

? COMPILE NUCANDE/TSHARER WITH TSPOL LIBRARY

PACKET 1041
INPUT 614 CARDS FROM ZIP
TIME 1740
DATE 77012 WEDNESDAY, 01/12/77

*compilation listing of
CANDE*

1/12/77

*** MESSAGE OF THE DAY ***

WELCOME TO B5700 TIME SHARING,
MAY YOUR DATA BE DEPENDABLE AND YOUR PROGRAMS PROVEN,
FROM THE STAFF OF THE BIG BAD BURROUGHS.

*** BURROUGHS B5700 TSMCP MARK XVI.0.69 AND INTRINSICS MARK XVI.0.00 ***

? COMPILE NUCANDE/TSHARER WITH TSPOL LIBRARY

? TSPOL STACK= 1024, COMMENT TSPOL STACK AINOT BIG ENOUGH
? TSPOL FILE TAPE= SYMBOL/CANDE DISK SERIAL
? CORE= 3000, COMMENT SETS CORE EST FOR CANDE TO 3000 WORDS
? STACK= 256, CANDE STACK NEEDS NO MORE THAN 256 WDS

? DATA CARD

4:TSPOL/NUCANDE= 4 BOJ 1740 08/28/76
CDB IN CARD DB:TSPOL/NUCANDE= 4
DKA IN SER SYMBOL CANDE:TSPOL/NUCANDE= 4
PBD1041 OUT 011 LINE:TSPOL/NUCANDE= 4
DKA OUT SER DSK1 0000000:TSPOL/NUCANDE= 4
DKA OUT SER DSK2 0000000:TSPOL/NUCANDE= 4
DKA OUT RDM NUCANDE TSHARER:TSPOL/NUCANDE= 4

4TI

TIME FOR TSPOL/NUCANDE= 4 IS 4:46 IN 10:34
DKA LOK NUCANDE TSHARER:TSPOL/NUCANDE= 4
CDB REL CARD DB:TSPOL/NUCANDE= 4

?END,

DKA REL SYMBOL CANDE:TSPOL/NUCANDE= 4
DKA OUT SER DSRT1 0000000:TSPOL/NUCANDE= 4
#NO USER DISK:TSPOL/NUCANDE= 4= 1056 SEGS.

DS OK

#NO USER DISK:TSPOL/NUCANDE= 4= 1056 SEGS.

4OK

DKA OUT SER DSRT2 0000000:TSPOL/NUCANDE= 4
DSRT1/0000000/0000000= 0 SEGS--CREATED 01/12/77 AT 17:58:59:51
DKA REL DSRT1 0000000:TSPOL/NUCANDE= 4
DSRT2/0000000/0000000= 0 SEGS--CREATED 01/12/77 AT 18:02:03:38

DKA REL DSRT2 0000000:TSPOL/NUCANDE= 4
DKA OUT SER DSRT1 0000000:TSPOL/NUCANDE= 4
DKA OUT SER DSRT2 0000000:TSPOL/NUCANDE= 4

#NO USER DISK:TSPOL/NUCANDE= 4= 901 SEGS.

DS OK

#NO USER DISK:TSPOL/NUCANDE= 4= 901 SEGS.

4OK

#NO USER DISK:TSPOL/NUCANDE= 4= 901 SEGS.

DS OK
#NO USER DISK:ITSPOL/NUCANDE= 4= 901 SEGS.
DS OK
#NO USER DISK:ITSPOL/NUCANDE= 4= 901 SEGS.
DS OK
#NO USER DISK:ITSPOL/NUCANDE= 4= 901 SEGS.
DS OK
#NO USER DISK:ITSPOL/NUCANDE= 4= 901 SEGS.
4DS
-OPRTR DS=ED:ITSPOL/NUCANDE= 4, S= 170, A= 42:1
DSK2/0000000/0000000= 600 SEGS--CREATED 01/12/77 AT 17:40:39:55
DKA REL DSK2 0000000:ITSPOL/NUCANDE= 4
PBD1041 REL 011 LINE 13502:ITSPOL/NUCANDE= 4
DSK1/0000000/0000000= 600 SEGS--CREATED 01/12/77 AT 17:40:22:39
DKA REL DSK1 0000000:ITSPOL/NUCANDE= 4
DSRT2/0000000/0000000= 1056 SEGS--CREATED 01/12/77 AT 18:04:06:28
DKA REL DSRT2 0000000:ITSPOL/NUCANDE= 4
DSRT1/0000000/0000000= 1056 SEGS--CREATED 01/12/77 AT 18:02:31:22
DKA REL DSRT1 0000000:ITSPOL/NUCANDE= 4
ITSPOL/NUCANDE= 4, PST= 11:04 DS=ED
PKT#1041 REMOVED

NUCANDE/TSHARER
=====

COMMENT: * TITLE: B5500/B5700 MARK XVI SYSTEM RELEASE *
 * FILE ID: SYMBOL/CANDE TAPE ID: SYMBOL2/FILE000 *
 * THIS MATERIAL IS PROPRIETARY TO BURROUGHS CORPORATION *
 * AND IS NOT TO BE REPRODUCED, USED, OR DISCLOSED *
 * EXCEPT IN ACCORDANCE WITH PROGRAM LINCENSE OR UPON *
 * WRITTEN AUTHORIZATION OF THE PATENT DIVISION OF *
 * BURROUGHS CORPORATION, DETROIT, MICHIGAN 48232 *
 * *
 * COPYRIGHT (C) 1971, 1972, 1974 *
 * BURROUGHS CORPORATION *
 * AA320206 AA386657 AA332366 *;

00000100 T 0000
 00000101 T 0000
 00000102 T 0000
 00000103 T 0000
 00000104 T 0000
 00000105 T 0000
 00000106 T 0000
 00000107 T 0000
 00000108 T 0000
 00000109 T 0000
 00000110 T 0000
 00001000 T 0000

BEGIN % CANDE/TSHARER LAST REVISION 11-72

START OF SEGMENT ***** 2

DEFINE	VFINISHED	= 00#;		00001100	T	0000
DEFINE	VHELLO	= 01#;		00001200	T	0000
DEFINE	VBREAK	= 02#;		00001300	T	0000
DEFINE	VWRU	= 03#;		00001400	T	0000
DEFINE	VLINECLEAR	= 04#;		00001500	T	0000
DEFINE	VDISKCHUNK	= 05#;		00001600	T	0000
DEFINE	VDISCONNECT	= 06#;		00001700	T	0000
DEFINE	VINPUTDATA	= 07#;		00001800	T	0000
DEFINE	VBREAKCLEAR	= 08#;		00001900	T	0000
DEFINE	VRESTARTIT	= 09#;		00001950	T	0000
DEFINE	VMCPUSE	= 10#;		00002000	T	0000
DEFINE	VLISTNEWS	= 12#;	% DUMMY CALL IN HITHERE	00002050	T	0000
DEFINE	VMCPMSG	= 14#;		00002100	T	0000
DEFINE	VCOBLIT	= 15#;		00002110	T	0000
DEFINE	VMCPOK	= 16#;		00002120	T	0000
DEFINE	VTEACHER	= 25#;		00002200	T	0000
DEFINE	VINPUT	= 26#;		00002300	T	0000
DEFINE	VDISPATCH	= 27#;		00002400	T	0000
DEFINE	VDISKREAD	= 28#;		00002500	T	0000
DEFINE	VRESTART	= 29#;		00002600	T	0000
DEFINE	VDOOPS	= 30#;		00002700	T	0000
DEFINE	VDOGATCHER	= 31#;		00002800	T	0000
DEFINE	VCOMPILEIT	= 32#;		00002900	T	0000
DEFINE	VSETUPFILES	= 33#;		00003000	T	0000
DEFINE	VERROR	= 34#;		00003100	T	0000
DEFINE	VCLOSEWORKTABLE	= 35#;		00003200	T	0000
DEFINE	VCHNGPSWD	= 36#;		00003300	T	0000
DEFINE	VTIME	= 37#;		00003301	T	0000
DEFINE	VCC	= 38#;		00003310	T	0000
DEFINE	VCHNGNAME	= 39#;		00003312	T	0000
DEFINE	VMONITORR	= 40#;		00003314	T	0000
DEFINE	VSET	= 41#;		00003316	T	0000
DEFINE	VSSFILE	= 42#;		00003318	T	0000
DEFINE	VEQUATE	= 43#;	% LABEL EQUATION	00003320	T	0000

```

DEFINE VSTATUSCHECK = 44#;
DEFINE VSSMSG = 50#;
DEFINE VPDIT = 51#;
DEFINE VCOMY = 52#;
DEFINE VSEQ = 53#;
DEFINE VBYE = 54#;
DEFINE VMAKE = 55#;
DEFINE VLIST = 56#;
DEFINE VEXECUTE = 57#;
DEFINE VLOAD = 58#;
DEFINE VSAVE = 59#;
DEFINE VREMOVE = 61#;
DEFINE VCHANGE = 62#;
DEFINE VRENAME = 63#;
DEFINE VWHATS = 64#;
DEFINE VTYPE = 65#;
DEFINE VGUARD = 67#;
DEFINE VLOCK = 68#;
DEFINE VTAPE = 69#;
DEFINE VPRINT = 70#;
DEFINE VCHARGE = 71#;
DEFINE VSCHEDULE = 72#;
DEFINE VSTATUSSTOP = 73#;
DEFINE VLFILES = 74#;
DEFINE VUPDATE = 75#;
DEFINE VREPLACE = 76#;
DEFINE CALL(CALL1)=IAM:=0#; % FOR UNUSED CASE STMTS
%
INTEGER COMMON;
SAVE ARRAY L[0:30];
DEFINE STUFF=L#;
DEFINE FF = [18:15]#, CF = [33:15]#; %F FIELD & C FIELD
DEFINE CR=0#, LF=3"100001"#, CRLF=1#, NOCRLF=3"100000" #;
DEFINE NO = NOT#;
DEFINE MAXLINES = 12#; % # OF LOGICAL LINES TO BE HANDLED. %0903-
DEFINE IDLETIME = 360000#;
DEFINE LLWORDS =100#; % NO OF WORDS PER LOGICAL LINE,
DEFINE TBLSZ =20#; % WRKTBL SEGS MAX (17+CHKPOINT),
REAL DISKBOTTOM; % FOR DIRECTORY SEARCHES,

DEFINE NUMOFVERBS = 49#,
NUMOFTYPES =11#, % NUMBER, OF TYPE WORDS
% VOID
% VOID
NUMOFRESWRDS = 30 #; % NO. OF RESERVED WORDS (NOT VERBS)
BOOLEAN FLAGS;
%
DEFINE WECANUSEDATA = FLAGS,[1:1] #
, NODATA = FLAGS,[2:1] #
, NODATAREQUESTED = FLAGS,[3:1] #
, NOFILES = FLAGS,[4:1] #
, NOTANKS = FLAGS,[5:1] #
, WENEEDTOCOOLIT = FLAGS,[6:1]#
, TIMETOOKLINES = FLAGS,[7:1]#
;BOOLEAN DOGSAROUND %%%%%%%%%%% TEMP %%%%%%%%%%%
;
DEFINE TELETYPE = 0#,

```

```

00003330 T 0000
00003400 T 0000
00003500 T 0000
00003600 T 0000
00003700 T 0000
00003800 T 0000
00003900 T 0000
00004000 T 0000
00004100 T 0000
00004200 T 0000
00004300 T 0000
00004500 T 0000
00004600 T 0000
00004700 T 0000
00004800 T 0000
00004900 T 0000
00005100 T 0000
00005200 T 0000
00005300 T 0000
00005400 T 0000
00005450 T 0000
00005500 T 0000
00005600 T 0000
00005700 T 0000
00006390 T 0000
00006395 T 0000
00006400 T 0000
00006500 T 0000
00006600 T 0000
00006700 T 0000
00006800 T 0001
00007600 T 0001
00007700 T 0001
00007800 T 0001
00007900 P 0001
00008000 T 0001
00008100 T 0001
00008200 T 0001
00008300 T 0001
00008400 T 0001
00008500 T 0001
00008600 T 0001
00008700 T 0001
00008750 T 0001
00008800 T 0001
00008900 T 0001
00009000 T 0001
00009100 T 0001
00009200 T 0001
00009300 T 0001
00009400 T 0001
00009500 T 0001
00009510 T 0001
00009520 T 0001
00009600 T 0001
00009700 T 0001
00009730 T 0001

```

```

CONRAC      = 1#;
TC500       = 2#;
BIDS        = 3#;

%PAGE
%
SAVE ARRAY  TTYINPUT [0:29];          % INPUT DATA FROM LINEMTNC.
SAVE ARRAY  WORK [0:59];              % I/O PROCESSING AREA
REAL        DATAOFFSET, DATAWORDS, DATADISKLOC;
REAL        ROWINUSE, DOGCATCHERS, ROWCOUNT, TIMENOW;
REAL        NUDDCOUNT;
REAL        MSGFILELOC;              % CONTAINS ABS DISK LOC MESSAGES.
REAL        TANKFILE;                % DISK ADDR OF TANK/DISK.
REAL        TANKSIZE;                % ROW SIZE OF TANK FILE.

% VOID
REAL        LL                       , % LOGICAL LINE NO CURRENTLY BEING HANDLED.
           LLPREV                    , % LOGICAL LINE NO PREVIOUSLY HANDLED.
           CLOCK                     , % CONTAINS PROCESSOR TIME USED.
           CLOCKPREV                 , % PREVIOUS PROCESSOR TIME USED.
           WAITBIT                   , % WAIT INDICATOR FOR NEXTEVENT.
           REASON                    , % REASON FOR THIS EVENT ENTRY.
           ASTOR                     , % CONTAINS # FOR TTY RESPONSE
           DATE                      , % DATE--MM/DD/YY

% TO VOID A CARD
           IAM                       , % CALLING PROCEDURE NO. OR NEXT OPERATION.
           DOING                     , % CALLER'S CASE OR SWITCH VALUE.
           WANT                      , % REQUESTED OPERATION NO.
           ERRSW                     , % IF AN ERROR IS ENCOUNTERED.
           BASE                      , % BASE OF LL SCRATCH AREA.
           TEMP                      , % TEMPORARY WORDS USED BY A PROCEDURE.
           PARAM                     , % PARAMETER COUNT BEING PASSED TO PROCEDURE
           WAITFOR                   , % CONDITION REQD BEFORE RETURNING.
           PREVRW                    , % POINTS AT LATEST RCW FOR THIS LINE.
           USERBASE,                , % BASE ADDR OF USERS/CANDE
           SREG                      ; % "STACK" REGISTER FOR ACCESSING ARRAY WDS.

%
REAL        SCW, RCW;                % STACK CONTROL WORD, RETURN CONTROL WORD.
REAL        STACKSIZE, PROCESSLIMIT, IOLIMIT, CORESIZE, COMMONVALUE;
REAL        WORKADRS, WORKENDADRS;  % POINTERS TO WORK ARRAY

DEFINE %
SCWWAITBITS = [10:7] #, %
SCWWAIT     = SCW, [10:7] #, %
SCWPREV     = SCW, [17:8] #, %
SCWBASE     = SCW, [25:8] #, %
SCWDOING    = SCW, [33:7] #, %
SCWIAM      = SCW, [40:8] #, %

%
RCWSREG     = RCW, [03:8] #, %
RCWPARAM    = RCW, [11:7] #, %
RCWTEMP     = RCW, [18:7] #, %
RCWBASE     = RCW, [25:8] #, %
RCWDOING    = RCW, [33:7] #, %
RCWIAM      = RCW, [40:8] #, %

DEFINE ESPTABLESIZE = 40#; % INCREASE THIS IF "NO ROOM" ERRORS
ARRAY ESPTABLE[0:ESPTABLESIZE]; % ESP SEGMENTS FOR LABEL EQUATION
ARRAY CTRANDBASE [0:15];
ARRAY DOGHOUSE [0:MAXLINES];
ALPHA ARRAY VERBTABLE [0:2*NUMOFVERBS];

