DECUS
PROGRAM LIBRARY

DECUS NO.	8-587
TITLE	FORTRAN-D 4K OVERLAYINGS TO CHAIN PROGRAMS
AUTHOR	Leopoldo Bertacche
COMPANY	Submitted by: Alessandro Zanon Instituto Nazionale Fisica Nucleare (INFN) Legnaro (Padova), Italy
DATE	October 12, 1972
SOURCE LANGUAGE	PAL-D

FORTRAN-D 4K OVERLAYINGS TO CHAIN PROGRAMS

ABSTRACT

The overlayings add five new statements to Fortran-D 4K supplied by Digital Equipment Corporation.
This new version of Fortran is designed to segment large programs and/or to use a large amount of data.

EXPLANATION OF NEW STATEMENTS AND THEIR USE

The first two statements "CALL N" and "RETURN" consent the use of subroutines (see DECUSCOPE, Vol. 9, No. 3, page 6, by Willen L.).

The "CHAIN NAME" statement is used to chain the current segment with the segment called "NAME".
The filname "NAME" has the same restrictions imposed by DECtape Monitor System.
These overlayings allow to use the whole DECtape No. 7 to store a large amount of data. The DECtape is not used in the usual file oriented way for I/O of data and the same location (of DECtape) may be used whether by "WRITE" or by "READ" statements.

The "REWIND" statement is used to read again the data beginning from the first block. The data are stored in the DECtape in sequential way (not directly addressable).

The "BLOCK OUT" statement is used whenever we wish to read data immediately after their writing; this statement is necessary because the I/O routines transfer only full blocks of data, from tape to I/O buffers (and vice versa).

To write and use chaining programs, for every segment, proceed in this way:

1) Use Edit to create the source segment.

   Remember that:

   a) the first instruction of every segment must be "DEFINE DECTAPE";

   b) the "DIMENSION" statement allows to create a COMMON area between the chaining segments. This statement must precede the first executable instruction in every segment and the common variable (integer, floating or subscritted) must be written on the same sequence;

   c) the "STOP" statement must be used only on the last segment; or if an error condition occurs;

   d) the "REDWIN" statement "READ 3, f, list" must precede the first instruction of the program if we wish to read data written by the program.

2) Use FORTRAN-D Compiler to convert the source into object segment. The filname must be the same employed on "CHAIN" statement of calling segment.

3) To execute Fortran program, call FOSL as usual; reply to its requests of input with the name of the first segment of the program, and then to the option request with "S".

   EXAMPLE:

```
. FOSL
* IN-S:        (filname of the first segment)
*
* OP-S
↑              (↑P Typed here)
```

The FOSL loads the first segment and executes it.
The next segment, if there is one, is loaded and started

automatically when the "CHAIN" statement is met.
If the segment is not in the system, FOSL answers with the unfound name followed by a question mark.

EXAMPLE OF USE

If a program is divided in two segments called "PRW1" and "PRW2" in the following form:

Segment PRW1

```
DEFINE DECTAPE
DIMENSION A(3) , FXQ(15) , BP(27)
      .
      .
      .
      .
      .
      .
CHAIN PRW2
END
```

Segment PRW2

```
DEFINE DECTAPE
DIMENSION B(3), AQZ(15) , PR(27)
      .
      .
      .
      .
      .
      .
STOP
END
```

to execute all the Fortran program it is necessary to load the first segment (PRW1) with the FOSL.
Infact, during the execution of PRW1 when the "CHAIN PRW2" statement is met, the segment PRW2 is loaded and then executed.

The "DIMENSION" statement creates a common area for the variable (A and B; FXQ; BP and PR).

