THE UNIVERSITY OF MICHIGAN

Memorandum 14

DEXEDITOR

enneth
K. Burkhalter

CONCOMP: Research in Conversational Use of Computers
F.H. Westervelt, Director
ORA Project 07449

supported by

DEPARTMENT OF DEFENSE
ADVANCED RESEARCH PROJECTS AGENCY
WASHINGTON, D.C.

CONTRACT NO. DA-49-083 OSA-3050
ARPA ORDER NO. 716

administered through:

OFFICE OF RESEARCH ADMINISTRATION ANN ARBOR

February 1968

engn

UMRC858

TABLE OF CONTENTS

DEXEDITOR

K. Burkhalter

I. INTRODUCTION

DEXEDIT is the object module of a program that converts free-form DEXEMBLER* statements to fixed-format form, thereby permitting the user to write PDP assembly language statements in free form. Although the DEXEMBLER compiler will accept free-form input, the DEXEDIT formatting routines will make the source listings more readable. Several additional DEXEDIT features allow more convenient comments to be inserted into the source listings.

The DEXEDITOR has a built-in command language interpreter which recognizes $ commands imbedded in the input stream. These commands allow listings to be turned on and off, the tab columns to be set, special comment functions to be accomplished, and a PAL (DEC assembler) converter of a very simple nature to be turned on and off.

II. USAGE

The program is invoked by the appropriate RUN command specifying *DEXEDIT as the file where the object cards are to be found.

* Powers, Michael, PDP-8 Assembler, Concomp Project Technical Memorandum 12, University of Michigan, Ann Arbor, November 1967, 13 pp.

Logical I/O Units Referenced:

SCARDS: Free-form input lines to be converted. At the current time (16 January 1968) this I/O unit must be given as "5" rather than SCARDS, however, it will soon be defaulted by FORTRAN to the value assigned to SCARDS, and SCARDS may thus be given rather than 5.

SPRINT: Error comments, plus listing (in formatted form) if $LIST was specified. All trailing blanks but one are stripped off outbound lines.

SPUNCH: Fixed-format output lines for processing by DEXEMBLER compiler. All trailing blanks but one are stripped off outbound lines of text.

Examples:
```
$RUN *DEXEDIT; 5=*SOURCE* SPUNCH=-PROG
$RUN *DEXEDIT; 5=INFILE SPRINT=*SINK* SPUNCH=OUTFILE
```

III. COMMAND LANGUAGE INTERPRETER: DESCRIPTION

The DEXEDITOR contains a command language interpreter which decodes the text on all cards, within the input stream, which have a "$" in column one. Currently six (6) commands are recognized and are described below. Only the first and last letters of each command are decoded, thus LIST may be abbreviated to "LT." The default cases for the commands are: UNLIST, PALOFF, and the TABS set to columns 8, 16, and 35.

<u>LIST</u>

Causes the same output that appears at the SPUNCH device to also be directed to the SPRINT file/device, with the following exceptions: the pseudo ops EJECT and SKIP n are executed rather than being just printed. Thus the listing is nearer to the final assembly. Also fatal and nonfatal errors are flagged on the SPRINT listing.

<u>UNLIST</u>

Turns off the listing facility described. Listings may be turned on and off as often as desired.

<u>PALON</u>

Turns on a simple translator which will convert a great deal of a PAL source into a DEXEMBLER-compatible source. It handles such things as the indirect flag (I to *) and the translation of periods into asterisks. Non-memory reference instructions separated by spaces will have a "+" inserted into the space position for the DEXEMBLER. In addition, a line of the form "*300" will be converted to the appropriate DEXEMBLER pseudo op " ORG 300." A line of expressions terminated by semicolons (;) will be broken up into individual lines of text. Thus "TAD ONE; RTR; SMA" becomes

```
TAD     ONE
RTR
SMA
```

The converter is not able to handle such things as literals and defined constants, due to the symbol table requirements and the different allowable structures between the two assemblers. These lines will have to be found and corrected by hand, or they may be left to be flagged by the DEXEMBLER.

PALOFF

Turns off the PAL conversion routines. Conversion, as in listing, may be turned on and off as desired.