```

```

00009740 T 0001
00009750 T 0001
00009760 T 0001
00009800 T 0001
00009900 T 0001
00010000 T 0001
00010100 T 0003
00010200 T 0005
00010300 T 0005
00010350 T 0005
00010400 T 0005
00010500 T 0005
00010600 T 0005
00010700 T 0005
00010800 T 0005
00010900 T 0005
00011000 T 0005
00011100 T 0005
00011200 T 0005
00011300 T 0005
00011400 T 0005
00011500 T 0005
00011600 P 0005
00011700 T 0005
00011800 T 0005
00011900 T 0005
00012000 T 0005
00012100 T 0005
00012200 T 0005
00012300 T 0005
00012400 T 0005
00012500 T 0005
00012510 C 0005
00012600 T 0005
00012700 T 0005
00012800 T 0005
00012810 T 0005
00012820 T 0005
00012900 T 0005
00013000 T 0005
00013100 T 0005
00013200 T 0005
00013300 T 0005
00013400 T 0005
00013500 T 0005
00013600 T 0005
00013700 T 0005
00013800 T 0005
00013900 T 0005
00014000 T 0005
00014100 T 0005
00014200 T 0005
00014300 T 0005
00014310 T 0005
00014500 T 0007
00014600 T 0008
00014700 T 0010

```

```

TYPETABLE [0:NUMOFTYPES],
RESWRDTABLE [0:NUMOFRESWRDS+1]];
REAL ARRAY EVENTS [0:4]; % OUR EVENT AREA,
REAL ARRAY EVENT [0:20]; % MCP EVENT AREA.
DEFINE %
    EVENTLL = EVENTS[0],[25:8] #
    , EVENTREASON = EVENTS[0],[18:7] #
    ; %
REAL ARRAY LINE [0:MAXLINES,0:LLWORDS]; % LOGICAL LINE INFO,
COMMENT: THE FOLLOWING DEFINES ARE USED TO ACCESS LINE ARRAYWORDS;
DEFINE TBASE = 0#; % BASE FOR TBL ENTRIES
DEFINE SBASE = 49#; % PROCEDURE STACK BASE
DEFINE %
    LLINFO = A[11] #, % LINE CONTROL INFORMATION,
    LLDISKREAD = A[12] #, % ADDRESS OF DISK SEG TO BE READ,
    WAITINGSCW = A[13] #, % HOLDS CURR SCW DURING A WAIT,
    MSGPINTER = A[14] #, %
    MSGPTR = A[14] #, %
    MSGPTRLOC = 14 #, %
    USERLOC = A[15] #, % ADDR OF USER REC
    USERCODE = A[16] #, %
    UCLOC = 16# #, %
    RLCW = A[17] #, % STORED VALUE OF CURRENT LCW
    CHARGE = A[18] #, % CHARGE CODE IF USED,
    LOGONTIME = A[19] #, %
    PTIME = A[20] #, % PROCESSOR TIME USED ON THIS LL,
    PTLOC = 20 #, % LOCATION OF THIS LINES PTIME,
    INPUTCTR = A[21] #, % NUMBER OF INPUTS WE RECVD,
    FILENAME = A[22] #, % NAME OF THE "WORK" FILE
    DEVICE = A[23] #, % TYPE OF REMOTE TERMINAL
    PREVMSG = A[24] #, % MESSAGE BEFORE NO DISK OCCURRED
    INCREMENT = A[24] #, %
    SEQLAST = A[25] #, %
    SEQIN = A[26] #, %
    LASTRECORD = A[27] #, % DATA FILE RECORD COUNT
    TABLEINFO = A[28] #, % CONTROL WORD FOR WORK TABLE
    UPDATEBIT = A[28],[23:1] #, % UPDATE IN PROGRESS
    TPKREL = TABLEINFO,[24:6] #, % REL SEG ADDR
    TPREC = TABLEINFO,[30:2] #, % ID WD REC IN SEG
    TENTRY = TABLEINFO,[32:4] #, % WHICH OF 10 IN REC
    TPCOUNT = TABLEINFO,[36:12] #, % NUMBER OF ENTRIES
    WRKTBLADR = A[29] #, % ABS DISK ADDR FOR WORK FILE
    TPKADR = WRKTBLADR + TPKREL #, % CURR WRK FILE ADDR
    FILETYPE = A[30] #, % TYPE OF FILE WORKING ON
    SOURCEFILE = A[31] #, % NAME OF CURRENT SOURCE FILE
    ESP1 = A[32] #, % ADDRESS OF FIRST ESP SEG
    ESP2 = A[33] #, % SECOND ESP SEGMENT
    RUN1 = A[34] #, % FIRST NAME OF THING RUNNING
    RUN2 = A[35] #, % LAST NAME OF THING RUNNING
    OBJECTFILE = A[36] #, % NAME OF CURRENT OBJECT FILE
    LIBMSGCTR = A[37] #, % NO OF LIBMTNCE MSGS,
    ETIME = A[38] #, % EXECUTION TIME,
    SCHEDNAME = A[39] #, % NAME OF SCHEDULE FILE
    TIMELIMITS = A[39] #, % AUTHORIZED TIME PERIOD
    NOTIFYBIT = A[39],[2:1] #, % "ON" WHEN NOTIFIED
    NOTIFIED = BOOLEAN(NOTIFYBIT) #,
    NOTIFYTIME = A[40] #,

```

```

00014800 T 0014
00014900 T 0015
00015000 T 0018
00015100 T 0020
00015200 T 0022
00015300 T 0022
00015400 T 0022
00015500 T 0022
00015600 T 0022
00015700 T 0024
00015800 T 0024
00015900 T 0024
00016000 T 0024
00016100 T 0024
00016200 T 0024
00016300 T 0024
00016400 T 0024
00016500 T 0024
00016510 T 0024
00016600 T 0024
00016700 T 0024
00016800 T 0024
00016900 T 0024
00017000 T 0024
00017100 T 0024
00017200 T 0024
00017300 T 0024
00017400 T 0024
00017500 T 0024
00017600 T 0024
00017650 T 0024
00017700 T 0024
00017800 T 0024
00017900 T 0024
00018000 T 0024
00018100 T 0024
00018110 T 0024
00018200 T 0024
00018300 T 0024
00018400 T 0024
00018500 T 0024
00018600 T 0024
00018700 T 0024
00018800 T 0024
00018900 T 0024
00019000 T 0024
00019100 T 0024
00019200 T 0024
00019300 T 0024
00019400 T 0024
00019500 T 0024
00019550 T 0024
00019560 T 0024
00019600 T 0024
00019602 T 0024
00019604 T 0024
00019606 T 0024

```

```

TIMERRESTRICTBIT = A[39],[3:1] #,
RESTRICTEDLANGUAGES = A[41] #,
RESTRICTEDVERBS = A[42] #, % AND A[43]
GRACEPERIOD = 0.25 #,
IOTOT = A[44] #, % TOTAL IO TIME FOR THIS LL
COMMONCELL = A[45] #,
CHKSEQNMBR = A[46] #, % NEXT SEQ NO. FOR BLOCK XMISSN
OPTIONBITS = A[47] #, % USERS "SPECIAL" OPTIONS
ESPWORD = A[48] #, % LABEL EQUATION INFO
ESPFLAG = ESPWORD.[1:1] #,
ESPCOUNT = ESPWORD.[2:8] #,
FIRSTESP = ESPWORD.[20:28] #,

```

```

00019608 T 0024
00019610 T 0024
00019620 T 0024
00019622 T 0024
00019624 T 0024
00019626 T 0024
00019627 T 0024
00019628 T 0024
00019630 T 0024
00019632 T 0024
00019634 T 0024
00019636 T 0024
00019700 T 0024
00019800 T 0024
00019900 T 0026
00020000 T 0026
00020100 T 0026
00020200 T 0026
00020300 T 0026
00020400 T 0026
00020500 T 0026
00020600 T 0026
00020700 T 0026
00020800 T 0026
00020900 T 0026
00021000 T 0026
00021100 T 0026
00021200 T 0026
00021300 T 0026
00021400 T 0026
00021500 T 0026
00021600 T 0026
00021700 T 0026
00021800 T 0026
00021900 T 0026
00022000 T 0026
00022100 T 0026
00022200 T 0026
00022300 T 0026
00022400 T 0026
00022500 T 0026
00022510 T 0026
00022520 T 0026
00022530 T 0026
00022540 T 0026
00022550 T 0026
00022580 T 0026
00022585 T 0026
00022590 T 0026
00022600 T 0026
00022700 T 0026
00022800 T 0026
00022900 T 0026
00023000 T 0026
00023100 T 0026
00023200 T 0026
00023202 T 0026

```

```

*
BOOLEAN ARRAY LLCONTROL [0:MAXLINES]; % LOGICAL LINE CONTROL.
COMMENT: THE FOLLOWING DEFINES ACCESS LLCONTROL;
DEFINE LCW = LLCONTROL[LL] #, % LINE CONTROL WORD.
LOGGEDON = LCW.[01:1] # %
, ALLDONE = LCW.[02:1] # %
, BUSYLINE = LCW.[03:1] # %
, IDLELINE = NOT BUSYLINE # %
, INITIATEDWHILEBUSY = LCW.[04:1] # %
, SEQMODE = LCW.[05:1] # %
, EQUATED = LCW.[06:1] # % LABEL EQUATED
, DATAENABLED = LCW.[07:1] # %
, INQUIRY = LCW.[08:1] # %
, RESTARTNEEDED = LCW.[09:1] # %
, WRITEASTERISK = LCW.[10:1] # %
, FILESOK = LCW.[11:1] # %
, GOODOBJ = LCW.[12:1] # %
, BOJMSG = LCW.[13:1] # %
, EOJMSG = LCW.[14:1] # %
, RUNNING = LCW.[15:1] # %
, COMPILING = LCW.[16:1] # %
, LIBMTC = LCW.[17:1] # %
, ALLMCPMSG = LCW.[13:5] # %
, DISCONNECTING = LCW.[18:1] # %
, BREAKORWRU = LCW.[19:1] # %
% 1:15 1:34 1:47:150
, RESTART = LCW.[20:1] # %
, SAVEDWHILEINITBUSY = LCW.[21:1] # %
, DOGLICENSE = LCW.[22:1] # %
, LOGGINGON = LCW.[23:1] # %
, NODISKMCP = LCW.[24:1] # %
, CONTINUEBIT = LCW.[25:1] # %
, MONITORBIT = LCW.[26:1] # %
, NOSTOP = LCW.[27:1] # % XDONT STOP SCHED. ON ERR
, SCHEDULELINE = LCW.[28:1] # %
, SCHEDULELINE = REAL(SCHEDULELINE) # %
, NOSAVE = LCW.[29:1] # %
, ALLTHINGS = LCW.[30:18] # %
, THINGSLAST = LCW.[30:06] # %
, THINGSLINK = LCW.[36:06] # %
, THINGSTODO = LCW.[42:6] # %
;
DEFINE B = BOOLEAN #;
DEFINE R = REAL #;
DEFINE CONCISEBIT = OPTIONBITS.[1:1] #,

```

```

CONCISE          = B(CONCISEBIT) #,
HELPFULBIT      = OPTIONBITS,[2:1] #,
HELPFUL         = B(HELPFULBIT) #,
QUICKLOGBIT     = OPTIONBITS,[3:1] #,
QUICKLOG        = B(QUICKLOGBIT) #,
QUICKBYEBIT     = OPTIONBITS,[4:1] #,
QUICKBYE        = B(QUICKBYEBIT) #,
ALLOWMSGBIT    = OPTIONBITS,[5:1] #,
ALLOWMSG        = B(ALLOWMSGBIT) #,
CCLONGBIT       = OPTIONBITS,[6:1] #,
CCLONG          = B(CCLONGBIT) #,
HELPTOGGLE      = OPTIONBITS,[7:1] #,
NORETRY         = OPTIONBITS,[8:1] #,
UNLOCKTOGGLE    = OPTIONBITS,[9:1] #,
DEFINE LEFTARROW = "+" #;
DEFINE QMARK    = 12 #;           % 12 = 14 OCT = QUESTION MARK
DEFINE WORDS    = "W" #, CHARS = "C" #;
DEFINE INC(INC1) = INC1 := INC1 + 1 #;
DEFINE UPS      = SI := SI + 1 #,
               UPD = DI := DI + 1 #,
               UPT = TALLY := TALLY + 1 #;
DEFINE INCB(INCB1) = INCB1 := BOOLEAN (REAL (INCB1) + 1) #;
DEFINE DECR(DECR1) = DECR1 := DECR1 - 1 #;
DEFINE DECB(DEC1) = DEC1 := BOOLEAN (REAL (DEC1) - 1) #;
DEFINE MATCH(MATCH1,MATCH2) = IF REAL(BOOLEAN(MATCH1)
                                EQV BOOLEAN(MATCH2))
                                = REAL(NOT FALSE) THEN #;
DEFINE COMPAR(COMPAR1,COMPAR2) =
REAL(BOOLEAN(COMPAR1) EQV BOOLEAN(COMPAR2)) = REAL(NOT FALSE) #;
DEFINE TEN8 = 10000000 #; % TEN TO THE EIGHTH
DEFINE GOERR(GOERR1,GOERR2) = BEGIN ERRSW := GOERR1; MSGPOINTER := %
GOERR2; WANT := WAITFOR := 0; IAM := VERROR; GO EXIT; END #;
DEFINE SCHEDERROR =
BEGIN
LINECLEAR(LCW,A[*]); SCWBASE := BASE := SBASE;
WORK[0] := REAL(NOT FALSE); MSGPTR := 0;
IAM := VTEACHER; DOING := 1;
GO TO EXIT;
END #;
DEFINE SCRAMBLE(SCRAMBLE1,SCRAMBLE2) = (-2 *
((SCRAMBLE1,[1:23] + SCRAMBLE1,[24:24]) MOD MODULUS * MODULUS +
(SCRAMBLE2,[1:23] + SCRAMBLE2,[24:24]) MOD MODULUS) +
DISKBOTTOM) #,
MODULUS = 13 #, DIRMOD = 169 #;
REAL SYSTEM;
ARRAY SCANPTR[0:(LLWORDS - SBASE)]; % POINTERS TO WORK ARRAY
*****
COMMENT DOCUMENT
THE ARRAY TOBEDONE IS USED TO STORE THINGS THAT HAVE TO BE
DONE FOR A LINE BUT MUST WAIT BECAUSE THE LINE IS BUSY AND
CANNOT HANDLE THIS INFO NOW,
TOBECONTROL = CONTROL WORD AS FOLLOWS
3:6 = COUNT OF NUMBER OF ROWS USED,
12:36 = BITS INDICATING ROW USED IF ON, BIT 47 = ROW 0,
EACH ROW CONTAINS:
WORD 0 = 3:6 = NEXT ROW WITH INFO FOR THIS LOGICAL LINE,
12:36 = AND WORD TO TURN OFF BIT IN TOBECONTROL.