The use of the overlayings and of the new statements is demonstrated by the program which fits a Gaussian to set of experimental data, put as example.

```
.EDIT
*OUT-T:
*
*IN-S:GAUS
*
*OPT-B

*R

*L
C       EXAMPLE PROGRAM TO USE NEW STATMENTS.

C       TO FIT A GAUSSIAN TO A SET OF DATA
C       READ DATA FROM THE TTY AND STORE THEM
C       ON THE DECTAPE NR. 7 .
C       THIS SEGMENT IS CALLED  "GAUS"

        DEFINE DECTAPE
        DIMENSION NC(1),A(15)
        DO 10 I=1,15
        A(I)=0
10      CONTINUE
        TYPE 200
        ACCEPT 230,NC
        TYPE 220
        DO 20 I=1,NC
        TYPE 210
        ACCEPT 240,X,Y
        WRITE 3,240,X,Y
20      CONTINUE
        BLOCK OUT
        REWIND
        DO 40 J=1,NC
        READ 3,240,X,Y
        DO 30 I=4,14,5
        ESP=(I-4)/5
        CALL 100
        A(I)=A(I)+Y*LOGF(Y)*W
30      CONTINUE
        DO 40 K=1,11,5
        DO 40 I=0,2
        ESP=I+(K-1)/5
        CALL 100
        A(I+K)=A(I+K)+Y*W
40      CONTINUE
        TYPE 210
        CHAIN GAU1


C       SUBROUTINE

100     W=1
        S=1
        IF (X) 110,140,120
110     S=-1
120     W=ABSF(X)**ESP
        L=ESP/2.
        B=L*2
        IF (B-ESP) 130,140,130
130     W=W*S
140     RETURN
```

```
200       FORMAT (/,"FIT TO A GAUSSIAN",/,/"NUMBER OF CHANNELS=")
210       FORMAT (/)
220       FORMAT (/,/,"X  Y",/,)
230       FORMAT (I)
240       FORMAT (E,E)

          END

*
.FORT
*OUT-S:GAUS
*
*IN-S:GAUS
*
↑
.EDIT
*OUT-T:
*
*IN-S:GAU1
*
*OPT-B

*R

*L
C         THIS SEGMENT CALLED " GAU1 " EVALUATES
C         SOLUTIONS OF SYSTEM WITH GAUSS-JORDAN

          DEFINE DECTAPE
          DIMENSION NC(1),A(15)
          DO 90 I=1,3
          P=0
          DO 30 J=0,2
          IF (A((J+1)*5)) 10,10,30
10        DO 30 K=1,3
          IF (A(J*5+K)*A(J*5+K)-P*P) 30,30,20
20        JP=J
          KQ=K
          P=A(J*5+KQ)
30        CONTINUE
          A((JP+1)*5)=KQ
          IF (P) 50,40,50
40        TYPE 200
          STOP
50        DO 60 K=1,4
          A(JP*5+K)=A(JP*5+K)/P
60        CONTINUE
          DO 90 J=0,2
          P=A(J*5+KQ)
          IF (J-JP) 70,90,70
70        DO 80 K=1,4
          A(J*5+K)=A(J*5+K)-A(JP*5+K)*P
80        CONTINUE
          A(J*5+KQ)=0
90        CONTINUE
          DO 100J=5,15,5
          K=A(J)
          A(K)=A(J-1)
100       CONTINUE
          CHAIN GAU2
200       FORMAT (/,"NOT SOLUTIONS",/)
          END
```



```
*
```

```
.FORT
*OUT-S:GAU1
*
*IN-S:GAU1
*
↑
.EDIT
*OUT-T:
*
*IN-S:GAU2
*
*OPT-B

*R

*L
C       THIS SEGMENT CALLED "GAU2" EVALUATES
C       THE PARAMETERS OF THE GAUSSIAN
C        CENTER
C        WIDTH
C        AMPLITUDE
C        AREA
C       THEN THE VALUE "FX" OF FUNCTION AT THE
C       "X" COORDINATE OF EACH DATA POINT IS CALCULATED.

C       AT LEAST IT EVALUATES THE RATIO OF CHI-SQUARE
C       WITH THE DEGREES OF FREEDOM

        DEFINE DECTAPE
        DIMENSION NC(1),A(15)
        PK=-A(3)
        CENT=-A(2)*.5/A(3)
        SK=-A(3)*CENT*CENT+A(1)
        AMPL=EXPF(SK)
        WIDT=SQTF(-.5/A(3))
        AREA=AMPL*WIDT*SQTF(2.*3.14159)
        TYPE 200,CENT,WIDT,AMPL,AREA
        REWIND
        CHSQ=0
        DO 10 J=1,NC
        READ 3,210,X,Y
        FX=AMPL*EXPF(-PK*(X-CENT)*(X-CENT))
        TYPE 220,X,FX
        CHSQ=RCHSQ+(Y-FX)*(Y-FX)/Y
10      CONTINUE
        DF=NC-3
        CHDF=CHSQ/DF
        TYPE 230,CHSQ,DF,CHDF
        STOP
200     FORMAT (/,"CENTER        = ",E,/,"WIDTH  = ",E,/'
        "AMPLI. = ",E,/,"AREA    = ",E,/,/)
210     FORMAT (E,E)
220     FORMAT (/," X= ",E,"     FX= ",E)
230     FORMAT (/,/,"CHI-SQUARE   =",E,/,"DEGREES OF FREEDOM= ",E,/'
        "CHI-SQUARE/DEGREES OF FREEDOM =",E,/)
        END

