```
/ .MACRO SRC FOR CS2: REDUCED VERSION
/ KIT WHITFIELD  29/11/71
/
/
/ STREAMS ARE INPUT/OUTPUT(1:4) RATHER THAN (0:3)
/ LINE-BY-LINE OPTION NOT AVAILABLE
/ .TRACE MODIFIED TO BE CONDITIONAL ON ASSEMBLY PARAMETER
/ RATHER THAN THE ACC. SWITCHES
/ SELECT INPUT/OUTPUT => SELIN/SELOUT, STOP = .EXIT
/
        ** INTERNAL **
        .DEFIN %M2,NAME,EXT  / GENERATE 3-WORD DIRECTORY ENTRY BLOCK
%X=.;   .SIXBT NAME
        0; .LOC %X+2; .SIXBT EXT'SRC'
        .LOC %X+3
        .ENDM
/
/
/ DEFINE I/O STREAM BASES AND VALIDITY CHECK
INPUT=0;OUTPUT=4
        .DEFIN %T1,STREAM;      .IFNZR -OUTPUT-1&STREAM /IF \OUTPUT&STREAM
** FAULTY PARAM **; .ENDC
        .ENDM
/
/
/       ** PART1 NOT INCLUDED **
/
/       ** PART 2 **
        .DEFIN EVENT,NUMBER,TEXT,?TAG
        .GLOBL %EV; LAC (NUMBER); JMS* %EV; JMP TAG
        .ASCII '@TEXT@'<15>
TAG=.
        .ENDM
        .DEFIN ON,EVENT,ACTION,?T1,?T2
        .GLOBL %EV,%ON
        LAC (EVENT); JMS* %ON; JMP T2+1
        0
T1      0
        ACTION
        DZM T1-1        / UNSET BUSY MARKER
T2      JMP* T1
        .ENDM
/
/       ** PART 3 NOT INCLUDED **
/
/       ** PART 4 **
        .DEFIN ATTACH,STREAM,NO,NAME,EXT
        .GLOBL %AT; %T1 STREAM
```

```
%X=NO-1;%X=OUTPUT-1&%X+STREAM    / INPUT/OUTPUT(1:4) NOT (0:3)
        LAC (%X);%X=%X+1; .IODEV %X
        JMS* %AT; .DSA (.+2); JMP .+4; %M2 NAME,EXT
        .ENDM

        .DEFIN RSYM
        .GLOBL %AT,%RS; JMS* %RS
        .ENDM

        .DEFIN PSYM
        .GLOBL %AT,%PS; JMS* %PS
        .ENDM

        .DEFIN SELIN
        .GLOBL %AT,%SL,%IO
        TAD (-1); JMS* %SL; .DSA %IO     / STREAMS 1:4 NOT 0:3
        .ENDM

        .DEFIN SELOUT
        .GLOBL %AT,%SL,%OO
        TAD (-1); JMS* %SL; XCT %OO      / STREAMS 1:4 NOT 0:3
        .ENDM

        .DEFIN CLOSE,STREAM
        .GLOBL %AT,%CL; %T1 STREAM
        .IFZER STREAM; .GLOBL %IO; LAC %IO; .ENDC
        .IFNZR STREAM; .GLOBL %OO; LAC %OO; .ENDC
        JMS* %CL
        .ENDM


        ** PART 5 (MODIFIED) **
        .IFUND NOTRACE
        .DEFIN .TRACE
        .GLOBL %I9; JMS* %I9
        .ENDM
        .ENDC
        .IFDEF NOTRACE
        .DEFIN .TRACE
        .GLOBL %I9; NOP
        .ENDM
        .ENDC


        .DEFIN STOP
        CLOSE OUTPUT
        .EXIT
        .ENDM
```

```
/       PREDEFINED MACROS FOR KM9-15
/       EXTENDED MONITOR PACKAGE(EMOP)
/
/       KIT WHITFIELD     31/10/71
/
/  TO SAVE SYMBOL TABLE SPACE, EITHER DEC-STYLE LINE BY LINE I/O
/  OR CHARACTER BY CHARACTER I/O COMMANDS ARE AVAILABLE BUT NOT
/  BOTH.  THIS SELECTION IS MADE BY SETTING '%X' TO ZERO
/  AND NON-ZERO RESPECTIVELY; THE DEFAULT VALUE IS
/  ZERO SET BY THE '.IFUND' LINE BELOW.
/  NOTE THAT THE STREAM SELECTION ALSO DEPENDS ON THIS.  CHARACTER
/  COMMANDS GO WITH INDEPENDENTLY NUMBERED INPUT AND OUTPUT STREAMS; LINE-
/  BY-LINE COMMANDS DO NOT.
/
/  THE SEPARATELY NUMBERED I/O STREAMS ARE INPUT(0:3) AND OUTPUT(0:3)
/  MAPPED ONTO .DAT 1-4 AND .DAT 5-10 RESPECTIVELY.
/
/       WHERE THE THE GLOBAL REFERENCE IS ONE
/       POSSIBLE ENTRY POINT OF A MULTIPLE
/       ENTRY POINT MODULE, THE NAME OF THE
/       MODULE IS GIVEN FIRST FOLLOWED BY THE
/       NAME OF THE PARTICULAR ENTRY POINT.
/       E.G.    .GLOBL %AT,%RS  DENOTES
/       ENTRY '%RS' OF MODULE '%AT'.
/
        .IFUND %X  ;%X=0;    .ENDC  / OMIT STREAM MACRO'S BY DEFAULT
/
/
/       ** INTERNAL **
/
 .IFZER %X
        .DEFIN %M1,I,J,K
        .GLOBL I; LAC J; JMS* I; .DSA K
        .ENDM
 .ENDC
/
        .DEFIN %M2,NAME,EXT     / GENERATE A 3-WORD DIRECTORY ENTRY BLOCK
%X=.;   .SIXBT NAME
        0; .LOC %X+2; .SIXBT EXT'SRC'
        .LOC %X+3
        .ENDM
/
 .IFNZR %X      / DEFINE I/O STREAM BASES AND VALIDITY CHECK
INPUT=0;OUTPUT=4
/
        .DEFIN %T1,STREAM; .IFNZR -OUTPUT-1&STREAM / IF \OUTPUT&STREAM
** FAULTY PARAM; .ENDC
        .ENDM
 .ENDC
/
/
/
/       ** PART 1 **
 .IFZER %X
        .DEFIN PACK,PNTR1,PNTR2; %M1 %PK,PNTR1,PNTR2
        .ENDM
/
```

```
        .DEFIN UNPACK,PNTR1,PNTR2; %M1 %UN,PNTR1,PNTR2
        .ENDM
/
        .DEFIN SEEK DAT,NAME,EXT; %M1 %SE,DAT,(.+2)
        JMP .+4; %M2 NAME,EXT
        .ENDM
/
        .DEFIN ENTER DAT,NAME,EXT; %M1 %SE,DAT,<XCT (.+2)> / XCT=400000(8)
        JMP .+4; %M2 NAME,EXT
        .ENDM
/
        .DEFIN CLOSE,DAT; .GLOBL %CS; LAC DAT; JMS* %CS
        .ENDM
/
/
        .DEFIN READ,DAT,PNTR; %M1 %RD,DAT,PNTR
        .ENDM
/
        .DEFIN WRITE,DAT,PNTR; %M1 %WR,DAT,PNTR
        .ENDM
 .ENDC
/
/
/
/       ** PART 2 **
/
        .DEFIN EVENT,NUMBER,TEXT,?TAG
        .GLOBL %EV; LAC (NUMBER); JMS* %EV; JMP TAG
        .ASCII '@TEXT@'<15>
TAG=.
        .ENDM
/
        .DEFIN ON,EVENT,ACTION,?T1,?T2
        .GLOBL %EV,%ON
        LAC (EVENT); JMS* %ON; JMP T2+1
        0
T1      0
        ACTION
        DZM T1-1          / UNSET 'BUSY' MARKER
T2      JMP* T1
        .ENDM
/
/
/       ** PART 3 **
/
        .DEFIN ALLOT,SIZE,PNTR
        .GLOBL %AL; LAC SIZE; JMS* %AL; DAC PNTR
        .ENDM
/
        .DEFIN FREE,PNTR
        .GLOBL %AL,%FR; LAC PNTR; JMS* %FR
        .ENDM
/
/
/
/       ** PART 4 **
 .IFNZR %X
```

```
        .DEFIN ATTACH,STREAM,NO,NAME,EXT
        .GLOBL %AT; %T1 STREAM
%X=OUTPUT-1&NO+STREAM; LAC (%X);%X=%X+1; .IODEV %X
        JMS* %AT; .DSA (.+2); JMP .+4; %M2 NAME,EXT
        .ENDM
/
        .DEFIN RSYM,I
        .GLOBL %AT,%RS; JMS* %RS; DAC I
        .ENDM
/
        .DEFIN PSYM,I
        .GLOBL %AT,%PS; LAC I; JMS* %PS
        .ENDM
/
        .DEFIN SELECT,STREAM,NO
        .GLOBL %AT,%SL; %T1 STREAM
        LAC NO; JMS* %SL
        .IFZER STREAM; .GLOBL %IO; .DSA %IO; .ENDC
        .IFNZR STREAM; .GLOBL %OO; XCT %OO; .ENDC  / XCT=400000(8)
        .ENDM
/
        .DEFIN CLOSE,STREAM
        .GLOBL %AT,%CL; %T1 STREAM
        .IFZER STREAM; .GLOBL %IO; LAC %IO; .ENDC
        .IFNZR STREAM; .GLOBL %OO; LAC %OO; .ENDC
        JMS* %CL
        .ENDM
 .ENDC
/
/
/
/         ** PART 5 **
/
        .DEFIN .TRACE
        .GLOBL %I9; LAS; SZA; JMS* %I9
        .ENDM
/
```

```
PAGE    1      %RD     SRC       *** EMOP: PART 1 ***

                                        .TITLE  *** EMOP: PART 1 ***
                                        .GLOBL %RD
                                /       READ A LA DEC BUT WITH DYNAMIC PARAMETERS
                                / CALLING SEQUENCE:
                                /       LAC <DAT>; JMS* %RD; .DSA <ADDR OF POINTER>
                                / ON EXIT ACCUMULATOR IS NORMALLY 777777(8).  AC. = 0
                                / SIGNALS END-OF-FILE.
