                          0000  ;
                          0000  ; Input Parameters:
                          0000  ;       PROMPT(AP)  -  Address of ASCIZ prompt string
              00000004    0000          PROMPT  =  4
                          0000  ;
                          0000  ;       SIZE(AP)    -  Maximum length of input string
              00000008    0000          SIZE    =  8
                          0000  ;                       Note: if size is zero, then nothing is read
                          0000  ;                             and only the prompt string is written.
                          0000  ;
                          0000  ;       BUF(AP)     -  Address of input buffer
              0000000C    0000          BUF     =  12
                          0000  ;
                          0000  ; Output Parameters:
                          0000  ;       R0 - Completion status code
                          0000  ;
                          0000  ;       Buffer located by BUF(AP) will be filled with the string
                          0000  ;       read as an ASCIC string.
                          0000  ;
                          0000  ;--
                          0000
                          0000  BOO$READPROMPT::
                  0004    0000          .WORD   ^M<R2>
                          0002
         0008'CF    B5    0002          TSTW    W^CHANNEL               ; Channel assigned yet?
              18    12    0006          BNEQ    10$                     ; Yes
                          0008          $ASSIGN_S       CHAN = W^CHANNEL,-  ; No, assign it
                          0008                          DEVNAM = DEVNAM_DSC,-
                          0008                          ACMODE = #3     ; Allow access from user mode
           73 50    E9    001D          BLBC    R0,90$                  ; Error
                          0020
04 BC   FFFF 8F  00  3A   0020  10$:    LOCC    #0,#^XFFFF,@PROMPT(AP)  ; Locate end of prompt string
      51   04 AC    C2    0027          SUBL    PROMPT(AP),R1           ; R1 = size of prompt string
      50   08 AC    D0    002B          MOVL    SIZE(AP),R0             ; R0 = size of input buffer
              38    13    002F          BEQL    20$                     ; No input buffer
      52   0C AC    D0    0031          MOVL    BUF(AP),R2              ; R2 = address of input buffer
                          0035
                          0035          $QIOW_S CHAN = W^CHANNEL,-      ; Prompt and read
                          0035                  FUNC = #IO$_READPROMPT,-
                          0035                  IOSB = W^IOSTBLK,-
                          0035                  P1 = 1(R2),-            ; Address of input buffer
                          0035                  P2 = R0,-               ; Size of input buffer
                          0035                  P5 = PROMPT(AP),-       ; Address of prompt buffer
                          0035                  P6 = R1                 ; Size of prompt buffer
```

```
                36 50   E9  005A   146          BLBC    R0,90$                  ; Error
        50    0000'CF   3C  005D   147          MOVZWL  W^IOSTBLK,R0            ; Get I/O status block
        62    0002'CF   90  0062   148          MOVB    W^IOSTBLK+2,(R2)        ; Store size of input line
                   2A   11  0067   149          BRB     90$
                            0069   150
                            0069   151 20$:     $QIOW_S CHAN = W^CHANNEL,-      ; Write prompt string, no input
                            0069   152                  FUNC = #IO$_WRITEVBLK,-
                            0069   153                  IOSB = W^IOSTBLK,-
                            0069   154                  P1 = @PROMPT(AP),-      ; Address of prompt buffer
                            0069   155                  P2 = R1                 ; Size of prompt buffer
                05 50   E9  008B   156          BLBC    R0,90$                  ; Error
        50    0000'CF   3C  008E   157          MOVZWL  W^IOSTBLK,R0            ; Get I/O status block
                            0093   158
                        04  0093   159 90$:     RET
                            0094   160
```