*
```

```
.FORT
*OUT-S:GAU2
*
*IN-S:GAU2
*
↑
```

```
.FOSL
*IN-S:GAUS
*
*OPT-S
↑

FIT TO A GAUSSIAN

NUMBER OF CHANNELS=17

X          Y

Ø          6
1          11
2          17
3          25
4          34
5          43
6          52
7          58
8          61
9          58
1Ø         52
11         43
12         34
13         25
14         17
15         11
16         6
↑
↑

CENTER   =   Ø.8Ø4772E+1
WIDTH    =   Ø.371148E+1
AMPLI.   =   Ø.6Ø41Ø5E+2
AREA     =   Ø.562Ø17E+3
```

```
X=   0.000000E+0        FX=   0.575653E+1
X=   0.100000E+1        FX=   0.995682E+1
X=   0.200000E+1        FX=   0.160159E+2
X=   0.300000E+1        FX=   0.239585E+2
X=   0.400000E+1        FX=   0.333301E+2
X=   0.500000E+1        FX=   0.431210E+2
X=   0.600000E+1        FX=   0.518815E+2
X=   0.700000E+1        FX=   0.580508E+2
X=   0.800000E+1        FX=   0.604055E+2
X=   0.900000E+1        FX=   0.584544E+2
X=   0.100000E+2        FX=   0.526054E+2
X=   0.110000E+2        FX=   0.440267E+2
X=   0.120000E+2        FX=   0.342668E+2
X=   0.130000E+2        FX=   0.248030E+2
X=   0.140000E+2        FX=   0.166958E+2
X=   0.150000E+2        FX=   0.104516E+2
X=   0.160000E+2        FX=   0.608461E+1

CHI-SQUARE     = 0.628437E+1
DEGREES OF FREEDOM=  0.140000E+2
CHI-SQUARE/DEGREES OF FREEDOM = 0.448884E+0
!
.
```

```
                /COMPILER MODIFICATIONS TO DEFINE "RETURN";
                /"CALL"; "CHAIN"; "REWIND" AND "BLOCK OUT"
                /STATMENTS
                /
                /SYSTEM DEFINITONS:
                        PUNCH=4460
                        OVER31=6307
                        END=101
                        BEGN=100
                        NEWL=46
                        SVSRN=127
                        GOO=5200
                        TPIN=5600
                        *5365
5365  7440              SZA      /IS IT AN "0"?
5366  5374              JMP .+6 /NO
                        *5374
5374  1377              TAD C7
5375  5776              JMP I .+1  /SECOND CHARACTER IS "A" OR "H"?
5376  3770              3770
5377  0007  C7,         7
                        *3770
3770  7640              SZA CLA
3771  5373              JMP .+2  /SECOND CHARACTER IS "A"
3772  5777              JMP I .+5  /SECOND CHARACTER IS "H"
3773  1376              TAD .+3    /TAKE CALL=61
3774  5775              JMP I .+1  /JUMP TO THE NEXT HOLE FOR PATCH
3775  5173              5173       /ON 5173
3776  0061              61         /CALL=61
3777  4370              4370   /PATCH ON 4370 OVERLAY BLOCK FOR CHAIN
                        *4370
4370  1375              TAD C6111
4371  3774              DCA I RRS
4372  4527              JMS I Z SVSRN /CHAIN STATMENT ROUTINE
4373  0003              3
4374  6512  RRS,         6512
4375  6111  C6111,      6111       /STARTING ADDRESS OF ROUTINE
                        *6311      /OVERLAY BLOCK NUMBER 2
6311  1370  RCHAIN,     TAD R63    /TAKE CHAIN=63
6312  4460              PUNCH      /PUNCH NEXT INST. CHAIN
6313  1367              TAD C4      /PREPARE FOR SKIPPING 4 LETTERS
6314  4707              JMS I OVER31 /STEP 4-CHARACTER THRU STATMENT
6315  1100              TAD Z BEGN
6316  3101              DCA Z END
6317  4341  LIKK,       JMS LGET    /GET A CHARACTER
6320  7106              RTL CLL
6321  7006              RTL
6322  7006              RTL
6323  3371              DCA WORD1  /SAVE 1ST. LEFT HALF
6324  4341              JMS LGET
6325  1371              TAD WORD1
6326  3371              DCA WORD1  /SAVE 1ST. PACKED WORD
6327  4341              JMS LGET
6330  7106              RTL CLL
```

```
6331  7006          RTL
6332  7006          RTL
6333  3372          DCA WORD2  /SAVE SECOND LEFT HALF
6334  4341          JMS LGET
6335  1372          TAD WORD2
6336  3372          DCA WORD2  /SAVE SECOND PACKED WORD
6337  4341          JMS LGET  /LOOK FOR CR
6340  5353          JMP TOMN   /MORE THAN 4-CHARACTER
6341  0000  LGET,    0
6342  1501          TAD I Z END /GET A CHARACTER
6343  3373          DCA LCHAR    /SAVE IT
6344  2101          ISZ END
6345  1373          TAD LCHAR   /IS IT A CR?
6346  7450          SNA
6347  5362          JMP PUC     /YES: END OF NAME
6350  1374  LOK,    TAD P40      /CREATE STRIPPED
6351  0375          AND C77     /ASCII+40
