DECUS
PROGRAM LIBRARY

DECUS NO.	8-176
TITLE	PAL CHOP
AUTHOR	Edward A. Taft III
COMPANY	St. Mark's School Southboro, Massachusetts
DATE	April 2, 1969
SOURCE LANGUAGE	Machine Language

PAL CHOP

DECUS Program Library Write-up DECUS No. 8-176

DESCRIPTION

PAL CHOP is a program for making minimum length copies of PAL source tapes. It does so by removing all comments; tabs, rubouts, blank tape, and multiple spaces and carriage return/line feeds. The final result is functionally identical to the input program but with all comments and formatting removed.

This program is especially useful while writing immensely large programs. Such a program is usually written and debugged in sections. Once a section is operating properly there is no further need for the comments and formatting ; the only reason for retaining the ASCII source is that it is needed for assembly with sections that are still being written and debugged.

A shorter source tape is not only more convenient to handle; it also takes less time to assemble. Users with a Disk/DECtape monitor and limited amounts of storage can save even more time and effort by saving their source on disk or tape; editing and assembly are immensely speeded up. PAL CHOP can be used to make a condensed copy of the program under development, which, when saved, will take much less space than it would if saved in its entirety.

Condensed source programs have the advantage over BIN object tapes in that they can still be edited easily; if such changes work, they can always be added to the original (unCHOPped) source.

PAL CHOP accepts ASCII source tapes one "page" at a time (EDIT punches CTRL/L at the end of each "page") at either low or high speed and punches the condensed tape on the low or high speed punch. A pseudo interrupt system is used to ensure that I/O proceeds at full speed.

EQUIPMENT

PDP-5, PDP-8, PDP-8/I, or PDP-8/L with an ASR-33 and the optional use of a high speed reader and punch.

LOADING

PAL CHOP is loaded with the binary loader. It will operate properly in any field.

The start address is 0200.

PAL CHOP occupies locations 10-366, and uses locations 400-1177 as a buffer.

OPERATION

After loading PAL CHOP, set 0200 into the switch register and press LOAD ADDRESS.

Set switches properly for the following options:

Bit 0: 0 = Input from the high speed reader.
1 = Input from the Teletype.

Bit 1: 0 = Output to the high speed punch.
1 = Output to the Teletype.

Bit 2: 0 = Process the input tape.
1 = Punch 6 inches of blank tape and stop.

Place PAL source tape in selected reader and activate reader and punch. Press START. The tape will be processed until a CTRL/L (the end-file character) is encountered. Three inches of blank leader will be punched at the beginning and end of the copy.

To process another tape (or another section of the same tape) press CONTINUE. An extra 6 inches of blank tape may be punched wherever desired by setting bit 2 and pressing CONTINUE.

Restart: If the program (locations 10-366) has not been disturbed it may be re-started at location 0200.

EXPLANATION

Input/Output

This program uses a pseudo interrupt input/output system. The computer interrupt is not used. This system ensures that the output buffer will not overflow, output will take place at maximum speed, and input will be smooth (reader either stopped or running at full speed; no stall between characters).

A three-page output buffer is maintained. As the input tape is processed the results are placed in the buffer sequentially until it has been filled. At this point processing is suspended until the output routine has emptied the first two pages of the buffer. The remaining page is then shifted to the beginning of the buffer and processing is resumed. The reader runs at full speed during processing. When the high speed reader is being used for input, processing proceeds at several times the output speed and the buffer is filled quickly.

Internal Control

There are four control subroutines, as follows:

CHECK. Output status check and buffer control. If punch flag is up, check the number of characters in the buffer; if there are any, test the value of pointer OUTXR1 to see if the first two pages are empty. If so, shift the last page to the first page and reset the pointers. Finally, punch the next character in the buffer.

PUNCH. Character output routine. Place the contents of CHAR into the output buffer. If this character fills the buffer, continue CHECKing until the first 2 pages of the buffer have been output and the last page has been reset.

READ. Character input routine. Continue CHECKing until the reader flag is raised, then read the character into location CHAR.