                                /     THIS ROUTINE DOESN'T DISTINGUISH BETWEEN IOPS
                                / CODES 5 AND 6(END-OF-FILE/END-OF-MEDIUM)
                                /
    00000 R 000000 A            %RD
    00001 R 500030 R                    AND (777); DAC WAIT  ;    / DAT SLOT NO.
    00002 R 040016 R
    00003 R 340031 R                    TAD (2000); DAC READ  ; / DAT SLOT + IOPS ASCII
    00004 R 040012 R
    00005 R 220000 R                    LAC* %RD; ISZ %RD / GET ADDRESS OF BUFFER PNTR
    00006 R 440000 R
    00007 R 040014 R                    DAC READ+2; LAC* READ+2; DAC READ+2 / GET VALUE
    00010 R 220014 R
    00011 R 040014 R
    00012 R                     READ    .READ 0,2,0,200
    00012 R 002000 A *G                 CAL+2*1000 0&777
    00013 R 000010 A *G                 10
    00014 R 000000 A *G                 0
                     *G                 .DEC
    00015 R 777470 A *G                 -200
    00016 R                     WAIT    .WAIT 0
    00016 R 000000 A *G                 CAL 0&777
    00017 R 000012 A *G                 12
    00020 R 220014 R                    LAC* READ+2; AND (17)
    00021 R 500032 R
    00022 R 540033 R                    SAD (5); SKP
    00023 R 741000 A
    00024 R 540034 R                    SAD (6); CLA!SKP
    00025 R 751000 A
    00026 R 750001 A                    CLA!CMA       / AC = 777777(OK),000000(EOF)
    00027 R 620000 R                    JMP* %RD
          000000 A                      .END
    00030 R 000777 A *L
    00031 R 002000 A *L
    00032 R 000017 A *L
    00033 R 000005 A *L
    00034 R 000006 A *L
            SIZE=00035      NO ERROR LINES
```

```
                                .TITLE  *** EMOP: PART 1 ***
                                .GLOBL %WR
                        /       WRITE A LA DEC BUT WITH DYNAMIC PARAMETERS
                        / CALLING SEQUENCE:
                        /       LAC <DAT>; JMS* %WR; .DSA <ADDR OF POINTER>
                        /
                        / NO ERROR CONDITIONS AT THIS LEVEL
                        /
00000 R 000000 A        %WR
00001 R 500021 R                AND (777); DAC WAIT
00002 R 040016 R
00003 R 340022 R                TAD (2000); DAC WRITE
00004 R 040012 R
00005 R 220000 R                LAC* %WR; ISZ %WR
00006 R 440000 R
00007 R 040014 R                DAC WRITE+2; LAC* WRITE+2; DAC WRITE+2
00010 R 220014 R
00011 R 040014 R
00012 R                 WRITE   .WRITE 0,2,0,200
00012 R 002000 A *G             CAL+2*1000 0&777
00013 R 000011 A *G             11
00014 R 000000 A *G             0
                        *G      .DEC
00015 R 777470 A *G             -200
00016 R                 WAIT    .WAIT 0
00016 R 000000 A *G             CAL 0&777
00017 R 000012 A *G             12
00020 R 620000 R                JMP* %WR
        000000 A                .END
00021 R 000777 A *L
00022 R 002000 A *L
            SIZE=00023      NO ERROR LINES
```

```
                                .TITLE  *** EMOP: PART 1 ***
                                .GLOBL %SE
                        /       SEEK/ENTER A LA DEC BUT WITH DYNAMIC PARAMETERS
                        /       ALSO GENERATES .INIT AUTOMATICALLY
                        /
                        / CALLING SEQUENCE:
                        /       LAC <DAT>; JMS* %SE; S+<ADDR OF POINTER TO DIREC
                        / WHERE:
                        /       S = 0 FOR INPUT FILE(SEEK)
                        /       S = 400000 FOR OUTPUT FILE(ENTER)
                        /
                        /       NO ERROR CONDITIONS AT THIS LEVEL
                        /
00000 R 000000 A        %SE
00001 R 500032 R                AND (777); DAC SEEK     / SET .DAT SLOT NO.
00002 R 040025 R
                                / LOAD POINTER TO POINTER TO DIRECTORY ENTRY BLO
00003 R 220000 R                LAC* %SE; ISZ %SE; DAC #P
00004 R 440000 R
00005 R 040031 R
00006 R 740010 A                RAL                     / INPUT(0) OR OUTPUT(1)
00007 R 200025 R                LAC SEEK; SZL; XOR (1000); DAC INIT
00010 R 741400 A
00011 R 240033 R
00012 R 040021 R
00013 R 200034 R                LAC (3); SZL; TAD (1); DAC SEEK+1
00014 R 741400 A
00015 R 340035 R
00016 R 040026 R
00017 R 220031 R                LAC* P; DAC SEEK+2      / ENTRY BLOCK POINTER
00020 R 040027 R
                        /
00021 R                 INIT    .INIT 0,0,0
00021 R 000000 A *G             CAL+0*1000 0&777
00022 R 000001 A *G             1
00023 R 000000 A *G             0+0
00024 R 000000 A *G             0
                        /
00025 R                 SEEK    .SEEK 0,0
00025 R 000000 A *G             CAL 0&777
00026 R 000003 A *G             3
00027 R 000000 A *G             0
                        /
00030 R 620000 R                JMP* %SE
        000000 A                .END
00032 R 000777 A *L
00033 R 001000 A *L
00034 R 000003 A *L
00035 R 000001 A *L
            SIZE=00036      NO ERROR LINES
```

```
                                .TITLE  *** EMOP: PART 1 ***
                                .GLOBL %CS
                        /       CLOSE A LA DEC WITH DYNAMIC PARAMETER
                        / CALLING SEQUENCE:
                        /       LAC <DAT>; JMS* %CS
                        /
                        /       NO ERROR CONDITIONS AT THIS LEVEL
                        /
00000 R 000000 A        %CS
00001 R 500006 R                AND (777); DAC CLOSE
00002 R 040003 R
00003 R                 CLOSE   .CLOSE 0
00003 R 000000 A *G             CAL 0&777
00004 R 000006 A *G             6
00005 R 620000 R                JMP* %CS
        000000 A                .END
00006 R 000777 A *L
            SIZE=00007      NO ERROR LINES
```

```
PAGE    1       %PK     SRC         *** EMOP: PART 1 ***

                                            .TITLE  *** EMOP: PART 1 ***
                                            .GLOBL %PK
                                    /
                                    / PACK THE FORMAT GENERATED BY '%UN' INTO DEC'S
                                    / 5/7 ASCII AND SET THE HEADER PAIR APPROPRIATELY.
                                    /
                                    / CALLING SEQUENCE:
                                    /       LAC <FROM>; JMS* %PK; .DSA <TO>
                                    / WHERE <FROM> AND <TO> ARE NAMES OF POINTERS TO THE SOU
                                    / SINK BUFFER AREAS.
                                    /
                                    /
                                    /
                                            .DEFIN  LEFT4
                                            RTL; RTL
                                            .ENDM
                                    /
                                            .DEFIN  RIGHT3
                                            CLL!RAR; CLL!RAR; CLL!RAR
                                            .ENDM
                                    /
            100000 R        LEFT7=JMS .; 0
    00000 R 000000 A
    00001 R 744010 A                        CLL!RAL; RTL; RTL; RTL
    00002 R 742010 A
    00003 R 742010 A
    00004 R 742010 A
    00005 R 620000 R                        JMP* LEFT7&7777
                                    /
            100006 R        GET=JMS .; 0
    00006 R 000000 A
    00007 R 440115 R                        ISZ #FROM; LAC* FROM; AND (177); DAC #TMP
    00010 R 220115 R
    00011 R 500122 R
    00012 R 040117 R
    00013 R 200115 R                        LAC FROM; TAD #MLIMIT; SMA!CLL; JMP .+3
    00014 R 340116 R
    00015 R 744100 A
    00016 R 600021 R
    00017 R 200117 R                        LAC TMP; JMP* GET&7777
    00020 R 620006 R
    00021 R 751200 A                        SNA!CLA; LAC (15); JMP* GET&7777
    00022 R 200123 R
    00023 R 620006 R
                                    /
                                    /
    00024 R 000000 A        %PK
    00025 R 040115 R                        DAC FROM
    00026 R 360115 R                        TAD* FROM; CMA; DAC MLIMIT
    00027 R 740001 A
    00030 R 040116 R
    00031 R 220024 R                        LAC* %PK; ISZ %PK
    00032 R 440024 R
    00033 R 040120 R                        DAC #TO; LAC* TO; DAC TO
    00034 R 220120 R
```

```
00035 R 040120 R
00036 R 340124 R                TAD (2); DAC #T1
00037 R 040121 R
00040 R 100006 R     L1         GET; LEFT7
00041 R 100000 R
00042 R 040113 R                DAC #A
00043 R 100006 R                GET; XOR A; LEFT4
00044 R 240113 R

00045 R 742010 A *G             RTL; RTL
00046 R 742010 A *G
00047 R 040113 R                DAC A
00050 R 100006 R                GET; DAC #B; RIGHT3
00051 R 040114 R

00052 R 744020 A *G             CLL!RAR; CLL!RAR; CLL!RAR
00053 R 744020 A *G
00054 R 744020 A *G
00055 R 240113 R                XOR A; DAC* T1; ISZ T1
00056 R 060121 R
00057 R 440121 R
00060 R 200114 R                LAC B; AND (7); LEFT7
00061 R 500125 R
00062 R 100000 R
00063 R 040114 R                DAC B
00064 R 100006 R                GET; XOR B; LEFT7
00065 R 240114 R
00066 R 100000 R
00067 R 040114 R                DAC B
00070 R 100006 R                GET; XOR B
00071 R 240114 R
00072 R 744010 A                CLL!RAL; DAC* T1; ISZ T1
00073 R 060121 R
00074 R 440121 R
00075 R 200115 R                LAC FROM; TAD MLIMIT; SPA; JMP L1
00076 R 340116 R
00077 R 741100 A
00100 R 600040 R
                        /
00101 R 200120 R                LAC TO; CMA; TAD (1); TAD T1
00102 R 740001 A
00103 R 340126 R
00104 R 340121 R
00105 R 100000 R                LEFT7; RAL; AND (377000)
00106 R 740010 A
00107 R 500127 R
00110 R 340124 R                TAD (2); DAC* TO / PAIR COUNT + IOPS ^SCII(=2)
00111 R 060120 R
00112 R 620024 R                JMP* %PK
                        /
        000000 A                .END
00122 R 000177 A *L
00123 R 000015 A *L
00124 R 000002 A *L
00125 R 000007 A *L
```

```
00126 R 000001 A *L
00127 R 377000 A *L
            SIZE=00130      NO ERROR LINES
```

```
                                .TITLE  *** EMOP: PART 1 ***
                                .GLOBL %UN
                        /
                        / UNPACK A DEC 5/7('IOPS') ASCII LINE BUFFER
                        / INTO CONSECUTIVE MACHINE WORDS EACH
                        / CONTAINING ONE CHARACTER.  A SINGLE HEADER
                        / WORD IS SET TO CONTAIN THE (+VE) NUMBER OF
                        / CHARACTERS IN THE BUFFER, I.E. TO ONE LESS
                        / THAN THE TOTAL LENGTH OF THE BUFFER COUNTING
                        / THE HEADER.
                        /  SINCE THE RESULTING FORMAT IS LESS COMPACT
                        / THAN 5/7 ASCII, CARE MUST BE TAKEN THAT CORRUPTION
                        / DOES NOT OCCUR IF THE TWO BUFFERS ARE NOT
                        / DISTINCT.
                        /
                        / CALLING SEQUENCE:
                        /       LAC <FROM>; JMS* %UN; .DSA <TO>
                        / WHERE <FROM> AND <TO> ARE NAMES OF POINTERS
                        / OF THE SOURCE AND SINK BUFFERS RESP.
                        /
                        /
00000 R 000000 A        %UN
00001 R 040071 R                DAC #FROM
00002 R 220000 R                LAC* %UN; ISZ %UN; DAC #TO
00003 R 440000 R
00004 R 040072 R
00005 R 220072 R                LAC* TO; DAC TO; DAC #T1
00006 R 040072 R
00007 R 040074 R
00010 R 220071 R                LAC* FROM; AND (377000)
00011 R 500077 R
00012 R 744003 A                CLL!CML!CMA; RTR; RTR; RTR; RTR
00013 R 742020 A
00014 R 742020 A
00015 R 742020 A
00016 R 742020 A
00017 R 340100 R                TAD (2); DAC #TWPC
00020 R 040073 R
                                / 'TWPC' HOLDS TWICE WORD PAIR COUNT
                                / FOR LINE LESS HEADER PAIR
00021 R 440071 R                ISZ FROM; ISZ FROM
00022 R 440071 R
00023 R 440073 R        L1      ISZ TWPC; JMP L3
00024 R 600033 R
00025 R 200074 R        L2      LAC T1; CMA; TAD TO; CMA; DAC* TO
00026 R 740001 A
00027 R 340072 R
00030 R 740001 A
00031 R 060072 R
00032 R 620000 R                JMP* %UN
00033 R 220071 R        L3      LAC* FROM; ISZ FROM; DAC #WD1
00034 R 440071 R
00035 R 040075 R
00036 R 220071 R                LAC* FROM; ISZ FROM; DAC #WD2
00037 R 440071 R
```

```
00040 R 040076 R
00041 R 777773 A                LAW -5; DAC #CNT5
00042 R 040067 R
00043 R 777770 A     L5         LAW -10; DAC #CNT7.5
00044 R 040070 R
00045 R 200076 R     L6         LAC WD2; RAL
00046 R 740010 A
00047 R 440070 R                ISZ CNT7.5; SKP; JMP L7
00050 R 741000 A
00051 R 600057 R
00052 R 040076 R                DAC WD2
00053 R 200075 R                LAC WD1; RAL; DAC WD1
00054 R 740010 A
00055 R 040075 R
00056 R 600045 R                JMP L6
00057 R 500101 R     L7         AND (177); SAD (15); JMP L2
00060 R 540102 R
00061 R 600025 R
00062 R 440074 R                ISZ T1; DAC* T1
00063 R 060074 R
00064 R 440067 R                ISZ CNT5; JMP L5
00065 R 600043 R
00066 R 600023 R                JMP L1
                                /
        000000 A                .END
00077 R 377000 A *L
00100 R 000002 A *L
00101 R 000177 A *L
00102 R 000015 A *L
            SIZE=00103      NO ERROR LINES
```

```
                        .TITLE  *** EMOP: PART 2 ***
                        .GLOBL %EV,%ON
                    / THIS CONTINGENCY PACKAGE CONSISTS OF TWO
                    / PARTS:  '%ON' WHICH SETS UP THE ACTION TO BE
                    / TAKEN WHEN SOME PARTICULAR EVENT IS SIGNALLED
                    / AND  '%EV'  WHICH SUPERVISES THE TRANSFER OF
                    / CONTROL TO THE APPROPRIATE 'ACTION' AND THEN
                    / THE RETURN TO THE ORIGINAL SEQUENCE.