6352  5741          JMP I LGET
6353  7200  TOMN,   CLA    /TOO MANY CHARACTERS
6354  1501          TAD I Z END
6355  3373          DCA LCHAR
6356  2101          ISZ END
6357  1373          TAD LCHAR
6360  7640          SZA CLA
6361  5353          JMP TOMN
6362  1371  PUC,    TAD WORD1
6363  4460          PUNCH
6364  1372          TAD WORD2
6365  4460          PUNCH
6366  5446          JMP I Z NEWL  /NEW LINE
6367  0004  C4,      4
6370  0063  R63,     63
6371  0000  WORD1,   0
6372  0000  WORD2,   0
6373  0000  LCHAR,   0
6374  0040  P40,     40
6375  0077  C77,     77
                    *5173
5173  4460          PUNCH  /PUNCH NEXT INST. CALL
5174  7240          CLA CMA /PREPARE FOR SKIPPING 3 LETTERS
5175  5776          JMP I .+1 /READ STATMENT NUMBER AS IN GO TO
5176  5207          GOO+7
                    *5145
5145  4561          IBIN  /OTHER OUTLET FOR IBIN
                    *5165
5165  4566          RETO  /OTHER OUTLET FOR READ AND RETURN
                    *4560
4560  1365  RETU,   TAD CRET /FORM RETURN=62
4561  1364  IBIN,   TAD CIBI  /FORM IBIN INSTR.=60
4562  4460          PUNCH      /PUNCH IT
4563  5446          JMP I NEWL  /NEW LINE
4564  0060  CIBI,   60       /=IBIN
4565  0002  CRET,   2         /RETURN=60+2
4566  2101  RETO,   ISZ ED
4567  2101          ISZ END  /SKIP OTHER 2 LETTERS
```

```
4570  1501          TAD I END  /TEST IF 3RD. LETTER IS "A" OR NOT
4571  1377          TAD M301   /-"A"
4572  7440          SZA
4573  5776          JMP I .+3 /TEST IF 3RD LETTER IS "T" OR "W"
4574  5775          JMP I .+1  /READ
4575  5600          TPIN       /COMPILE IT ON TPIN AS USUAL
4576  4170          4170
4577  7477  M301,    -301       /-A
                    *5635
5635  0002  C4A,     2          /SKIP OVER REMAINING 2 CHARACTERS
                    *4170
4170  1376          TAD M23
4171  7650          SNA CLA
4172  5775          JMP I .+3 /3RD CHARACTER IS "T"
4173  5774          JMP I .+1 /3RD CHARACTER IS "W"
4174  3171          3171       /ROUTINE TO COMPILE REWIND
4175  4560          4560       /ROUTINE TO COMPILE RETURN
4176  7755  M23,     -23
                    *3171
3171  1374  REWI,    TAD C57  /FORM REWIND=57
3172  4460          PUNCH    /PUNCH IT
3173  5446          JMP I NEWL   /NEW LINE
3174  0057  C57,     57
                    *4360
4360  1366          TAD C6056  /NEW ORIGIN
                    *4366
4366  6056  C6056,  6056
```

```
BEGN   0100
CIBI   4564
CRET   4565
C4     6367
C4A    5635
C57    3174
C6056  4366
C6111  4375
C7     5377
C77    6375
END    0101
GOO    5200
IBIN   4561
LCHAR  6373
LGET   6341
LIKK   6317
LOK    6350
M23    4176
M301   4577
NEWL   0046
OVER31 6307
PUC    6362
PUNCH  4460
P40    6374
RCHAIN 6311
RETO   4566
RETU   4560
```

```
REWI    3171
RRS     4374
R63     6370
SVSRN   0127
TOMN    6353
TPIN    5600
WORD1   6371
WORD2   6372

.LOAD
*IN-D7:.FT.,S:F02
*
*
ST=
↑↑↑
!SAVE .FT.!200-7377;0
.
```

```
                /FORTRAN D OPERATING SYSTEM LOADER
                /(FOSL) MODIFICATIONS TO SERVICE
                /"CHAIN"; "REWIND" AND "BLOCK OUT"
                /STATMENTS
                /
                /SYSTEM DEFINITIONS:
                TMES=560
                TTYOUT=514
                C200=73
                ILC=14
                IOERR=4000
                DNSRN=23
                SBSRN=24
                AX0=10
                AX1=11
                LTST=20
                LNDX=21
                BUFADR=43
                BUFOBJ=44
                FH2A=301
                OBJBLK=27
                MPNT=15
                LNKWD=42
                GTWD=41
                SYSIO=77
                MONRET=76
                GETSW1=743
                GETLNK=734
                INTI=151
                TEMB=16
                TEMC=17
                PUTBLK=754
                PUTLNK=756
                PUT=745
                PUTEOF=101
                PUTSW1=777
                DUMMY=476
                NULL=113
                SWCH=636
                TPWD=637
                PSW1=707
                TEMPZ=711
                        *205
0205  5476              JMP I Z MONRET./NO, RETURN TO MONITOR
                        *256
0256  1273  FH3,       TAD C705  /PREPARE FOR THE I/O
0257  3674              DCA I PTW
0260  1275              TAD C703
0261  3676              DCA I GTW
0262  7001              IAC    /GET BLOCK NR=1
0263  3442              DCA I LNKWD
0264  4441              JMS I GTWD /PRIME THE INPUT BUFFER
0265  7300              CLA CLL
0266  7001              IAC   /GET DUMMY LOCATION
```

```
0267  3677          DCA I DUMW
0270  1300          TAD C2  /SET LINK FOR NEXT BLOCK
0271  3442          DCA I LNKWD
0272  5301          JMP FH2A
0273  0705  C705,    705
0274  5353  PTW,     753+4400
0275  0703  C703,    703
0276  5331  GTW,     731+4400
0277  5076  DUMW,    DUMMY+4400
0300  0002  C2,      2
                    *404
0404  4477          JMS I Z SYSIO /READ A BLOCK FROM SYS DEV
                    *442
0442  5670          JMP I P200 /BEGIN PROGRAM EXECUTION (.OS.)
                    /CHAIN STATMENT ROUTINE
0443  1414  CCHAIN, TAD I Z ILC /GET 1ST CHARACTER OF FILE NAME
0444  3266          DCA FILN1
0445  1414          TAD I Z ILC /GET LAST CHARACTERS OF FILE NAME
0446  3267          DCA FILN2
0447  1271          TAD FBLKK
0450  3253  FGO,    DCA FCORE-1
0451  4477          JMS I Z SYSIO/CALL THE FOSL
0452  0003          3              /FUNCTION-READ
 0 453  0000          0
0454  0000  FCORE,   0
0455  0000  FLINK,   0
0456  4000          IOERR
0457  1254          TAD FCORE
0460  1073          TAD Z C200  /INCREMENT CORE LOCATION
0461  3254          DCA FCORE
0462  1255          TAD FLINK  /GET NEXT BLOCK NUMBER
0463  7440          SZA
0464  5250          JMP FGO    /CONTINUE LOADING
0465  5554          JMP I Z SSTR /RETURN FROM FOSL CALL
0466  0000  FILN1,   0        /STORE 1ST LETTERS OF FILE NAME
0467  0000  FILN2,   0        /STORE LAST LETTERS OF FILE NAME
0470  2772  P200,    2772
0471  0000  FBLKK,   0        /BLOCK NUMBER OF FOSL+1
                    *472
0472  0001  SAVB,    1        /1ST BLOCK OF I/O IN D7
0473  6065  C6065,   6065
0474  3200  M4600,   -4600
0475  5000  C5000,   5000
                    *402
0402  4000          IOERR   /TRIED TO READ TOO MANY BLOCKS
                    *411
0411  4000          IOERR  /SYSTEM DEVICE READ ERROR
                    *423
0423  4000          IOERR
                    *154
0154  2000  SSTR,    KMOVE
0155  1600  LLKK,    1600
                    *200
0200  4555  BEGIN,   JMS I Z LLKK
                    *1056
```

```
1056  1272          TAD DN200   /YES,GET 1ST SAM BLOCK NR
                    *1072
1072  0200  DN200,  200  /BLOCK NR OF 1ST SAM BLOCK
                    *1600
1600  0000  LKLK,   0
1601  3027          DCA OBJBLK
1602  1027          TAD OBJBLK /IS IT COMPILE &GO?
1603  7640          SZA CLA
1604  5600          JMP I LKLK /YES
1605  4423          JMS I DNSRN  /GET SAM NUMBER
1606  4657          4657         /FOSL
1607  6354          6354
1610  4336          JMS WRITE /FOSL IS NOT FOUND IN SYS DEV
1611  4424          JMS I SBSRN  /GET BLOCK NUMBER
1612  5210          JMP .-2
1613  7001          IAC      /INCREMENT BLOCK NR
1614  3707          DCA I FFBLK /SAVE BLOCK NUMBER+1
1615  5600          JMP I LKLK  /RETURN
1616  1712  LKO,    TAD I BL      /BLOCK NR OF FOSL
1617  3707          DCA I FFBLK  /SAVE IT ON 471
1620  1710          TAD I FNAM1
1621  3240          DCA NAM1
1622  1711          TAD I FNAM2
1623  3241          DCA NAM2
1624  1323          TAD TDN  /DNSRCH ROUTINE MODIFICATIONS
1625  3722          DCA I POTR  /POINTER
1626  2322          ISZ POTR
1627  1324          TAD ADN
1630  3722          DCA I POTR
1631  2322          ISZ POTR
1632  1325          TAD NDN
1633  3722          DCA I POTR
1634  2322          ISZ POTR
1635  1326          TAD SCL
1636  3722          DCA I POTR
1637  4423          JMS I DNSRN  /GET SAM NUMBER OF
1640  0000  NAM1,    0            /NEXT FILE NAME
1641  0000  NAM2,    0
1642  4336          JMS WRITE  /FILE IS NOT FOUND IN SYS DEV
1643  4424          JMS I SBSRN  /GET BLOCK NUMBER
1644  5242          JMP .-2
1645  3713          DCA I C2601  /SAVE IT
1646  1314          TAD C6000
1647  3715          DCA I C2600
1650  7240          CLA CMA
1651  3716          DCA I C2602
1652  1327          TAD SMC  /PUT INTO ROUTINE SBSRN FOR SYS
1653  3721          DCA I POTR1
1654  1513          TAD I Z NULL  /IS IT I/O?
1655  7440          SZA
1656  4730          JMS I SASW  /YES: SAVE FLAG