SORT. Sort the contents of CHAR against a list, as follows:

```
      SORT
       LIST-1
      (return here if no match)

     .........

LIST, character
      pointer
      character
      pointer
      ....
      negative number to end list
```

SORT compares CHAR successively to each character in the list. If a match is found, control is transferred to the location specified by the pointer after the character. If no match is found, control is returned to the calling sequence.

The rest of the program is straightforward. Further information may be gained from the listing which is available for a $5.00 handling charge.

PROBLEMS AND RESTRICTIONS

Because PAL CHOP eliminates all tabs, users who replace separating spaces with tabs in their programs will find that the output from PAL CHOP will not work. Whereas an instruction such as TAD(space)ADDRSS will be copied properly, the instruction TAD(tab)ADDRSS will be rendered improperly as TADADDRSS. The places in which this will cause problems are as follows:

1. Between an operator and its operand.

2. Between combined microinstructions.

3. Between the word TEXT and the delimiting character marking the start of the text field.

Spaces must be present in the above three cases. In all other situations, spaces and tabs are optional and the multiple spaces and tabs are removed by PAL CHOP.

There is a minor bug in the routine that identifies and handles the word TEXT. As a result, comments immediately following the words T, TE, or TEX will not be deleted. This is of no consequence, and, in the unlikely event that such a situation should appear, the resulting tape will still assemble properly.