TABS

The tabs which determine the fixed-form output are preset to columns 1, 8, 16, and 35. Of these, the latter three may be reset to any desired value by a $TABS command. Any combination of those three values may be left out. Trailing commas resulting from the omission of items may be left off; leading and internal commas are required. After reading a TABS command the system responds with a comment designating the new values.

BOX

Used for titling pages or routines, this command draws a box of asterisks around the text remaining between the BOX command and column 72 of the source line. The line of text so determined is tabbed to the second tab field (column 16 for default tabs). Any text which would extend beyond column 69 of the output line is truncated. If the BOX command is followed by another BOX command that next line is placed under the first. The bottom of the box is closed only when no more contiguous BOX commands are found.

Commands may be concatenated on a single card separated by blanks or commas, with the exception of the BOX command which, if so done, must be the last command on the card, otherwise the following commands will be BOXED rather than executed.

Example: $TABS,, 28 LIST PALON

IV. INPUT FORMAT DESCRIPTION

To allow easy use from those terminals with facilities for both upper- and lower-case letters, all alphabetic input is automatically converted to capital letters as it is read from the SCARDS device. If PAL is on, the normal rules of PAL hold. These can be reviewed in DEC publication 8-3-S. With PAL off, the following conventions hold.

Comment Lines: A line is a comment when the character in column 1 of the source line is an asterisk (*). This type of line is transmitted directly to the output device. If the character in column 1 should be a semicolon (;) then a special comment state is entered and the remaining text is tabbed to the second tab field with an asterisk inserted in column 1. This is handy for subroutine and other labeling.

Text Lines: If an alphabetic character is found in column 1, it is presumed to be the first character of a statement label. The first blank terminates the statement label and starts a hunt for the operator field. If the first character of the input line (column 1) is a blank or comma then it is assumed there is not a statement label and a search for the operator field is begun.

The operator field is delimited by the first nonspecial character* at the left and the first blank or semicolon on the right. The semicolon serves as a comment delimiter. All text following it is tabbed to the third tab field (default column 35) and the remainder of the card is outputted directly.

If the operator field is not terminated directly with a semicolon, then a search for the expression field is begun. If a semicolon is found first then, as before, the remaining text is tabbed to the comment field. Otherwise the scan continues until the first nonspecial character is found. The expression field is terminated by a blank or semicolon. Any text remaining, if any, is taken as a comment.

If the source cards contain sequenced ID it is copied to the sink line images. If there is no ID, then the output lines are trimmed of trailing blanks before being transmitted to the SINK device.