                    / A CHECK IS CARRIED OUT TO TRAP RE-ENTRANT
                    / CALLS ON ANY ACTION; IF SUCH A CALL IS
                    / DETECTED, THE PROGRAM IS TERMINATED.  IF
                    / THE ACTION CAUSES A BRANCH OUT OF ITSELF
                    / TO SOME OTHER PART OF THE PROGRAM AND
                    / DOES NOT RETURN, THE EVENT WHICH
                    / CAUSED THIS ACTION TO BE ENTERED IN THE
                    / FIRST PLACE SHOULD BE RESET WITH AN %ON
                    / STATEMENT TO PREVENT A SPURIOUS 'RE-ENTRANT CALL'
                    / FAULT SUBSEQUENTLY OCCURRING IF AND WHEN
                    / THAT ACTION IS RE-ACTIVATED.
                    /
                    / * E V E N T 0 IS UNTRAPPABLE AND INVARIABLY
                    / CAUSES THE MONITOR TO BE RELOADED.
                    / AS SUCH, IT CAN BE USED TO SIGNAL THE
                    / OCCURRENCE OF IRRECOVERABLE ERRORS.
                    /
                    / N.B.   NOTE THAT AN EVENT NO. DEFINES
                    / A CATEGORY WHICH MAY CONTAIN ANY NUMBER
                    / OF DISTINCT ENTITIES.  IF THE EVENT IS NOT
                    / TRAPPED, THESE ENTITIES CAN BE DISTINGUISHED
                    / BY THE MESSAGE PASSED WHEN SIGNALLING
                    / THE OCCURRENCE OF A MEMBER OF THE CATEGORY
                    /
                    / %ON: SET UP CONTINGENCY ROUTINE:
                    / CALLING SEQUENCE:
                    /       LAC <EVENT NO>; JMS* %ON; JMP T2+1
                    /       0       / RE-ENTRANT CALL MARKER
                    / T1    0
                    /       <ACTION>
                    /       DZM T1-1        / CLEAR MARKER
                    / T2    JMP* T1
                    /
                    /
                        .DEC
                    / DEFINE NO. OF DISTINCT EVENTS AVAILABLE
    000020 A        TABSZE=16       / ** MUST BE A POWER OF 2 **
                        .OCT
                    /
                    /
00000 R 000000 A    %ON
00001 R 741300 A        SPA!SNA; JMP* %ON       / CAN'T SET <= 0
00002 R 620000 R
00003 R 500276 R        AND (TABSZE-1); TAD (IVTAB); DAC .Y
00004 R 340277 R
00005 R 040274 R
00006 R 200000 R        LAC %ON; AND (77777);!TAD (1); DAC* .Y
```

```
00007 R 500300 R
00010 R 340301 R
00011 R 060274 R
00012 R 040274 R            DAC Y; DZM* Y    / CLEAR 'BUSY' FLAG
00013 R 160274 R
00014 R 620000 R            JMP* %ON
                            .EJECT
```

```
                        / %EV: SIGNAL OCCURRENCE OF SPECIFIED EVENT
                        / CALLING SEQUENCE:
                        /       LAC <EVENT NO>; JMS* %EV; JMP TAG
                        /       .ASCII '@<TEXT>@'<15>   / FAULT MESSAGE
                        / TAG=.
                        /
                        / NOTE THAT *EVENT 0 IS TREATED AS ANY
                        / OTHER HERE BUT THAT IT IS IMPOSSIBLE
                        / TO SET A TRAP FOR IT USING %ON.  IT
                        / ALWAYS THEREFORE CAUSES THE MONITOR TO
                        / BE RELOADED AFTER PRINTING THE FAULT
                        / MESSAGE.
                        /
00015 R 000000 A        %EV
00016 R 500276 R                AND (TABSZE-1); DAC #NUMBER
00017 R 040270 R
00020 R 340277 R                TAD (TVTAB); DAC Y
00021 R 040274 R
00022 R 220274 R                LAC* Y; SPA!SNA; JMP NOTSET
00023 R 741300 A
00024 R 600057 R
                                / TRAP IS SET IF AC > 0
00025 R 040274 R                DAC Y; LAC* Y; SZA; JMP RECURS
00026 R 220274 R
00027 R 740200 A
00030 R 600061 R
00031 R 200015 R                LAC %EV
00032 R 060274 R                DAC* Y; ISZ Y
00033 R 440274 R
00034 R 060274 R                DAC* Y; ISZ Y
00035 R 440274 R
00036 R 620274 R                JMP* Y
                                /
                                .REPT TABSZE
00037 R 777777 A        TVTAB   -1
00040 R 777777 A *R
00041 R 777777 A *R
00042 R 777777 A *R
00043 R 777777 A *R
00044 R 777777 A *R
00045 R 777777 A *R
00046 R 777777 A *R
00047 R 777777 A *R
00050 R 777777 A *R
00051 R 777777 A *R
00052 R 777777 A *R
00053 R 777777 A *R
00054 R 777777 A *R
00055 R 777777 A *R
00056 R 777777 A *R
                        /
00057 R 200015 R        NOTSET  LAC %EV; SKP
00060 R 741000 A
00061 R 200302 R        RECURS  LAC (RECM-1)
00062 R 040271 R                DAC #P
```

```
00063 R 200303 R                LAC (M3-1); DAC #Q
00064 R 040272 R
00065 R 777766 A                LAW -12; DAC Y  / MAX OF 24 CHARACTERS
00066 R 040274 R
00067 R 440271 R                ISZ P; ISZ Q
00070 R 440272 R
00071 R 220271 R                LAC* P; DAC* Q
00072 R 060272 R
00073 R 440274 R                ISZ Y; JMP .-5
00074 R 600067 R
                                / MAKE SURE OF CARRIAGE RETURN
00075 R 220272 R                LAC* Q; AND (774000); XOR (32); DAC* Q
00076 R 500304 R
00077 R 240305 R
00100 R 060272 R
                                /
00101 R 200306 R                LAC (M1); DAC P; TAD (1); DAC Q
00102 R 040271 R
00103 R 340301 R
00104 R 040272 R
00105 R 160271 R                DZM* P; LAC (124*200+40*2); DAC* Q       / 'T'
00106 R 200307 R
00107 R 060272 R
00110 R 777734 A                LAW -44; JMS DSHIFT  ;  / RIGHT 1
00111 R 100236 R
00112 R 200270 R                LAC NUMBER; JMS BCD; JMS DLJ
00113 R 100172 R
00114 R 100251 R
00115 R 200310 R                LAC (M2); DAC P; TAD (1); DAC Q
00116 R 040271 R
00117 R 340301 R
00120 R 040272 R
00121 R 160271 R                DZM* P; DZM* Q
00122 R 160272 R
00123 R 777776 A                LAW -2; TAD %EV; AND (77777)
00124 R 340015 R
00125 R 500300 R
00126 R 100172 R                JMS BCD; JMS DLJ
00127 R 100251 R
                                .WRITE -3,2,MO,100
00130 R 002775 A *G             CAL+2*1000 -3&777
00131 R 000011 A *G             11
00132 R 000140 R *G             MO
                 *G             .DEC
00133 R 777634 A *G             -100
                                .WAIT -3
00134 R 000775 A *G             CAL -3&777
00135 R 000012 A *G             12
                                .EXIT
00136 R 000000 A *G             CAL
00137 R 000015 A *G             15
                                /
00140 R 032002 A     MO         32002; 0
00141 R 000000 A
00142 R 252132 A                .ASCII '*EVEN'
```

```
00143 R 642634 A
00144 R 000000 A     M1         0; 0
00145 R 000000 A
00146 R 000000 A     M2         0; 0
00147 R 000000 A
00150 R 201004 A                .ASCII '      '  / FIVE SPACES
00151 R 020100 A
00152 R        A     M3         .BLOCK 12
00164 R 251252 A     RECM       .ASCII '**RE-ENTERED**'<15>
00165 R 242532 A
00166 R 426352 A
00167 R 442644 A
00170 R 426105 A
00171 R 225032 A
                     /
                     /
00172 R 000000 A     BCD
00173 R 040274 R                DAC Y
00174 R 777773 A                LAW -5; DAC #Z
00175 R 040275 R
00176 R 200274 R                LAC Y
00177 R 500311 R     BCD1       AND (700000); SZA; JMP BCD2
00200 R 740200 A
00201 R 600210 R
00202 R 200274 R                LAC Y; RAL; RTL; DAC Y
00203 R 740010 A
00204 R 742010 A
00205 R 040274 R
00206 R 440275 R                ISZ Z; JMP BCD1
00207 R 600177 R
                     /
00210 R 777777 A     BCD2       LAW -1; TAD Z; DAC Z
00211 R 340275 R
00212 R 040275 R
00213 R 200274 R     BCD3       LAC Y; RTL; RTL; DAC #TMP
00214 R 742010 A
00215 R 742010 A
00216 R 040273 R
00217 R 740020 A                RAR; DAC Y
00220 R 040274 R
00221 R 200273 R                LAC TMP; AND (7); TAD (60); DAC TMP
00222 R 500312 R
00223 R 340313 R
00224 R 040273 R
00225 R 777771 A                LAW -7; JMS DSHIFT
00226 R 100236 R
00227 R 777600 A                LAW 777600; AND* Q; XOR TMP; DAC* Q
00230 R 520272 R
00231 R 240273 R
00232 R 060272 R
00233 R 440275 R                ISZ Z; JMP BCD3
00234 R 600213 R
00235 R 620172 R                JMP* BCD
                     /
00236 R 000000 A     DSHIFT
```

```
00237 R 040267 R                DAC #CNTR
00240 R 220272 R                LAC* Q; RAL; DAC* Q
00241 R 740010 A
00242 R 060272 R
00243 R 220271 R                LAC* P; RAL; DAC* P
00244 R 740010 A
00245 R 060271 R
00246 R 440267 R                ISZ CNTR; JMP .-7
00247 R 600240 R
00250 R 620236 R                JMP* DSHIFT
                                /
00251 R 000000 A    DLJ
00252 R 754001 A                CLL!CLA!CMA; JMS DSHIFT
00253 R 100236 R
00254 R 774000 A    DLJ1        LAW 774000; AND* P; SZA!CLL; JMP* DLJ
00255 R 520271 R
00256 R 744200 A
00257 R 620251 R
00260 R 777771 A                LAW -7; JMS DSHIFT
00261 R 100236 R
00262 R 777400 A                LAW 777400; AND* Q; XOR (100); DAC* Q
00263 R 520272 R
00264 R 240314 R
00265 R 060272 R
                                / (100) = ' '<<1
00266 R 600254 R                JMP DLJ1
                                /
        000000 A                .END
00276 R 000017 A *L
00277 R 000037 R *L
00300 R 077777 A *L
00301 R 000001 A *L
00302 R 000163 R *L
00303 R 000151 R *L
00304 R 774000 A *L
00305 R 000032 A *L
00306 R 000144 R *L
00307 R 052100 A *L
00310 R 000146 R *L
00311 R 700000 A *L
00312 R 000007 A *L
00313 R 000060 A *L
00314 R 000100 A *L
            SIZE=00316      NO ERROR LINES
```

```
                                .TITLE  *** EMOP: PART 3 ***
                                .GLOBL %AL,%FR
                        / %AL: STORAGE ALLOCATOR.