The input tape must be in correct PAL format, and must also be in the format in which EDIT outputs tapes. Nonstandard items, such as spurious CTRL/L characters and CR's and LF's not in CR-LF combinations, will cause trouble.

```
/
/
/PAL CHOP        4/69
/
/BY EDWARD A. TAFT, III
/       ST. MARK'S SCHOOL
/       SOUTHBOROUGH, MASS. 01772
/
/FOR MAKING MINIMUM LENGTH COPIES OF PAL SOURCE TAPES
/
/FOR ANY PDP-8 FAMILY COMPUTER
/HIGH-SPEED READER AND PUNCH OPTIONAL
/
/REMOVES ALL BLANK TAPE, COMMENTS, TABS, MULTIPLE
/SPACES, AND MULTIPLE CR-LF'S
/
/CORE USAGE
/       PROGRAM          10-377
/       OUTPUT BUFFER    400-1177
/
/START ADDRESS: 0200
/
/SWITCH REGISTER OPTIONS:
/BIT 0: 0=HIGH SPEED READER INPUT
/       1=TELETYPE INPUT
/BIT 1: 0=HIGH SPEED PUNCH OUTPUT
/       1=TELETYPE OUTPUT
/BIT 2: 0=PROCESS INPUT TAPE
/       1=PUNCH 6 INCHES OF LEADER
/
/TO START ANOTHER RUN, RESET SWITCHES IF
/NECESSARY AND PRESS CONTINUE
/
```

```
            *10
            /AUTOINDEX REGISTERS
0010  0000  OUTXR1, 0          /OUTPUT BUFFER POINTERS
0011  0000  OUTXR2, 0
0012  0000  SORTXR, 0          /USED BY SORT
0013  0000  XR1,    0          /MISC
            *20
            /CONSTANTS
0020  6601  PCHECK, -OUTBUF-577
0021  7001  BCHECK, -OUTBUF-377
0022  0377  BUFBEG, OUTBUF-1
0023  7742  LEADC,  -36
            /VARIABLES
0024  0000  CHAR,   0          /CURRENT INPUT CHARACTER
0025  0000  LASTC,  0          /LAST CHARACTER PUNCHED
0026  0000  TEMP,   0          /TEMPORARY REGISTER
0027  0000  COUNT,  0          / - (# OF CHARACTERS IN OUTPUT BUFFER)
            /
```

```
                READ=JMS .
0075    0000    XREAD,  0                       /READ A CHARACTER INTO CHAR - "READ"
0076    4030            CHECK
0077    6011    INSKP,  KSF
0100    5076            JMP .-2
0101    6016    INPUT,  KRB RFC
0102    3024            DCA CHAR
0103    4106            SORT                    /IGNORE BLANK, 200, TAB, RUBOUT
0104    0342             RLIST-1
0105    5475            JMP I XREAD
                /
                SORT=JMS .
0106    0000    XSORT,  0                       /SORT CHAR AGAINST A LIST - "SORT"
0107    1506            TAD I XSORT             /GET LIST ADDRESS
0110    2106            ISZ XSORT
0111    3012            DCA SORTXR
0112    1412            TAD I SORTXR            /COMPARISON CHARACTER
0113    7510            SPA
0114    5125            JMP NOTF                /LIST ENDED BY NEGATIVE #
0115    7041            CIA
0116    1024            TAD CHAR
0117    7650            SNA CLA
0120    5123            JMP .+3
0121    2012            ISZ SORTXR              /NOT A MATCH, TRY NEXT
0122    5112            JMP XSORT+4
0123    1412            TAD I SORTXR            /MATCH, GET POINTER
0124    3106            DCA XSORT
0125    7300    NOTF,   CLA CLL
0126    5506            JMP I XSORT
                LEADER=JMS .
                /
0127    0000    XLEAD,  0                       /PUNCH 3 INCHES BLANK TAPE - "LEADER
0130    3024            DCA CHAR
0131    1023            TAD LEADC               /-30(10)
0132    3026            DCA TEMP
0133    4062            PUNCH
0134    2026            ISZ TEMP
0135    5133            JMP .-2
0136    5527            JMP I XLEAD
                *171
                /PUNCH BLANK TAPE, COMPLETE OUTPUT, AND HALT
0171    4062    ENDFIL, PUNCH                   /END OF INPUT
0172    4127            LEADER
0173    4030            CHECK
0174    1027            TAD COUNT               /WAIT FOR OUTPUT TO END
0175    7640            SZA CLA
0176    5173            JMP .-3
0177    7402            HLT
                /
```

```
                /CONTROL SUBROUTINES
                /
                CHECK=JMS .
0030  0000  XCHECK, 0                    /CHECK OUTPUT STATUS - "CHECK"
0031  6021  OUTSKP, PSF
0032  5430         JMP I .-2             /PUNCH NOT READY