* Special characters being those used by the editor, such as asterisk, comma, or dollar sign.

```
      IMPLICIT INTEGER*2 (A-Z)
C     *************** STATE TABLES ***************
C     ***** DEXEDIT STATE TABLE
C     ***** COLUMNS=STATES(0-7)
C     ***** ROWS=INPUTS ($ * , ; BLANK TEXT)
      INTEGER*2 STATE(8,6)/11,10,10,10,10,10,07,07,
     1                     08,10,10,03,05,05,07,07,
     2                     02,02,10,10,10,10,07,07,
     3                     09,10,10,06,06,06,06,07,
     4                     02,02,02,04,04,06,06,07,
     6                     01,01,03,03,05,05,07,07/
C     ***** PAL STATE TABLE
C     ***** COLUMNS=STATES(0-7)
C     ***** ROWS=INPUTS (* , ; / =)
      INTEGER*2 PSTATE(8,5)/05,09,09,09,02,02,02,04,
     1                      01,07,07,07,09,09,09,04,
     2                      09,09,09,06,06,06,06,04,
     3                      08,09,09,03,03,03,03,04,
     4                      09,10,10,10,09,09,09,04/
      DIMENSION LINE(80),C(80),TAB(3),COL(4)
      DIMENSION CMMDS(6)
      DIMENSION OPS(50)
      DIMENSION EJECT(5),SPACE(5),ORG(3),EQU(3)
      DATA BREAK/', '/,SCOLON/'; '/,COMMA/', '/
      DATA PERIOD/'. '/,SLASH/'/ '/,EYE/'I '/
      DATA BLANK/'  '/,ASTER/'* '/,DOLLAR/'$ '/
      DATA MINUS/'- '/,PLUS/'+ '/,EQUAL/'= '/
      DATA EJECT/'E ','J ','E ','C ','T '/
      DATA SPACE/'S ','P ','A ','C ','E '/
      DATA ORG/'O ','R ','G '/
      DATA EQU/'E ','Q ','U '/
      DATA HF040/ZF040/,H8140/Z8140/,HA940/ZA940/,H4000/Z4000/
      DATA HF940/ZF940/,HC140/ZC140/,HE940/ZE940/,H0040/Z0040/
      DATA HFF00/ZFF00/
      DATA TAB/8,16,35/
C     ***** COMMANDS: LIST,UNLIST,TABS,PALON,PALOFF,BOX
      DATA CMMDS/'LT','UT','TS','PN','PF','BX'/
      DATA OPS/6,3,'A ','N ','D ',3,'T ','A ','D ',
     1         3,'I ','S ','Z ',3,'D ','C ','A ',
     2         3,'J ','M ','S ',3,'J ','M ','P '/
      LOGICAL PSW,LIST,PAL,BOX
      LOGICAL PSEUDO,INIT,MORPAL,MEMREF
      LOGICAL LEQU
C     ***** PROGRAM INITIALIZATION
      LCNT=0
      ERROR=0
      FIELD=0
      LIST=.FALSE.
      LEQU=.FALSE.
      PSW=.FALSE.
      PAL=.FALSE.
      BOX=.FALSE.
      INIT=.TRUE.
20    READ (5,1001,END=100) C
      DO 201 I=1,80
201   IF (C(I).GE.H8140.AND.C(I).LE.HA940) C(I)=C(I)+H4000
      LCNT=LCNT+1
      I=1
      ***** NOW BLANK OUTPUT LINE IMAGE
```

```
21      DO 211 L=1,80
211     LINE(L)=BLANK
        S=0
        ICOL=0
        LASTL=1
        MEMREF=.FALSE.
        MORPAL=.FALSE.
        PSEUDO=.FALSE.
        L=1
220     IF (C(I).NE.DOLLAR) GO TO 222
        JUMP=STATE(S+1,1)
        GO TO (31,32,33,34,35,36,37,38,39,40,41), JUMP
222     IF (.NOT.BOX) GO TO 223
        PUNCH 1009
        IF (LIST) PRINT 1009
        BOX=.FALSE.
223     IF (INIT.AND.LIST) PRINT 1011
        INIT=.FALSE.
        IF (C(I).NE.ASTER) GO TO 23
        IF (.NOT.PAL) GO TO 2231
        JUMP=PSTATE(S+1,1)
        GO TO (32,35,36,37,65,66,67,68,40,70), JUMP