```

```

00023204 T 0026
00023206 T 0026
00023208 T 0026
00023210 T 0026
00023212 T 0026
00023214 T 0026
00023216 T 0026
00023218 T 0026
00023220 T 0026
00023230 T 0026
00023240 T 0026
00023250 T 0026
00023260 T 0026
00023270 T 0026
00023300 T 0026
00023400 T 0026
00023500 T 0026
00023600 T 0026
00023700 T 0026
00023800 T 0026
00023900 T 0026
00024000 T 0026
00024100 T 0026
00024200 T 0026
00024300 T 0026
00024400 T 0026
00024500 T 0026
00024510 T 0026
00024520 T 0026
00024600 T 0026
00024700 T 0026
00024800 T 0026
00024810 T 0026
00024820 T 0026
00024830 T 0026
00024840 T 0026
00024850 T 0026
00024860 T 0026
00024870 T 0026
00025000 T 0026
00025100 T 0026
00025200 T 0026
00025300 T 0026
00025400 T 0026
00025500 T 0026
00026000 T 0026
00027000 T 0029
00027110 T 0029
00027120 T 0029
00027130 T 0029
00027140 T 0029
00027150 T 0029
00027160 T 0029
00027170 T 0029
00027180 T 0029
00027190 T 0029
00027200 T 0029

```



```

1-15= 10 WORDS INPUTDATA AND 5 WORDS OF EVENTS INFO,
;
REAL          TOBECUNROL;
REAL ARRAY    TOBEDONE [0:35,0:20];
%*****%
          BEGIN % TO SEGMENT THE INIT CODE.
%*****%
PROCEDURE     DISKIO (REASON)      " 33:7=REASON NO, 40:8= LOGICAL LINE.
          "(READFLAG)      " 1 = READ, 0 = WRITE.
          "(A)              " DISK I/O AREA.
          "(OFFSET)        " OFFSET INTO I/O AREA.
          "(SIZE)          " NO OF WORDS TO READ/WRITE.
          "(DISKADDRESS);

VALUE        % REASON, READFLAG, OFFSET, SIZE, DISKADDRESS;
REAL         REASON, READFLAG, OFFSET, SIZE, DISKADDRESS;
REAL         A; % ARRAY[0] + OFFSET = AREA WHERE I/O STARTS.
          %NOTE% % IF OFFSET NEQ 0 THEN A+OFFSET-1 GETS DESTROYED.****
          COMMUNICATE (=1);

```

```

00027210 T 0029
00027220 T 0029
00027300 T 0029
00027400 T 0029
00027500 T 0031
00029900 T 0031
00030000 T 0031
00030100 T 0031
START OF SEGMENT ***** 3
00030200 T 0000
00030300 T 0000
00030400 T 0000
00030500 T 0000
00030600 T 0000
00030700 T 0000
00030800 T 0000
00030900 T 0000
00031000 T 0000
00031100 T 0000
00031200 T 0000

```

```

%*****%
REAL COMM2; % USED TO STORE COMMUNICATE(=2) TO RELEASE CODE SEGMENT
%*****%
PROCEDURE     NEXTEVENT (WAITBIT,CLOCK,A); % A[0]= 0:1-FLAG BIT = 0,
          %              %              1:1-SIGN BIT = 0,
          %              %              2:2-DO NOT TOUCH,
          %              %              18:7-REASON NO.
REAL         WAITBIT, CLOCK;
ARRAY        A[0];
          BEGIN LABEL DUMMY;
          COMMUNICATE (=2); % 25:8-LOGICAL LINE.
          END;

```

```

00031210 T 0001
00031220 T 0001
00031300 T 0001
00031400 T 0001
00031500 T 0001
00031600 T 0001
00031700 T 0001
00031750 T 0001
START OF SEGMENT ***** 4
00031800 T 0000
00031850 T 0000
4 IS 1 LONG, NEXT SEG 3

```

```

COMMENT      %
          WHEN THIS PROCEDURE IS CALLED:
          WAITBIT MUST BE 1 IF WE WILL WAIT FOR AN EVENT,
          0 IF WE DO NOT WISH TO WAIT.
          WHEN THE MCP RETURNS TO THE PROGRAM:
          WAITBIT WILL BE 1 IF AN EVENT IS BEING RETURNED,
          0 IF NO EVENT AND NO WAIT;
%*****%
REAL PROCEDURE GETESPDISK; COMMUNICATE (=3);

```

```

00031900 T 0001
00032000 T 0001
00032100 T 0001
00032200 T 0001
00032300 T 0001
00032400 T 0001
00032500 T 0001
00032600 T 0001
00032700 T 0001

```

```

%*****%
PROCEDURE FORGETESPDISK(ADDR); VALUE ADDR; REAL ADDR; COMMUNICATE(=4);

```

```

00032800 T 0004
00032900 T 0004

```

Acute Business Forms, Inc.

```

*****%
PROCEDURE REQUESTINPUT(BUFF); ARRAY BUFF[0]; COMMUNICATE(-5);
00033000 T 0006
00033100 T 0006

*****%
% REAL PROCEDURE GETUSERDISK (N); VALUE N; REAL N; COMMUNICATE (-6);
00033200 T 0007
00033300 T 0007
*****%
PROCEDURE FORGETUSERDISK (A,N); VALUE A,N; REAL A,N; COMMUNICATE (-7);
00033400 T 0007
00033500 T 0007

*****%
PROCEDURE DISKWAIT(IO,AREA,WDS,ADDR);
VALUE IO,WDS,ADDR;
REAL IO,WDS,ADDR;
ARRAY AREA[0];
COMMUNICATE (-8);
00033600 T 0008
00033700 T 0008
00033800 T 0008
00033900 T 0008
00034000 T 0008
00034100 T 0008

*****%
% PROCEDURE CONTROLCARD (R,A); VALUE R; ARRAY A[0]; REAL R;
% COMMUNICATE (-9); % REASON=R.[33:7], LL=R.[40:8]
00034200 T 0009
00034300 T 0009
00034400 T 0009
*****%
% PROCEDURE SETUSER (CODE); VALUE CODE; REAL CODE; COMMUNICATE (-10);
00034500 T 0009
00034600 T 0009
*****%
DEFINE DISKOUT(DISKOUT1,DISKOUT2)= %
DISKIO(LL&13[33:41:7],0,WORK[0],0,DISKOUT2,DISKOUT1)#;
00034700 T 0009
00034800 T 0009
00034900 T 0009
*****%
DEFINE DISKWRITE(DISKWRITE1) = %
DISKIO(LL&13[33:41:7],00,WORK[0],00,30,DISKWRITE1) #;
00035000 T 0009
00035100 T 0009
00035200 T 0009
*****%
PROCEDURE TWXOUT (R,A,N,T); VALUE R,N,T; REAL R,A,N,T;
00035300 T 0009
00035400 T 0009
IF NOT DISCONNECTING THEN % WRITES ARE O.K.
00035500 T 0009
COMMUNICATE (-12); % T.CF=# OF LFS,T.FF=1 FOR NO CR
00035600 T 0010

*****%
REAL LITOUTSTORE; % KEEPS CHARACTERS BEING OUTPUT BY LITOUT
00035700 T 0011
*****%
PROCEDURE LITOUT (X,CR); VALUE X,CR; REAL X,CR;
00035750 T 0011
00035775 T 0011
00035800 T 0011
*****%
BEGIN
00035900 T 0011
00036000 T 0011
STREAM PROCEDURE BLANK(X,Y);
00036050 T 0011
00036100 T 0000
00036150 T 0000
00036200 T 0001
00036250 T 0004

```

START OF SEGMENT ***** 5

```

BLANK(X, LITOUTSTORE);
TWXOUT(LL, LITOUTSTORE, 8, CR);
END LITOUT;

```

```

00036300 T 0004
00036350 T 0006
00036400 T 0007
5 IS      8 LONG, NEXT SEG 3

```

```

*****
PROCEDURE CG15 (R,F,N,L,U,T,X); VALUE R,F,N,L,U,T,X;
      REAL R,F,N,L,U,T,X;
      GCOMMUNICATE (-15);

```

```

00036500 T 0011
00036600 T 0011
00036700 T 0011
00036800 T 0011

```

```

DEFINE COM15(COM151,COM152,COM153,COM154,COM155,COM156)
      =CC15(COM151,COM152,COM153,COM154,COM155,COM156,0);
DEFINE COM157=CC15;

```

```

00036820 T 0013
00036840 T 0013
00036860 T 0013
00036900 T 0013
00037000 T 0013
00037100 T 0013

```

```

*****
DEFINE FIND(FIND1,FIND2,FIND3,FIND4,FIND5)
      =COM15(FIND1,FIND2,FIND3,FIND4,FIND5,15);

```

```

00037200 T 0013
00037300 T 0013
00037400 T 0013
00037500 T 0013
00037600 T 0013
00037700 T 0013
00037800 T 0013
00037900 T 0013
00038000 T 0013

```

```

-----
COMMENT--ON RETURN:
EVENTS[1]= PROTECT CODE (SAME AS ALGOL SEARCH)
      [2],[1:1]= INTERLOCK
      ,[18:15]= HEADER ADDR
      ,[36:6]= FILE TYPE
      ,[42:6]= OPEN COUNT
      [3]= EOF COUNT
      [4]= ADDR OF 1ST RECORD;

```

```

00038100 T 0013
00038200 T 0013
00038300 T 0013
00038400 T 0013
00038500 T 0013

```

```

*****
DEFINE MAKEFILE =COM15;
      % EVENTS[1]= ADDR OF 1ST RECORD
      % EVENTS[2]= HEADER ADDR

```

```

00038510 T 0013
00038520 T 0013
00038530 T 0013
00038540 T 0013
00038550 T 0013
00038560 T 0013
00038900 T 0013
00039000 T 0013
00039050 T 0013

```

```

*****
DEFINE STAATUS(STAATUS1,STAATUS2,STAATUS3,STAATUS4)=
      CC15(STAATUS1,13,STAATUS2,STAATUS3,STAATUS4,0,0);
      % ON RETURN:
      % EVENTS[1]=PROCESS TIME ( -1 IF NOT RUNNING )
      % EVENTS[2]=IO TIME
      % EVENTS[3]=CONTENTS OF R+27 ( IF COMPILING )

```

```

START OF SEGMENT ***** 6

```

```

*****
PROCEDURE LOGON (LCW,A); BOOLEAN LCW; ARRAY A[0];
      BEGIN LABEL DUMMY;

      LOGGEDON := TRUE;
      LOGONTIME := TIME(1); % SAVE START TIME.
      ESP1 := GETESPDISK; ESP2 := GETESPDISK;
      IF NOT SCHEDULELINE THEN % LET SCHED CALL LOGON.
      COM15(LL&13[33:41:7],2,USERCODE,CHARGE,0,IDLETIME);
      END LOGON;

```

```

00039100 T 0000
00039150 T 0002
00039200 T 0004
00039290 T 0007
00039300 T 0008
00039400 T 0012
6 IS      14 LONG, NEXT SEG 3

```

```

*****

```

```

00039500 T 0013

```

```

PROCEDURE LOGOFF(LCW,A); BOOLEAN LCW; ARRAY A[0];
BEGIN LABEL DUMMY;

IF LOGGEDON THEN %ESP TO GIVE BACK,
BEGIN LOGGEDON := FALSE;
IF ESP1 NEQ 0 THEN FORGETESPDISK(ESP1);
IF ESP2 NEQ 0 THEN FORGETESPDISK(ESP2);
COM15(LL&13[33:41:7],2,USERCODE,CHARGE,[1:41],1,3600);
END IF LOGGEDON;
END LOGOFF;

```

```

00039600 T 0013
00039610 T 0013
START OF SEGMENT ***** 7
00039620 T 0000
00039700 T 0000
00039800 T 0003
00039900 T 0006
00040000 T 0008
00040050 T 0013
00040100 T 0013
7 IS 15 LONG, NEXT SEG 3

```

```

*****
DEFINE SCHEDGOTO(SCHEDGOTO1)=
COM15(LL&13[33:41:7],9,SCHEDGOTO1,0,0,0);
*****
DEFINE REQUESTIP(REQUESTIP1)=
COM15(REQUESTIP1&13[33:41:7],10,0,0,0,0);
*****
DEFINE SCHEDSTAT(SCHEDSTAT1)=
COM15(LL&IAM[33:41:7],11,SCHEDSTAT1,USERCODE,0,0);
*****
DEFINE SCHEDSTOP(SCHEDSTOP1)=
COM15(LL&IAM[33:41:7],12,SCHEDSTOP1,USERCODE,0,0);
*****
PROCEDURE DISCONNECTED(LL); VALUE LL; REAL LL; COMMUNICATE (-16);

```

```

00040200 T 0013
00040205 T 0013
00040210 T 0013
00040215 T 0013
00040220 T 0013
00040225 T 0013
00040230 T 0013
00040235 T 0013
00040240 T 0013
00040245 T 0013
00040250 T 0013
00040255 T 0013
00040260 T 0013
00040300 T 0013

```

```

PROCEDURE RUNJOB(L,A,D); VALUE L,D; REAL L,A,D; COMMUNICATE(-19);

```

```

00040400 T 0014

```

```

PROCEDURE READTAPE(R,N,F,U); VALUE R,N,F,U; REAL R,N,F,U;
COMMUNICATE(-15);

```

```

00040500 T 0015
00040600 T 0015

```

```

*****
DEFINE REMOVEFILE(REMOVEFILE1,REMOVEFILE2,REMOVEFILE3)=
COM15(REMOVEFILE1,3,REMOVEFILE2,REMOVEFILE3,0,0);
*****
DEFINE CHANGEFILE(CHANGEFILE1,CHANGEFILE2,CHANGEFILE3,
CHANGEFILE4,CHANGEFILE5)=CC15 (CHANGEFILE1,5,CHANGEFILE2,
CHANGEFILE3,CHANGEFILE4,CHANGEFILE5,USERCODE) #;
*****
DEFINE REPLACE(REPLACE1,REPLACE2,REPLACE3,REPLACE4,REPLACES5,REPLACE6)=
COM157 (REPLACE1,4,REPLACE2,REPLACE3,REPLACE4,REPLACES5,REPLACE6);
*****
DEFINE SECURE (SECURE1,SECURE2,SECURE3,SECURE4) =
COM157 (LL&14 [33:41:7],7,SECURE1,SECURE2,USERCODE,SECURE3,%0516=

```

```

00040700 T 0016
00040800 T 0016
00040900 T 0016
00041000 T 0016
00041100 T 0016
00041200 P 0016
00041300 P 0016
00041400 T 0016
00041500 T 0016
00041600 T 0016
00041700 T 0016
00041800 P 0016
00041900 P 0016

```

*All times
00036500
to 00037200*

```

SECURE4) #;
*****
PROCEDURE CCCOM(LL,UC,C); VALUE LL,UC,C; REAL LL,UC,C;
COMMUNICATE(-17); % CARRIAGE CONTROL COMMUNICATE (C=1, LONG)

```

```

%0516-
00042000 P 0016
00043000 T 0016
00043100 T 0016
00043200 T 0016

```

```

*****
BEGIN % TO SEGMENT THE NEGATIVE COMMUNICATES;
*****
STREAM PROCEDURE MOVE (N)"NO OF"(WC)"W=WDS, C=CHR FROM"(A)" TO "(B);

```

```

00049700 T 0017
00049800 T 0017
00049900 T 0017
00050000 T 0017

```

START OF SEGMENT ***** 8

```

*****%
VALUE      N, WC;
BEGIN
LOCAL      T;
            SI+LOC N; SI+SI+6; DI+LOC T; DI+DI+7; DS+CHR;
            DI+ B; SI+LOC WC; SI+SI+7;
            IF SC= "W" THEN
BEGIN      SI+ A; DS+ N WDS;
            T(DS+32 WDS; DS+32 WDS);
END        OF WORD MOVER
            ELSE
BEGIN      SI+ A; DS+ N CHR;
            T(DS+32 CHR; DS+32 CHR);
END        OF CHAR MOVER
END        OF MOVE;

```

```

00050100 T 0000
00050200 T 0000
00050300 T 0000
00050400 T 0000
00050500 T 0000
00050600 T 0001
00050700 T 0002
00050800 T 0002
00050900 T 0003
00051000 T 0005
00051100 T 0005
00051200 T 0005
00051300 T 0006
00051400 T 0008
00051500 T 0008

```

```

*****
STREAM PROCEDURE BLANK(X);
*****
BEGIN
    DI:=X; SI:=X;
    ?(IF SC="0" THEN ELSE JUMP OUT; DS:=LIT" "; UPS);
END STREAM BLANK;

```

```

00051600 T 0008
00051700 T 0008
00051800 T 0008
00051900 T 0008
00052000 T 0009
00052100 T 0009
00052200 T 0011

```

```

*****%
INTEGER STREAM PROCEDURE DECCONV (OCTV);
VALUE      OCTV;
BEGIN      SI+LOC OCTV; DI+LOC DECCONV; DS+8 DEC;
END        OF DECIMAL CONVERT;

```

```

00054300 T 0012
00054400 T 0012
00054500 T 0012
00054600 T 0012
00054700 T 0013

```

```

*****
PROCEDURE FIXUSER (CNTR,A,EXIT,BUMPCNTR); VALUE BUMPCNTR;
INTEGER CNTR; ARRAY A [0];
LABEL EXIT; BOOLEAN BUMPCNTR;
BEGIN

```

```

%0501-
%0501-
%0501-
%0501-
00054702 C 0014
00054704 C 0014
00054706 C 0014
00054708 C 0014
00054710 C 0014

```

```

REAL WORD;
INTEGER I,IBY6,LENGTH;
INTEGER STREAM PROCEDURE COUNTZERO (A); VALUE A;
  BEGIN
  SI:=LOC A; TALLY:=0;
  8 (IF SC EQL "0" THEN BEGIN UPS; UPT END ELSE JUMP OUT);
  COUNTZERO:=TALLY;
  END OF COUNTZERO;

```

```

%0501- 00054712 C 0014
START OF SEGMENT ***** 9
%0501- 00054714 C 0000
%0501- 00054716 C 0000
%0501- 00054718 C 0000
%0501- 00054720 C 0000
%0501- 00054722 C 0000
%0501- 00054724 C 0004
%0501- 00054726 C 0004