0033  1011         TAD OUTXR2            /GET # OF CHARACTERS
0034  7141         CLL CIA               /LEFT IN OUTPUT BUFFER
0035  1010         TAD OUTXR1
0036  3027         DCA COUNT
0037  7430         SZL
0040  5430         JMP I XCHECK          /BUFFER EMPTY
0041  1010         TAD OUTXR1            /NOT EMPTY: ARE 1ST TWO
0042  1021         TAD PCHECK            /PAGES EMPTY
0043  7710         SPA CLA
0044  5056         JMP OUT               /NO, PUNCH NEXT CHARACTER
0045  1022         TAD BUFBEG            /YES, SHIFT CONTENTS TO BEGINNING
0046  3011         DCA OUTXR2            /OF OUTPUT BUFFER
0047  1410         TAD I OUTXR1
0050  3411         DCA I OUTXR2
0051  2027         ISZ COUNT
0052  5047         JMP .-3
0053  1022         TAD BUFBEG            /RESET POINTER
0054  3010         DCA OUTXR1
0055  5033         JMP OUTSKP+2
0056  1410  OUT,    TAD I OUTXR1          /PUNCH NEXT BUFFER CHARACTER
0057  6026  OUTPUT, PLS
0060  7300         CLA CLL
0061  5430         JMP I XCHECK
                /
                PUNCH=JMS .
0062  0000  XPUNCH, 0                    /OUTPUT CHAR - "PUNCH"
0063  1024         TAD CHAR              /APPEND IT TO BUFFER
0064  3025         DCA LASTC
0065  1024         TAD CHAR
0066  3411         DCA I OUTXR2
0067  4030         CHECK                 /END OF SPACE IN BUFFER?
0070  1011         TAD OUTXR2
0071  1020         TAD PCHECK
0072  7710         SPA CLA
0073  5462         JMP I XPUNCH          /NO
0074  5067         JMP .-5               /YES, LET PUNCH CATCH UP
                /
```

```
0200  6002  START,  IOF              /INITIALIZE I/O
0201  6046           TLS
0202  6026           PLS
0203  6032           KCC
0204  6014           RFC
0205  7604           LAS              /GET OPTION BITS
0206  7104           CLL RAL
0207  7710           SPA CLA          /TEST BIT 1 (OUTPUT)
0210  1334           TAD C20
0211  1332           TAD CPSF
0212  3031           DCA OUTSKP       /6041 (TSF) OR 6021 (PSF)
0213  1031           TAD OUTSKP
0214  1335           TAD C5
0215  3057           DCA OUTPUT       /6046 (TLS) OR 6026 (PLS)
0216  7430           SZL              /TEST BIT 0 (INPUT)
0217  1334           TAD C20
0220  1333           TAD CRSF
0221  3077           DCA INSKP        /6031 (KSF) OR 6011 (RSF)
0222  1077           TAD INSKP
0223  1335           TAD C5
0224  3101           DCA INPUT        /6036 (KRB) OR 6016 (RRB RFC)
0225  1022           TAD BUFBEG       /INITIALIZE OUTPUT BUFFER
0226  3010           DCA OUTXR1
0227  1022           TAD BUFBEG
0230  3011           DCA OUTXR2
0231  2027           ISZ COUNT
0232  4127           LEADER
0233  7604           LAS              /TEST OPTION BIT 2
0234  7106           CLL RTL
0235  7710           SPA CLA
0236  5171           JMP ENDFIL       /=1; PUNCH LEADER AND QUIT
                /
```

```
                    /NORMAL INPUT AND TEST
0237    4075    NEXTC,  READ    /READ TAPE
0240    4106            SORT    /LOOK FOR /, T, CTRL/L, SPACE, CR
0241    0353             CLIST-1
0242    4062            PUNCH   /NONE OF THOSE, CONTINUE
0243    5237            JMP NEXTC
                    /SLASH FOUND, SKIP ENSUING COMMENT
0244    4075    COMMNT, READ
0245    1024            TAD CHAR
0246    1353            TAD MCR
0247    7640            SZA CLA
0250    5244            JMP COMMNT
0251    5314            JMP CARRTN      /FOUND CR, RESUME NORMAL SEARCH
                    /T FOUND, TEST FOR "TEXT"
0252    1025    TXLOOK, TAD LASTC
0253    3026            DCA TEMP
0254    4062            PUNCH
0255    1026            TAD TEMP        /MAKE SURE PRECEDING CHARACTER IS
0256    3024            DCA CHAR        /COMMA, SPACE, LF, OR BLANK
0257    4106            SORT
0260    0321             TLIST-1
0261    5237            JMP NEXTC       /NONE OF THOSE, NOT LEGAL "TEXT"
0262    1336    TXT2,   TAD TXPTR       /OK, SET UP POINTER TO TEST