2231    JUMP=STATE(S+1,2)
        GO TO (31,32,33,34,35,36,37,38,39,40), JUMP
23      IF(C(I).NE.COMMA) GO TO 24
        IF (.NOT.PAL) GO TO 231
        JUMP=PSTATE(S+1,2)
        GO TO (32,35,36,37,65,66,67,68,40,70), JUMP
231     JUMP=STATE(S+1,3)
        GO TO (31,32,33,34,35,36,37,38,39,40), JUMP
24      IF(C(I).NE.SCOLON) GO TO 25
        IF (.NOT.PAL) GO TO 241
        JUMP=PSTATE(S+1,3)
        GO TO (32,35,36,37,65,66,67,68,40,70), JUMP
241     JUMP=STATE(S+1,4)
        GO TO (31,32,33,34,35,36,37,38,39,40), JUMP
25      IF(C(I).NE.BLANK) GO TO 26
        JUMP=STATE(S+1,5)
        GO TO (31,32,33,34,35,36,37,38,39,40), JUMP
26      IF (C(I).NE.EQUAL.OR..NOT.PAL) GO TO 261
        JUMP=PSTATE(S+1,5)
        GO TO (32,35,36,37,65,66,67,68,40,70), JUMP
261     IF (C(I).NE.SLASH.OR..NOT.PAL) GO TO 27
        JUMP=PSTATE(S+1,4)
        GO TO (32,35,36,37,65,66,67,68,40,70), JUMP
C       ***** IF IN PAL STATE,ASSUME TEXT BELONGS TO
C       ***** OPER FIELD UNTIL A COMMA IS FOUND.
27      IF (PAL.AND.S.EQ.0) S=2
        JUMP=STATE(S+1,6)
        GO TO (31,32,33,34,35,36,37,38,39,40), JUMP
C       ***** STATEMENT LABLE STATE
31      S=1
        IF (L.LE.8) GO TO 45
        IF (PSW) GO TO 46
        PSW=.TRUE.
        ICOL=ICOL+1
        COL(ICOL)=L
        GO TO 46
C       ***** LOOK FOR OPERATOR FIELD STATE
32      S=2
```

```
      GO TO 46
C     ***** OPERATOR FIELD STATE
33    IF (S.EQ.3) GO TO 45
C     ***** IF PAL, IS THIS A MEMORY REF INSTRUCTION
      IF (.NOT.PAL) GO TO 3300
      FINOPS=OPS(1)
      OPEND=1
      DO 3306 CNT=1,FINOPS
      K=OPEND+1
      OPEND=OPS(K)+K
      K=K+1
      KS=K
      DO 3305 K=KS,OPEND
      IF (C(I+K-KS).NE.OPS(K)) GO TO 3306
3305  CONTINUE
      MEMREF=.TRUE.
      GO TO 3304
3306  CONTINUE
3300  DO 3301 K=1,5
      IF (C(I-1+K).NE.EJECT(K)) GO TO 3302
3301  CONTINUE
      GO TO 334
3302  DO 3303 K=1,5
      IF (C(I-1+K).NE.SPACE(K)) GO TO 3304
3303  CONTINUE
      GO TO 335
3304  IF (L.GE.TAB(1)) GO TO 331
336   L=TAB(1)
      GO TO 332
331   IF (PSW) GO TO 333
      ICOL=ICOL+1
      COL(ICOL)=L-1
      PSW=.TRUE.
333   L=L+1
332   S=3
      GO TO 45
334   PRINT 1011
      PSEUDO=.TRUE.
      GO TO 336
335   NUM=0
      K=I+6
      DO 3351 K=K,72
C     ***** IF C(K) IS A NUMBER, GO TO 3352
      IF (C(K).LE.HF940.AND.C(K).GE.HF040) GO TO 3352
      IF (NUM.NE.0) GO TO 3353
3351  CONTINUE
C     ***** DEFAULT 1 SPACE
      NUM=1
      GO TO 3353
3352  NUM=(NUM*10)+((C(K)-HF040)/2**8)
      GO TO 3351
3353  DO 3354 J=1,NUM
3354  PRINT 1012
      PSEUDO=.TRUE.
      GO TO 336
C     ***** LOOK FOR EXPRESSION FIELD STATE
34    S=4
      GO TO 46
C     ***** EXPRESSION FIELD STATE
35    IF (S.EQ.5) GO TO 45
```

```
      IF (.NOT.PAL) GO TO 356
      IF (MEMREF) GO TO 355
      LINE(L)=PLUS
      L=L+1
      S=5
      GO TO 45
  355 IF (C(I).EQ.PERIOD) C(I)=ASTER
      IF (C(I).EQ.EYE.AND.C(I+1).EQ.BLANK) GO TO 354