                        / BLOCKS OF STORAGE ARE GIVEN TO THE USER PROGRAM
                        / ON REQUEST FROM THE FREE SPACE AVAILABLE
                        / BELOW C(.SCOM+3), ALLOCATING DOWNWARDS
                        / TOWARDS C(.SCOM+2).
                        / A HOUSEKEEPING TABLE IS BUILT UPWARDS FROM
                        / C(.SCOM+2).
                        / NONE OF THE .SCOM REGISTERS IS ALTERED.
                        /
                        / CALLING SEQUENCE:
                        /       LAC <N>; JMS* %AL; DAC <P>
                        / I.E.  ALLOCATE <N> WORDS OF STORE TO POINTER <P>
                        /
                        / * E V E N T 1 IS SIGNALLED ON STORE OVERFLOW
                        /
        000100 A        .SCOM=100
                        /
                        /
00000 R 000000 A        %AL
00001 R 040160 R                DAC #N
00002 R 777777 A        L1      LAW -1; SAD BUSYP; JMP SETUP    / INITIALISE?
00003 R 540060 R
00004 R 600052 R
00005 R 340160 R                TAD N; CMA; TAD BUSYP; DAC #Y
00006 R 740001 A
00007 R 340060 R
00010 R 040164 R
00011 R 200163 R                LAC #TABTOP; DAC #P
00012 R 040161 R
00013 R 220161 R                LAC* P; CMA; DAC #Z; JMP L3
00014 R 740001 A
00015 R 040165 R
00016 R 600023 R
00017 R 440161 R        L2      ISZ P; LAC* P; SNA; JMP L4
00020 R 220161 R
00021 R 741200 A
00022 R 600027 R
00023 R 440165 R        L3      ISZ Z; JMP L2
00024 R 600017 R
00025 R 440161 R                ISZ P; ISZ* TABTOP
00026 R 460163 R
00027 R 200161 R        L4      LAC P; CMA; TAD Y; SMA; JMP L5
00030 R 740001 A
00031 R 340164 R
00032 R 740100 A
00033 R 600046 R
                                EVENT 1,<STORE OVERFLOW>
                    *G          .GLOBL %EV; LAC (1); JMS* %EV; JMP . 0002
00034 R 200167 R    *G
00035 R 120166 E    *G
00036 R 600045 R    *G
00037 R 516511 A    *G          .ASCII 'STORE OVERFLOW'<15>
00040 R 751212 A    *G
```

```
00041 R 202372 A *G
00042 R 642644 A *G
00043 R 432311 A *G
00044 R 753432 A *G
        000045 R *G ..0002=.
00045 R 600002 R            JMP L1  /TRY AGAIN
                    /
00046 R 200164 R    L5      LAC Y; DAC* P; DAC BUSYP
00047 R 060161 R
00050 R 040060 R
00051 R 620000 R            JMP* %AL
                    /
00052 R 220170 R    SETUP   LAC* (.SCOM+2); DAC TABTOP
00053 R 040163 R
00054 R 220171 R            LAC* (.SCOM+3); DAC BUSYP
00055 R 040060 R
00056 R 160163 R            DZM* TABTOP
00057 R 600002 R            JMP L1
                    /
00060 R 777777 A    BUSYP   -1
                            .EJECT
```

```
                        / %FR:  FREE STORAGE ALLOCATED BY %AL
                        / CALLING SEQUENCE:
                        /       LAC <PNTR>; JMS* %FR
                        / I.E. FREE BLOCK OF STORAGE POINTED AT BY <PNTR>
                        /
                        / * E V E N T O  IS SIGNALLED IF <PNTR> DOES
                        / NOT APPEAR IN INTERNAL HOUSEKEEPING TABLE.
                        / THIS IS ASSUMED TO BE AN IRRECOVERABLE
                        / ERROR, EITHER BY MACHINE OR PROGRAMMER.
                        /
                        /
00061 R 000000 A        %FR
00062 R 040161 R                DAC P
00063 R 220163 R                LAC* TABTOP; SPA!SNA!CMA; JMP L6A
00064 R 741301 A
00065 R 600105 R
00066 R 040162 R                DAC #R; DAC #Z  / DEFINES ALLOC. TABLE SIZE
00067 R 040165 R
00070 R 200163 R                LAC TABTOP; DAC Y
00071 R 040164 R
00072 R 140160 R                DZM N; LAC P; JMP .+4
00073 R 200161 R
00074 R 600100 R
00075 R 440164 R                ISZ Y; SAD* Y; ISZ N
00076 R 560164 R
00077 R 440160 R
00100 R 440162 R                ISZ R; JMP .-4
00101 R 600075 R
00102 R 200160 R                LAC N; SAD (1); JMP L6B / TABLE EMPTY?
00103 R 540167 R
00104 R 600120 R
00105 R                 L6A     EVENT 0,<FREE IMPOSSIBLE>
                    *G          .GLOBL %EV; LAC (0); JMS* %EV; JMP . 0005
00105 R 200172 R    *G
00106 R 120166 E    *G
00107 R 600120 R    *G
00110 R 432450 A    *G          .ASCII 'FREE IMPOSSIBLE'<15>
00111 R 542500 A    *G
00112 R 446332 A    *G
00113 R 047646 A    *G
00114 R 516230 A    *G
00115 R 246212 A    *G
00116 R 064000 A    *G
00117 R 000000 A    *G
          000120 R  *G  ..0005=.
00120 R 200163 R        L6B     LAC TABTOP; DAC Y
00121 R 040164 R
00122 R 220171 R                LAC* (.SCOM+3); CMA; DAC N      / -(((.SCOM+3)+1
00123 R 740001 A
00124 R 040160 R
00125 R 600143 R                JMP L8
00126 R 440164 R        L7      ISZ Y
00127 R 200161 R                LAC P; SAD* Y; DZM* Y   / CLEAR TABLE ENTRY = P
00130 R 560164 R
00131 R 160164 R
```

```
00132 R 220164 R              LAC* Y; SPA!SNA; JMP L9 / IGNORE ENTRY <= 0
00133 R 741300 A
00134 R 600145 R
00135 R 340160 R              TAD N; SMA!CLA!CMA; JMP L8
00136 R 750101 A
00137 R 600143 R
00140 R 360164 R              TAD* Y; CMA; DAC N
00141 R 740001 A
00142 R 040160 R
00143 R 200165 R     L8       LAC Z; DAC #R
00144 R 040162 R
00145 R 440165 R     L9       ISZ Z; JMP L7
00146 R 600126 R
00147 R 200162 R              LAC R; TAD (1); TAD* TABTOP; DAC* TABTOP
00150 R 340167 R
00151 R 360163 R
00152 R 060163 R
00153 R 777777 A              LAW -1; TAD N; CMA; DAC BUSYP    / RECOVER SPACE
00154 R 340160 R
00155 R 740001 A
00156 R 040060 R
00157 R 620061 R              JMP* %FR
          000000 A            .END
00166 R 000166 E *E
00167 R 000001 A *L
00170 R 000102 A *L
00171 R 000103 A *L
00172 R 000000 A *L
              SIZE=00173      NO ERROR LINES
```

```
PAGE    1       %AT     SRC         *** EMOP: PART 4 ***

                                .TITLE  *** EMOP: PART 4 ***
                        /
                        / CHARACTER STREAM PACKAGE.  CONTAINS:
                        /       %AT:    ATTACH INPUT/OUTPUT
                        /       %SL:    SELECT INPUT/OUTPUT
                        /       %RS:    READ SYMBOL
                        /       %PS:    PRINT SYMBOL
                        /       %CL:    CLOSE INPUT/OUTPUT
                        /
                        /
                                .GLOBL  %AT,%SL,%RS,%PS,%CL
                                .GLOBL %IO,%RD  / RSYM ONLY
                                .GLOBL %OO,%WR  / PSYM ONLY
                        /
            000004 A    OUTPUT=4            / * MUST MATCH DEFN. IN MACRO FILE *
                        /
                                .DEC ;BUFSZE=80; .OCT
            000120 A

                        /
            000100 A    .SCOM=100
                        /
                        / SAVES LITERALS IF THESE TABLES ARE AT
                        / THE BEGINNING.
    00000 R         A   CO      .BLOCK 5
                                / THIS TABLE IS USED AS WORK SPACE,
                                / IT'S ELEMENTS CORRESPONDING TO
                                / I/O(0:3) IN RSYM/PSYM.
                                / ELEMENTS(1:4) ARE SAVED IN %CT
                                / WHENEVER STREAMS ARE SWITCHED.
                                / TABLE CONTENTS ARE:
                                / (0)  STREAM NO. (0:OUTPUT-1)
                                / (1)  BUFFER START ADDRESS
                                / (2)  BUFFER CURRENT POSITION
                                / (3)  BUFFER END ADDRESS
                                / (4)  FILE DIRECTORY BLOCK POINTER
                        /
                        / ALLOCATE AREA FOR TABLE TO HOLD
                        / STATUS INFORMATION FOR CURRENTLY
                        / DE-SELECTED BUT STILL ACTIVE STREAMS
                        / SPACE IS REQUIRED BOTH FOR INPUT AND
                        / OUTPUT STREAMS AND 4 WORDS ARE REQUIRED
                        / FOR EACH STREAM
                        /
            000040 A    %CTSZE=2*OUTPUT*4       / DEFINE TABLE SIZE
                                .REPT %CTSZE
    00005 R 777777 A    %CT     -1
                        /
                        /
                                .DEFIN COPY,N,FROM,TO
                        %X=N;   LAW -%X; JMS CPY; LAC FROM; LAC TO
                                .ENDM
    00045 R 000000 A    CPY
    00046 R 040600 R            DAC #CPY3
    00047 R 420045 R            XCT* CPY; ISZ CPY; DAC #CPY1
```

```
00050 R 440045 R
00051 R 040576 R
00052 R 420045 R            XCT* CPY; ISZ CPY; DAC #CPY2
00053 R 440045 R
00054 R 040577 R
00055 R 220576 R            LAC* CPY1; DAC* CPY2; ISZ CPY1; ISZ CPY2
00056 R 060577 R
00057 R 440576 R
00060 R 440577 R
00061 R 440600 R            ISZ CPY3; JMP .-5
00062 R 600055 R
00063 R 620045 R            JMP* CPY
                    /
                    /
                    / COMPUTE CHECKSUM FOR STREAM; SAVE TABLE(%CT)
00064 R 000000 A    CHECK
00065 R 777740 A            LAW -%CTSZE; DAC CPY1
00066 R 040576 R
00067 R 200615 R            LAC (%CT); DAC CPY2
00070 R 040577 R
00071 R 320577 R            ADD* CPY2; ISZ CPY2
00072 R 440577 R
00073 R 440576 R            ISZ CPY1; JMP .-3
00074 R 600071 R
00075 R 620064 R            JMP* CHECK
                            .EJECT
```

```
                        / %AT:  ATTACH A DIRECTORY ENTRY BLOCK TO A SPECIFIED
                        / LOGICAL I/O STREAM.  THIS ESTABLISHES THE NAME OF
                        / THE FILE REFERRRED TO IMPLICITLY BY SUBSEQUENT
                        / I/O COMMANDS.
                        /
                        /
                        / CALLING SEQUENCE:
                        /       LAC <STREAM NO>; JMS* [%AT
                        /       .DSA <ADDR OF POINTER TO DIRECTORY ENTRY BLOCK>
                        /
                        /
00076 R 000000 A        %AT                         / ATTACH FILE DIRECTORY BLOCK
00077 R 500616 R                AND (2*OUTPUT-1)
00100 R 744010 A                CLL!RAL; CLL!RAL; TAD [(%CT); DAC #P
00101 R 744010 A
00102 R 340615 R
00103 R 040603 R
00104 R 160603 R                DZM* P; ISZ P; DZM* P; ISZ P; DZM* P; ISZ P
00105 R 440603 R
00106 R 160603 R
00107 R 440603 R
00110 R 160603 R
00111 R 440603 R
00112 R 220076 R                LAC* %AT; ISZ %AT; DAC #Q
00113 R 440076 R
00114 R 040604 R
00115 R 220604 R                LAC* Q; AND (77777); DAC* P
00116 R 500617 R
00117 R 060603 R
00120 R 100064 R                JMS CHECK; DAC #CHKSUM
00121 R 040575 R
00122 R 620076 R                JMP* %AT
                                .EJECT
```

```
                        / %SL:  SELECT I/O STREAM ALLOCATING BUFFER SPACE IF NEC
                        / ALSO DOES .INIT AND .SEEK/.ENTER IF NECESSARY.
                        /
                        / CALLING SEQUENCE:
                        /       LAC <STREAM NO>; JMS* %SL
                        /       .DSA %IO (INPUT)  OR .XCT %OO (OUTPUT)
                        /
                        / E.G. LAC (3); JMS* %SL; .DSA %IO
                        /
                        /
00123 R 000000 A    %SL                         / SELECT SPECIFIED I/O STREAM
00124 R 040602 R            DAC #N
00125 R 200574 R            LAC LINBUF; SMA; JMP LO
00126 R 740100 A
00127 R 600133 R
                            / ALLOCATE LINE-BUFFER IF NOT ALREADY DONE
                            ALLOT (BUFSZE*2/5+3),LINBUF
00133 R 220123 R    LO      LAC* %SL; ISZ %SL; DAC #IO; RAL
00134 R 440123 R
00135 R 040601 R
00136 R 740010 A
00137 R 220601 R            LAC* IO; DAC IO
00140 R 040601 R
00141 R 200602 R            LAC N; AND (OUTPUT-1); SZL; TAD (OUTPUT); DAC N
00142 R 500621 R
00143 R 741400 A
00144 R 340622 R
00145 R 040602 R
                            COPY 4,IO,(CO)
00152 R 200000 R            LAC CO; SAD (654321); JMP L2    / NOTHING CURRENT
00153 R 540624 R
00154 R 600210 R
00155 R 540602 R            SAD N; JMP* %SL         / REDUNDANT 'SELECT'
00156 R 620123 R
00157 R 100064 R            JMS CHECK; SAD CHKSUM; JMP L1
00160 R 540575 R
00161 R 600177 R
00162 R             CORUPT  EVENT 0,<STREAM TABLES CORRUPT>
                        /
00177 R 200000 R    L1      LAC CO
00200 R 744010 A            CLL!RAL; CLL!RAL; TAD (%CT); DAC P
00201 R 744010 A
00202 R 340615 R
00203 R 040603 R
                            COPY 3,(CO+1),P             / SAVE OLD STATUS
00210 R 200602 R    L2      LAC N; DAC CO           / SET UP NEW STATUS
00211 R 040000 R
00212 R 744010 A            CLL!RAL; CLL!RAL; TAD (%CT); DAC P
00213 R 744010 A
00214 R 340615 R
00215 R 040603 R
                            COPY 4,P,(CO+1)         / N.B. INCLUDING DIRECTORY
00222 R 200001 R            LAC CO+1; SMA!SZA; JMP L4
00223 R 740300 A
00224 R 600304 R
```

```
                                        ALLOT (BUFSZE),CO+1
00230 R 200001 R                        LAC CO+1; DAC CO+2
00231 R 040002 R
00232 R 200630 R                        LAC (BUFSZE-1); DAC* CO+1
00233 R 060001 R
00234 R 200000 R                        LAC CO+0; AND (OUTPUT); SZA; LAC (1000)  / INIT
00235 R 500622 R
00236 R 740200 A
00237 R 200631 R
00240 R 340000 R                        TAD CO+0; TAD (1); DAC INIT
00241 R 340632 R
00242 R 040275 R
00243 R 500633 R                        AND (777); DAC IOM
00244 R 040301 R
00245 R 240275 R                        XOR INIT; SZA!CLL; CML
00246 R 744200 A
00247 R 740002 A
00250 R 200621 R                        LAC (3); SZL; TAD (1); DAC IOM+1    / SEEK OR ENT
00251 R 741400 A
00252 R 340632 R
00253 R 040302 R
00254 R 200002 R                        LAC CO+2; SZL; TAD* CO+1; DAC CO+3 / INITIALISAT
00255 R 741400 A
00256 R 360001 R
00257 R 040003 R
                                        / INPUT: I1=I2=I3,  OUTPUT: I1=I2, I3=I2+(BUFSZE
                        /
00260 R 200004 R                        LAC CO+4; SMA!SZA; JMP L3
00261 R 740300 A
00262 R 600274 R
                                        EVENT 0,<NOT ATTACHED>              / NOT ATTACHED
                        /
00274 R 040303 R        L3              DAC IOM+2
00275 R                 INIT            .INIT 0,0,0
00301 R                 IOM             .SEEK 0,0
00304 R                 L4              COPY 4,(CO),IO          / SET NEW STATUS
00310 R 100064 R                        JMS CHECK; DAC CHKSUM
00311 R 040575 R
00312 R 620123 R                        JMP* %SL
                                        .EJECT
```

```
                        / %RS:  READ SYMBOL.  FETCH THE NEXT CHARACTER
                        / FROM THE CURRENTLY SELECTED INPUT STREAM AND
                        / MOVE FORWARD ONE CHARACTER.  CONSECUTIVE CALLS
                        / FETCH CONSECUTIVE CHARACTERS.
                        /
                        / CALLING SEQUENCE:
                        /       JMS* %RS; DAC <NAME>
                        /
                        /
                        / * E V E N T  2  SIGNALS  'END-OF-FILE'
                        / A PROGRAM ERROR RESULTING IN AN ATTEMPT TO READ WHEN N
                        / STREAM IS SELECTED IS SIGNALLED BY * E V E N T  0
                        /
                        /
                        / N.B.  I1,I2,I3 OCCUPY %IO+1,%IO+2,%IO+3 RESPECTIVELY
                        /
00313 R 000000 A        %RS
00314 R 200412 R        RSL0    LAC %IO; AND (7); SAD %IO; JMP RSL1
00315 R 500616 R
00316 R 540412 R
00317 R 600333 R
                                EVENT 0,<INPUT UNSELECTED>
00333 R 200414 R        RSL1    LAC I2; SAD I3; JMP RSL2
00334 R 540415 R
00335 R 600342 R
00336 R 440414 R                ISZ I2; LAC* I2; AND (177); JMP* %RS
00337 R 220414 R
00340 R 500634 R
00341 R 620313 R
                        /
                        / I2  &  I3  ARE CURRENTLY FREE.  USE THEN AS WORK LOCAT
00342 R 200412 R        RSL2    LAC %IO; TAD (1); DAC I2        / INPUT .DAT SLO
00343 R 340632 R
00344 R 040414 R
                        / N.B.    LINBUF  IS A POINTER TO SPACE
                        /       ACQUIRED FROM THE STORAGE ALLOCATOR,
                        /       NOT THE BUFFER ITSELF.
                                READ I2,LINBUF
00350 R 740200 A                SZA; JMP RSL3
00351 R 600366 R
                                EVENT 2,<INPUT ENDED>:          / INPUT ENDED
00363 R 200414 R                LAC I2; DAC I3; JMP RSL1        / TRY AGAIN
00364 R 040415 R
00365 R 600333 R
                        /
00366 R                 RSL3    UNPACK LINBUF,I1
00371 R 200413 R                LAC I1; DAC I2; TAD* I1; DAC I3
00372 R 040414 R
00373 R 360413 R
00374 R 040415 R
00375 R 220415 R        RSL4    LAC* I3; SAD (40); JMP RSL6     / REMOVE TRAILIN
00376 R 540636 R
00377 R 600404 R
00400 R 200637 R        RSL5    LAC (12); ISZ I3; DAC* I3; JMP RSL1
00401 R 440415 R
```

```
00402 R 060415 R
00403 R 600333 R
00404 R 777777 A     RSL6     LAW -1; TAD I3; DAC I3
00405 R 340415 R
00406 R 040415 R
00407 R 540413 R              SAD I1; JMP RSL4; JMP RSL5
00410 R 600375 R
00411 R 600400 R
                      /
00412 R 654321 A     %IO      654321           / INITIALLY DESELECTED
00413 R 000000 A     I1
00414 R 000000 A     I2
00415 R 000000 A     I3
                              .EJECT
```

```
                        / %PS:  PRINT SYMBOL.  OUTPUT A SINGLE SYMBOL TO THE OUT
                        / STREAM CURRENTLY SELECTED.
                        /
                        / * E V E N T  0  IS USED TO SIGNAL A PROGRAM ERROR RESU
                        / IN AN ATTEMPT TO WRITE WHEN NO OUTPUT STREAM IS SELECT
                        /
                        /
                        / N.B.  O1,O2,O3 OCCUPY %OO+1,%OO+2,%OO+3 RESPECTIVELY
                        /
00416 R 000000 A        %PS
00417 R 040605 R                DAC #TMP
00420 R 200475 R                LAC %OO; AND (7); SAD %OO; JMP PSL0
00421 R 500616 R
00422 R 540475 R
00423 R 600437 R
                                EVENT 0,<OUTPUT UNSELECTED>
00437 R 200605 R        PSL0    LAC TMP
00440 R 500634 R                 AND (177); SAD (12); JMP PSL1
00441 R 540637 R
00442 R 600451 R
00443 R 440477 R                 ISZ O2; DAC* O2
00444 R 060477 R
00445 R 200477 R                 LAC O2; SAD O3; JMP PSL1
00446 R 540500 R
00447 R 600451 R
00450 R 620416 R                 JMP* %PS
                        /
00451 R 200477 R        PSL1     LAC O2; CMA; TAD O1; CMA; DAC* O1
00452 R 740001 A
00453 R 340476 R
00454 R 740001 A
00455 R 060476 R
                        / O2  &  O3  ARE CURRENTLY NON-BUSY. USE THEM FOR WORK S
                        / AS IN %RS
                        /
00456 R 200475 R                 LAC %OO; TAD (1); DAC O2
00457 R 340632 R
00460 R 040477 R
                                 PACK O1,LINBUF
                                 WRITE O2,LINBUF
00467 R 200476 R                 LAC O1; DAC O2; TAD (BUFSZE-1); DAC O3
00470 R 040477 R
00471 R 340630 R
00472 R 040500 R
00473 R 160476 R                 DZM* O1          / COMPLETELY RESET BUFFER:- IT'S
00474 R 620416 R                 JMP* %PS
                        /
00475 R 654321 A        %OO      654321  / INITIALLY DESELECTED
00476 R 000000 A        O1
00477 R 000000 A        O2
00500 R 000000 A        O3
                                 .EJECT
```

```
                        / %CL:  CLOSE OFF I/O STREAM, FLUSH BUFFER, DE-ALLOCATE
                        / CALLING SEQUENCE:
                        /       LAC <ADDR %IO> (INPUT)  OR  LAC <ADDR %OO> (OUTP
                        /       JMS* %CL
                        /
                        /
00501 R 000000 A        %CL                         / CLOSE OFF FILE
00502 R 040601 R                DAC IO
00503 R 540640 R                SAD (%IO); JMP L4.9
00504 R 600520 R
00505 R 540641 R                SAD (%OO); JMP L4.9
00506 R 600520 R
                                EVENT 0,<FAULTY CLOSE>
                        /
00520 R 220601 R        L4.9    LAC* IO; SAD (654321); JMP* %CL
00521 R 540624 R
00522 R 620501 R
00523 R 100064 R                JMS CHECK; SAD CHKSUM; SKP; JMP CORUPT
00524 R 540575 R
00525 R 741000 A
00526 R 600162 R
                                COPY 4,IO,(CO)
00533 R 200000 R                LAC CO+0
00534 R 500622 R                AND (OUTPUT); SNA; JMP L5     / INPUT FILE ->
00535 R 741200 A
00536 R 600544 R
00537 R 200001 R                LAC CO+1; SAD CO+2; JMP L5   / BUFFER EMPTY ->
00540 R 540002 R
00541 R 600544 R
                                / CLOSE CURRENT LINE WITH NL CHARACTER
00542 R 200637 R                LAC (12); JMS %PS
00543 R 100416 R
00544 R 200000 R        L5      LAC CO+0; AND (2*OUTPUT-1)
00545 R 500616 R
00546 R 744010 A                CLL!RAL; CLL!RAL; TAD (%CT); DAC P
00547 R 744010 A
00550 R 340615 R
00551 R 040603 R
00552 R 200000 R                LAC CO+0; TAD (1); AND (777); DAC .+1
00553 R 340632 R
00554 R 500633 R
00555 R 040556 R
                                .CLOSE 0        / CLOSE OFF IOPS HANDLED
00560 R 200624 R                LAC (654321); DAC* IO       / DESELECT CURRENT STR
00561 R 060601 R
                                FREE CO+1                   / FREE BUFFER AREA ATTACHE
00564 R 140001 R                DZM CO+1                    / AND ZERO CORRESPONDING
                                COPY 3,(CO+1),P         / CLEAR OUT ITS TABLE ENTRY
00571 R 100064 R                JMS CHECK; DAC CHKSUM
00572 R 040575 R
00573 R 620501 R                JMP* %CL
                        /
                                / 'LINBUF' IS A POINTER TO SPACE
                                / ACQUIRED FROM THE ALLOCATOR.  INITIALISED
                                / TO -1 TO MARK INITIAL ABSENCE OF THIS BUFFER
```

```
00574 R 777777 A     LINBUF   -1
                              /
                              /
        000000 A                .END
00606 R 000606 E *E
00607 R 000607 E *E
00610 R 000610 E *E
00611 R 000611 E *E
00612 R 000612 E *E
00613 R 000613 E *E
00614 R 000614 E *E
00615 R 000005 R *L
00616 R 000007 A *L
00617 R 077777 A *L
00620 R 000043 A *L
00621 R 000003 A *L
00622 R 000004 A *L
00623 R 000000 R *L
00624 R 654321 A *L
00625 R 000000 A *L
00626 R 000001 R *L
00627 R 000120 A *L
00630 R 000117 A *L
00631 R 001000 A *L
00632 R 000001 A *L
00633 R 000777 A *L
00634 R 000177 A *L
00635 R 000002 A *L
00636 R 000040 A *L
00637 R 000012 A *L
00640 R 000412 R *L
00641 R 000475 R *L
            SIZE=00642     NO ERROR LINES
```

```
                .TITLE  INT9:-  INTERPRETIVE PDP-9 SIMULATOR
                /
                /
                /   *** KM9-15 VERSION ***
                / INT9 INTERPRETS RELOCATED BINARY
                / MACHINE CODE, SPLITTING THE STORE
                / INTO THREE ADDRESSING ZONES
                / KNOWN INTERNALLY AS 'A','B' AND 'C'.
                / IN 'C' CAL'S ARE ILLEGAL AND ONLY
                / WITHIN-C ADDRESS REFERENCES ARE PERMITTED
                / EXCEPT THAT:
                / 1) .EXIT( CAL; 15) IS PERMITTED
                / 2) JMS INTO 'B' IS PERMITTED
                /
                / IN 'B' ANYTHING GOES.
                /
                / NORMALLY 'C' AND 'B' ARE USER PROGRAM
                / AND LIBRARY ITEMS RESPECTIVELY.
                /
                / AT ALL TIMES, EAE INSTRUCTIONS AND
                / DBR NOT FOLLOWED BY JMP* ARE ILLEGAL
                / AND ARE TRAPPED.
                /
                /
                / IN THE KM9-15 ENVIRONMENT, INT9 IS
                / LOADED BETWEEN THE USER PROGRAMS
                / AND THE LIBRARY ITEMS. THE BOUNDARIES
                / OF THE ZONES ARE AS FOLLOWS:
                / ZONE B: LO - .SCOM+3, HI - INT9-1.
                / ZONE C: LO - END OF INT9+1, HI - .SCOM
                / ZONE A: ALL THE REST
                /
                / TO INTERRUPT HANDLING ROUTINES
                / INT9 APPEARS AS THE 'USER' PROGRAM
                / AND NO SPECIAL PRECAUTIONS NEED
                / BE TAKEN WHEN RUNNING DEVICE
                / HANDLERS INTERPRETIVELY. THIS IS
                / SO BECAUSE THE INTERRUPT DRIVEN
                / PORTION OF THE HANDLER IS NOT
                / INTERPRETED AND THEREFORE RUNS
                / AT FULL MACHINE SPEED.
                        .IFDEF EMOP
                / NOTE THAT THE INT9 PROGRAM UNIT(%I9)
                / MUST BE EITHER IN A 'USER LIBRARY'(.LIBR5)
                / OR AT THE VERY BEGINNING OF THE SYSTEM
                / LIBRARY(.LIBR) TO ENSURE THE CORRECT
                / RELATIVE POSITIONING OF THE VARIOUS
                / ITEMS IN CORE.
                        .ENDC
                /
                / WHATEVER ENVIRONMENT IT IS RUN IN, THE
                / BASIC BEHAVIOUR OF THE PROGRAM
                / IS TO INTERPRET BINARY MACHINE CODE
                / UNTIL A FAULT, HOWEVER THAT MAY BE DEFINED,
                / IS DISCOVERED.  WHILE DOING THIS, A CIRCULAR
```

```
                        / LIST IS MAINTAINED OF THE LAST 20(10) MEMORY
                        / REFERENCE INSTRUCTIONS EXECUTED INCLUDING
                        / A TRANSFER VECTOR IF INDIRECT, AND
                        / THE CONTENTS OF THE ADDRESSED LOCATION
                        / ON DISCOVERING A FAULT, THIS LIST
                        / IS OUTPUT TO THE CONSOLE TYPEWRITER BEFORE
                        / TERMINATING THE RUN BY RELOADING
                        / THE SUPERVISOR.
                        /
                        /
                        /
                        /       ** ASSEMBLY OPTIONS **
                        /
                        / DEFINITION OF THE ASSEMBLY SYMBOL 'EMOP'
                        / CHANGES THE GLOBAL NAME OF THIS PROGRAM
                        / TO '%I9' SO THAT IT CAN BE USED AS
                        / PART OF THE EMOP PACKAGE.  OTHERWISE
                        / IT IS A FREE-STANDING UNIT CALLED 'INT9'
                        / AND THE CALL IS:
                        /           .GLOBL INT9; JMS* INT9
                        /
                        /
                        /
                        /
                        /
                        / ONE AUTO-INDEX REGISTER IS REQUIRED
                        / TO WORK THE CIRCULAR BUFFER IN WHICH
                        / THE RECORD OF THE LAST 20 MR INSTRUCTIONS
                        / EXECUTED IS HELD.  THIS IS REFERRED TO
                        / SYMBOLICALLY AS  STP.
                        /
000017 A                STP=17
                        /
                        /
                        /
                        / THE RUN OF A PROGRAM MAY BE TERMINATED
                        / BY THE USER CHANGING THE SETTING OF
                        / THE ACCUMULATOR SWITCHES.
                        / SWTEST=SZA: ABORT ON ANY CHANGE
                        / SWTEST=SPA: ABORT ON SIGN BIT ONLY
                        / SWTEST=SKP: NEVER ABORT, DISABLE OPTION
                        /
                                / SUPPLY DEFAULT
                                .IFUND SWTEST  ;SWTEST=SZA;  .ENDC

                        /
                        /
                        /
                        /
                        / WHEN TESTING INTERRUPT-DRIVEN DEVICE
                        / HANDLERS, IT MAY BE NECESSARY FOR THE
                        / PROGRAM CALLED SEGMENT TO PERFORM DATA
                        / ACCESSES ON LOCATION 0.  DEFINING
                        / DAZERO=<ANY NON-ZERO VALUE> WILL
                        / PERMIT SUCH ACCESSES. NOTE THAT IT DOES
```

```
/ NOT PERMIT USE OF LOC. 0 AS A
/ TRANSFER VECTOR OR AS THE OBJECT OF
/ A JMP,JMS OR XCT.
        / SUPPLY DEFAULT
        .IFUND DAZERO   ;DAZERO=0;   .ENDC

/
/
/
/
/
/
/
/
/ ADDRESS BOUND CHECKING REQUIRES:
/   LO <= ADDR <= HI
/ I.E.   0 <= (ADDR-LO)
/ AND: (ADDR-HI) <= 0
/ OR (ADDR-LO)+LO-HI <= 0
/
/  EFFICIENCY CAN BE IMPROVED BUT
/ DEBUGGING MADE MORE DIFFICULT IF
/ INSTEAD OF LO,HI THEMSELVES WE
/ STORE -LO,+LO-HI.
/ THE FOLLOWING MACRO IS EFFECTIVELY
/  -> GOTO UNLESS LO <= AC <= HI
/
        .IFDEF DEBUG
        .DEFIN  IFNBET LO,HI,GOTO
    CMA; DAC TMP
    TAD LO; SMA; GOTO
    LAC HI; TAD TMP; TAD (1); SPA; GOTO
        .ENDM
        .ENDC
/ ELSE
        .IFUND DEBUG
        .DEFIN  IFNBET LO,HI,GOTO
    TAD LO; SPA; GOTO
    TAD HI; SMA!SZA; GOTO
        .ENDM
/
/ THIS MACRO MODIFIES LO,HI FROM THEIR
/ ACTUAL VALUES TO THOSE REQUIRED
/ BY THE FAST VERSION OF 'IFNBET'
        .DEFIN  MODIFY,LO,HI
        / HI =+LO-HI
    LAC LO; CMA; TAD HI; CMA
    DAC HI
        / LO = -LO
    LAW -1; TAD LO; CMA; DAC LO
        .ENDM
        .ENDC
/
        .DEFIN  TEXT,STRING
    JMS PRTEXT
```

```
        .SIXBT '@STRING@'
            .ENDM
/
/
/
/
/
/
/ THERE ARE TWO CALLS ON THIS MACRO, ONE
/ BEFORE AND ONE AFTER LABEL 'BEGIN'.  DIFFERENT
/ EXPANSIONS ARE REQUIRED IN THE TWO PLACES.
/ SINCE 'BEGIN' IS UNDEFINED AT THE FIRST POINT OF CALL,
/ IT IS NECESSARY TO SUPPLY A DUMMY VALUE FOR THE FIRST
/ PASS TO AVOID PHASE ERRORS.
/ ** NOTE THAT ALL ABORT CALLS
/ GO VIA TAG '%ABORT'.  IF INT9 IS LOADED
/ AS A FREE-STANDING UNIT WITH DDT,
/ A BREAK-POINT ON THIS TAG WILL CATCH
/ THESE CALLS AFTER THE 'RUN TERMINATED'
/ MESSAGE FROM INT9. **
/
            .DEFIN ABORT
            .IFUND BEGIN  ;BEGIN=.+1; .ENDC / SET PASS 1 IN
        .IFPOZ .-BEGIN    / WHICH CALL IS IT?
        CAF; ION
%ABORT=.;   .EXIT         / GO ENTER BOOTSTRAP
        .ENDC
        .IFNEG .-BEGIN
        JMP %ABORT-2
        .ENDC
            .ENDM
            .EJECT
```

```
                                / THIS AREA INITIALLY HOLDS THE ONCE-ONLY
                                / SETUP ROUTINE APPEARING BELOW.
                                /
                                / SUBSEQUENTLY IT IS USED AS A 80(10) WORD
                                / CIRCULAR BUFFER WHICH HOLDS A RECORD
                                / OF THE LAST 20 MEMORY REFERENCE INSTRUCTIONS
                                / EXECUTED, TO BE PRINTED IF AND WHEN
                                / A FAILURE CONDITION IS DETECTED.
                                  /
00000 R 600060 R    INT9        JMP ERR1        / MUST ENTER BY 'JMS' NOT 'JMP'
00001 R 703302 A                CAF             / OVERWRITTEN BELOW
                    /
                                / SET UP EXTERNAL LINKAGES EITHER FOR
                                / STAND-ALONE USE OR AS PART OF EXTENDED
                                / MONITOR PACKAGE
                                .IFUND EMOP; .GLOBL INT9; .ENDC

                                .IFDEF EMOP; .GLOBL %I9  ;%I9=INT9;  .ENDC

        000000 R

                                /
        000002 R    STK=.       / START OF BUFFER AREA
                    /
                    /
                                / SET UP SIMULATED PC
00002 R 200000 R                LAC INT9; AND (77777); DAC PC
00003 R 500734 R
00004 R 040126 R
                                / REMEMBER CALLING ADDRESS
00005 R 340054 R                TAD MINUS1; DAC CALLER
00006 R 040130 R
00007 R 220043 R                LAC* LIT2; DAC BB  / DEFINE LOWER AND ...
00010 R 040122 R
00011 R 200044 R                LAC LIT3; DAC TB  / ... TOP LIBRARY AREA BOUND
00012 R 040123 R
00013 R 200045 R                LAC SIZLIT; DAC BC    / DEFINE LOWER AND ...
00014 R 040124 R
00015 R 220042 R                LAC* LIT1; DAC TC  / ... TOP USER AREA BOUND
00016 R 040125 R
                                .IFUND DEBUG
                                / MODIFY VALUES IN BC,TC,BB,TB IF
                                / THOSE REQUIRED BY THE FAST VERSION
                                / OF 'IFNBET'.
                                MODIFY BC,TC
                                MODIFY BB,TB
                                .ENDC
00041 R 600046 R                JMP SIZLIT+1
00042 R 000100 A    LIT1        .SCOM       ;.SCOM=100
        000100 A
00043 R 000103 A    LIT2        .SCOM+3
00044 R 777777 R    LIT3        INT9-1
00045 R 000764 R    SIZLIT      .SIZE
                    /
                    /
```

```
00046 R 750004 A                LAS; DAC SW     / REMEMBER AC. SWITCHES
00047 R 040127 R
00050 R 200735 R                LAC (STK-1); DAC* (STP)
00051 R 060736 R
00052 R 200057 R                LAC LIT4; DAC INT9+1    / CAN'T BE RE-ENTERED
00053 R 040001 R
00054 R 777777 A    MINUS1      LAW -1; ION; JMP AC
00055 R 700042 A
00056 R 600131 R
00057 R 620000 R    LIT4        JMP* INT9
                    /
00060 R 700002 A    ERR1        IOF; CAF
00061 R 703302 A
                                TEXT <%INT9 CALLED AS MAIN PGM@>
                                ABORT
                    /
                    /
                                .IFNOZ  STK+120-.    / IF <= ZERO
                    ****** SETUP ROUTINE TOO BIG.  MORE THAN 120 OCTAL LOCAT
                                .END
                                .ENDC
00122 R                         .LOC STK+120
                    /
                    /
                    / THE ORDER OF THE FOLLOWING VARIABLES IS SIGNIFICANT
00122 R 000000 A    BB;TB;BC;TC;PC
00123 R 000000 A
00124 R 000000 A
00125 R 000000 A
00126 R 000000 A
                    /
00127 R 000000 A    SW;CALLER
00130 R 000000 A
                    /
00131 R 060017 A    AC          DAC* STP     / FINAL BIT OF SETUP ROUTINE
00132 R 220736 R    LINK        LAC* (STP)
00133 R 240737 R    IONCNT      XOR (STK+117)
00134 R 750201 A    EFADDR      SZA!CLA!CMA
00135 R 600131 R    CI          JMP .-4
                    /INITIALISE STATUS INDICATORS
00136 R 140136 R    DBRFLG      DZM DBRFLG
00137 R 140133 R    SELECT      DZM IONCNT
00140 R 140132 R    TMP         DZM LINK
                    /    *** END OF SETUP ROUTINE ***
                                .EJECT
```

```
                        /    *** INTERPRETER PROPER STARTS HERE ***
00141 R 200133 R    BEGIN   LAC IONCNT; SMA!LAS; JMP .+4
00142 R 750104 A
00143 R 600147 R
00144 R 440133 R            ISZ IONCNT; SKP; ION
00145 R 741000 A
00146 R 700042 A
                            / CHECK AC. SWITCHES:- ABORT IF CHANGED
00147 R 240127 R            XOR SW; SWTEST; JMP ERROR+1
00150 R 741100 A
00151 R 600461 R
00152 R 220126 R            LAC* PC
00153 R 040135 R    XCTXCT  DAC CI; DAC TMP  /  TMP USED BY OPR & IOT
00154 R 040140 R
00155 R 500740 R            AND (OPR)
                            / TREAT OPR(74) AND IOT(70) AS SPECIAL CASES
00156 R 540740 R            SAD (OPR); JMP XCTOPR
00157 R 600323 R
00160 R 540741 R            SAD (IOT); JMP XCTIOT
00161 R 600341 R
00162 R 140137 R            DZM SELECT          / SWITCH TABLE SELECTOR
00163 R 200126 R            LAC PC; IFNBET  BC,TC,<JMP NOTINC>

00172 R 777762 A            LAW TABLE1-TABLE2; DAC SELECT
00173 R 040137 R
00174 R 200126 R    NOTINC  LAC PC; AND (060000); DAC TMP
00175 R 500742 R
00176 R 040140 R
                            / EFADDR = (PC & 0600000) + (CI & 17777)
00177 R 200135 R            LAC CI; AND (17777); TAD TMP; DAC EFADDR
00200 R 500743 R
00201 R 340140 R
00202 R 040134 R
00203 R 200135 R            LAC CI; AND (020000); SNA; JMP*DREF
00204 R 500744 R
00205 R 741200 A
00206 R 600237 R
                            / THE CODE BETWEEN HERE AND 'DREF'
                            / SIMULATES THE DEFER CYCLE OF INDIRECT
                            / MEMORY REFERENCE INSTRUCTIONS, CHECKING
                            / CHECKING ADDRESS BOUNDS IF APPROPRIATE.
00207 R 200137 R            LAC SELECT; SNA; JMP INB
00210 R 741200 A
00211 R 600221 R
00212 R 200134 R            LAC EFADDR; IFNBET BC,TC,<JMP ERROR>

                            / CHECK FOR FIRST INDIRECT REFERENCE
                            / AFTER A DBR. IF THIS IS IT, RESET
                            / LINK FROM SIGN-BIT OF TRANSFER
                            / VECTOR. CHECKED ELSEWHERE FOR
                            / ANYTHING OTHER THAN XCT OR JMP*
                            / FOLLOWING THE DBR.
00221 R 440136 R    INB     ISZ DBRFLG; JMP NODBR  / JUST HAD A DBR?
00222 R 600233 R
                            / YES: IS THIS, THE FIRST INDIRECT
```

```
                                        / REFERENCE AFTER THE SAID DBR, A JMP* ?
00223 R 760000 A                        LAW 760000; AND CI
00224 R 500135 R
00225 R 540745 R                        SAD (JMP*); SKP; JMP ERROR
00226 R 741000 A
00227 R 600460 R
00230 R 220134 R                        LAC* EFADDR; AND (400000); DAC LINK
00231 R 500746 R
00232 R 040132 R
00233 R 140136 R    NODBR               DZM DBRFLG    / SO'S NO OVERFLOW PROBLEMS
00234 R 220134 R                        LAC* EFADDR; AND (77777); DAC EFADDR
00235 R 500734 R
00236 R 040134 R
                                        / END OF 'DEFER' CYCLE
00237 R 200135 R    DREF                LAC CI; AND (740000)
00240 R 500740 R
00241 R 744010 A                        CLL!RAL; RTL; RTL
00242 R 742010 A
00243 R 742010 A
00244 R 340747 R                        TAD (TABLE2)
00245 R 340137 R                        TAD SELECT
00246 R 040140 R                        DAC TMP; JMP* TMP
00247 R 620140 R
                        /
                        /
                        /  DISPATCH TABLES FOR MEMORY REFERENCE INSTRUCTIONS
                        / IF PC IS IN USER AREA(ZONE C), TABLE1 IS USED
                        / OTHERWISE TABLE2
                        /
00250 R 600365 R    TABLE1              JMP C00
00251 R 600304 R                        JMP C04
00252 R 600377 R                        JMP C10
00253 R 600304 R                        JMP C14
00254 R 600304 R                        JMP C20
00255 R 600304 R                        JMP C24
00256 R 600304 R                        JMP C30
00257 R 600304 R                        JMP C34
00260 R 600427 R                        JMP C40
00261 R 600304 R                        JMP C44
00262 R 600304 R                        JMP C50
00263 R 600304 R                        JMP C54
00264 R 600445 R                        JMP C60
00265 R 600460 R                        JMP C64
                        /
00266 R 600374 R    TABLE2              JMP C00A
00267 R 600315 R                        JMP C04A
00270 R 600416 R                        JMP C10A
00271 R 600315 R                        JMP C14A
00272 R 600315 R                        JMP C20A
00273 R 600315 R                        JMP C24A
00274 R 600315 R                        JMP C30A
00275 R 600315 R                        JMP C34A
00276 R 600436 R                        JMP C40A
00277 R 600315 R                        JMP C44A
00300 R 600315 R                        JMP C50A
```

```
00301 R 600315 R                JMP C54A
00302 R 600454 R                JMP C60A
00303 R 600460 R                JMP C64A
                        /
                        /
                        /
                        / EXECUTE NON-BRANCHING MEMORY REFERENCE INSTRUCTION
00304 R 200134 R        XCTMR   LAC EFADDR
                                .IFNZR  DAZERO  / PERMIT TO ACCESS LOC. 0?
00305 R 741200 A                SNA; JMP XCTMRA / YES ->
00306 R 600315 R
                                .ENDC
                                IFNBET BC,TC,<JMP ERROR>
                                 / ENTER AT 'XCTMRA' IF 'EFADDR' IS NOT
                                 / TO BE CHECKED
00315 R 100620 R        XCTMRA  JMS SAVE
                                 / MAKE UP <FN>* EFADDR THEN XCT IT
00316 R 200135 R                LAC CI; AND (OPR)
00317 R 500740 R
00320 R 340750 R                TAD (CAL* EFADDR); DAC TMP; JMP DO
00321 R 040140 R
00322 R 600326 R
00323 R 200136 R        XCTOPR  LAC DBRFLG; SZA; JMP ERROR  / DBR OUTSTANDING?
00324 R 740200 A
00325 R 600460 R
00326 R 200132 R        DO      LAC LINK; RAL; LAC AC
00327 R 740010 A
00330 R 200131 R
                                / 'TMP' CONTAINS CURRENT INSTRUCTION. SET
                                / ABOVE FOR MEMORY REF. CODES AND AT 'XCTXCT+1'
                                / OPERATE AND IOT GROUPS
00331 R 400140 R                XCT TMP; JMP DONE  / JUMP IF NO SKIP
00332 R 600334 R
00333 R 440126 R                ISZ PC      / SIMULATE SKIP
                        /
00334 R 040131 R        DONE    DAC AC; CLA!RAR; DAC LINK
00335 R 750020 A
00336 R 040132 R
00337 R 440126 R        NXTCYC  ISZ PC; JMP BEGIN
00340 R 600141 R
                        /
                        /
                        /
                        / THE PROGRAM INTERRUPT IS TURNED OFF
                        / FOR ONE PSEUDO-MACHINE CYCLE AFTER EACH
                        / IOT.  AT 'BEGIN', 'IONCNT' IS INCREMENTED
                        / IF NEGATIVE AND THE INTERRUPT
                        / ENABLED WHEN 'IONCNT' GOES POSITIVE.
                        / IF SET NON-NEGATIVE, IT IS NEVER
                        / INCREMENTED.
                        / 'IOF' AND 'ION' ARE SPECIAL CASES
                        / BECAUSE FOR THE FORMER, THE INTERRUPT MUST
                        / NOT BE RE-ENABLED AT THE END OF THE
                        / CURRENT CYCLE, AND FOR THE LATTER IT MUST
                        / ALWAYS BE RE-ENABLED IRRESPECTIVE OF IT'S
```

```
                        / PREVIOUS STATE.  IONCNT > 0 MARKS EXPLICIT
                        / IOF WHICH CAN ONLY BE CANCELLED BY AN
                        / EXPLICIT ION.