1657  1010  FMOVE,  TAD Z AX0   /MOVE I/O ROUTINE
1660  1011          TAD Z AX1
1661  1020          TAD LTST
1662  7440          SZA
```

```
1663  5720          JMP I MONITR
1664  1410  KFH1,   TAD I Z AX0
1665  3411          DCA I Z AX1
1666  2021          ISZ LNDX
1667  5264          JMP KFH1
1670  1513          TAD I Z NULL  /IS IT I/O?
1671  7440          SZA
1672  4731          JMS I PUSW /YES: PUT FLAG FOR NEXT STEP
1673  1043          TAD BUFADR  /SET UP FOR SYSTEM DEVICE INPUT
1674  3444          DCA I BUFOBJ
1675  1073          TAD Z C200
1676  1332          TAD P5
1677  3733          DCA I IP200  /SET NEW START OF .OS.
1700  1334          TAD P5535    /JMP I Z ENFI
1701  3335          DCA P5107    /TTYIN+6
1702  4777          JMS I TMSI  /TYPE
1703  0336          336          /↑
1704  0215          215          /CR
1705  4212          4212        /LF
1706  5717          JMP I KFH2A
1707  0471  FFBLK,  FBLKK   /471
1710  5066  FNAM1,  FILN1+4400  /466+4400
1711  5067  FNAM2,  FILN2+4400  /467+4400
1712  5071  BL,     FBLKK+4400  /471+4400
1713  2601  C2601,  2601
1714  6000  C6000,  6000
1715  2600  C2600,  2600
1716  2602  C2602,  2602
1717  0301  KFH2A,  FH2A
1720  7600  MONITR, 7600
1721  1054  POTR1,  1054
1722  1051  POTR,   1051
1723  1700  TDN,    1700     /TAD I DNT3
1724  0302  ADN,    302      /AND DN6000
1725  7000  NDN,    7000     /NOP
1726  7650  SCL,    7650     /SNA CLA
1727  7700  SMC,    7700     /SMA CLA
1730  3000  SASW,   SSW
1731  3075  PUSW,   PSW
1732  0005  P5,     5
1733  5070  IP200,  P200+4400
1734  5535  P5535,  5535
1735  5107  P5107,  5107
1736  0000  WRITE,  0   /ROUTINE TO TYPE ERROR
1737  7346          CLA CLL CMA RTL  /SET AC=-3
1740  1336          TAD WRITE
1741 3336          DCA WRITE
1742  4777          JMS I TMSI /TYPE
1743  0215          215         /CR
1744  4212          4212       /LF
1745  4355          JMS PRINT  /TYPE 1ST CHARACTERS
1746  2336          ISZ WRITE
1747  4355          JMS PRINT  /TYPE LAST CHARACTERS
1750  1374          TAD SPACE
1751  4776          JMS I TYPE
```

```
1752  1373          TAD QMARK
1753  4776          JMS I TYPE
1754  5720          JMP I MONITR
1755  0000  PRINT,  0
1756  1736          TAD I WRITE
1757  7106          RTL CLL
1760  7006          RTL
1761  7006          RTL
1762  7004          RAL
1763  0375          AND P77
1764  1374          TAD SPACE
1765  4776          JMS I TYPE
1766  1736          TAD I WRITE
1767  0375          AND P77
1770  1374          TAD SPACE
1771  4776          JMS I TYPE
1772  5755          JMP I PRINT
1773  0277  QMARK,  277
1774  0240  SPACE,  240
1775  0077  P77,    77
1776  0514  TYPE,   TTYOUT
1777  0560  TMSI,   TMES
                    *754
0754  0001  PUTBLK, 1
                    *756
0756  0002  PUTLNK, 2    /NEXT BLOCK NR
                    *762
0762  1354          TAD PUTBLK  /GET NEXT BLOCK
0763  7001          IAC
0764  3354          DCA PUTBLK
0765  1356          TAD PUTLNK  /GET NEXT LINK
0766  7450          SNA      /IS THIS LINK ZERO?
0767  5501          JMP I PUTEOF  /YES, FINISH PROGRAM EXEC.
0770  7001          IAC
0771  3356          DCA PUTLNK
0772  1355          TAD PUTBLK+1 /SET BUFFER POINT
0773  3375          DCA BOUT
0774  5745          JMP I PUT
0775  5600  BOUT,   5600
0776  0000  BINP,   0
                    *746
0746  3775          DCA I BOUT  /SAVE WORD IN BUFFER
0747  2375          ISZ BOUT
                    *713
0713  1776          TAD I BINP  /GET WORD FROM BUFFER
0714  2376          ISZ BINP
                    *723
0723  3376          DCA BINP
                    *2000
                    /MOVE SSW, PSW AND REW ROUTINES
2000  1226  KMOVE,  TAD SAX1  /PREPARE AUTOINDEX
2001  3016          DCA Z TEMB
2002  1227          TAD SAX2
2003  3017          DCA Z TEMC
2004  1416  FX0,    TAD I TEMB  /MOVE SSW ROUTINE (2200-3000)
```

```
2005  3417          DCA I TEMC
2006  2230          ISZ SAZ
2007  5204          JMP FX0
2010  7300          CLA CLL   /PREPARE AUTOINDEX
2011  1231          TAD REX1
2012  3016          DCA Z TEMB
2013  1232          TAD REX2
2014  3017          DCA Z TEMC
2015  1416  FX1,    TAD I TEMB  /MOVE REW ROUTINE (2400-6000)
2016  3417          DCA I TEMC
2017  2233          ISZ REZ