  356 IF (L.GE.TAB(2)) GO TO 351
      L=TAB(2)
      GO TO 352
  351 ICOL=ICOL+1
      COL(ICOL)=L-1
      PSW=.TRUE.
  357 L=L+1
  352 S=5
      GO TO 45
  354 LINE(L)=ASTER
      S=4
      LASTL=L
      GO TO 451
C     ***** LOOK FOR COMMENT FIELD STATE
   36 IF ((PAL.AND.C(I).EQ.SLASH).OR.(.NOT.PAL)) GO TO 361
      S=4
      GO TO 46
  361 S=6
      GO TO 46
C     ***** COMMENT FIELD STATE
   37 IF (S.EQ.7) GO TO 45
      IF (L.GE.TAB(3)) GO TO 371
      L=TAB(3)
      GO TO 372
  371 ICOL=ICOL+1
      COL(ICOL)=L-1
      PSW=.TRUE.
  373 L=L+1
      PSW=.TRUE.
  372 S=7
   45 LINE(L)=C(I)
      IF (C(I).NE.BLANK) LASTL=L
  451 L=L+1
      IF (L.GE.73) GO TO 461
   46 I=I+1
      IF (I.LE.72) GO TO 220
  461 IF (I.EQ.72) GO TO 464
      I=I+1
      DO 462 I=I,72
      IF (C(I).NE.BLANK) GO TO 463
  462 CONTINUE
      GO TO 464
  463 PSW=.TRUE.
      ICOL=ICOL+1
      COL(ICOL)=L-1
C     ***** TEXT FINISHED.COPY ID FIELD
  464 L=73
      DO 47 I=73,80
      IF (C(I).EQ.BLANK) GO TO 47
      LINE(L)=C(I)
      LASTL=L
      L=L+1
```

```
47     CONTINUE
C      ***** TACK ONE TRAILING BLANK ONTO LINE
48     IF (LASTL.EQ.1.AND.LINE(1).EQ.BLANK) GO TO 481
       LASTL=LASTL+1
481    PUNCH 1001,(LINE(L),L=1,LASTL)
       IF (LIST.AND..NOT.PSEUDO) PRINT 1001,(LINE(L),L=1,LASTL)
       IF (.NOT.PSW.AND..NOT.MORPAL) GO TO 20
       IF (.NOT.PSW.AND.MORPAL) GO TO 21
       PRINT 1003,LCNT,(COL(K),K=1,ICOL)
       FIELD=FIELD+1
       PSW=.FALSE.
       IF (MORPAL) GO TO 21
       GO TO 20
C
C      ***** PAL SPECIAL SECTIONS
C
C      ***** CHANGE INITIAL * TO "ORG"
65     L=TAB(1)
       DO 651 J=1,3
651    LINE(L-1+J)=ORG(J)
C      ***** FAKE IT,FOR NO "+"
652    MEMREF=.TRUE.
       S=4
       LASTL=L
       GO TO 451
C      ***** STRIP MULTIPLE LINES WITHIN ONE
56     LASTL=L-1
       MORPAL=.TRUE.
       I=I+1
       GO TO 48
C      *****COMMA OR = FOUND AFTER FIRST TEXT FIELD
C      ***** THUS MOVE OPER BACK TO LABEL FIELD.
67     START=TAB(1)
       DO 671 J=START,LASTL
671    LINE(J-TAB(1)+1)=LINE(J)
       TEMP=LASTL
       LASTL=LASTL-START+1
C      ***** NOW BLANK OUT PREVIOUS OPER FIELD
       START=LASTL+1
       DO 672 J=START,TEMP
672    LINE(J)=BLANK
       L=LASTL+1
       IF (LEQU) GO TO 673
       GO TO 32
673    L=TAB(1)
       DO 674 J=1,3
674    LINE(L-1+J)=EQU(J)
       LEQU=.FALSE.
       GO TO 652
C      ***** PAL COMMENT CARD
68     C(1)=ASTER
       GO TO 38
C      ***** CHANGE = TO "EQU"
70     LEQU=.TRUE.
       GO TO 67
C
C      ***** COMMAND LANGUAGE DECODER
41     I=1
411    I=I+1
       DO 412 I=I,72
```

```
C       ***** IF C(I) IS A LETTER, GO TO 413
        IF (C(I).GE.HC140.AND.C(I).LE.HE940) GO TO 413
 412    CONTINUE
        GO TO 20
 413    CLIS=I
        DO 414 I=I,72
        IF (C(I).EQ.BLANK.OR.C(I).EQ.COMMA) GO TO 415
 414    CONTINUE
C       ***** HAVE COMMAND, NOW FIND ENTRY POINT
 415    CLIE=I-1