0263    3013            DCA XR1         /FOR "EXT "
0264    4075    TNEXT,  READ
0265    4062            PUNCH
0266    1413            TAD I XR1
0267    7450            SNA
0270    5275            JMP TFOUND      /LIST ENDS WITH 0
0271    1024            TAD CHAR
0272    7640            SZA CLA
0273    5237            JMP NEXTC       /NOT THE WORD "TEXT"
0274    5264            JMP TNEXT       /OK SO FAR
0275    1024    TFOUND, TAD CHAR        /GET DELIMITING CHARACTER
0276    7041            CIA
0277    3026            DCA TEMP
0300    4075    LITRAL, READ            /NOW COPY INPUT LITERALLY UNTIL
0301    4062            PUNCH           /2ND DELIMITER
0302    1026            TAD TEMP
0303    1024            TAD CHAR
0304    7640            SZA CLA
0305    5300            JMP LITRAL
0306    5244            JMP COMMNT      /IGNORE INPUT AFTER TEXT FIELD
                /
```

```
                    /FOUND SPACE, ELIMINATE ANY MULTIPLE SPACES
0307  1025  SPACE,   TAD LASTC        /WAS PREVIOUS CHAR A SPACE?
0310  1342           TAD MSPC
0311  7640           SZA CLA
0312  5242           JMP NEXTC+3      /NO, PRINT SPACE
0313  5237           JMP NEXTC        /YES, SKIP SPACE
                    /FOUND CR, ELIMINATE MULTIPLE CR-LF'S
0314  1025  CARRTN,  TAD LASTC        /WAS PREVIOUS CHAR A LF?
0315  1366           TAD MLF
0316  7640           SZA CLA
0317  5242           JMP NEXTC+3      /NO, PRINT CR-LF
0320  4075           READ
0321  5237           JMP NEXTC        /YES, SKIP CR-LF
                    /

                    /SORTING LISTS
                    /
                    /CHARACTERS LEGALLY OCCURRING BEFORE "TEXT"
0322  0000  TLIST,   000      /BLANK
0323  0262            TXT2
0324  0254           254      /COMMA
0325  0262            TXT2
0326  0240           240      /SPACE
0327  0262            TXT2
0330  0212           212      /LINE FEED
0331  0262            TXT2
                    /CONSTANTS FOR I/O INITIALIZATION
0332  6021  CPSF,    PSF
0333  6011  CRSF,    RSF
0334  0020  C20,     20
0335  0005  C5,      5
                    /TEST CONSTANTS FOR "TEXT"
0336  0336  TXPTR,   .
0337  7473  TXLIST,  -305     /E
0340  7450           -330     /X
0341  7454           -324     /T
0342  7540  MSPC,    -240     /SPACE
                    /INPUT CHARACTERS TO BE IGNORED
0343  0000  RLIST,   000      /BLANK
0344  0076            XREAD+1
0345  0200           200      /LEADER
0346  0076            XREAD+1
0347  0211           211      /TAB
0350  0076            XREAD+1
0351  0377           377      /RUBOUT
0352  0076            XREAD+1
0353  7563  MCR,     -215
                    /NORMAL SEARCH CHARACTERS
0354  0257  CLIST,   257      /SLASH
0355  0244            COMMNT
0356  0324           324      /T
0357  0252            TXLOOK
0360  0214           214      /CTRL/L
0361  0171            ENDFIL
0362  0240           240      /SPACE
0363  0307            SPACE
0364  0215           215      /CARRIAGE RETURN
0365  0314            CARRTN
0366  7566  MLF,     -212
                    /
                    OUTBUF=400       /3-PAGE OUTPUT BUFFER
```

```
PCHECK 0021
BUFREG 0022
CARRTN 0314
CHAR   0024
CHECK  4030
CLIST  0354
COMMNT 0244
COUNT  0027
CPSF   0332
CRSF   0333
C20    0334
C5     0335
ENDFIL 0171
INPUT  0101
INSKP  0077
LASTC  0025
LEADC  0023
LEADER 4127
LITRAL 0300
MCR    0353
MLF    0366
MSPC   0342
NEXTC  0237
NOTF   0125
OUT    0056
OUTBUF 0400
OUTPUT 0057
OUTSKP 0031
OUTXR1 0010
OUTXR2 0011
PCHECK 0020
PUNCH  4062
READ   4075
RLIST  0343
SORT   4106
SORTXR 0012
SPACE  0307
START  0200
TEMP   0026
TFOUND 0275
TLIST  0322
TNEXT  0264
TXLIST 0337
TXLOOK 0252
TXPTR  0336
TXT2   0262
XCHECK 0030
XLEAD  0127
XPUNCH 0062
XREAD  0075
XR1    0013
XSORT  0106
```

VIII. EXAMPLE OF USE

PAL CHOP was used with the high speed reader and the teleprinter to produce the following CHOPped copy of the first two sections of PAL CHOP ASCII: