FLOD1:SYMB

```
*)9REG

% =====================================
% = ROUTINE   6.1                     =
% =====================================

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% ROUTINE TO LAY OUT A SINTRAN-III SYSTEM ON DISKETTE.              %%
%% THE ROUTINE USES TWO DISKETTES                                     %%
%% FIRST DISKETTE CONTAINS MACM AND FIRST SINTRAN-FILE                %%
%% SECOND DISKETTE CONTAINS SECOND SINTRAN-FILE , SYMBOL-LIST AND     %%
%% PATCHFILE FOR THIS VERSION.                                        %%
%%                                                                    %%
%% THIS ROUTINE MUST BE USED FOR S-III VERSION 80.02.01.A             %%
%%                                                                    %%
%% ROUTINE LAST MODIFIED: 80.03.20  BY: LA                            %%
%%                                                                    %%
%% ROUTINE IS CALLED FROM FORTRAN:     CALL FLODU (JBNO)              %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


*)9ENT FLODU




%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%           DATA AND DEFENITION PART                                 %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

BASE AA
INTEGER FORM,CUNIT,CHCB,SAVX,SAVT,BUFFP:=BUFFER
INTEGER DESTFILE,SOURCEFILE
INTEGER POINTER OUTTEXT:=9OUTTEXT,READTEXT:=8READTEXT,INBT:=9INBT
INTEGER POINTER OUTBT:=9OUTBT,MOVCHAR:=9MOVCHAR, ASK1:=9ASK1
INTEGER POINTER STSPACE:=9STSPACE,STAPO:=9STAPO,STCHAR:=9STCHAR
INTEGER POINTER MON70:=9MON70,COPYTEXT:=TEXTCOPY,COPYFILE:=FILECOPY
INTEGER POINTER LINK1,LINK2,LINK3
ESAB

INTEGER ADIRNAM:='$COMMON PART OF DIRECTORY NAME: '
INTEGER ADEVNAM:='$DEVICE NAME: '
INTEGER AUNITNO:='$DEVICE UNIT: '
INTEGER APERIFILE:='$PERIPHERAL FILE NAME: '
INTEGER AFORMAT:='$$FORMATING THE DISKETTE? '
INTEGER DIONE:='-I'
INTEGER DITWO:='-II'
INTEGER CREDIR:='CREATE-DIRECTORY'
INTEGER ENTDIR:='ENTER-DIRECTORY'
INTEGER CREUSER:='CREATE-USER'
INTEGER GIVUSER:='GIVE-USER-SPACE'
INTEGER ALLFILE:='ALLOCATE-FILE'
INTEGER DEVFU:='DEVICE-FUNCTION'
INTEGER BOOTSTRAP:=' DUMP-BOOT FLOPPY-MONITOR'
INTEGER FORMAT:='FORMAT'
INTEGER RELDIR:='RELEASE-DIRECTORY'
INTEGER SYSTEM:='SYSTEM'
INTEGER BITFADR:='231B'
INTEGER NOPAGES:='148'
INTEGER SINT1:=':SYSTEM)SINTRAN-I:BPUN'
INTEGER SINT2:=':SYSTEM)SINTRAN-II:BPUN'
INTEGER MACM:=':SYSTEM)MACM-1718H:BPUN'
INTEGER PATCHF:=':SYSTEM)PATCHFILE:SYMB'
INTEGER SYMBLI:=':SYSTEM)SYMBOL-LIST:SYMB'
INTEGER AMACM:=':SYSTEM)MACM-1718H:BPUN,128D,20D'
INTEGER ASIN1:=':SYSTEM)SINTRAN-I:BPUN,1,127D'
INTEGER ASIN2:=':SYSTEM)SINTRAN-II:BPUN,1,100D'
INTEGER APATCHF:=':SYSTEM)PATCHFILE:SYMB,101D,30D'
INTEGER ASYMBLI:=':SYSTEM)SYMBOL-LIST:SYMB,131D,16D'
INTEGER FITERM:='$--- FIRST FLOPPY FINISHED ---$'
INTEGER CHAMESS:='$PLEASE CHANGE DISKETTES TO CONTINUE DUMPING.'
INTEGER FINISH:='$--- DUMPING OF SINTRAN FINISHED ---$'
INTEGER ARRAY DIRNAM(40),PERIFILE(40),DEVNAM(40),BUFFER(140),YESNO(40)
INTEGER SAVB,SAVL
DISP 0; INTEGER POINTER JOBNO; PSID     % PARAMETER POINTER
DISP 7; INTEGER FINO; PSID              % DISPLACEMENT IN FILE NAMES

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% CONFIGURATION INDEPENDENT FILES.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%          RT-LOADER
INTEGER BPUN1:='(SINTRAN)BPUN1:SYMB'
%%          FILE-SYSTEM PART OF FILE-SYSTEM-SEGMENT
INTEGER BPUN2:='(SINTRAN)BPUN2:SYMB'
%%          SINTRAN PART OF SYSTEM SEGM. AND OP. SEGM.
INTEGER BPUN4:='(SINTRAN)BPUN4:SYMB'
INTEGER ULST1:='(SINTRAN)ULIST1:SYMB'
INTEGER ULST2:='(SINTRAN)ULIST2:SYMB'
INTEGER ULST4:='(SINTRAN)ULIST4:SYMB'
INTEGER SMACM:='(SINTRAN)MACM-1718H:BPUN'
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% CONFIGURATION DEPENDANT FILES. (USER-NAME MUST BE 7 CHARACTERS.)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%          ALL RESIDENT
INTEGER BPUN5:='(AUX-SIN)BPUN-00-5:SYMB'
%%          FILE-SYSTEM PART OF SINTRAN PART II
INTEGER BPUN6:='(AUX-SIN)BPUN-00-6:SYMB'
%%          FILE-SYSTEM PART OF SYSTEM SEGM.
INTEGER BPUN7:='(AUX-SIN)BPUN-00-7:SYMB'
%%          SINTRAN COMM. ROUTINES (REMOTE-LOAD)
INTEGER BPUN8:='(AUX-SIN)BPUN-00-8:SYMB'
%%          POF
INTEGER BPUNA:='(AUX-SIN)BPUN-00-A:BPUN'
%%          START-PROGRAM
INTEGER BPUNB:='(AUX-SIN)BPUN-00-B:BPUN'
%%          ERROR SEGM.
INTEGER BPUNE:='(AUX-SIN)BPUN-00-E:BPUN'
%%          SPOOLING-PROGRAM SEGM.
INTEGER BPUNX:='(AUX-SIN)BPUN-00-X:BPUN'
INTEGER L5IST:='(AUX-SIN)LIST-00:SYMB'
INTEGER SPATCHF:='(AUX-SIN)PATCHES-80A:SYMB'

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% JOB NUMBERS IN ASCII. (00-99)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
@ICR;
INTEGER ARRAY JBNO:=(
30060,30061,30062,30063,30064,30065,30066,30067,30070,30071,
30460,30461,30462,30463,30464,30465,30466,30467,30470,30471,
31060,31061,31062,31063,31064,31065,31066,31067,31070,31071,
31460,31461,31462,31463,31464,31465,31466,31467,31470,31471,
32060,32061,32062,32063,32064,32065,32066,32067,32070,32071,
32460,32461,32462,32463,32464,32465,32466,32467,32470,32471,
33060,33061,33062,33063,33064,33065,33066,33067,33070,33071,
33460,33461,33462,33463,33464,33465,33466,33467,33470,33471,
34060,34061,34062,34063,34064,34065,34066,34067,34070,34071,
34460,34461,34462,34463,34464,34465,34466,34467,34470,34471);
@CR;


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% ENTRYPOINT CALLED FROM FORTRAN.
 SUBR   FLODU
 FLODU: A:=:B=:SAVB
        A:=:L=:SAVL

        X:=JOBNO; A:=JBNO(X)             % GET PARAMETER AND ASCII EQUIV.
        X:="BPUN5"; A=:X.FINO            % MODIFY FILENAMES WITH
        X:="BPUN6"; A=:X.FINO            % JOB-NUMBER.
        X:="BPUN7"; A=:X.FINO            %
        X:="BPUN8"; A=:X.FINO            %
        X:="BPUNA"; A=:X.FINO            %
        X:="BPUNB"; A=:X.FINO            %
        X:="BPUNE"; A=:X.FINO            %
        X:="BPUNX"; A=:X.FINO            %
        X:="L5IST"; A=:X.FINO            %

        CALL DUMP
        SAVB=:B
        SAVL=:L
        EXIT
 RBUS
 %% RETURN TO FORTRAN.
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


 SUBR DUMP
 DUMP: "AA"=:B                           % B-REG POINTS TO BASE AA
        A:=L=:"LINK2"
        0=:FORM=:CUNIT



 % READ DIRECTORY NAME
        "ADIRNAM"; CALL OUTTEXT; T:="DIRNAM"; CALL READTEXT

 % READ DEVICE NAME
        "ADEVNAM"; CALL OUTTEXT; T:="DEVNAM"; CALL READTEXT

 % READ UNIT NUMBER
 ASKUNIT: "AUNITNO"; CALL OUTTEXT; CALL INBT
        IF A<##0 OR A>##2 AND A><##N GO ASKUNIT
        A=:CUNIT; DO CALL INBT WHILE A><15 OD

 % READ THE PERIPHERAL FILE NAME OF THE FLOPPY DISC
        "APERIFILE"; CALL OUTTEXT; T:="PERIFILE"; CALL READTEXT

 %==========================================================
 %      D U M P I N G   F L O P P Y - I
 %==========================================================
```

```
% ASK IF DISKETTE SHOULD BE FORMATED
        CALL ASK1


% FORMAT THE DISKETTE IF FORM><0