        CMMD=(C(CLIS)-H0040)+(C(CLIE)/2**8-HFF00-1)
        DO 416 J=1,6
        IF (CMMD.EQ.CMMDS(J)) GO TO 417
 416    CONTINUE
C       ***** INVALID COMMAND
        IF (CLIE-CLIS.GT.40) CLIE=CLIS+40
        PRINT 1010,(C(I),I=CLIS,CLIE)
        GO TO 411
 417    GO TO (421,422,423,424,425,426),J
C       ***** COMMAND...LIST
 421    LIST=.TRUE.
        GO TO 411
C       ***** COMMAND...UNLIST
 422    LIST=.FALSE.
        GO TO 411
C       ***** COMMAND...TABS
C       ***** TAB SETUP
 423    NUM=0
        T=1
        DO 4235 I=I,72
C       ***** IF C(I) ALPHABETIC, GO TO 4236
        IF (C(I).GE.HC140.AND.C(I).LE.HE940) GO TO 4236
C       ***** IF C(I) .NOT. A NUMBER, GO TO 4231
        IF (C(I).GT.HF940.OR.C(I).LT.HF040) GO TO 4231
        NUM=(NUM*10)+((C(I)-HF040)/2**8)
        GO TO 4235
 4231   IF (C(I).NE.COMMA) GO TO 4232
        IF (NUM.EQ.0) GO TO 4234
        GO TO 4233
 4232   IF (C(I).EQ.BLANK.AND.NUM.EQ.0) GO TO 4235
 4233   TAB(T)=NUM
        NUM=0
 4234   T=T+1
        IF (T.EQ.4) GO TO 4236
 4235   CONTINUE
 4236   PRINT 1007,TAB
        GO TO 411
C       ***** COMMAND...PALON
 424    PAL=.TRUE.
        GO TO 411
C       ***** COMMAND...PALOFF
 425    PAL=.FALSE.
        GO TO 411
C       ***** COMMAND...BOX
 426    IF (BOX) GO TO 4261
        IF (INIT.AND.LIST) PRINT 1011
        INIT=.FALSE.
        PUNCH 1008
        IF (LIST) PRINT 1008
        BOX=.TRUE.
```

```
4261  I=I+1
      START=TAB(2)
      DO 4262 L=START,69
      LINE(L)=C(I)
      I=I+1
      IF (I.GT.72) GO TO 4263
4262  CONTINUE
4263  LINE(1)=ASTER
      LINE(71)=ASTER
      PUNCH 1000,(LINE(L),L=1,71)
      IF (LIST) PRINT 1000,(LINE(L),L=1,71)
      GO TO 20
C
C     ***** TERMINATION MESSAGE
100   IF (BOX) PUNCH 1009
      IF (BOX.AND.LIST) PRINT 1009
      PRINT 1006,ERROR,FIELD,LCNT
      STOP
C     ***** PRINT COMMENT CARD
38    PUNCH 1001,C
      IF (LIST) PRINT 1001,C
      GO TO 20
C     ***** INDENTED (SPECIAL) COMMENT
39    LINE(1)=ASTER
      L=TAB(2)
      DO 391 I=2,80
      LINE(L)=C(I)
      IF (C(I).NE.BLANK) LASTL=L
391   L=L+1
      GO TO 48
C     ***** ERROR MESSAGE
40    IF (LIST) PRINT 1001,C
      PRINT 1002,ERROR,I
      ERROR=ERROR+1
      PUNCH 1004,ERROR,ERROR
      IF (ERROR.LE.10) GO TO 20
      PRINT 1005
      STOP
C
C     ***** FORMATS
C
1000  FORMAT(71A1)
1001  FORMAT(80A1)
1002  FORMAT('E',I3,T7,'**** ILLEGAL SYMBOL IN COLUMN',I3)
1003  FORMAT('**** FIELD OVERFLOW',4X,'LINE',
     1I5,4X,'COLUMN(S)',I3,3(',',I3))
1004  FORMAT('ERROR',I3,1X,62(1H*),T73,'ERROR',I3)
1005  FORMAT(/52H**** THIS ISN'T ASSEMBLY LANGUAGE, IT'S MADNESS ****)
1006  FORMAT(1H1/'ALL INPUT HAS BEEN PROCESSED'/
     1'FLAGS (ERROR',I4,4X,'FIELD',I4,1H),10X,
     2'LINES PROCESSED',I5/)
1007  FORMAT('TABS SET TO',I3,1H,,I3,5H, AND,I3)
1008  FORMAT(71(1H*)/1H*,T71,1H*)
1009  FORMAT(1H*,T71,1H*/71(1H*))
1010  FORMAT('**** INVALID COMMAND....',40A1)
1011  FORMAT(1H1)
1012  FORMAT(1H )
      END
```