2020  5215          JMP FX1
2021  7300          CLA CLL
2022  3016          DCA Z TEMB
2023  3017          DCA Z TEMC
2024  5625          JMP I SSTT
2025  1616  SSTT,   LKO
2026  2177  SAX1,   2177
2027  2777  SAX2,   2777
2030  7600  SAZ,    -200
2031  2377  REX1,   2377
2032  5777  REX2,   5777
2033  7755  REZ,    -23
                    *6000
                    /ROUTINE FOR REWIND STATMENT
6000  4477  REW,    JMS I SYSIO  /READ IN THE BUFFER
6001  0703          703
6002  0001  RBLK,   1
6003  5400  RCOR,   5400/ CORE ADDRESS
6004  0000  RLNK,   0
6005  4000          IOERR
6006  1076          TAD MONRET
6007  3615          DCA I RGTSW
6010  1203          TAD RCOR
6011  3617          DCA I BINW
6012  1204          TAD RLNK   /SAVE BLOCK NR OF THE NEXT BLOCK
6013  3616          DCA I TLNK
6014  5551          JMP I Z INTI /GET NEXT INSTRUCTION
6015  5343  RGTSW,  GETSW1+4400
6016  5334  TLNK,   GETLNK+4400
6017  5376  BINW,   BINP+4400
                    /ROUTINE FOR "BLOCK OUT"STATMENT
6020  1650  BLOT,   TAD I SBOT  /SAVE BOUT
6021  3251          DCA SR1
6022  1653          TAD I SPBK  /SAVE PUTBLK
6023  3255          DCA SR3
6024  4240          JMS FILB
6025  1251          TAD SR1
6026  3650          DCA I SBOT
6027  7132          CLL CML RTR  /2000
6030  1251          TAD SR1
6031  3652          DCA I SPW1  /GET PUTSW1
6032  1255          TAD SR3
6033  3653          DCA I SPBK
6034  1255          TAD SR3
```

```
6035  7001          IAC
6036  3654          DCA I SPLK
6037  5551          JMP I Z INTI /GET NEXT INSTRUCTION
6040  0000  FILB,   0          /ROUTINE TO FILL UP LAST BUFFER
6041  4647          JMS I PUTF
6042  1652          TAD I SPW1
6043  1073          TAD Z C200
6044  7640          SZA CLA
6045  5241          JMP .-4
6046  5640          JMP I FILB
6047  5345  PUTF,   PUT+4400
6050  5375  SBOT,   BOUT+4400
6051  0000  SR1,    0
6052  5377  SPW1,   PUTSW1+4400
6053  5354  SPBK,   PUTBLK+4400
6054  5356  SPLK,   PUTLNK+4400
6055  0000  SR3,    0
              *3000
              /ROUTINE TO SAVE SWITCHES FOR NEXT TIME
3000  0000  SSW,    0
3001  1643          TAD I SWBO  /SAVE BOUT
3002  3244          DCA SWT1
3003  1244          TAD SWT1
3004  1272          TAD M2
3005  3274          DCA SWT12
3006  1674          TAD I SWT12
3007  1273          TAD M7712
3010  7640          SZA CLA
3011  5214          JMP GOW
3012  1274          TAD SWT12
3013  3244          DCA SWT1
3014  1645  GOW,    TAD I SWG  /SAVE GETSW1
3015  3246          DCA SWT2
3016  1647          TAD I SWGL  /SAVE GETLNK
3017  3250          DCA SWT3
3020  1651          TAD I SWBI  /SAVE BINP
3021  3252          DCA SWT4
3022  1653          TAD I SWP  /SAVE PUTSW1
3023  3254          DCA SWT5
3024  1655          TAD I SWPB  /SAVE PUTBLK
3025  3256          DCA SWT6
3026  1657          TAD I SWPL  /SAVE PUTLNK
3027  3260          DCA SWT7
3030  7001          IAC    /SET DUMMY=1
3031  3661          DCA I SWDM
3032  1662          TAD I SCH  /SAVE SWCH
3033  3263          DCA SWT8
3034  1664          TAD I TPD  /SAVE TPWD
3035  3265          DCA SWT9
3036  1666          TAD I PS1  /SAVE PSW1
3037  3267          DCA SWT0
3040  1670          TAD I TEPZ  /SAVE TEMPZ
3041  3271          DCA SWT11
3042  5600          JMP I SSW
3043  5375  SWBO,   BOUT+4400
```

```
3044  0000  SWT1,   0
3045  5343  SWG,    GETSW1+4400
3046  0000  SWT2,   0
3047  5334  SWGL,   GETLNK+4400
3050  0000  SWT3,   0
3051  5376  SWBI,   BINP+4400
3052  0000  SWT4,   0
3053  5377  SWP,    PUTSW1+4400
3054  0000  SWT5,   0
3055  5354  SWPB,   PUTBLK+4400
3056  0000  SWT6,   0
3057  5356  SWPL,   PUTLNK+4400
3060  0000  SWT7,   0
3061  0476  SWDM,   DUMMY
3062  5236  SCH,    SWCH+4400
3063  0000  SWT8,   0
3064  5237  TPD,    TPWD+4400
3065  0000  SWT9,   0
3066  5307  PS1,    PSW1+4400
3067  0000  SWT0,   0
3070  5311  TEPZ,   TEMPZ+4400
3071  0000  SWT11,  0
3072  7776  M2,     -2
3073  0066  M7712,  -7712
3074  0000  SWT12,  0
            /ROUTINE TO PUT SWITCHES
3075  0000  PSW,    0
3076  1244          TAD SWT1
3077  3643          DCA I SWBO
3100  1246          TAD SWT2
3101  3645          DCA I SWG
3102  1250          TAD SWT3
3103  3647          DCA I SWGL
3104  1252          TAD SWT4
3105  3651          DCA I SWBI
3106  1254          TAD SWT5