        IF FORM><0 THEN
              0=:CHCB; T:="DEVFU"; CALL MOVCHAR; CALL STSPACE
              T:="PERIFI"; CALL MOVCHAR; CALL STSPACE
              T:="FORMAT"; CALL MOVCHAR; CALL STAPO
              CALL MON70; GO L1; *)FILL
        FI
L1:     0=:CHCB=:FORM

% CREATE DIRECTORY
 T:="CREDIR"; CALL MOVCHAR; CALL STSPACE
 T:="DIRNAM"; CALL MOVCHAR; T:="DIONE"; CALL MOVCHAR
 CALL STSPACE; T:="DEVNAM"; CALL MOVCHAR
 IF CUNIT><##N THEN
  CALL STSPACE; CUNIT; CALL STCHAR
 FI; CALL STSPACE; T:="BITFADR"; CALL MOVCHAR
 CALL STAPO; CALL MON70

% ENTER DIRECTORY
        0=:CHCB
        T:="ENTDIR"; CALL MOVCHAR; CALL STSPACE
        T:="DIRNAM"; CALL MOVCHAR; CALL STSPACE
        T:="DEVNAM"; CALL MOVCHAR
        IF CUNIT><##N THEN
              CALL STSPACE; CUNIT; CALL STCHAR
        FI; CALL STAPO; CALL MON70

% CREATE USER
 0=:CHCB
 T:="CREUSER"; CALL MOVCHAR; CALL STSPACE
 T:="DIRNAM"; CALL MOVCHAR; ##:; CALL STCHAR
 T:="SYSTEM"; CALL MOVCHAR; CALL STAPO
 CALL MON70; GO L2; *)FILL

% GIVE USER SPACE
L2: 0=:CHCB
 T:="GIVUSER"; CALL MOVCHAR; CALL STSPACE
 T:="DIRNAM"; CALL MOVCHAR; ##:; CALL STCHAR
 T:="SYSTEM"; CALL MOVCHAR; CALL STSPACE
 T:="NOPAGES"; CALL MOVCHAR; CALL STAPO
 CALL MON70

% ALLOCATE THE MACM-FILE
        0=:CHCB
        T:="ALLFILE"; CALL MOVCHAR; CALL STSPACE
        ##(; CALL STCHAR; T:="DIRNAM"; CALL MOVCHAR
        T:="DIONE"; CALL MOVCHAR; T:="AMACM"; CALL MOVCHAR
        CALL STAPO; CALL MON70

 % ALLOCATE THE SINTRAN-I FILE
        0=:CHCB
         T:="ALLFILE"; CALL MOVCHAR; CALL STSPACE
         ##(; CALL STCHAR; T:="DIRNAM"; CALL MOVCHAR
         T:="DIONE"; CALL MOVCHAR
         T:="ASIN1"; CALL MOVCHAR; CALL STAPO; CALL MON70

 % COPY MACM TO THE DISKETTE

        0=:CHCB
        X:="SMACM"; T:=1; *MON 50; MON 65
        A=:SOURCEFILE; 0=:CHCB
        ##(; CALL STCHAR; T:="DIRNAM"; CALL MOVCHAR
        T:="MACM"; CALL MOVCHAR; CALL STAPO
        X:=BUFFP; T:=0; *MON 50; MON 65
        A=:DESTFILE; CALL COPYFILE
        T:=SOURCEFILE; *MON 43; MON 65
        T:=DESTFILE; *MON 43; MON 65

% OPEN THE FILE SINTRAN-I ON DISKETTE
        0=:CHCB
        ##(; CALL STCHAR; T:="DIRNAM"; CALL MOVCHAR
        T:="DIONE"; CALL MOVCHAR; T:="SINT1"; CALL MOVCHAR
        CALL STAPO; X:=BUFFP; T:=0; *MON 50; MON 65
        A=:DESTFILE

% COPY TEXT TO THE SINTRAN FILE
        "TX010"; CALL COPYTEXT
        "TX020"; CALL COPYTEXT; GO LF1; *)FILL

% OPEN THE FIRST BINARY FILE (FILE SYSTEM)
LF1:    X:="BPUN5"; T:=1; *MON 50; MON 65
        A=:SOURCEFILE; CALL COPYFILE

% COPY TEXT TO THE DISKETTE
        "TX030"; CALL COPYTEXT

% COPY NEXT BINARY FILE TO DISKETTE
T:=SOURCEFILE; *MON 43; MON 65
X:="BPUN7"; T:=1; *MON 50; MON 65
A=:SOURCEFILE; CALL COPYFILE

% COPY TEXT TO THE DISKETTE
```

```
        "TX040"; CALL COPYTEXT

 % COPY SWAP-DRIVER TO DISKETTE
        T:=SOURCEFILE; *MON 43; MON 65
        X:="BPUN8"; T:=1; *MON 50; MON 65
        A=:SOURCEFILE; CALL COPYFILE

        "TX050"; CALL COPYTEXT

 % OPEN THE NEXT BINARY FILE AND COPY TO DISKETTE
        T:=SOURCEFILE; *MON 43; MON 65
        X:="BPUN6"; T:=1; *MON 50; MON 65
        A=:SOURCEFILE; CALL COPYFILE

 % COPY TEXT TO THE DISKETTE
        "TX060"; CALL COPYTEXT

 % OPEN THE NEXT BINARY FILE AND COPY TO DISKETTE
        T:=SOURCEFILE; *MON 43; MON 65
        X:="LSIST"; T:=1; *MON 50; MON 65
        A=:SOURCEFILE; CALL COPYFILE

 % COPY TEXT TO THE DISKETTE
        "TX070"; CALL COPYTEXT
        GO L3; *)FILL