```

```

IF (WORD:=A [INC (SREG)],[1:5] EQL 0 THEN
  BEGIN
  WORD:=DECCONV (WORD);
  IF LENGTH:=A [INC (SREG)],[3:3] EQL 0 THEN
    GOERR (WORD,8);
  IF I:=8-COUNTZERO (WORD) > 6 THEN
    GOERR (WORD,8);
  IF I EQL 0 THEN I:=1;
  IF LENGTH:=I+LENGTH > 7 THEN
    GOERR (WORD,8);
  IBY6:=I * 6;
  A [SREG]:=0 & WORD [6:48-IBY6:IBY6] &
    A [SREG] [6+IBY6:6:(7-I)*6] & LENGTH [3:45:3];
  DECR (SREG);
  IF BUMPCNTR THEN
    INC (CNTR);
  END
ELSE
  DECR (SREG);
END OF FIXUSER;

```

```

%0501- 00054728 C 0005
%0501- 00054730 C 0009
%0501- 00054732 C 0009
%0501- 00054734 C 0011
%0501- 00054736 C 0014
%0501- 00054738 C 0020
%0501- 00054740 C 0022
%0501- 00054741 C 0028
%0501- 00054742 C 0030
%0501- 00054744 C 0032
%0501- 00054746 C 0038
%0501- 00054748 C 0039
%0501- 00054750 C 0042
%0501- 00054752 C 0047
%0501- 00054754 C 0049
%0501- 00054756 C 0049
%0501- 00054758 C 0051
%0501- 00054760 C 0051
%0501- 00054762 C 0051
%0501- 00054764 C 0053

```

9 IS 56 LONG, NEXT SEG 8

```

*****
PROCEDURE BLACKOUT;
*****
BEGIN REAL I;

STREAM PROCEDURE BO(I,W); VALUE I;
  BEGIN
  SI:=LOC I; SI:=SI+7; DI:=W; DS:=CHR;
  7(SI:=SI-1; DS:=CHR);
  END STREAM PROCEDURE BO;

```

```

00054800 T 0014
00054900 T 0014
00055000 T 0014
00055100 T 0014
START OF SEGMENT ***** 10
00055200 T 0000
00055300 T 0000
00055400 T 0000
00055500 T 0001
00055600 T 0002

```

```

FOR I:="M","W","#", "S", "e" DO
  BEGIN
  BO(I,WORK);
  TWXOUT(LL,WORK[0],8,CR);
  END;

```

```

00055700 T 0002
00055800 T 0013
00055900 T 0013
00056000 T 0014
00056100 T 0016

```

Moore Business Forms, Inc. 14-1

END PROCEDURE BLACKOUT;

00056200 T 0017
10 IS 20 LONG, NEXT SEG 8

STREAM PROCEDURE CLEAR(A,N); VALUE N;

BEGIN
COMMENT CLEARS A + N WORDS STARTING AT A + 1;
DI:=A; DS:=8 LIT "0000"; SI:=A; DS:=N WDS;
END CLEAR;

00056300 T 0014
00056400 T 0014
00056500 T 0014
00056600 T 0014
00056700 T 0015
00056800 T 0015
00056900 T 0017

STREAM PROCEDURE TIMEUSED(W,HMSTX); VALUE HMSTX;

BEGIN LOCAL SV;
DI:=W; SI:=LOC HMSTX;
DS:=2 CHR; DS:=6 LIT " HRS, ";
DS:=2 CHR; DS:=6 LIT " MIN, ";
DS:=2 CHR; DS:=1 LIT " .";
DS:=1 CHR; DS:=5 LIT " SEC. ";
DS:=25LIT " ";
DI:=W; DS:=18FILL; SV:=DI;
SI:=SV; DI:=W; DS:=LIT " "; DS:=25CHR; DS:=LIT LEFTARROW;
% VOID
END TIMEUSED;

00057000 T 0017
00057100 T 0017
00057200 T 0017
00057300 T 0017
00057400 T 0018
00057500 T 0018
00057600 T 0019
00057700 T 0021
00057800 T 0021
00057900 T 0023
00058000 T 0026
00058100 T 0027
00058200 T 0029
00058300 T 0029

STREAM PROCEDURE TIMEOFDAY(W,HMSTX); VALUE HMSTX;

BEGIN
DI:=W; DS:=LIT " "; SI:=LOC HMSTX;
DS:=2 CHR; DS:=LIT "!"; DS:=2 CHR;
SI:=SI+3;
IF SC="A" THEN DS:=4 LIT " AM." ELSE DS:=4 LIT " PM.";
DI:=W; DS:=2FILL;
END TIMEOFDAY;

00058400 T 0029
00058500 T 0029
00058600 T 0029
00058700 T 0029
00058800 T 0030
00058900 T 0031
00059000 T 0032
00059100 T 0032
00059200 T 0034
00059300 T 0035

REAL PROCEDURE TIMECONV(T,B); VALUE T,B; REAL T; BOOLEAN B;

BEGIN
REAL AP;

INTEGER HRS, MIN, SEC, TSEC;
REAL STREAM PROCEDURE ST(H,M,S,T,X);

00059400 T 0035
00059500 T 0035
00059600 T 0035
00059700 T 0035
00059800 T 0035
START OF SEGMENT ***** 11
00059900 T 0000
00060000 T 0000

```

BEGIN
  BI:=LOC ST; SI:=H; DS:=2 DEC; SI:=M; DS:=2 DEC;
  SI:=S; DS:=2 DEC; SI:=T; DS:=1 DEC;
  SI:=X; SI:=SI+7; DS:=CHR;
END STREAM ST;

```

```

00060100 T 0000
00060200 T 0000
00060300 T 0001
00060400 T 0002
00060500 T 0003

```

```

HRS := T DIV 216000;
MIN :=(T DIV 3600) MOD 60;
SEC :=(T DIV 60) MOD 60;
TSEC:=(T DIV 6) MOD 10;
IF HRS GTR 11 THEN AP:= "P" ELSE AP:= "A";
IF HRS GTR 12 THEN HRS:=HRS-12;
IF B THEN IF HRS LSS 01 THEN HRS:= 12;
TIMECONV:=ST(HRS,MIN,SEC,TSEC,AP);
END TIMECONV;

```

```

00060600 T 0004
00060700 T 0005
00060800 T 0007
00060900 T 0008
00061000 T 0010
00061100 T 0015
00061200 T 0018
00061300 T 0021
00061400 T 0023

```

11 IS 28 LONG, NEXT SEG 8

```

*****%
REAL PROCEDURE MAKEFN(FT,LL); VALUE LL,FT; REAL LL,FT;
*****%

```

```

00061800 T 0035
00061900 T 0035
00062000 T 0035
00062100 T 0035
00062200 T 0035

```

```

BEGIN
REAL STREAM PROCEDURE FN(FT,LL); VALUE LL,FT;

```

START OF SEGMENT ***** 12

```

BEGIN
SI:=LOC FT; SI:=SI+5; DI:=LOC FN; DS:=3CHR;
SI:=LOC LL; DS:=3DEC; DS:=2LIT" ";
END STREAM PROCEDURE FN;

```

```

00062300 T 0000
00062400 T 0000
00062500 T 0001
00062600 T 0002

```

```

MAKEFN:=FN(FT,(LL+SYSTEM));
END PROCEDURE MAKEFN;

```

```

00062700 T 0003
00062800 T 0005
12 IS 8 LONG, NEXT SEG 8

```

```

*****%
PROCEDURE @CHKPT (A); ARRAY A[0];
*****%

```

```

00062900 T 0035
00063000 T 0035
00063100 T 0035
00063200 T 0035

```

```

BEGIN LABEL DUMMY;

```

START OF SEGMENT ***** 13

```

RLCW := REAL(LCW); % SAVE TOGGLES IN STACK
MOVE(40,WORDS,A[BASE],WORK[10*TPREC]);
DISKOUT(T@DKADR,10*TPREC+40);
END CHKPT;

```

```

00063210 T 0000
00063300 T 0001
00063400 T 0005
00063500 T 0011

```

13 IS 12 LONG, NEXT SEG 8


```

*****
PROCEDURE CANCELCKPT(A); ARRAY A[0];
  BEGIN LABEL DUMMY;

  RLCW := REAL(LCW); % SAVE TOGGLES IN STACK
  TABLEINFO I:= 0;
  A[TBASE+10] := TEN8;
  MOVE(30,WORDS,A[TBASE+10],WORK[0]);
  DISKWRITE(WRKTBLADR);
  END CANCELCKPT;

```

```

00063600 T 0035
00063700 T 0035
00063800 T 0035
START OF SEGMENT ***** 14
00063810 T 0000
00063900 T 0001
00064000 T 0002
00064100 T 0004
00064200 T 0007
00064300 T 0010
14 IS 12 LONG, NEXT SEG 8

```

```

*****
REAL PROCEDURE DATECONV(DATE); VALUE DATE; REAL DATE;
*****
  BEGIN
  REAL M,D,%; ARRAY L[0:15];

  %.....
  INTEGER STREAM PROCEDURE OCTCONV (DECV);
  VALUE DECV;
  BEGIN SI+LOC DECV; DI+LOC OCTCONV; DS+8 OCT;
  END OF OCTAL CONVERT;

```

```

00064400 T 0035
00064500 T 0035
00064600 T 0035
00064700 T 0035
00064800 T 0035
START OF SEGMENT ***** 15
00064900 T 0001
00065000 T 0001
00065100 T 0001
00065200 T 0001
00065300 T 0003

```

```

%.....
  FILL L[*] WITH 0,31,60,91,121,152,182,213,244,274,305,335,366;

  M:=DECCONV(DATE); D:=OCTCONV(M,[30:18]); Y:=OCTCONV(M,[18:12]);
  IF D GTR 60 THEN IF Y,[46:2] NEQ 0 THEN D:=D+1;
  Y:=M,[18:12]; M:=0; DO M:=M+1 UNTIL D LEQ L[M] OR M=12;
  D:=DECCONV(D=L[M-1]); M:=DECCONV(M);
  DATECONV:="0/00/00"&M[1:37:11]&D[18:36:12]&Y[36:36:12];
  END PROCEDURE DATECONV;

```

```

00065400 T 0004
00065500 T 0004
START OF SEGMENT ***** 16
16 IS 13 LONG, NEXT SEG 15
00065600 T 0006
00065700 T 0013
00065800 T 0017
00065900 T 0023
00066000 T 0028
00066100 T 0032
15 IS 39 LONG, NEXT SEG 8

```

```

*****
PROCEDURE EILSAWAY (OBJTOO,USERCODE); VALUE OBJTOO,USERCODE;
  BOOLEAN OBJTOO; REAL USERCODE;
*****
  BEGIN
  REAL T1,T2;

  T1:=LL & 13 [33:41:7];
  REMOVEFILE (T1,T2:=MAKEFN ("1S",LL),USERCODE);
  IF OBJTOO THEN REMOVEFILE (T1,T2,[6:36],USERCODE);
  REMOVEFILE (T1,T2 & "P" [12:42:6],USERCODE);
  REMOVEFILE (T1,T2 & "T" [12:42:6],USERCODE);

```

```

00066150 C 0035
00066200 C 0035
00066225 C 0035
00066250 C 0035
00066300 C 0035
00066350 C 0035
START OF SEGMENT ***** 17
00066400 C 0000
00066450 C 0001
00066500 C 0005
00066550 C 0008
00066600 C 0012

```

Micro Business Forms, Inc. 51 2402

```

REMOVEFILE (T1,T2 & "Q" [12:42:6],USERCODE);
T2,[36:12]:=1;
REMOVEFILE (T1,T2 & "P" [12:42:6],USERCODE);
END OF FILES AWAY;

```

```

%0527= 00066650 C 0015
%0527= 00066700 C 0018
%0527= 00066750 C 0020
%0527= 00066800 C 0023

```

17 IS 26 LONG, NEXT SEG 8

```

*****
DEFINE MAKEDATE =
DATE!="/00/00"&(DATE!=TIME(5))[1:13:11]&DATE[18:24:12]&DATE[36:36:12]#;
*****
REAL STREAM PROCEDURE FNONLY(W1,W2,F,S); VALUE F,S;
*****
BEGIN
LABEL SPRED, EXIT;
DI := W1;
DS := 5 LIT "FILE!";
SI := LOC F; UPS;
TALLY := 5;
CI := CI + S;
GO TO SPRED;
DS := 7 CHR;
TALLY := 12;
GO TO EXIT;
SPRED:
6(IF SC = ALPHA THEN ELSE JUMP OUT;
DS := CHR; UPT);
EXIT:
DS := 2 LIT " ="; TALLY := TALLY + 2;
FNONLY := TALLY;
S := DI;
SI := LOC S;
DI := W2;
DS := WDS;
END OF FNONLY;

```

```

%0401=
%0401=

```

```

00066900 T 0035
00067000 T 0035
00067100 T 0035
00068700 T 0035
00068800 T 0035
00068900 T 0035
00069000 T 0035
00069100 T 0036
00069200 T 0036
00069300 T 0036
00069400 T 0037
00069500 T 0037
00069600 T 0038
00069700 T 0038
00069800 P 0038
00069900 P 0039
00070000 T 0039
00070100 T 0039
00070200 T 0039
00070300 T 0041
00070400 T 0043
00070500 T 0043
00070600 T 0043
00070700 T 0044
00070800 T 0044
00070900 T 0044
00071000 T 0044
00071100 T 0045

```

```

*****
PROCEDURE GENFILE (FN, FT); VALUE FN, FT; REAL FN, FT;
*****
BEGIN
REAL X, D;

```

```

00073500 T 0046
00073600 T 0046
00073700 T 0046
00073800 T 0046
00073900 T 0046

```

START OF SEGMENT ***** 18

```

X.....
REAL STREAM PROCEDURE TYPEONLY (W,T,S); VALUE W,T,S;
BEGIN
LABEL SPRED, EXIT;
LOCAL X;
DI := W;
DS := 6 LIT " TYPE:"; TALLY := 10;
SI := LOC T;
UPS;
CI := CI + S;

```

```

00074000 T 0000
00074100 T 0000
00074200 T 0000
00074300 T 0000
00074400 T 0000
00074500 T 0000
00074600 T 0000
00074700 T 0001
00074800 T 0001
00074900 T 0002

```

```

GO TO SPRED;
DS := 7 CHR;
TALLY := 17;
GO TO EXIT;
SPRED:
7(IF SC = ALPHA THEN ELSE JUMP OUT;
DS := CHR; UPT);
DS := LIT " ";
EXIT:
DS := 4 LIT " == ";
TYPEONLY := TALLY;
END OF TYPEONLY;

```

```

00075000 T 0002
00075100 T 0002
00075200 T 0003
00075300 T 0003
00075400 T 0003
00075500 T 0003
00075600 T 0005
00075700 T 0007
00075800 T 0007
00075900 T 0007
00076000 T 0008
00076100 T 0009

```

```

X.....
X := FNONLY (L[0], D, FN, 0);
X := X + TYPEONLY (D, FT, 0);
TWXOUT (LL, L[0], X, NOCRLF);
END OF GENFILINE;

```

```

00076200 T 0010
00076300 T 0010
00076400 T 0012
00076500 T 0015
00076600 T 0017

```

18 IS 21 LONG, NEXT SEG 8

```

*****
PROCEDURE ERROR(S,LCW,A); VALUE S; REAL S; BOOLEAN LCW; ARRAY A[0];
FORWARD;
*****
REAL PROCEDURE DOGSLOOSE;
*****
BEGIN
REAL LL,T;
LABEL COLLECT,EXIT;
FOR LL:=MAXLINES STEP =1 UNTIL 0 DO
IF T:=DOGHOUSE[LL] NEQ 0 THEN
IF CTRANDBASE[ROWINUSE] NEQ T THEN
IF NO DOGLICENSE THEN GO TO COLLECT;
DOGSAROUND := DOGCATCHERS GTR 0;
WECANUSEDATA := NO DOGSAROUND;
IF WECANUSEDATA THEN % O.K. TO RETURN DISK.
BEGIN % FIXUP TANK/DISK HEADER.
DISKWAIT(1,WORK[*],30,TANKFILE);
CLEAR(WORK[10],15);
WORK[ROWINUSE+10]:=CTRANDBASE[ROWINUSE];
WORK[6] := ROWINUSE & (ROWCOUNT:=1) [18:33:15];
DISKWAIT(0,WORK[*],30,TANKFILE);
FOR T:=0 STEP 1 UNTIL 15 DO
BEGIN % DONT GIVE AWAY THE CURRENT ROW.
IF T NEQ ROWINUSE AND CTRANDBASE[T] NEQ 0 THEN
BEGIN % GIVE BACK UNUSED DISK.
FORGETUSERDISK(CTRANDBASE[T],TANKSIZE);
CTRANDBASE[T] := 0;
END GIVEBACK;
END FOR LOOP;