3107  3653          DCA I SWP
3110  1256          TAD SWT6
3111  3655          DCA I SWPB
3112  1260          TAD SWT7
3113  3657          DCA I SWPL
3114  1263          TAD SWT8
3115  3662          DCA I SCH
3116  1265          TAD SWT9
3117  3664          DCA I TPD
3120  1267          TAD SWT0
3121  3666          DCA I PS1
3122  1271          TAD SWT11
3123  3670          DCA I TEPZ
3124  1332          TAD P705
3125  3733          DCA I PWT
3126  1331          TAD P703
3127  3734          DCA I GWT
3130  5675          JMP I PSW
3131  0703  P703,   703
```

```
3132  0705  P705,    705
3133  5353  PWT,     753+4400
3134  5331  GWT,     731+4400

ADN     1724
AX0     0010
AX1     0011
BEGIN   0200
BINP    0776
BINW    6017
BL      1712
BLOT    6020
BOUT    0775
BUFADR  0043
BUFOBJ  0044
CCHAIN  0443
C2      0300
C200    0073
C2600   1715
C2601   1713
C2602   1716
C5000   0475
C6000   1714
C6065   0473
C703    0275
C705    0273
DNSRN   0023
DN200   1072
DUMMY   0476
DUMW    0277
FBLKK   0471
FCORE   0454
FFBLK   1707
FGO     0450
FH2A    0301
FH3     0256
FILB    6040
FILN1   0466
FILN2   0467
FLINK   0455
FMOVE   1657
FNAM1   1710
FNAM2   1711
FX0     2004
FX1     2015
GETLNK  0734
GETSW1  0743
GOW     3014
GTW     0276
GTWD    0041
GWT     3134
ILC     0014
INTI    0151
IOERR   4000
IP200   1733
```

```
KFH1    1664
KFH2A   1717
KMOVE   2000
LKLK    1600
LKO     1616
LLKK    0155
LNDX    0021
LNKWD   0042
LTST    0020
MONITR  1720
MONRET  0076
MPNT    0015
M2      3072
M4600   0474
M7712   3073
NAM1    1640
NAM2    1641
NDN     1725
NULL    0113
OBJBLK  0027
POTR    1722
POTR1   1721
PRINT   1755
PSW     3075
PSW1    0707
PS1     3066
PTW     0274
PUSW    1731
PUT     0745
PUTBLK  0754
PUTEOF  0101
PUTF    6047
PUTLNK  0756
PUTSW1  0777
PWT     3133
P200    0470
P5      1732
P5107   1735
P5535   1734
P703    3131
P705    3132
P77     1775
QMARK   1773
RBLK    6002
RCOR    6003
REW     6000
REX1    2031
REX2    2032
REZ     2033
RGTSW   6015
RLNK    6004
SASW    1730
SAVB    0472
SAX1    2026
SAX2    2027
```

```
SAZ     2030
SBOT    6050
SBSRN   0024
SCH     3062
SCL     1726
SMC     1727
SPACE   1774
SPBK    6053
SPLK    6054
SPW1    6052
SR1     6051
SR3     6055
SSTR    0154
SSTT    2025
SSW     3000
SWBI    3051
SWBO    3043
SWCH    0636
SWDM    3061
SWG     3045
SWGL    3047
SWP     3053
SWPB    3055
SWPL    3057
SWT0    3067
SWT1    3044
SWT11   3071
SWT12   3074
SWT2    3046
SWT3    3050
SWT4    3052
SWT5    3054
SWT6    3056
SWT7    3060
SWT8    3063
SWT9    3065
SYSIO   0077
TDN     1723
TEMB    0016
TEMC    0017
TEMPZ   0711
TEPZ    3070
TLNK    6016
TMES    0560
TMSI    1777
TPD     3064
TPWD    0637
TTYOUT  0514
TYPE    1776
WRITE   1736

.LOAD
*IN-D7:FOSL,S:FO5
*
*
```

```
ST=
↑↑↑
!SAVE 'FOSL!0-2177,3000,6000;200
.
```

```
                /OPERATING SYSTEM MODIFICATIONS TO SERVICE
                 /"CALL"; "RETURN"; "CHAIN"; "REWIND" AND
                 /"BLOCK OUT" STATMENTS.
                 /
                 /SYSTEM DEFINITIONS:
                         NSUB=2670
                         LSUB=2677
                         CCHAIN=5043
                         REW=6000
                         BLOT=6020
                         *3057
3057   6000              REW    /GO TO REWIND INSTR.
3060   6020              BLOT   /GO TOBLOCK OUT INSTR.
3061   3775              CALLIN  /GO TO CALL INST.
3062   2174              RETIN  /GO TO RETURN INSTR.
3063   5043              CCHAIN  /GO TO CHAIN INSTR.
                         *3775
3775   1056   CALLIN,    TAD PTW /TAKE POINTER
3776   5777              JMP I .+1  /STACK RETURN WITH NSUB
3777   2670              NSUB
                         PTW=56
                         *2174
2174   5775   RETIN,     JMP I .+1 /RETURN WITH THE HELP OF LSUB
2175   2677              LSUB
                         *312
0312   7540              SMA SZA  /IS IT ON TABLE?
                         *2765
2765   0200              200
                         *36
0036   6055              6055
                         *5067
5067   4711   ADSMFX,    5111-200  /RETURN TO THE MONITOR

ADSMFX 5067
BLOT   6020
CALLIN 3775
CCHAIN 5043
LSUB   2677
NSUB   2670
PTW    0056
RETIN  2174
REW    6000

.LOAD
*IN-D7:.OS.,S:FO3
*
*
ST=
↑↑↑
↓SAVE .OS.!0-5177;0
.
```