 % COPY THE NEXT BINARY FILE TO THE DISKETTE
 L3:    T:=SOURCEFILE; *MON 43; MON 65
        X:="BPUN2"; T:=1; *MON 50; MON 65
        A=:SOURCEFILE; CALL COPYFILE

 % COPY TEXT TO DISKETTE
        "TX080"; CALL COPYTEXT

 % ULIST2 TO DISKETTE
        T:=SOURCEFILE; *MON 43; MON 65
        X:="ULST2"; T:=1; *MON 50; MON 65
        A=:SOURCEFILE; CALL COPYFILE

 % COPY TEXT TO THE DISKETTE
        "TX090"; CALL COPYTEXT

 % COPY NEXT BINARY FILE TO DISKETTE
        T:=SOURCEFILE;*MON 43;MON 65
        X:="BPUNE";T:=1;*MON 50;MON 65
        A=:SOURCEFILE;CALL COPYFILE

 %TEXT TO DISKETTE
        "TX100"; CALL COPYTEXT

 % NEXT BINARY FILE TO DISKETTE
        T:=SOURCEFILE; * MON 43; MON 65
        X:="BPUN1"; T:=1; *MON 50; MON 65
        A=:SOURCEFILE; CALL COPYFILE

 % TEXT TO DISKETTE
        "TX110"; CALL COPYTEXT

 % ULIST1 TO DISKETTE
        T:=SOURCEFILE; *MON 43; MON 65
        X:="ULST1"; T:=1; *MON 50; MON 65
        A=:SOURCEFILE; CALL COPYFILE; GO L4; *)FILL

 % TEXT TO DISKETTE
 L4:    "TX120"; CALL COPYTEXT

 % CLOSE ALL FILES AND RELEASE DIRECTORY
        T:=-1; *MON 43; MON 65
        0=:CHCB
        T:="RELDIR"; CALL MOVCHAR; CALL STSPACE; T:="DIRNAM"; CALL MOVCHAR
        T:="DIONE"; CALL MOVCHAR; CALL STAPO; CALL MON70

 % DUMP BOOTSTRAP ON DISKETTE
        0=:CHCB
        T:="DEVFU"; CALL MOVCHAR; CALL STSPACE
        T:="PERIFI"; CALL MOVCHAR; T:="BOOTSTRAP"; CALL MOVCHAR
        CALL STAPO; CALL MON70

 % WRITE MESSAGE TO OPERATOR TO INSERT NEW DISKETTE
        "FITERM"; CALL OUTTEXT
        "CHAMESS"; CALL OUTTEXT


 %=================================================================
 %      D U M P I N G   F L O P P Y - I I
 %=================================================================

 % ASK IF DISKETTE SHOULD BE FORMATED
        CALL ASK1


 % FORMAT THE DISKETTE IF FORM><0
        IF FORM><0 THEN
              0=:CHCB; T:="DEVFU"; CALL MOVCHAR; CALL STSPACE
              T:="PERIFI"; CALL MOVCHAR; CALL STSPACE
              T:="FORMAT"; CALL MOVCHAR; CALL STAPO
```

```
                CALL MON70; GO L5; *)FILL
        FI
L5:     0=:CHCB

% CREATE DIRECTORY
        T:="CREDIR"; CALL MOVCHAR; CALL STSPACE
        T:="DIRNAM"; CALL MOVCHAR; T:="DITWO"; CALL MOVCHAR
        CALL STSPACE; T:="DEVNAM"; CALL MOVCHAR
        IF CUNIT><##N THEN
                CALL STSPACE; CUNIT; CALL STCHAR
        FI
        CALL STSPACE; T:="BITFADR"; CALL MOVCHAR
        CALL STAPO; CALL MON70

% ENTER DIRECTORY
        0=:CHCB
        T:="ENTDIR"; CALL MOVCHAR; CALL STSPACE
        T:="DIRNAM"; CALL MOVCHAR; CALL STSPACE
        T:="DEVNAM"; CALL MOVCHAR
        IF CUNIT><##N THEN
                CALL STSPACE; CUNIT; CALL STCHAR
        FI; CALL STAPO; CALL MON70

% CREATE USER
        0=:CHCB
        T:="CREUSER"; CALL MOVCHAR; CALL STSPACE
        T:="DIRNAM"; CALL MOVCHAR; ##:; CALL STCHAR
        T:="SYSTEM"; CALL MOVCHAR; CALL STAPO
        CALL MON70; GO L6; *)FILL

% GIVE USER SPACE
L6:     0=:CHCB
        T:="GIVUSER"; CALL MOVCHAR; CALL STSPACE
        T:="DIRNAM"; CALL MOVCHAR; ##:; CALL STCHAR
        T:="SYSTEM"; CALL MOVCHAR; CALL STSPACE
        T:="NOPAGES"; CALL MOVCHAR; CALL STAPO
        CALL MON70

% ALLOCATE THE SINTRAN-II FILE
        0=:CHCB
        T:="ALLFILE"; CALL MOVCHAR; CALL STSPACE
        ##(; CALL STCHAR; T:="DIRNAM"; CALL MOVCHAR
        T:="DITWO"; CALL MOVCHAR; T:="ASIN2"; CALL MOVCHAR
        CALL STAPO; CALL MON70

% ALLOCATE THE PATCHFILE
        0=:CHCB
         T:="ALLFILE"; CALL MOVCHAR; CALL STSPACE
         ##(; CALL STCHAR; T:="DIRNAM"; CALL MOVCHAR
         T:="DITWO"; CALL MOVCHAR
         T:="APATCHF"; CALL MOVCHAR; CALL STAPO; CALL MON70
        GO LF2; *)FILL

% ALLOCATE THE SYMBOL-FILE
LF2:    0=:CHCB
         T:="ALLFILE"; CALL MOVCHAR; CALL STSPACE
         ##(; CALL STCHAR; T:="DIRNAM"; CALL MOVCHAR
         T:="DITWO"; CALL MOVCHAR
         T:="ASYMBLI"; CALL MOVCHAR; CALL STAPO; CALL MON70

% OPEN THE FILE SINTRAN-II ON DISKETTE
        0=:CHCB
        ##(; CALL STCHAR; T:="DIRNAM"; CALL MOVCHAR
        T:="DITWO"; CALL MOVCHAR; T:="SINT2"; CALL MOVCHAR
        CALL STAPO; X:=BUFFP; T:=0; *MON 50; MON 65
        A=:DESTFILE

% TEXT TO THE DISKETTE
        "TX130"; CALL COPYTEXT

% OPEN THE FIRST BINARY-FILE
        X:="BPUNX"; T:=1; *MON 50; MON 65
        A=:SOURCEFILE; CALL COPYFILE
        "TX140"; CALL COPYTEXT

% NEXT BINARY-FILE TO DISKETTE
        T:=SOURCEFILE; *MON 43; MON 65
        X:="BPUNA"; T:=1; *MON 50; MON 65
        A=:SOURCEFILE; CALL COPYFILE
        "TX150"; CALL COPYTEXT

% NEXT BINARY FILE TO DISKETTE
        T:=SOURCEFILE; *MON 43; MON 65
        X:="BPUN4"; T:=1; *MON 50; MON 65
        A=:SOURCEFILE; CALL COPYFILE

% TEXT TO DISKETTE
        "TX160"; CALL COPYTEXT

% ULIST4 TO DISKETTE
        T:=SOURCEFILE; *MON 43; MON 65
        X:="ULST4"; T:=1; *MON 50; MON 65
        A=:SOURCEFILE; CALL COPYFILE
        "TX170"; CALL COPYTEXT; GO LF3; *)FILL

% NEXT BINARY-FILE TO DISKETTE
```

```
LF3:    T:=SOURCEFILE; *MON 43; MON 65
        X:="BPUN8"; T:=1; *MON 50; MON 65
        A=:SOURCEFILE; CALL COPYFILE

% TEXT TO DISKETTE
        "TX180"; CALL COPYTEXT

% CLOSE DESTINATION AND SOURCEFILE
        T:=DESTFILE; *MON 43; MON 65
        T:=SOURCEFILE; *MON 43; MON 65

% COPY SYMBOL-LIST TO DISKETTE
        0=:CHCB
        X:="LSIST"; T:=1; *MON 50; MON 65
        A=:SOURCEFILE; 0=:CHCB
        ##(; CALL STCHAR; T:="DIRNAM"; CALL MOVCHAR
        T:="SYMBLI"; CALL MOVCHAR; CALL STAPO
        X:=BUFFP; T:=0; *MON 50; MON 65
        A=:DESTFILE; CALL COPYFILE
        "TX000"; CALL COPYTEXT
        T:=SOURCEFILE; *MON 43; MON 65
        T:=DESTFILE; *MON 43; MON 65

% COPY PATCH-FILE TO DISKETTE
        0=:CHCB
        X:="SPATCHF"; T:=1; *MON 50; MON 65
        A=:SOURCEFILE; 0=:CHCB
        ##(; CALL STCHAR; T:="DIRNAM"; CALL MOVCHAR
        T:="PATCHF"; CALL MOVCHAR; CALL STAPO
        X:=BUFFP; T:=0; *MON 50; MON 65
        A=:DESTFILE; CALL COPYFILE
        T:=SOURCEFILE; *MON 43; MON 65
        T:=DESTFILE; *MON 43; MON 65

% RELEASE DIRECTORY
        0=:CHCB
        T:="RELDIR"; CALL MOVCHAR; CALL STSPACE
        T:="DIRNAM"; CALL MOVCHAR; T:="DITWO"; CALL MOVCHAR
        CALL STAPO; CALL MON70


% SUBROUTINE TERMINATES
        "FINISH"; CALL CUTTEXT
        GO LINK2

RBUS

%
% 9ASK1
% SUBROUTINE TO ASK IF DISKETTE SHOULD BE FORMATED
%
% RETURN:    0 IN FORM IF NO, 1 IF YES
%

SUBR 9ASK1
9ASK1:  A:=L=:"LINK3"
         "AFORMAT"; CALL OUTTEXT; T:="YESNO"; CALL READTEXT
        X:=0; "YESNO"=:0; *LBYT; AAX 1
        IF A=##Y THEN
              T:=0; *LBYT; AAX 1
              IF A><##E AND A><##' GO 9ASK1
              IF A=##' GO YES
              T:=0; *LBYT; AAX 1
              IF A ><##S AND A><##' GO 9ASK1
              IF A=##' GO YES
              T:=0; *LBYT; AAX 1
              IF A><##' GO 9ASK1
YES:          MIN FORM
        ELSE
              IF A=##N THEN
                  T:=0; *LBYT; AAX 1
                  IF A><##O AND A><##' GO 9ASK1
                  IF A=##' GO LX
                  T:=0; *LBYT
                  IF A><##' GO 9ASK1
              ELSE
                  GO 9ASK1
        FI; FI
LX:     GO LINK3
RBUS


%
% READTEXT
% SUBROUTINE TO READ A TEXT STRING FROM THE TERMINAL
% THE INPUT SHOULD BE TERMINATED WITH CARRIAGE RETURN
%
% ENTRY: T=ADDRESS OF THE ARRAY WHERE THE TEXT STRING SHOULD
%         BE STORED
%
% RETURN: TEXT STRING TERMINATED WITH THE CHARACTER "!" IN THE ARRAY
%

SUBR &READTEXT
&READTEXT: A:=L=:"LINK1"
        X:=0; T=:0
```

```
        DO CALL INBT WHILE A><15
              IF A=21 THEN ##_; CALL OUTBT; X:=0; GO NEXT FI
              IF A=1 THEN ##^; CALL OUTBT; X-1; GO NEXT FI
              IF X>100 GO NEXT
              T:=D; *SBYT; AAX 1
NEXT:   OD; T:=D; ##'; *SBYT
        GO LINK1
RBUS

%
% INBT
% SUBROUTINE TO READ ONE CHARACTER FROM THE TERMINAL
%
% RETURN: A=CHARACTER
%

SUBR 9INBT
9INBT:  T=:SAVT:=1; *MON 1; MON 65
        A/0177; T:=SAVT; EXIT
RBUS


%
% MOVCHAR
% SUBROUTINE TO MOVE CHARACTERS FROM A TEXT STRING TO AN ARRAY
%
% ENTRY: T=ADDRESS OF TEXT STRING
%

SUBR 9MOVCHAR
9MOVCHAR: A:=L=:"LINK1"
        T=:D; X:=0
        DO T:=D; *LBYT
           WHILE A><##'; CALL STCHAR; X+1
        OD; GO LINK1
RBUS

%
% STCHAR
% SUBROUTINE TO STORE ONE CHARACTER IN THE ARRAY NAMED BUFFER
%
% ENTRY: A=CHARACTER
%

SUBR 9STCHAR
9STCHAR: X=:SAVX:=CHCB; T:=BUFFP; *SBYT
        MIN CHCB; X:=SAVX; EXIT
RBUS

%
% STAPO - STSPACE
% SUBROUTINES TO STORE A SPACE OR A "'" IN THE ARRAY NAMED BUFFER
%

SUBR 9STAPO,9STSPACE
9STAPO: ##'; GO STCHAR
9STSPACE: 40; GO STCHAR
RBUS

%
% OUTTEXT
% SUBROUTINE TO PRINT A TEXT STRING ON THE TERMINAL
%
% ENTRY: A=ADDRESS OF THE TEXT STRING
%

SUBR 9OUTTEXT
9OUTTEXT: T:=L=:"LINK1"
        A=:D; X:=0
        DO T:=D; *LBYT
           WHILE A><##'
              IF A=##$ THEN 15; CALL OUTBT; 12 FI
              CALL OUTBT; X+1
        OD; GO LINK1
RBUS


%
% OUTBT
% SUBROUTINE TO OUTPUT ONE CHARACTER ON THE TERMINAL
%
% ENTRY: A=CHARACTER
%

SUBR 9OUTBT
9OUTBT: T=:SAVT:=1; *MON 2; MON 65
        EXIT
RBUS

%
% MON70
% SUBROUTINE TO EXECUTE THE MONITOR CALL COMND (MON 70)
%

SUBR 9MON70
9MON70: A:=BUFFP; *MON 70; EXIT
```

```
 RBUS


 %
 % COPYTEXT
 % SUBROUTINE TO COPY A TEXT STRING TO THE DISKETTE
 %
 % ENTRY: A=ADDRESS OF TEXT STRING
 %

 SUBR TEXTCOPY
 TEXTCOPY: A=:D; X:=0
        DO
              T:=D; *LBYT; AAX 1
          WHILE A><##'
              * BSET ZRO 70 DA; BSKP ZRO 60 DA; BSET BCM 70 DA
              * BSKP ZRO 50 DA; BSET BCM 70 DA; BSKP ZRO 40 DA
              * BSET BCM 70 DA; BSKP ZRO 30 DA; BSET BCM 70 DA
              * BSKP ZRO 20 DA; BSET BCM 70 DA; BSKP ZRO 10 DA
              * BSET BCM 70 DA; BSKP ZRO 00 DA; BSET BCM 70 DA
              T:=DESTFILE; *MON 2; MON 65
        OD; EXIT
 RBUS


 %
 % COPYFILE
 % SUBROUTINE TO COPY A FILE TO THE DISKETTE
 %

 SUBR FILECOPY
 FILECOPY: DO
              T:=SOURCEFILE; *MON 1; JMP OUT
              T:=DESTFILE; *MON 2; MON 65
        OD
 OUT:   IF A=3 THEN EXIT FI
        *MON 65
 RBUS


 @MAC
 )9RLPL
 )9SCLC

 %=================================================================
 %=================================================================
 %
 %      T E X T - S T R I N G S   T O   B E   C O P I E D
 %            T O   T H E   D I S K E T T E S
 %
 %      THE TEXT-STRINGS ARE LISTED IN THE SAME ORDER AS THEY
 %      WILL BE COPIED TO THE DISKETTES.