```
                              0094  .SBTTL RIO$OUTPUT_LINE - Output one line
                              0094
                              0094  ;+
                              0094  ; This routine is in RMSCONIO for SYSGEN, is used here to map STASYSGEN
                              0094  ; calls to this routine into calls to BOO$READPROMPT.
                              0094  ;
                              0094  ; Inputs:
                              0094  ;       RIO$GW_OUTLEN - length of string to output
                              0094  ;       RIO$AB_BUFFER - buffer to output
                              0094  ;-
                              0094
                              0094  RIO$OUTPUT_LINE::
                              0094
            7E     51     7D  0094          MOVQ    R1,-(SP)                ; Save R1,R2
   51   00000017'EF       3C  0097          MOVZWL  RIO$GW_OUTLEN,R1        ; Set length
   52   00000021'EF       9E  009E          MOVAB   RIO$AB_BUFFER,R2        ; Set address
             51   6241    9E  00A5          MOVAB   (R2)[R1],R1             ; Set address of end of string
   61   00000A0D 8F       D0  00A9          MOVL    #^X00000A0D,(R1)        ; Set CR, LF, zero byte at end
                              00B0
                   7E     7C  00B0          CLRQ    -(SP)                   ; Null read buffer
                   52     DD  00B2          PUSHL   R2                      ; Address of string
   FFFFFF45 EF     03     FB  00B4          CALLS   #3,L^BOO$READPROMPT     ; Output string
                              00BB
            51     8E     7D  00BB          MOVQ    (SP)+,R1                ; Restore R1,R2
                          05  00BE          RSB                             ; Return
                              00BF
                              00BF          .END
```

```
$$T1              = 00000001
BOO$READPROMPT      00000000 RG     02
BUF               = 0000000C
CHANNEL             00000008 R      01
DEVNAM_DSC          0000000A R      01
IO$_READPROMPT      ********    X   02
IO$_WRITEVBLK       ********    X   02
IOSTBLK             00000000 R      01
PROMPT            = 00000004
RIO$AB_BUFFER       00000021 RG     01
RIO$AB_OUTBUF       00000019 RG     01
RIO$GW_OUTLEN       00000017 RG     01
RIO$OUTPUT_LINE     00000094 RG     02
SIZE              = 00000008
SYS$ASSIGN          ********   GX   02
SYS$QIOW            ********   GX   02


                                          +----------------+
                                          ! Psect synopsis !
                                          +----------------+

PSECT name                       Allocation             PSECT No.  Attributes
----------                       ----------             ---------  ----------
.  ABS  .                        00000000  (    0.)     00 (  0.)  NOPIC   USR   CON   ABS   LCL NOSHR NOEXE NORD  NOWRT NOVEC BYTE
BOO$SYSGEN                       00000121  (  289.)     01 (  1.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD    WRT NOVEC WORD
BOO$READPROMPT                   000000BF  (  191.)     02 (  2.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD  NOWRT NOVEC BYTE


                                        +------------------------+
                                        ! Performance indicators !
                                        +------------------------+

Phase                      Page faults    CPU Time       Elapsed Time
-----                      -----------    --------       ------------
Initialization                      30    00:00:00.09    00:00:00.79
Command processing                 128    00:00:00.65    00:00:02.88
Pass 1                             129    00:00:01.08    00:00:03.45
Symbol table sort                    0    00:00:00.01    00:00:00.01
Pass 2                              48    00:00:00.45    00:00:01.08
Symbol table output                  3    00:00:00.02    00:00:00.02
Psect synopsis output                2    00:00:00.01    00:00:00.02
Cross-reference output               0    00:00:00.00    00:00:00.00
Assembler run totals               342    00:00:02.31    00:00:08.25

The working set limit was 900 pages.
4406 bytes (9 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 16 non-local and 3 local symbols.
188 source lines were read in Pass 1, producing 13 object records in Pass 2.
6 pages of virtual memory were used to define 6 macros.
```

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

Macro library name                  Macros defined
------------------                  --------------
_$255$DUA28:[BOOTS.OBJ]BOOTS.MLB;1               0
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                   0
_$255$DUA28:[SYSLIB]STARLET.MLB;2                6
TOTALS (all libraries)                           6

70 GETS were required to define 6 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:READPRMPT/OBJ=OBJ$:READPRMPT MSRC$:READPRMPT/UPDATE=(ENH$:READPRMPT)+EXECML$/LIB+LIB$:BOOTS.MLB/LIB
```

0039 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

READLBN
LIS

RTFILREAD
LIS

SHARE
LIS

QVSS
LIS

RXBTDRIVR
LIS

MBBTDRIVR
LIS

SCSLOADER
LIS

RMSCONIO
LIS

READDRIV
LIS

PABTDRIVR
LIS

PUBTDRIVR
LIS

PUTERROR
LIS

READPRMPT
LIS