```

```

00078500 T 0046
00078600 T 0046
00078800 T 0046
00078900 T 0046
00079000 T 0046
00079100 T 0046
00079200 T 0046
00079300 T 0046

```

START OF SEGMENT ***** 19

```

00079400 T 0000
00079600 T 0000
00079700 T 0001
00079800 T 0002
00079900 T 0004
00080000 T 0008
00080100 T 0009
00080200 T 0011
00080300 T 0012
00080400 T 0012
00080500 T 0014
00080600 T 0016
00080700 T 0018
00080800 T 0020
00080900 T 0022
00081000 T 0023
00081100 T 0023
00081200 T 0025
00081300 T 0025
00081400 T 0026
00081500 T 0028
00081600 T 0028

```

```

END FIXUP TANK HEADER;
GO EXIT;
COLLECT:
DOGCATCHERS := DOGCATCHERS+1;
DOGLICENSE := TRUE;
DOGSLOOSE := LL;
EXIT:END DOGSLOOSE;

```

```

00081700 T 0030
00081800 T 0030
00081900 T 0030
00082000 T 0031
00082100 T 0032
00082200 T 0034
00082300 T 0035

```

19 IS 39 LONG, NEXT SEG 8

```

*****
REAL PROCEDURE FINDALETTER(WORD, TABLE, TABLESIZE);
*****
VALUE WORD, TABLESIZE; REAL WORD, TABLE, TABLESIZE;
BEGIN LABEL DUMMY;

REAL STREAM PROCEDURE FNDALLETTER(WORD, TABLE, TABLESIZE);
VALUE WORD, TABLESIZE;
% ROUTINE COMPARES "WORD" WITH ENTRIES IN ARRAY "TABLE"
% 1ST CHR. OF EACH "TABLE" ENTRY=NO. OF CHRS. IN WORD ([1:1]=1 IF COMPILER)
% "TABLE" SHOULD BE "TABLE[1]" TO USE RETURNED VALUE AS INDEX TO "TABLE"
BEGIN LOCAL SV, T, SGN; LABEL FOUND, EXIT;
SI:=TABLE; SV:=SI; TALLY:=1;
TABLESIZE(DI:=LOC SGN; DS:=ZON; % SAVE ZONE BITS
SI:=SV; SI:=SI+1; DI:=LOC WORD; DI:=DI+1; % POINT TO 1ST CHRS.
IF SC = DC THEN JUMP OUT TO FOUND;
TALLY:=TALLY+1; SI:=SV; SI:=SI+8; SV:=SI);
TALLY:=0; FNDALLETTER:=TALLY; GO TO EXIT; % IF NOT FOUND
FOUND:
T:=TALLY; SI:=LOC T; DI:=LOC FNDALLETTER; DS:=WDS;
SI:=LOC SGN; DI:=LOC FNDALLETTER; DS:=ZON; % TRANSFER SIGN
EXIT:
END STREAM PROCEDURE FNDALLETTER;

```

```

00082400 T 0046
00082500 T 0046
00082510 T 0046
00082520 T 0046
00082600 T 0046
00082610 T 0000
00082700 T 0000
00082800 T 0000
00082900 T 0000
00083000 T 0000
00083100 T 0000
00083200 T 0000
00083300 T 0000
00083400 T 0002
00083500 T 0003
00083600 T 0004
00083700 T 0005
00083800 T 0006
00083900 T 0006
00084000 T 0008
00084100 T 0008
00084200 T 0008

```

START OF SEGMENT ***** 20

```

FINDALETTER:=FNDALLETTER(WORD, TABLE, TABLESIZE);
END PROCEDURE FINDALETTER;

```

```

00084210 T 0010
00084220 T 0013

```

20 IS 16 LONG, NEXT SEG 8

```

*****
REAL PROCEDURE FINDAWORD(WORD, TABLE, TABLESIZE);
*****
VALUE WORD, TABLESIZE; REAL WORD, TABLE, TABLESIZE;
BEGIN LABEL DUMMY;

REAL STREAM PROCEDURE FNDAWORD(WORD, TABLE, TABLESIZE);
VALUE WORD, TABLESIZE;
% ROUTINE COMPARES "WORD" WITH ENTRIES IN ARRAY "TABLE"
% 1ST CHR. OF EACH "TABLE" ENTRY=NO. OF CHRS. IN WORD ([1:1]=1 IF COMPILER)
% 1ST CHR. OF "WORD" = NO. OF CHRS. INPUT BY USER ( MAX. = 7 )

```

```

00084300 T 0046
00084400 T 0046
00084410 T 0046
00084420 T 0046
00084500 T 0046
00084510 T 0000
00084600 T 0000
00084700 T 0000
00084800 T 0000
00084900 T 0000

```

START OF SEGMENT ***** 21

```

* "TABLE" SHOULD BE "TABLE[1]" TO USE RETURNED VALUE AS INDEX TO "TABLE"
BEGIN LOCAL SV,WRDSIZ,TBLSIZ,SGN; LABEL FOUND,EXIT;
SI:=LOC WORD; DI:=LOC WRDSIZ; DI:=DI+7; DS:=CHR; % WORD SIZE
SI:=TABLE; SV:=SI; TALLY:=1;
TABLESIZE(DI:=LOC SGN; DS:=ZON; % SAVE ZONE BITS
SI:=SV; DI:=LOC TBLSIZ; DI:=DI+7; DS:=NUM; % LENGTH OF TABLE ENTRY
SI:=LOC WRDSIZ; SI:=SI+7; DI:=LOC TBLSIZ; DI:=DI+7;
IF SC GEQ DC THEN % WRDSIZ AT LEAST AS LARGE AS TBLSIZ
BEGIN
SI:=SV; SI:=SI+1; DI:=LOC WORD; DI:=DI+1; % POINT TO ACTUAL CHRS.
IF WRDSIZ SC = DC THEN JUMP OUT TO FOUND;
END;
TALLY:=TALLY+1; SI:=SV; SI:=SI+8; SV:=SI;
TALLY:=0; FNDAWORD:=TALLY; GO TO EXIT; % IF NOT FOUND
FOUND:
WRDSIZ:=TALLY; SI:=LOC WRDSIZ; DI:=LOC FNDAWORD; DS:=WDS;
SI:=LOC SGN; DI:=LOC FNDAWORD; DS:=ZON; % TRANSFER SIGN
EXIT:
END STREAM PROCEDURE FINDAWORD;

```

```

00085000 T 0000
00085100 T 0000
00085200 T 0000
00085300 T 0001
00085400 T 0001
00085500 T 0003
00085600 T 0004
00085700 T 0005
00085800 T 0005
00085900 T 0005
00086000 T 0007
00086100 T 0008
00086200 T 0008
00086300 T 0009
00086400 T 0010
00086500 T 0010
00086600 T 0012
00086700 T 0012
00086800 T 0012

```

```

FINDAWORD:=FNDAWORD(WORD, TABLE, TABLESIZE);
END PROCEDURE FINDAWORD;

```

```

00086810 T 0014
00086820 T 0017

```

21 IS 20 LONG, NEXT SEG 8

```

*****
PROCEDURE ENTERESP(LL,ADRS,A);VALUE LL,ADRS; REAL LL,ADRS; ARRAY A[0];
*****
% THIS PROCEDURE IS USED TO KEEP TRACK OF ESP DISK SEGMENTS ALLOCATED
% FOR LABEL EQUATION USEAGE. EACH LOGICAL LINE HAS A VARIABLE CALLED
% "ESPWORD" WHICH IS COMPRISED OF:
% ESPFLAG = ESPWORD.[1:1] ( ON WHEN ESPTABLE IS FULL )
% ESPCOUNT= ESPWORD.[2:8] ( NUMBER OF ENTRIES IN ESPTABLE )
% FIRSTESP= ESPWORD.[20:28] (1ST ESP ADDRESS FOR LBL, EQN. )
% EACH ENTRY IN "ESPTABLE" IS COMPRISED OF:
% ESPTABLE[I],[1:1] (MARKS THIS WORD IN USE )
% ESPTABLE[I],[2:8] (LOGICAL LINE USING THIS WORD )
% ESPTABLE[I],[20:28] (ADDRESS OF ESP SEGMENT IN USE )
% "ESPTABLE" IS DECLARED [0:ESPTABLESIZE]
% WHEN A POSITIVE VALUE FOR LL IS PASSED TO THIS PROCEDURE, AN
% ENTRY IS PLACED IN ESPTABLE FOR THAT LINE AND ESPCOUNT IS INCREMENTED.
% ESPFLAG IS TURNED ON IN THE EVENT THAT THE ESPTABLE IS COMPLETELY
% FILLED. WHEN A NEGATIVE VALUE FOR LL IS PASSED TO THIS PROCEDURE,
% AND A NON-NEGATIVE VALUE FOR ADRS IS PASSED, THE ENTRIES FOR THE
% SPECIFIED LINE ARE REMOVED FROM THE TABLE, AND ESPWORD IS RESET
% TO ZERO. WHEN BOTH LL AND ADRS ARE NEGATIVE, THE ENTRIES FOR THE
% SPECIFIED LINE ARE REMOVED FROM THE TABLE AND THE ESP SEGEMENTS
% ADDRESSED BY EACH TABLE ENTRY ARE RETURNED TO THE SYSTEM. ESPWORD
% IS ALSO RE-SET TO ZERO.
BEGIN
BOOLEAN DELINK,FORGET; REAL I,WORD; LABEL EXIT;

```

```

00087602 T 0046
00087603 T 0046
00087604 T 0046
00087605 T 0046
00087606 T 0046
00087607 T 0046
00087608 T 0046
00087609 T 0046
00087610 T 0046
00087611 T 0046
00087612 T 0046
00087613 T 0046
00087614 T 0046
00087615 T 0046
00087616 T 0046
00087617 T 0046
00087618 T 0046
00087619 T 0046
00087620 T 0046
00087621 T 0046
00087622 T 0046
00087623 T 0046
00087624 T 0046
00087625 T 0046
00087626 T 0046
00087628 T 0046

```

START OF SEGMENT ***** 22

```

DELINK := BOOLEAN(LL,[1:1]); LL:=ABS(LL);
FORGET := BOOLEAN(ADRS,[1:1]); ADRS:=ABS(ADRS);
FOR I:=0 STEP 1 UNTIL ESPTABLESIZE DO
IF DELINK THEN
BEGIN
IF BOOLEAN(WORD:=ESPTABLE[I],[1:1]) THEN % VALID ENTRY
IF WORD,[2:8]=LL THEN % ENTRY FOR THIS LINE
BEGIN
ESPCOUNT := ESPCOUNT - 1;
IF FORGET THEN
IF (WORD:=WORD,[20:28]) NEQ 0 THEN FORGETESPDISK(WORD);
ESPTABLE[I]:=0;
IF ESPCOUNT = 0 THEN % NO MORE ENTRIES FOR THIS LINE
BEGIN
ESPWORD := 0; GO TO EXIT;
END;
END; % IF ENTRY FOR THIS LINE
END % IF DELINK
ELSE
BEGIN % ENTERING NEW VALUE IN TABLE
IF NOT BOOLEAN(ESPTABLE[I],[1:1]) THEN % AVAILABLE WORD
BEGIN
ESPTABLE[I] := "ADRS & LL[2:40:8];
ESPCOUNT := ESPCOUNT + 1;
GO TO EXIT;
END; % IF AVAILABLE WORD
END; % IF ENTERING NEW VALUE
ESPFLAG := 1; % NO ROOM IN TABLES
EXIT;
END PROCEDURE ENTERESP;

```

```

00087630 T 0000
00087632 T 0002
00087634 T 0004
00087636 T 0006
00087638 T 0006
00087640 T 0006
00087642 T 0008
00087644 T 0010
00087646 T 0010
00087648 T 0014
00087650 T 0014
00087652 T 0018
00087654 T 0019
00087656 T 0020
00087658 T 0021
00087660 T 0023
00087662 T 0023
00087664 T 0023
00087666 T 0023
00087668 T 0023
00087670 T 0023
00087672 T 0024
00087674 T 0025
00087676 T 0027
00087678 T 0031
00087680 T 0032
00087682 T 0032
00087684 T 0034
00087686 T 0036
00087688 T 0037

```

22 IS 40 LONG, NEXT SEG 8

```

*****
REAL PROCEDURE SCAN(PTR,STACKWORD,COMPVALUE,WORKENDADRS,PNTR);
*****
VALUE COMPVALUE,WORKENDADRS;
REAL PTR,STAGKWORD,COMPVALUE,WORKENDADRS,PNTR;
BEGIN LABEL DUMMY;

REAL STREAM PROCEDURE SCN(PTR,STACKWORD,COMPVALUE,WORKENDADRS,PNTR);
VALUE COMPYALUE,WORKENDADRS;
BEGIN LOCAL SV,DV,T; LABEL BLNKS,EXIT;
% PUTS A TOKEN INTO STACKWORD
% RETURNS 1,2,3 OR 5 FOR ALPHANUMERIC,NUMERIC,SPECIAL,END OF RECORD
DI:=WORKENDADRS; DS:=LIT LEFTARROW; % MARK END OF RECORD
SI:=PTR; SII:=SI+5; SIII:=SC; % STARTING ADDRESS FOR SCAN
% VOID
BLNKS:
IF SC=" " THEN % IGNORE ALL BLANKS
BEGIN
SII:=SII+1; GO TO BLNKS;
END;
IF SC="," THEN % IGNORE ALL COMMAS
BEGIN

```

```

00087700 T 0046
00087800 T 0046
00087810 T 0046
00087820 T 0046
00087830 T 0046
00087900 T 0046
START OF SEGMENT ***** 23
00087910 T 0000
00088000 T 0000
00088100 T 0000
00088200 T 0000
00088300 T 0000
00088400 T 0000
00088500 T 0000
00088600 T 0001
00088700 T 0001
00088800 T 0001
00088900 T 0002
00089000 T 0002
00089100 T 0003
00089200 T 0003
00089300 T 0004

```

```

      SII=SI+1; GO TO BLNKS;
    END;
  SV:=SI; SII=LOC SV; DI:=PNTR; DS:=WDS; SII=SV; DI:=STACKWORD;
  IF SC = LEFTARROW THEN % CHECK FOR END OF RECORD
  BEGIN
    SV:=SI;      SII=LOC SV;      SII=SI+5;
    DV:=DI; DI:=LOC WORKENDADRS; DI:=DI+5;
    IF 3 SC = DC THEN % END OF RECORD
    BEGIN
      DI:=DV; DS:=7LIT"8000000"; DS:=LIT LEFTARROW;
      TALLY:=5; SCNI=TALLY; SII=SV; GO TO EXIT;
    END; %
    SII=SV; DI:=DV;
  END; % IF LEFTARROW
  IF SC=ALPHA THEN IF SC LEQ "9" THEN % LETTER OR DIGIT
  BEGIN
    IF SC GEQ "0" THEN IF SC LEQ "9" THEN;
    IF TOGGLE THEN % DIGIT STRING
    BEGIN
      TALLY:=2; SCNI=TALLY; % CODE FOR DIGIT STRING
      TALLY:=1; SII=SI+1;
      7(IF SC GEQ "0" THEN IF SC LEQ "9" THEN;
      IF TOGGLE THEN ELSE JUMP OUT;
      SII=SI+1; TALLY:=TALLY+1);
      T:=TALLY; SII=SI-T; DS:= T OCT; GO TO EXIT;
    END; % IF DIGIT STRING
    TALLY:=1; SCNI=TALLY; % CODE FOR ALPHANUMERIC STRING
    DV:=DI; DS:=8LIT"0"; DI:=DI-7; TALLY:=0;
    7(IF SC = ALPHA THEN ELSE JUMP OUT;
    IF SC GTR COMPVALUE THEN JUMP OUT;
    DS:=CHR; TALLY:=TALLY+1);
    T:=TALLY; SV:=SI; SII=LOC T; SII=SI+7; DI:=DV; DS:=CHR; % LENGTH
    SII=SV;
    63(IF SC=ALPHA THEN ELSE JUMP OUT;
    IF SC GTR COMPVALUE THEN JUMP OUT; SII=SI+1);
    GO TO EXIT;
  END; % IF ALPHA
  TALLY:=3; SCNI=TALLY; % CODE FOR SPECIAL CHARACTER
  DS:=7LIT"8000000"; DS:=CHR; % RIGHT JUSTIFIED
EXIT;
  SV:=SI; SII=LOC SV; DI:=PTR; DS:=WDS; % NEXT ADDRESS
  END STREAM PROCEDURE SCAN;

```

```

00089400 T 0004
00089500 T 0005
00089510 T 0005
00089600 T 0006
00089700 T 0007
00089800 T 0007
00089900 T 0008
00090000 T 0009
00090100 T 0009
00090200 T 0009
00090300 T 0012
00090400 T 0013
00090500 T 0013
00090600 T 0013
00090700 T 0013
00090800 T 0015
00090900 T 0015
00091000 T 0017
00091100 T 0017
00091200 T 0017
00091300 T 0018
00091400 T 0018
00091500 T 0020
00091600 T 0021
00091700 T 0023
00091800 T 0024
00091900 T 0024
00092000 T 0025
00092100 T 0027
00092200 T 0028
00092300 T 0030
00092400 T 0031
00092500 T 0032
00092600 T 0032
00092700 T 0034
00092800 T 0037
00092900 T 0037
00093000 T 0038
00093100 T 0038
00093200 T 0040
00093300 T 0040
00093400 T 0041

```

```

SCAN:=SCAN(PTR,STACKWORD,COMPVALUE,WORKENDADRS,PNTR);
END PROCEDURE SCAN;

```

```

00093410 T 0042
00093420 T 0046

```

23 IS 49 LONG, NEXT SEG 8

```

*****
BOOLEAN PROCEDURE YOUFINDAVERB(STACK,VERBLOC,NUMOFVERBS);
*****
VALUE NUMOFVERBS; REAL STACK,VERBLOC,NUMOFVERBS;

```

```

00093500 T 0046
00093600 T 0046
00093610 T 0046
00093620 T 0046

```

```

BEGIN
LABEL L;
BOOLEAN STREAM PROCEDURE YOUFNDAVERB(STACK,VERBLOC,NUMOFVERBS);
VALUE
    BEGIN
    LABEL          FOUND, AWAY;
    LOCAL         T, SAVD;
    DI := VERBLOC; SI := STACK;
    NUMOFVERBS(IF SC GEQ DC THEN
        BEGIN SAVD := DI; SI:=SI-1; DI := LOC T; DI := DI + 7;
        DS := GHR; DI := SAVD;
        IF T SC = DC THEN JUMP OUT TO FOUND;
        DI := SAVD;
        END;
    SI := STACK;
    DI := DI + 15);
    GO AWAY;
FOUND: SI := SAVD; SI := SI + 7; DI := STACK;
    DS := WDS; TALLY := 1; % REPLACE VERB WITH DEFINITION
    YOUFNDAVERB:=TALLY;
AWAY: END OF YOUFNDAVERB;

```

```

00093630 T 0046
00093640 T 0046
START OF SEGMENT ***** 24
00093700 T 0000
00093800 T 0000
00093900 T 0000
00094000 T 0000
00094100 T 0000
00094200 T 0000
00094300 T 0000
00094400 T 0002
00094500 T 0003
00094600 T 0004
00094700 T 0005
00094800 T 0005
00094900 T 0005
00095000 T 0005
00095100 T 0006
00095200 T 0006
00095300 T 0007
00095400 T 0008
00095500 T 0008

```

```

YOUFNDAVERB:=YOUFNDAVERB(STACK,VERBLOC,NUMOFVERBS);
END PROCEDURE YOUFNDAVERB;

```

```

00095510 T 0010
00095520 T 0012
24 IS 16 LONG, NEXT SEG 8

```

```

*****
REAL STREAM PROCEDURE ADDRESS(W);
    BEGIN SI:=W; ADDRESS:=SI; END;

```

```

00095600 T 0046
00095700 T 0046
00095800 T 0046

```

```

*****
STREAM PROCEDURE STICK(P,A,ENDADR); VALUE P,ENDADR;
*****
    BEGIN LOCAL SV,DV; LABEL EXIT;
    % TRANSFERS SS MESSAGE FROM WORK ARRAY
    DI:=A; DS:=BLIT" "; SI:=A; DS:=14 WDS; % BLANK 15 WORDS
    DI:=ENDADR; % INSERT RECORD MARKER DS:=LIT LEFTARROW;
    SI:=P; DI:=A;
    2(60(IF SC=LEFTARROW THEN
        BEGIN
        SV:=SI; SI:=LOC SV; SI:=SI+5;
        DV:=DI; DI:=LOC ENADR; DI:=DI+5;
        IF 3SC=DC THEN JUMP OUT 2 TO EXIT; % END OF RECORD
        SI:=SV; DI:=DV;
        END;
    DS:=CHR));
EXIT:

```

```

00095900 T 0047
00096000 T 0047
00096100 T 0047
00096200 T 0047
00096300 T 0048
00096400 T 0048
00096500 T 0050
00096600 T 0050
00096700 T 0050
00096800 T 0051
00096900 T 0051
00097000 T 0053
00097100 T 0053
00097200 T 0055
00097300 T 0055
00097400 T 0055
00097500 T 0056

```


END STREAM PROCEDURE STICK;

00097600 T 0056

BOOLEAN PROCEDURE FIXERROR(SADRS,EADRS);
VALUE SADRS,EADRS; REAL SADRS,EADRS;
BEGIN LABEL DUMMY;

00097610 T 0057
00097612 T 0057
00097613 T 0057
00097614 T 0057

START OF SEGMENT ***** 25

BOOLEAN STREAM PROCEDURE FXERROR(SADRS,EADRS); VALUE SADRS,EADRS;
% SADRS IS STARTING ADDRESS OF USERS INPUT RECORD
% EADRS IS ADDRESS OF END OF USERS INPUT
% ROUTINE CHECKS SYNTAX OF "FIX" REQUESTS
BEGIN LOCAL SV,D; LABEL EXIT;
SI:=SADRS; TALLY:=1;
20(IF SC LSS "0" THEN SI:=SI+1 ELSE JUMP OUT); % SCAN TO DIGIT
IF TOGGLE THEN GO EXIT; % NO DIGITS
8(IF SC LSS "0" THEN JUMP OUT;IF SC GTR "9" THEN JUMP OUT; SI:=SI+1);
20(IF SC=" " THEN SI:=SI+1 ELSE JUMP OUT); % SCAN TO CHR.
IF TOGGLE THEN GO TO EXIT; % EMPTY FIELD
DI:=LOC D; DI:=DI+7; DS:=CHR; % SAVE DELIMITER
IF SC=D THEN GO TO EXIT; % NO STRING
IF SC = LEFTARROW THEN
BEGIN
SV:=SI; SI:=LOC SV; SI:=SI+5; DI:=LOC EADRS; DI:=DI+5;
IF 3SC=DC THEN % LEFT ARROW REPLACEMENT
BEGIN TALLY:=0; GO EXIT; END ELSE SI:=SV;
END; % IF LEFTARROW
63(IF SC=LEFTARROW THEN % CHECK FOR END OF RECORD
BEGIN
SV:=SI; SI:=LOC SV; SI:=SI+5; DI:=LOC EADRS; DI:=DI+5;
IF 3SC=DC THEN JUMP OUT TO EXIT ELSE SI:=SV;
END;
IF SC NEQ D THEN SI:=SI+1 ELSE JUMP OUT);
IF TOGGLE THEN GO EXIT; % MISSING 2ND DELIMITER
TALLY:=0;
EXIT;
FXERROR:=TALLY;
END STREAM PROCEDURE FXERROR;

00097615 T 0000
00097616 T 0000
00097618 T 0000
00097620 T 0000
00097622 T 0000
00097624 T 0000
00097626 T 0000
00097628 T 0003
00097630 T 0003
00097632 T 0007
00097634 T 0009
00097636 T 0009
00097638 T 0010
00097640 T 0011
00097642 T 0012
00097644 T 0012
00097646 T 0013
00097648 T 0014
00097650 T 0015
00097652 T 0015
00097654 T 0016
00097656 T 0016
00097658 T 0018
00097660 T 0019
00097662 T 0019
00097664 T 0022
00097666 T 0022
00097668 T 0022
00097670 T 0022
00097672 T 0023

FIXERROR:=FXERROR(SADRS,EADRS);
END PROCEDURE FIXERROR;

00097674 T 0024
00097676 T 0027

25 IS 31 LONG, NEXT SEG 8

PROCEDURE SCANNER(STRADRS,A,LCW,ENDADRS); VALUE STRADRS,ENDADRS;

REAL STRADRS,ENDADRS; BOOLEAN LCW; ARRAY A(O);
BEGIN
% MAIN CANDE SCANNER

00097700 T 0057
00097800 T 0057
00097900 T 0057
00098000 T 0057
00098100 T 0057
00098200 T 0057

Moore Business Forms, Inc. 34

```

% RETURNS TYPE OF INPUT IN A[BASE],[3:3] =
% 0 = COMMENT ( INPUT STARTS WITH QUOTE )
% 1 = VERB ( INPUT STARTS WITH A VERB )
% 2 = NUMBER ( INPUT STARTS WITH A NUMBER )
% 3 = SPECIAL ( INPUT STARTS WITH A SPECIAL CHARACTER )
% 4 = SSMSG ( INPUT IS AN SS MESSAGE )
% 5 = NULL INPUT
% WHEN INPUT TYPE = 2 ( NUMBER ) =
% A[BASE+1] = OCTAL NUMBER
% A[BASE+2] =
% 0 = MORE INFORMATION FOLLOWS NUMBER
% 1 = NO INFORMATION FOLLOWS NUMBER ( DELETION )
% 3 = A "FIX" REQUEST
% RETURNS NUMBER OF TOKENS IN STACK IN A[BASE],[33:15]
% RETURNS NUMBER OF "VERBS" FOUND IN A[BASE],[18:15]
% FOR EACH VERB =
% A [VERBLOCATION ],[33:15] = NUMERIC CODE ASSOCIATED WITH VERB
% A [VERBLOCATION ],[18:15] = NUMBER OF PARAMETERS FOR THE VERB
REAL I, K, P, S, MS, TYPE, VERBS, PLUSORMINUS;

BOOLEAN ERRFLG,PLUS;
LABEL NULL,STARTCOM,LOOKFERPARAMS,TOOLONG,SNEAKIN,GENSEQ,
FIXCHECK,SETERROR,FINISH,EXIT;
DEFINE ERRV = 3"000000000000036"&3[1:46:2];
DEFINE SEQV = 3"000060000000065"&3[1:46:2];
MS I= S I= BASE + 1; % INITIALIZE STACK INDICES
P I= STRADR; % ADDRESS OF WORK[1];

NULL:
IF TYPE=SCAN(P,A[S],"Z",ENDADR,SCANPTR[S=SBASE])=5 THEN % EOR
BEGIN
A[BASE] I= 0 & 5[3:45:3]; % NULL RECORD
GO TO EXIT;
END;
IF TYPE=2 THEN % INPUT STARTS WITH NUMBER
BEGIN % CHECK FOR DELETION (NUMBER FOLLOWED BY END OF RECORD)
K I= P; A[S I= S+1] I=
IF SCAN(P,I,"Z",ENDADR,SCANPTR[S=SBASE])=5 AND K=P THEN 1 ELSE 0;
A[BASE] I= 2 & 2[3:45:3]; % OCTAL NUMBER CODE AND STACK SIZE
GO TO EXIT;
END; % IF RECOR D STARTS WITH NUMBER

%.....
IF TYPE=1 THEN % ALPHANUMERIC TOKEN
BEGIN
STARTCOM:
PLUSORMINUS I= 0;
IF YOUFINDAVERB(A[S],VERBTABLE[0],NUMOFVERBS) THEN % C&E VERB
BEGIN
MS I= S; % SAVE STACK LOCATION OF VERB
VERBS I= VERBS + 1; % INCREMENT VERB COUNT
IF A[S],[33:15]=VSSMSG THEN % SS MESSAGE
BEGIN
IF VERBS NEQ 1 THEN GO TO SETERROR; % MUST BE FIRST VERB
IF SCAN(P,A[S I= S+1],"9",ENDADR,SCANPTR[S=SBASE]) GTR 2 THEN
GO TO SETERROR;
STICK(P,L,ENDADR); % TRANSFER MESSAGE TO ARRAY "L"
A[BASE] I= 0 & 4[3:45:3]; % CODE FOR SSMSG
GO TO EXIT;

```

```

00098300 T 0057
00098400 T 0057
00098500 T 0057
00098600 T 0057
00098700 T 0057
00098800 T 0057
00098900 T 0057
00099000 T 0057
00099100 T 0057
00099200 T 0057
00099300 T 0057
00099400 T 0057
00099500 T 0057
00099600 T 0057
00099700 T 0057
00099800 T 0057
00099900 T 0057
00100000 T 0057
00100100 T 0057
START OF SEGMENT ***** 26
00100200 T 0000
00100300 T 0000
00100400 T 0000
00100500 T 0000
00100600 T 0000
00100700 T 0000
00100800 T 0001
00100900 T 0002
00101000 T 0003
00101100 T 0010
00101200 T 0011
00101300 T 0013
00101400 T 0013
00101500 T 0013
00101600 T 0014
00101700 T 0015
00101710 T 0017
00101800 T 0024
00101900 T 0027
00102000 T 0027
00102100 T 0027
00102200 T 0027
00102300 T 0028
00102400 T 0029
00102500 T 0029
00102600 T 0029
00102700 T 0033
00102800 T 0034
00102900 T 0035
00103000 T 0036
00103100 T 0037
00103200 T 0038
00103300 T 0039
00103310 T 0047
00103400 T 0048
00103500 T 0050
00103600 T 0052