 %      A COMMENT GIVING THE FILENAME INDICATES WHERE THIS FILE
 %      IS COPIED TO THE DISKETTE.
 %
 %=================================================================
 %=================================================================


 TX000, ')LINE
 '
 %=================================================================
 %      L A Y O U T   O N   D I S K E T T E - I
 %=================================================================

 TX010, '
 %% PATCHING PROCEDURE:
 %% ====================

 %% BEFORE 22! IS TYPED, PATCHES TO THE SYSTEM CAN BE DONE BY CALLING
 %% A  SET OF MACROES FOR THIS PURPOSE.
 %% THE MACROES ARE CALLED BY ENTERING THE MACRO-NAME FOLLOWED BY
 %% CARRIAGE-RETURN AND LINE-FEED.

 %% NAMES OF THE MACROES AND CORRESPONDING AREA FOR PATCHES:
 %% ------------------------------------------------------------
 %%     - PFILS                CORRECTIONS IN THE FILE-SYSTEM
 %%
 %%     - PRTLO                CORRECTIONS IN THE RT-LOADER
 %%
 %%     - PPOFF                CORRECTIONS IN THE "PAGING-OFF" AREA
 %%
 %%     - PRESI                CORRECTIONS IN RESIDENT, OP. SEGM
 %%
 %%     - PEND                 RESET CORE-IMAGE BEFORE STARTING!
 %%
 %% ------------------------------------------------------------

 %% *** I*M*P*O*R*T*A*N*T ***
 %%
 %% IF PATCH-MACROES HAVE BEEN USED, THE MACRO / PEND / MUST BE
 %% CALLED TO RESET CORE IMAGE!!
 %% ------------------------------------------------------------
 10,0$
 XXX=4000 % RJO 06/06/78!!!!!!!!!!
```

```
YYY=100
ZZZ=4000
)9TABL XXX YYY ZZZ
)CLEAR
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% NAMES OF )9BYTT PARAMETERS
%%
%%  A = CRMAX  ,  B =   CLM  ,  C = COADR  ,  D =  LONG  ,  E = XDSKT
%%  F = MSTYP  ,  G = DEVNO  ,  H =  DASA  ,  L = MACAD  ,  M =  DRES
%%  N =  BLTS  ,  O =  RTAD  ,  P = 99MRE  ,  Q=
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
A=175777
B=100
C=30000
D=30000
E=0

)MCDEF REMOV
F=2
G=500
H=10
M=2000
L=1000
N=M
O=4200
P=L
Q=3600
R

)MCDEF FIXED
F=2
G=500
H=100010
M=102000
L=101000
N=M
O=104200
P=L
Q=103600
R

)MCDEF DRUM
F=0
G=540
H=40
M=4000
L=2000
N=M
O=10400
P=L
Q=7400
R

)MCDEF BD33
F=3
G=1540
H=2
L=200
M=400
N=M
O=1040
P=L
Q=740
R

)MCDEF BD38
F=4
G=1540
H=2
L=200
M=400
N=M
O=1040
P=L
Q=740
R

)MCDEF BD66
F=3
G=1540
H=2
L=200
M=400
N=M
O=1040
P=L
Q=740
R

)MCDEF BD75
F=4
G=1540
H=2
L=200
M=400
N=M
```

```
O=1040
P=L
Q=740
$

)MCDEF BD288
)KILL E
F=5
G=1540
H=2
L=200
M=400
N=M
O=1040
P=L
Q=740
E=100000
$
)MCDEF BDREM
F=6
G=1540
H=2
L=200
M=400
N=M
O=1040
P=L
Q=740
$
)MCDEF BDFIX
F=6
G=1540
H=100002
L=100200
M=100400
N=M
O=101040
P=L
Q=100740
$
10,1$


%%  SINTRAN-III 80.02.01.A   LOAD:     SINTRAN-DISKETTE-I
%%  ==========================================================

%% CHOOSE MASS STORAGE BY GIVING ONE OF THE FOLLOWING CODES
%% FOLLOWED BY CARRIAGE RETURN AND LINE-FEED.

%% IF SYSTEM ON BIG-DISC  33 MEGABYTES, ------ TYPE: BD33
%% IF SYSTEM ON BIG-DISC  38 MEGABYTES, ------ TYPE: BD38
%% IF SYSTEM ON BIG-DISC  66 MEGABYTES, ------ TYPE: BD66
%% IF SYSTEM ON BIG-DISC  75 MEGABYTES, ------ TYPE: BD75
%% IF SYSTEM ON BIG-DISC 288 MEGABYTES, ------ TYPE: BD288
%% IF SYSTEM ON BIG REMOVABLE CARTRIGDE DISK,  TYPE: BDREM
%% IF SYSTEM ON BIG FIXED CARTRIGDE DISK, ---- TYPE: BDFIX
%% IF SYSTEM ON REMOVABLE CARTRIGDE DISK, ---- TYPE: REMOV
%% IF SYSTEM ON FIXED CARTRIGDE DISK, -------- TYPE: FIXED
%% IF SYSTEM ON DRUM, ----------------------- TYPE: DRUM

%% CHOOSE MASS-STORAGE DEVICE, AND TYPE 10,0$ TO CONTINUE.