                        / NOTE THAT:
                        /       IONCNT = -(N+1)
                        / CAN BE CONSTRUED AS A PROMISE TO
                        / RE-ENABLE THE INTERRUPT AFTER  N
                        / PSEUDO-MACHINE CYCLES.
                        /
00341 R 700002 A        XCTIOT  IOF; LAC IONCNT
00342 R 200133 R
00343 R 741300 A                SPA!SNA; LAW -2; DAC IONCNT
00344 R 777776 A
00345 R 040133 R
00346 R 200135 R                LAC CI
00347 R 540751 R                SAD (ION); JMP XCTION
00350 R 600362 R
00351 R 540752 R                SAD (IOF); JMP XCTIOF
00352 R 600360 R
00353 R 540753 R                SAD (DBR); SKP!CLA!CMA; JMP XCTOPR
00354 R 751001 A
00355 R 600323 R
00356 R 040136 R        XCTDBR  DAC DBRFLG      / SET MARKER FOR NEXT CYCLE
00357 R 600337 R                JMP NXTCYC  / START NEXT CYCLE.
00360 R 200754 R        XCTIOF  LAC (1); SKP
00361 R 741000 A
00362 R 777776 A        XCTION  LAW -2; DAC IONCNT
00363 R 040133 R
00364 R 600337 R                JMP NXTCYC
                        /
                        /
                        /
                        /
                        /           *** SIMULATION ROUTINES ARE DEFINED HERE ***
                        / CAL
00365 R 200126 R        C00     LAC PC; TAD (1); DAC TMP
00366 R 340754 R
00367 R 040140 R
00370 R 220140 R                LAC* TMP; SAD (15); JMP* PC  /;CAL IF .EXIT
00371 R 540755 R
00372 R 620126 R
00373 R 600460 R                JMP ERROR               / OTHERWISE IT'S AN ERROR
                                / IN ZONE B, TREAT 'CAL' AS 'JMS 20'
                                / NOTE THAT THIS WON'T WORK CORRECTLY
                                / WITH API ENABLED
00374 R 200756 R        C00A    LAC (20); DAC EFADDR
00375 R 040134 R
00376 R 600416 R                JMP C10A
                        /
                        /
                        / DAC
                000304 R        C04=XCTMR;C04A=XCTMRA
                000315 R
                        /
                        /
```

```
                        / JMS
00377 R 200134 R    C10     LAC EFADDR; IFNBET BC,TC,<JMP .+C3>

00406 R 600416 R            JMP C10A
00407 R 200134 R    TC3     LAC EFADDR; IFNBET BB,TB,<JMP ERROR>

00416 R 100620 R    C10A    JMS SAVE
00417 R 200126 R            LAC PC; TAD (1); XOR LINK; DAC* EFADDR
00420 R 340754 R
00421 R 240132 R
00422 R 060134 R
00423 R 200134 R            LAC EFADDR; TAD (1); DAC PC
00424 R 340754 R
00425 R 040126 R
00426 R 600141 R            JMP BEGIN
                        /
                        /
                        / DZM
          000304 R      C14=XCTMR;C14A=XCTMRA
          000315 R
                        /
                        /
                        / LAC
          000304 R      C20=XCTMR;C20A=XCTMRA
          000315 R
                        /
                        /
                        / XOR
          000304 R      C24=XCTMR;C24A=XCTMRA
          000315 R
                        /
                        /
                        / ADD
          000304 R      C30=XCTMR;C30A=XCTMRA
          000315 R
                        /
                        /
                        / TAD
          000304 R      C34=XCTMR;C34A=XCTMRA
          000315 R
                        /
                        /
                        / XCT
00427 R 200134 R    C40     LAC EFADDR; IFNBET BC,TC,<JMP ERROR>

00436 R 100620 R    C40A    JMS SAVE
00437 R 700002 A            IOF     / XCT CYCLE IS NON-INTERRUPTIBLE
00440 R 200133 R           LAC IONCNT; SPA!SNA!CLA!CMA; DAC IONCNT
00441 R 751301 A
00442 R 040133 R
00443 R 220134 R            LAC* EFADDR
00444 R 600153 R            JMP XCTXCT
                        /
                        /
                        / ISZ
```

```
                000304 R        C44=XCTMR;C44A=XCTMRA
                000315 R
                                /
                                /
                                / AND
                000304 R        C50=XCTMR;C50A=XCTMRA
                000315 R
                                /
                                /
                                / SAD
                000304 R        C54=XCTMR;C54A=XCTMRA
                000315 R
                                /
                                /
                                / JMP
00445 R 200134 R        C60     LAC EFADDR; IFNBET BC,TC,<JMP ERROR>

00454 R 100620 R        C60A    JMS SAVE
00455 R 200134 R                LAC EFADDR; DAC PC
00456 R 040126 R
00457 R 600141 R                JMP BEGIN
                                /
                                /
                                / EAE
                000460 R        C64=.;C64A=.    / I.E.  ERROR
                000460 R
                                /
                                /
                                /
                                /
                                /
00460 R 100620 R        ERROR   JMS SAVE
00461 R 700002 A                IOF
                                TEXT <!  FAILURE%%START ADDR=@>
00473 R 200130 R                LAC CALLER; JMS POCT
00474 R 100654 R
                                TEXT <%% ADDR.   INSTR.   (I,V.)  CONTENTS%@>
00513 R 777754 A                LAW -24; DAC CI
00514 R 040135 R
                                / USE CI AS WORK SPACE:- 24(8)=-20(10)
00515 R 220736 R        REPT    LAC* (STP); SAD (STK+117); LAC*(STK-1)
00516 R 540737 R
00517 R 200735 R
00520 R 060736 R                DAC* (STP)              / BUFFER IS CIRCULAR
00521 R 220017 A                LAC* STP; SMA; JMP ER1  / -1 IF NOT YET SET
00522 R 740100 A
00523 R 600530 R
00524 R 220017 A                LAC* STP  / SKIP:- IT'S AN AUTO-INDEX
00525 R 220017 A                LAC* STP; LAC* STP
00526 R 220017 A
00527 R 600560 R                JMP ONO
00530 R 100654 R        ER1     JMS POCT    / INSTRUCTION ADDR:
                                TEXT <   @>
00534 R 220017 A                LAC* STP; JMS POCT  / INSTRUCTION WORD
00535 R 100654 R
```

```
                                    TEXT <    @>
00541 R 220017 A                    LAC* STP; SMA; JMP REALTV
00542 R 740100 A
00543 R 600551 R
                                    TEXT <       @>
00550 R 741000 A                    SKP
00551 R 100654 R    REALTV          JMS POCT     / TRANSFER VECTOR =
                                    TEXT <    @>
00555 R 220017 A                    LAC* STP; JMS POCT    / ADDRESSED CELL
00556 R 100654 R
00557 R 100726 R                    JMS CRLF
00560 R 440135 R    ONO             ISZ CI; JMP REPT
00561 R 600515 R
                                    TEXT <%PC,LINK,AC=@>
00570 R 200126 R                    LAC PC; JMS POCT; LAW 40; JMS PSYM
00571 R 100654 R
00572 R 760040 A
00573 R 100640 R
00574 R 744000 A                    CLL; LAC LINK; RTL; TAD (60)
00575 R 200132 R
00576 R 742010 A
00577 R 340757 R
00600 R 100640 R                    JMS PSYM          / PRINT LINK AS '0' OR '1'
00601 R 760040 A                    LAW 40; JMS PSYM    / PRINT SPACE
00602 R 100640 R
00603 R 200131 R                    LAC AC; JMS POCT    /  THEN AC.
00604 R 100654 R
                                    TEXT <%RUN TERMINATED%@>
                                    ABORT
                    /
                    /
                    /
                    / *** SUBROUTINES FOLLOW ***
                    /
00620 R 000000 A    SAVE
00621 R 220736 R                    LAC* (STP); SAD (STK+117); LAC (STK-1)
00622 R 540737 R
00623 R 200735 R
00624 R 060736 R                    DAC* (STP)
00625 R 200126 R                    LAC PC; DAC* STP
00626 R 060017 A
00627 R 200135 R                    LAC CI; DAC* STP
00630 R 060017 A
00631 R 500744 R                    AND (020000); SZA!CLA!CMA; LAC EFADDR
00632 R 750201 A
00633 R 200134 R
                                    / STORE 777777 OR T.V.
00634 R 060017 A                    DAC* STP
00635 R 220134 R                    LAC* EFADDR; DAC* STP    / SAVE CONTENT
00636 R 060017 A
00637 R 620620 R                    JMP* SAVE
                    /
                    /
00640 R 000000 A    PSYM
00641 R 700401 A                    TSF; JMP PS2    / WAIT IF TT FLAG ISN'T SET
```

```
00642 R 600647 R
00643 R 700406 A     PS1        TLS; TSF; JMP .-1
00644 R 700401 A
00645 R 600644 R
00646 R 620640 R                JMP* PSYM
00647 R 140653 R     PS2        DZM TIMOUT; ISZ TIMOUT; JMP .-4
00650 R 440653 R
00651 R 600650 R
00652 R 600643 R                JMP PS1
00653 R 000000 A     TIMOUT
          700406 A   TLS=700406;TSF=700401
          700401 A
                     /
                     /
00654 R 000000 A     POCT
00655 R 040673 R                DAC P80
00656 R 777772 A                LAW -6; DAC P81
00657 R 040674 R
00660 R 200673 R     P8L        LAC P80; CLL!RAL; RTL; DAC P80
00661 R 744010 A
00662 R 742010 A
00663 R 040673 R
00664 R 740010 A                RAL; AND (7); TAD (60)
00665 R 500760 R
00666 R 340757 R
00667 R 100640 R                JMS PSYM
00670 R 440674 R                ISZ P81; JMP P8L
00671 R 600660 R
00672 R 620654 R                JMP* POCT
00673 R 000000 A     P80;P81
00674 R 000000 A
                     /
                     /
00675 R 000000 A     PRTEXT
00676 R 777775 A     TL1        LAW -3; DAC TCNT
00677 R 040673 R
00700 R 220675 R                LAC* TP; DAC TS; ISZ TP
00701 R 040674 R
00702 R 440675 R
00703 R 200674 R     TL2        LAC TS; SMA!CLL; CLL!CML  / BIT-7-TO-BE
00704 R 744100 A
00705 R 744002 A
00706 R 742010 A                RTL; RTL; RTL
00707 R 742010 A
00710 R 742010 A
00711 R 040674 R                DAC TS
00712 R 740010 A                RAL; AND (177)
00713 R 500761 R
00714 R 540762 R                SAD (100); JMP* PRTEXT  / 6-BIT ZERO TERMINATES
00715 R 620675 R
00716 R 540763 R                SAD (45); JMP TL4  / '%' -> NEWLINE
00717 R 600724 R
00720 R 100640 R                JMS PSYM
00721 R 440673 R     TL3        ISZ TCNT; JMP TL2
00722 R 600703 R
```

```
00723 R 600676 R                JMP TL1
00724 R 100726 R     TL4        JMS CRLF; JMP TL3
00725 R 600721 R
        000673 R     TCNT=P80;TS=P81;TP=PRTEXT
        000674 R
        000675 R
                     /
                     /
00726 R 000000 A     CRLF
00727 R 760015 A                LAW 15; JMS PSYM
00730 R 100640 R
00731 R 760012 A                LAW 12; JMS PSYM
00732 R 100640 R
00733 R 620726 R                JMP* CRLF
                     /
                     /
        000000 A                .END
00734 R 077777 A *L
00735 R 000001 R *L
00736 R 000017 A *L
00737 R 000121 R *L
00740 R 740000 A *L
00741 R 700000 A *L
00742 R 060000 A *L
00743 R 017777 A *L
00744 R 020000 A *L
00745 R 620000 A *L
00746 R 400000 A *L
00747 R 000266 R *L
00750 R 020134 R *L
00751 R 700042 A *L
00752 R 700002 A *L
00753 R 703344 A *L
00754 R 000001 A *L
00755 R 000015 A *L
00756 R 000020 A *L
00757 R 000060 A *L
00760 R 000007 A *L
00761 R 000177 A *L
00762 R 000100 A *L
00763 R 000045 A *L
            SIZE=00764      NO ERROR LINES
```