```

Moore-Busset 3-7500-10-67

END; % IF SS MESSAGE	00103700 T	0053
IF BOOLEAN(A[S],[8:1]) THEN % "SPECIAL" VERB = STOP SCANNING	00103800 T	0053
IF VERBS NEQ 1 THEN % "SPECIAL" VERBS NOT ALLOWED AFTER ";"	00103900 T	0054
BEGIN	00103910 T	0055
A[S]:=VERBTABLE[A[S],[9:9]]; GO TO SETERROR;	00103920 T	0056
END ELSE GO TO FINISH;	00103930 T	0058
LOOKFERPARAMS:	00104000 T	0058
IF (S:=S+1) GEQ PREVRCH THEN % DONT CLOBBER STACK	00104100 T	0059
BEGIN	00104200 T	0060
TOOLONG: ERRSW := "TOOMUCH";	00104300 T	0061
MSGPTR := 30;	00104400 T	0062
GO TO SETERROR;	00104500 T	0064
END;	00104600 T	0066
IF TYPE:=SCAN(P,A[S],"9",ENDADRS,SCANPTR[S-SBASE])=5 THEN	00104700 T	0066
GO TO FINISH;	00104710 T	0073
IF TYPE=3 THEN % PARAMETER IS SPECIAL CHARACTER	00104800 T	0074
BEGIN	00104900 T	0074
PLUSORMINUS := 0;	00105000 T	0075
IF A[S] = ";" THEN % END OF LOGICAL REQUEST	00105100 T	0076
BEGIN	00105200 T	0077
SNEAKIN: IF TYPE:=SCAN(P,A[S],"Z",ENDADRS,SCANPTR[S-SBASE])=5 THEN	00105300 T	0077
GO TO FINISH;	00105310 T	0085
IF TYPE=1 THEN % ALPHANUMERIC TOKEN FOLLOWS ";"	00105400 T	0086
BEGIN	00105500 T	0086
A[MS] := A[MS] & (S-MS=1)[18:33:15]; % PARAMETER COUNT	00105600 T	0087
MS := S; % NEXT VERB LOCATION	00105700 T	0090
GO TO STARTCOM;	00105800 T	0091
END; % IF ALPHANUMERIC STRING FOLLOWS ";"	00105900 T	0091
IF TYPE=3 THEN % SPECIAL CHARACTER FOLLOWS ";"	00106000 T	0091
BEGIN	00106100 T	0092
IF A[S]=";" THEN GO TO SNEAKIN; % REDUNDANT	00106200 T	0092
IF A[S]="'" THEN GO TO FINISH; % COMMENT AFTER ";"	00106300 T	0094
IF A[S]="+" THEN GO TO GENSEQ; % RESEQ SHORTHAND	00106400 T	0095
END; % IF SPECIAL CHARACTER FOLLOWS ";"	00106500 T	0097
GO TO SETERROR; % NOTHING ELSE ALLOWED AFTER ";"	00106600 T	0097
END; % IF PARAMETER = ";"	00106700 T	0097
IF A[S]="-" THEN PLUSORMINUS:=2 ELSE	00106800 T	0097
IF A[S]="+" THEN PLUSORMINUS:=3;	00106850 T	0100
GO LOOKFERPARAMS; % SAVE ALL SPECIAL CHARACTERS	00106900 T	0102
END; % IF PARAM IS A SPECIAL CHARACTER	00106950 T	0103
IF TYPE=1 THEN % ALPHANUMERIC PARAMETER	00107000 T	0103
BEGIN	00107050 T	0104
PLUSORMINUS:=0;	00107100 T	0104
GO LOOKFERPARAMS;	00107150 T	0105
END; % IF ALPHANUMERIC PARAMETER	00107200 T	0105
% TYPE = 2 , NUMERIC PARAMETER	00107250 T	0105
IF PLUSORMINUS GTR 0 THEN % SIGNED NUMBER	00107300 T	0105
BEGIN	00107350 T	0106
A[S] := A[S] & PLUSORMINUS[1:46:2]; % SIGNED NUMBER	00107400 T	0107
IF NOT PLUS THEN A[S:=S-1] := A[S+1]; PLUS := FALSE;	00107450 T	0109
END;	00107500 T	0114
PLUSORMINUS := 0;	00107550 T	0114
GO LOOKFERPARAMS;	00107600 T	0115
END; % IF A CANDE VERB	00107650 T	0115
MATCH(A[S],[6:42],"FIX ") GO TO FIXCHECK;	00107700 T	0115
GO TO SETERROR; % ALPHA STRING NOT A VERB OR A "FIX"	00107750 T	0119
END; % IF TOKEN IS ALPHANUMERIC	00107800 T	0121

```

IF TYPE=3 THEN % TOKEN IS A SPECIAL CHARACTER
  BEGIN
  IF A[S]="+" THEN % SEQUENCE REQUEST
    BEGIN
GENSEQ: PLUSORMINUS := 3; % NEXT PARAM WILL BE INCREMENT
    MS := S; % MARK STACK LOCATION OF "SEQ" VERB
    VERBS := VERBS + 1; PLUS := TRUE;
    A[S] := SEQV; % REPLACE "+" WITH SEQUENCE VERB CODE
    GO LOOKFERPARAMS;
    END; % IF; "+"
    IF A[S]="*" THEN % A "FIX" REQUEST
      BEGIN
FIXCHECK:
    IF TYPE:=SCAN(P,A[S],"9",ENDADRS,SCANPTR(S=SBASE)) NEQ 2 THEN
      GO TO SETERROR;
    IF (S:=S+1) GEQ PREVRCH THEN GO TO TOOLONG;
    A[S] := 3; % DENOTES A "FIX"
    A[BASE] := 2 & 2[3:45:3]; % "FIX" CODE AND PARAM. COUNT
    GO TO EXIT;
    END; % IF "FIX"
    IF A[S]=";" THEN GO TO NULL; % INPUT STARTS WITH ";"
    IF A[S]="'" THEN % COMMENT
      BEGIN
      A[BASE] := 0;
      GO TO EXIT;
      END;
SETERROR:
    A[MS+1] := A[S]; % PLACE ERRONEOUS TOKEN IN 1ST PARAM. POSITION
    A[MS] := ERRV; % CALL ON ERROR ROUTINE
    VERBS := VERBS + 1;
    ERRFLG := TRUE;
    S := MS + 2; % ADJUST FOR CORRECT PARAMETER COUNT
    END; % IF TOKEN IS A SPECIAL CHARACTER
FINISH:
    A[MS] := A[MS] & (S-MS-1)[18:33:15]; % PARAMETER COUNT
    A[BASE] := (S-BASE-1) & VERBS[18:33:15] &
    REAL(ERRFLG)[17:47:1] & 1[3:45:3];
EXIT:
  END PROCEDURE SCANNER;

```

```

00107850 T 0121
00107900 T 0121
00107950 T 0122
00108000 T 0123
00108050 T 0123
00108100 T 0124
00108150 T 0125
00108200 T 0127
00108250 T 0129
00108300 T 0132
00108350 T 0132
00108400 T 0133
00108450 T 0133
00108500 T 0134
00108510 T 0141
00108550 T 0142
00108600 T 0144
00108650 T 0145
00108700 T 0147
00108750 T 0148
00108800 T 0148
00108850 T 0149
00108900 T 0150
00108950 T 0151
00109000 T 0152
00109050 T 0153
00109100 T 0153
00109150 T 0153
00109200 T 0155
00109250 T 0157
00109300 T 0158
00109350 T 0159
00109400 T 0160
00109450 T 0160
00109500 T 0161
00109550 T 0164
00109600 T 0166
00109650 T 0168
00109700 T 0169

```

26 IS 174 LONG, NEXT SEG 8

```

*****
BOOLEAN STREAM PROCEDURE CHECKBIT(N,LST); VALUE N;
BEGIN SI:=LST; SKIP SB; SKIP N SB; IF SB THEN TALLY:=1;
CHECKBIT:=TALLY; END STREAM PROCEDURE CHECKBIT;

```

```

00109750 T 0057
00109800 T 0057
00109900 T 0057
00110000 T 0059

```

```

*****
REAL PROCEDURE CHANGEWHICH(A); ARRAY A[0];
BEGIN
REAL T, S, WHICH;

```

```

00200000 T 0061
00200100 T 0061
00200200 T 0061
00200300 T 0061

```

START OF SEGMENT ***** 27

```

LABEL BADPARAM,GETYPE,BADTYPE,BADWORD,EXIT;
BOOLEAN SRC,OBJ;
INTEGER L;
IF T := A[SREG]=BASE=A[BASE],FF = 0 THEN
BEGIN WHICH := 0; A[BASE] := "NOPARAM";
SREG := BASE; MSGPTR := 10;
GO EXIT
END;
S := SREG;
MATCH(VERBTABLE[A[S],[9:9]],[6:42],"TYPE ")
BEGIN WHICH:=1; GO GETYPE END;
INC(SREG);
IF T = 1 THEN
BEGIN
MATCH(WHICH:=0&A[SREG][6:6:42],"PASSWOR") WHICH:=5
ELSE MATCH(WHICH,"NAME ") WHICH:=6
ELSE GO BADPARAM;
GO EXIT;
END;
MATCH(A[SREG],[6:42],"TYPE ")
BEGIN WHICH := 1;
GETYPE;
MATCH(A[INC(SREG)],[6:42],"TO ") INC(SREG);
IF A[SREG],[3:3] NEQ 0 THEN
BEGIN
IF A[BASE+1] := ABS(FINDAWORD(A[SREG],TYPETABLE[1],
NUMOFTYPES)) = 0 THEN
BEGIN
BADTYPE;
WHICH := 0; MSGPTR := 20;
GO EXIT
END
ELSE
IF A[SREG] = ":" THEN
BEGIN
IF A[BASE+1] := ABS(FINDALETTER(A[INC(SREG)],
TYPETABLE[1], NUMOFTYPES))
= 0 THEN GO BADTYPE;
END ELSE GO BADPARAM;
GO EXIT
END;
% OF TYPE ANALYSIS
IF A[SREG],[3:3] = 0 THEN GO BADPARAM;
MATCH(A[SREG],[6:42],"SOURCE ")
BEGIN
IF OBJ THEN GO BADPARAM;
SRC := TRUE; INC(SREG);
END;
MATCH(A[SREG],[6:42],"OBJECT ")
BEGIN
IF SRC THEN GO BADPARAM;
OBJ := TRUE; INC(SREG);
END;
IF FINDAWORD(A[SREG], RESWRDTABLE[1],
NUMOFRESWRDS) NEQ 0 THEN
BADWORD:
BEGIN
WHICH:=0; MSGPTR:=29; GO EXIT;

```

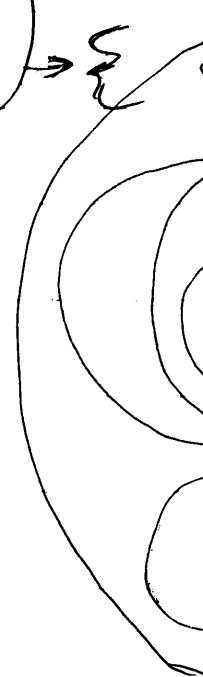
```

00200400 T 0000
00200410 T 0000
00200420 C 0000
00200500 T 0000
00200600 T 0003
00200700 T 0006
00200800 T 0008
00200900 T 0008
00201000 T 0010
00201010 C 0010
00201020 C 0014
00201100 T 0018
00201200 T 0019
00201300 T 0020
00201400 T 0020
00201500 T 0024
00201600 T 0029
00202100 T 0029
00202200 T 0032
00202300 T 0032
00202400 T 0032
00202500 T 0036
00202600 T 0037
00202700 T 0042
00202800 T 0044
00202900 T 0044
00203000 T 0047
00203100 T 0048
00203200 T 0049
00203300 T 0050
00203400 T 0052
00203500 T 0052
00203600 T 0055
00203700 T 0055
00203800 T 0056
00203900 T 0057
00204000 T 0059
00204100 T 0060
00204200 T 0062
00204300 T 0062
00204400 T 0063
00204500 T 0063
00204510 T 0065
00204515 T 0065
00204520 T 0068
00204525 T 0069
00204530 T 0071
00204535 T 0071
00204540 T 0071
00204545 T 0075
00204550 T 0076
00204555 T 0078
00204600 T 0078
00204700 T 0079
00204800 T 0080
00204900 T 0081
00205000 T 0081

```

redundant after
implementing the change
shown below

should be:
BEGIN
GETYPE;
WHICH := 1;



1472

```

END;
A[BASE] := A[SREG],[6:42]; % MUST BE FILE NAME
WHICH := 2;
IF A [INC(SREG)] EQL "/" THEN
  BEGIN
    IF L:=T=(SREG=S) EQL 0 THEN GO TO BADPARAM;
    IF L GEQ 2 THEN FIXUSER (S,A,EXIT,FALSE);
    IF A [INC(SREG)],[3:3] GTR 0 THEN
      A [BASE+5]:=A [SREG],[6:42]
    ELSE GO TO BADPARAM;
    INC (SREG);
  END
ELSE
  A [BASE+5]:=USERCODE;
  MATCH (A [SREG],[6:42],"TYPE ") GO GETYPE;
  MATCH(A[SREG],[6:42],"LOCK ")
  BEGIN WHICH := 4;
    INC(SREG);
  END ELSE WHICH := 3;
  MATCH(A[SREG],[6:42],"TO ") INC(SREG);
  IF A[SREG],[3:3] = 0 THEN GO BADPARAM;
  IF FINDAWORD(A[SREG], RESWRDTABLE[1],
    NUMOFRESWRDS) NEQ 0 THEN GO BADWORD;
  IF COMPAR(A[SREG],[6:42],"FACTOR ") OR
    COMPAR(A[SREG],[6:42],"SF ") THEN
  BEGIN INC(SREG); WHICH := 7; % CHANGING SAVE FACTOR
  MATCH(A[SREG],[6:42],"TO ") INC(SREG); % OPTIONAL WORD
  A[BASE+1]:=IF SRC THEN 1 ELSE IF OBJ THEN 2 ELSE 0;
  IF A[SREG],[3:3] = 0 THEN % NUMBER
  BEGIN A[BASE+2] := A[SREG]; GO TO EXIT;
  END ELSE GO BADPARAM;
  END; % IF "FACTOR"
  A [BASE+1]:=A [SREG],[6:42];
  IF A [INC(SREG)] EQL "/" THEN
    BEGIN
      IF L:=T=(SREG=S) EQL 0 THEN GO TO BADPARAM;
      IF L GEQ 2 THEN FIXUSER (S,A,EXIT,FALSE);
      IF A [INC(SREG)],[3:3] GTR 0 THEN
        A [BASE+6]:=A [SREG],[6:42]
      ELSE GO TO BADPARAM;
    END
  ELSE
    BEGIN DECR (SREG);
      A [BASE+6]:=USERCODE; END;
  END;
EXIT:
  IF WHICH NEQ 0 THEN
    IF T NEQ SREG - S THEN
      BEGIN WHICH := 0; SREG := S+T;
        MSGPTR := 10;
      END ELSE
      BEGIN
        IF WHICH NEQ 7 AND (SRC OR OBJ) THEN
          BEGIN
            A[SREG]:= (IF SRC THEN "SOURCE " ELSE "OBJECT ") & 6[3:45:3];
          BADPARAM:
            WHICH:=0; MSGPTR:=8;
          END;

```

OK after implementing
the change on lines
00202400 and
00202500
otherwise must set
WHICH := 1

```

00205100 T 0086
00205200 T 0086
00205300 T 0088
X0518- 00205400 P 0089
X0518- 00205405 C 0091
X0518- 00205410 C 0092
X0518- 00205415 C 0094
X0518- 00205420 C 0099
X0518- 00205425 C 0101
X0518- 00205430 C 0103
X0518- 00205435 C 0105
X0518- 00205440 C 0106
X0518- 00205445 C 0106
X0518- 00205450 C 0106
X0518- 00205455 C 0109
00205500 T 0112
00205600 T 0112
00205700 T 0116
00205800 T 0118
00205900 T 0121
00206000 T 0126
00206100 T 0128
00206200 T 0129
00206205 T 0131
00206210 T 0134
00206215 T 0137
00206220 T 0140
00206225 T 0144
00206230 T 0149
00206235 T 0151
00206240 T 0157
00206245 T 0157
X0518- 00206300 P 0157
X0518- 00206305 C 0160
X0518- 00206310 P 0162
X0518- 00206315 C 0162
X0518- 00206320 P 0165
X0518- 00206325 C 0169
X0518- 00206330 P 0172
X0518- 00206335 C 0174
X0518- 00206340 C 0176
X0518- 00206345 C 0176
X0518- 00206347 C 0176
X0518- 00206350 C 0177
00206400 T 0179
00206500 T 0180
00206600 T 0180
00206700 T 0182
00206800 T 0185
00206900 T 0186
00206910 T 0186
00206920 T 0186
00206930 T 0188
00206940 T 0189
00206950 T 0192
00206960 T 0193
00206970 T 0195

```

```

END;
ERRSW := A[SREG];
CHANGEWHICH := WHICH;
COMMENT COMPILER CHECK;
IF WHICH EQL 1 OR WHICH EQL 2 THEN % COMPILER NAME SPECIFIED
IF CHECKBIT((A[BASE+1]-1) DIV 1, RESTRICTEDLANGUAGES) THEN
BEGIN CHANGEWHICH := 0; MSGPTR :=49; END;
END OF CHANGEWHICH;

```

```

00206980 T 0195
00207000 T 0195
00207100 T 0196
00207110 T 0196
00207112 T 0196
00207114 T 0198
00207116 T 0202
00207200 T 0205

```

27 IS 213 LONG, NEXT SEG 8

```

%*****
BOOLEAN PROCEDURE TIMECHK(LCW,A); BOOLEAN LCW; ARRAY A[0];
%*****
COMMENT CHECK USERS TIME PERIOD FROM 24 BIT STRING;
BEGIN INTEGER TIM1,TIM2,SKP; BOOLEAN OK; LABEL EXIT;

```

```

00224400 T 0061
00224500 T 0061
00224600 T 0061
00224700 T 0061
00224800 T 0061

```

START OF SEGMENT ***** 28

```

%*****
BOOLEAN STREAM PROCEDURE TIMCK(STRG,SKP,TIM1); VALUE STRG;
%*****
BEGIN LOCAL SK; LABEL EXIT;
SI:=LOC STRG; SI:=SI+4; % POINT TO 24 BIT "TIME" STRING
DI:=LOC STRG; DS:=4 CHR; % EXTEND TO 48 BITS
SI:=SKP; DI:=LOC SK; DS:=WDS; % NO OF BITS TO SKIP (CRNT,TIME)
SI:=LOC STRG; SKIP SK SB; % POINT TO CURRENT HOUR
IF SB THEN TALLY:=1; TIMCK:=TALLY; TALLY:=SK;
24(IF SB THEN JUMP OUT ELSE % FIND 1ST "ON" BIT POSITION
BEGIN SKIP SB; TALLY:=TALLY+1; END); % COUNT EACH BIT SKIPPED
IF TOGGLE THEN ELSE GO TO EXIT; % NO BITS "ON"
SK:=TALLY; SI:=LOC SK; DI:=TIM1; DS:=WDS; % 1ST "ON" BIT POSITION
TALLY:=TALLY+1; % POSITION OF NEXT BIT
SK:=TALLY; SI:=LOC SK; DI:=SKP; DS:=WDS;
EXIT; END STREAM PROCEDURE TIMCK;