1,0$
'



TX020, '
)KILL REMOV FIXED DRUM BD33 BD38 BD66 BD75 BD288 BDREM BDFIX

)MCDEF PFILS
)KILL A
A=175777
)9BYTT F G C D B N L A L H
$

)MCDEF PRTLO
)KILL A
A=175777
)9BYTT F G C D B N O A L H
$

)MCDEF PPOFF
)KILL A
A=175777
)KILL BASE
)KILL DISP
BASE=9SMRE
DISP=40000
)9BYTT F G C D B N P A L H
$

)MCDEF PRESI
)KILL A
A=175777
)9BYTT F G C D B N M A L H
$
```

```
)MCDEF PEND
)9BYTT F G C D B N M A L H
)GJEM
)KILL A
A=77777
)9BYTT F G C D B N M A L H
B

)MCDEF PFTER
)9SBLO 4
)9ASSM 10,1
B

)MCDEF PFLP
)9SBLO 4
)9ASSM 10,5
B


)9BYTT F G C D B N M A L H
10,0,10$
)9READ
'


%*****************************************************
%       BPUN-01-5:SYMB
%*****************************************************

TX030, '
10,0,10$
)9READ
'

%*****************************************************
%       BPUN-01-7:SYMB
%*****************************************************

TX040, '
10,0,10$
)9READ
'

%*****************************************************
%       BPUN-01-8:BPUN
%*****************************************************

TX050, '
)9BYTT F G C D B N L A L H
10,0,10$
)9READ
'

%*****************************************************
%       BPUN-01-6:SYMB
%*****************************************************

TX060, '
10,0$
'

%*****************************************************
%       LIST-01:SYMB
%*****************************************************

TX070, '
10,0,10$
)9READ
'

%*****************************************************
%       BPUN2:SYMB
%*****************************************************

TX080, '
10,0$
'

%*****************************************************
%       ULIST2:SYMB
%*****************************************************

TX090, '
10,0,10$
)9READ
'

%*****************************************************
%       BPUN-01-E:BPUN
%*****************************************************

TX100, '
10,1$
%%
%% DUMPING OF FILE-SYSTEM DONE.
```

```
10,0$
)9BYTT F G C D B N O A L H
10,0,10$
)9READ
'

%*****************************************************
%       BPUN1:SYMB
%*****************************************************

TX110, '
10,0$
'

%*****************************************************
%       ULIST1:SYMB
%*****************************************************

TX120, '
10,1$
%%
%% DUMPING OF RT-LOADER DONE.
10,0$

)9BYTT F G C D B N Q A L H
10,1$

%% INSERT SINTRAN-DISKETTE-II IN FLOPPY-DISK UNIT 0
%% AND TYPE: 10,0$    TO CONTINUE LOADING
)9SBLO 4
'


%=====================================================================
%       L A Y O U T   D I S K E T T E - I I
%=====================================================================


TX130, '
10,1$
%%  SINTRAN-III 80.02.01.A   LOAD:     SINTRAN-DISKETTE-II
%%  ================================================
10,0,10$
)9READ
'

%*****************************************************
%       BPUN-01-X:BPUN
%*****************************************************

TX140, '
10,1$
%% DUMPING OF SPOOLING-PROGRAM DONE.
10,0$

)9BYTT F G C D B N P A L H
10,0,10$
)9READ
'

%*****************************************************
%       BPUN-01-A:BPUN
%*****************************************************

TX150, '
10,1$
%%
%% DUMPING OF "PAGING-OFF" AREA DONE.
10,0$

)9BYTT F G C D B N M A L H
10,0,10$
)9READ
'

%*****************************************************
%       BPUN4:SYMB
%*****************************************************


TX160, '
10,0$
'

%*****************************************************
%       ULIST4:SYMB
%*****************************************************

TX170, '
10,0,10$
)9READ
'

%*****************************************************
%       BPUN-01-8:SYMB
%*****************************************************
```

```
TX180, '
10,1$
%%
%% DUMPING OF SINTRAN DONE.
%%
%% )GJEM AND )9BYTT IS EXECUTED.
10,0$
30/M
2/E
)GJEM
)KILL A
A=77777
)9BYTT F G C D B N M A L H
10,1$

%% A CTOM2 DISKETTE IS REQUIRED WHEN YOU WANT TO MAKE
%% A "COLD START" OF THE SYSTEM.  (CTOM2/HENT START)
%%
%% THIS DISKETTE, SINTRAN-DISKETTE-II, WILL AUTOMATICALY
%% BE PREPARED AS CTOM2 DISKETTE FOR YOUR SYSTEM, AND
%% SHOULD BE USED AS CTOM2 IN FUTURE "COLD-STARTS".
%%
%% ADDITIONAL CTOM2 DISKETTES CAN BE MADE BY INSERTING
%% A NEW FLOPPY IN FLOPPY-DISK-1 UNIT 0  AND GIVE
%% THE COMMAND )CTOM2
%%
%% THE SINTRAN III SYSTEM MAY BE STARTED BY TYPING 22!
%%
)CTOM2
'
)9END
)9EOF
@
@EOF
```