```

```

00224900 T 0000
00225000 T 0000
00225100 T 0000
00225200 T 0000
00225300 T 0000
00225400 T 0000
00225500 T 0001
00225600 T 0001
00225700 T 0002
00225800 T 0004
00225900 T 0005
00226000 T 0007
00226100 T 0007
00226200 T 0008
00226300 T 0009
00226400 T 0010

```

```

%*****
REAL STREAM PROCEDURE NEXTIME(STRG,SKP,REM); VALUE STRG,SKP,REM;
%*****
BEGIN SI:=LOC STRG; SI:=SI+4; DI:=LOC STRG; DS:=4CHR; %EXTEND
DI:=LOC NEXTIME; SI:=LOC STRG; SKIP SKP SB; % POINT TO NEXT BIT
REM(IF SB THEN BEGIN SKIP SB; TALLY:=TALLY+1; END
ELSE JUMP OUT); % FIND NEXT "0" BIT
TALLY:=TALLY+SKP; NEXTIME:=TALLY;
END STREAM PROCEDURE NEXTIME;

```

```

00226500 T 0011
00226600 T 0011
00226700 T 0011
00226800 T 0011
00226900 T 0012
00227000 T 0013
00227100 T 0015
00227200 T 0017
00227300 T 0017

```

```

%*****
PROCEDURE TIMMSG(STRTIME,FINTIME,OK);
%*****
VALUE STRTIME,FINTIME,OK; INTEGER STRTIME,FINTIME; BOOLEAN OK;
BEGIN

```

```

00227400 T 0018
00227500 T 0018
00227600 T 0018
00227700 T 0018
00227800 T 0018

```

```

*****
STREAM PROCEDURE MSG1(WORK);

*****
BEGIN DI:=WORK;
  DS:=49LIT"SORRY, YOU ARE NOT SCHEDULED FOR TIME AT THIS HOUR";
END MSG1;

```

```

00227900 T 0018
00228000 T 0018
START OF SEGMENT ***** 29
00228100 T 0000
00228200 T 0000
00228300 T 0000
00228400 T 0006

```

```

*****
STREAM PROCEDURE MSG2(WORK, STRTIME, FINTIME); VALUE STRTIME, FINTIME;
*****
BEGIN DI:=WORK;
  DS:=33LIT"YOUR NEXT SCHEDULED TIME IS FROM ";
  SI:=LOC STRTIME; DS:=4 DEC; DS:=4 LIT " TO ";
  SI:=LOC FINTIME; DS:=4 DEC;
END MSG2;

```

```

00228500 T 0007
00228600 T 0007
00228700 T 0007
00228800 T 0007
00228900 T 0007
00229000 T 0011
00229100 T 0013
00229200 T 0013

```

```

*****
STREAM PROCEDURE MSG3(WORK);
*****
BEGIN DI:=WORK;
  DS:=40LIT"PLEASE CALL BACK AT YOUR SCHEDULED TIME.";
END MSG3;

```

```

00229300 T 0013
00229400 T 0013
00229500 T 0013
00229600 T 0013
00229700 T 0014
00229800 T 0019

```

```

*****
STREAM PROCEDURE MSG4(WORK, FINTIME); VALUE FINTIME;
*****
BEGIN DI:=WORK;
  DS:=42LIT"YOU ARE SCHEDULED TO USE THE SYSTEM UNTIL ";
  SI:=LOC FINTIME; DS:=4 DEC;
END MSG4;

```

```

00229900 T 0019
00230000 T 0019
00230100 T 0019
00230200 T 0019
00230300 T 0020
00230400 T 0025
00230500 T 0026

```

```

*****
IF OK THEN
  BEGIN MSG4(WORK, FINTIME); TWXOUT(LL, WORK[0], 46, 2); END ELSE
  BEGIN MSG1(WORK); TWXOUT(LL, WORK[0], 49, 2);
  MSG2(WORK, STRTIME, FINTIME); TWXOUT(LL, WORK[0], 45, 2);
  MSG3(WORK); TWXOUT(LL, WORK[0], 40, 2); END;
END TIMMSG;

```

```

00230600 T 0026
00230700 T 0026
00230800 T 0027
00230900 T 0030
00231000 T 0034
00231100 T 0037
00231200 T 0040
29 IS 41 LONG, NEXT SEG 28

```

```

*****

```

```

00231300 T 0018

```



```

*****
IF REAL(BOOLEAN(TIMELIMITS.[24:24]) EQV
  BOOLEAN(REAL(NOT FALSE).[24:24])) EQL REAL(NOT FALSE) % ALL ON
OR TIMELIMITS.[24:24] EQL 0 THEN % NO BITS ON
  BEGIN TIMECHK:=TRUE; TIMERESTRICTBIT:=REAL(FALSE); GO EXIT; END;
TIM1:=0; TIM2:=SKP:=TIME(1)/216000 = 0.5; % CURRENT TIME
OK:=TIMCK(TIMELIMITS,SKP,TIM1); TIM1:=(TIM1 MOD 24) * 100;
TIM2:=(NEXTTIME(TIMELIMITS,SKP,TIM2:=24-SKP+TIM2) MOD 24)*100;
TIMMSG(TIM1,TIM2,OK); % SEND APPROPRIATE MESSAGE
TIMERESTRICTBIT := REAL(TRUE); NOTIFYBIT := REAL(FALSE);
NOTIFYTIME := 0; TIMECHK := OK;
EXIT; END PROCEDURE TIMECHK;

```

```

00231400 T 0018
00231500 T 0018
00231600 T 0020
00231700 T 0021
00231800 T 0023
00231900 T 0028
00232000 T 0031
00232100 T 0036
00232200 T 0041
00232300 T 0042
00232400 T 0047
00232500 T 0049

```

28 IS 56 LONG, NEXT SEG 8

```

*****
PROCEDURE OUTPUTNUMBERS(LCW,A); BOOLEAN LCW; ARRAY A(O);
*****
BEGIN
% FORMATS SEQUENCE NUMBERS FOR SPECIAL DEVICES
% " " CLEARS SCREEN
% "4" HOMES CURSER ( CONRAC ONLY )
% "1" LINE ERASE ( CONRAC ONLY )
% ">" SHIFT OUT CHARACTER
% QMK SHIFT IN CHARACTER
% "x" SHIFT CONTROL CHARACTER
% "*" LINE FEED (CONRAC ONLY)
% "=" CARRIAGE RETURN
% ARROW END OF TEXT CHARACTER
REAL I, PTR, LST, INC; BOOLEAN LF;

```

```

00234600 T 0061
00234700 T 0061
00234800 T 0061
00234900 T 0061
00235000 T 0061
00235100 T 0061
00235200 T 0061
00235300 T 0061
00235400 T 0061
00235500 T 0061
00235600 T 0061
00235700 T 0061
00235800 T 0061
00235900 T 0061
00236000 T 0061

```

START OF SEGMENT ***** 30

```

%.....
REAL STREAM PROCEDURE ADDRESS(W);
  BEGIN SI:=W; ADDRESS:=SI; END;

```

```

00236100 T 0000
00236200 T 0000
00236300 T 0000

```

```

%.....
STREAM PROCEDURE CONVERTANDFORM(PTR,N); VALUE N;
%.....
BEGIN LOCAL V,SV;
% CONVERTS SEQUENCE NUMBER TO DECIMAL FORM AND BLANKS
% LEADING ZEROS, UPDATES PTR AT FINISH
SI:=LOC N; DI:=LOC V; DS:=8DEC; DI:=DI-8; DS:=7FILL;
SI:=LOC V; DI:=PTR; DI:=DI+5; DI:=DC;
8(IF SC=" " THEN SI:=SI+1 ELSE DS:=CHR);
SV:=DI; SI:=LOC SV; DI:=PTR; DS:=WDS;
END STREAM PROCEDURE CONVERTANDFORM;

```

```

00236400 T 0001
00236500 T 0001
00236600 T 0001
00236700 T 0001
00236800 T 0002
00236900 T 0002
00237000 T 0002
00237100 T 0003
00237200 T 0004
00237300 T 0006
00237400 T 0007

```

```

%.....
STREAM PROCEDURE PUTCHAR(PTR,N,CHAR); VALUE N,CHAR;

```

```

00237500 T 0007
00237600 T 0007

```

```

*.....
BEGIN LOCAL SV;
% TRANSFERS "N" CHARACTERS FROM "CHAR" AND UPDATES
% PTR AT FINISH
SI:=LOC CHAR; DI:=PTR; DI:=DI+5; DI:=DC;
SI:=SI+8; SI:=SI*N; DS:=N CHR;
SV:=DI; SI:=LOC SV; DI:=PTR; DS:=WDS;
END STREAM PROCEDURE PUTCHAR;

```

```

00237700 T 0007
00237800 T 0007
00237900 T 0008
00238000 T 0008
00238100 T 0008
00238200 T 0009
00238300 T 0010
00238400 T 0011

```

```

LST:=SEQLAST; INC:=INCREMENT; PTR:=ADDRESS(WORK[0]);
IF DEVICE=BIDS THEN PUTCHAR(PTR,1,""); % CLEAR THE SCREEN
IF (LF:=DEVICE=CONRAC) THEN PUTCHAR(PTR,1,"4"); % HOME THE CURSER
FOR I:=1 STEP 1 UNTIL 10 DO % WRITE OUT 10 SEQUENCE NUMBERS
BEGIN
PUTCHAR(PTR,2,">"); % SHIFT OUT, SHIFT CONTROL
CONVERTANDFORM(PTR,(LST:=LST+INC)); % FORMED SEQUENCE NUMBER
PUTCHAR(PTR,2,(12)&(" ")[36:42:6]); % SHIFT CONTROL, SHIFT OUT
IF I LSS 10 THEN % FORM CARRIAGE RETURN AND LINE FEED
BEGIN
PUTCHAR(PTR,1,""); % CARRIAGE RETURN
IF LF THEN PUTCHAR(PTR,1,"*"); % LINE FEED
END;
END I LOOP;
PUTCHAR(PTR,1,LEFTARROW); % TERMINATE STRING WITH ETX
CHKSEQNMBR:=LST; % LAST SEQUENCE NUMBER TO BE TRANSMITTED
TWXOUT(LL,WORK[0],200,-1); % SEND SEQUENCE STRING
END PROCEDURE OUTPUTNUMBERS;

```

```

00238500 T 0011
00238600 T 0016
00238700 T 0019
00238800 T 0022
00238900 T 0024
00239000 T 0024
00239100 T 0025
00239200 T 0028
00239300 T 0030
00239400 T 0031
00239500 T 0031
00239600 T 0033
00239700 T 0035
00239800 T 0035
00239900 T 0037
00240000 T 0039
00240100 T 0040
00240200 T 0042

```

30 IS 46 LONG, NEXT SEG 8

```

*****
% FORWARD DECLARATIONS GO HERE.
*****
PROCEDURE BYE(LCW,A); BOOLEAN LCW; ARRAY A[0]; FORWARD;
PROCEDURE GETSOMETHINGTODO(LCW,A); BOOLEAN LCW; ARRAY A[0]; FORWARD;
PROCEDURE SSMSG; FORWARD;
PROCEDURE REMEMBERTHIS(LCW); BOOLEAN LCW; FORWARD;
PROCEDURE MCPMSG(LCW,A); BOOLEAN LCW; ARRAY A[0]; FORWARD;
PROCEDURE LINECLEAR(LCW,A); BOOLEAN LCW; ARRAY A[0]; FORWARD;
PROCEDURE SETUPFILES(LCW,A); BOOLEAN LCW; ARRAY A[0]; FORWARD;
PROCEDURE DOGCATCHER(LCW,A); BOOLEAN LCW; ARRAY A[0]; FORWARD;
*****
REAL PROCEDURE GETPSWD(WORK,STACK,MAX,EADRS);
*****
VALUE MAX,EADRS; REAL STACK,MAX,EADRS,WORK;
BEGIN LABEL DUMMY;

REAL STREAM PROCEDURE GTPSWD(WORK,STACK,MAX,EADRS); VALUE MAX,EADRS;
*****
COMMENT SPECIAL SCANNER FOR OBTAINING PASSWORD AND/OR CHARGE
WHICH MAY START WITH A DIGIT AND CONTAIN SPECIAL CHARACTERS;
BEGIN LOCAL SV,DV,DN,H,N; LABEL LOOP,EXIT;

```

```

00900000 T 0061
00900100 T 0061
00900200 T 0061
00900300 T 0061
00900400 T 0061
00900500 T 0061
00900600 T 0061
00900700 T 0061
00900800 T 0061
00900900 T 0061
00900910 T 0061
00901000 T 0061
00901005 T 0061
00901006 T 0061
00901007 T 0061
00901008 T 0061
00901009 T 0000
00901010 T 0000
00901015 T 0000
00901020 T 0000
00901025 T 0000

```

START OF SEGMENT ***** 31

14121 Moore Business Forms, Inc. 87

```

DI:=EADRS; DS:=LIT LEFTARROW; % END OF RECORD MARK
SI:=WORK; DI:=STACK; DN:=DI;
63(IF SC=" " THEN SI:=SI+1 ELSE JUMP OUT);
IF TOGGLE THEN GO TO EXIT; % BLANK RECORD
IF SC="H" THEN % CHECK FOR "HELLO"
  BEGIN
  DI:=LOC H; DS:=5LIT"HELLO"; DI:=DI-5;
  IF 5SC=DC THEN ELSE SI:=SI-5;
  END;
MAX(DI:=DN; DS:=8LIT"0      "; DI:=DI-7;
LOOP:
IF SC=" " THEN BEGIN SI:=SI+1; GO TO LOOP; END;
IF SC="," THEN BEGIN SI:=SI+1; GO TO LOOP; END;
IF SC=LEFTARROW THEN % CHECK FOR END OF RECORD
  BEGIN
  SV:=SI; SI:=LOC SV; SI:=SI+5;
  DV:=DI; DI:=LOC EADRS; DI:=DI+5;
  IF 3SC=DC THEN JUMP OUT TO EXIT;
  SI:=SV; DI:=DV;
  END;
DS:=CHR; TALLY:=1;
6(IF SC=" " THEN JUMP OUT; IF SC="," THEN JUMP OUT;
IF SC=LEFTARROW THEN
  BEGIN
  SV:=SI; SI:=LOC SV; SI:=SI+5;
  DV:=DI; DI:=LOC EADRS; DI:=DI+5;
  IF 3SC=DC THEN;
  SI:=SV; DI:=DV; IF TOGGLE THEN JUMP OUT;
  END;
  DS:=CHR; TALLY:=TALLY+1;
H:=TALLY; SV:=SI; SI:=LOC H; SI:=SI+7; DI:=DN; DS:=CHR; % SIZE
DI:=DI+7; DN:=DI; SI:=SV;
TALLY:=N; TALLY:=TALLY+1; N:=TALLY); % END "MAX" LOOP
EXIT:
TALLY:=N; GTPSWD:=TALLY;
END STREAM PROCEDURE GTPSWD;

```

```

GETPSWD:=GTPSWD(WORK,STACK,MAX,EADRS);
END PROCEDURE GETPSWD;

```

```

*****
PROCEDURE      HITHERE (LCW, A); BOOLEAN LCW; ARRAY A[0];
*****
  BEGIN
  STREAM PROCEDURE  ANS(W);

  BEGIN DI := W;
  DS := 24 LIT "ENTER USER CODE, PLEASE=";
  END ANS;

```

```

00901030 T 0000
00901035 T 0000
00901040 T 0001
00901045 T 0004
00901050 T 0004
00901055 T 0005
00901060 T 0005
00901065 T 0007
00901070 T 0008
00901075 T 0008
00901080 T 0010
00901085 T 0010
00901090 T 0012
00901095 T 0014
00901100 T 0014
00901105 T 0014
00901110 T 0015
00901115 T 0016
00901120 T 0017
00901125 T 0018
00901130 T 0018
00901135 T 0018
00901140 T 0020
00901145 T 0021
00901150 T 0021
00901160 T 0022
00901170 T 0023
00901180 T 0023
00901190 T 0025
00901200 T 0025
00901210 T 0026
00901220 T 0027
00901230 T 0028
00901240 T 0030
00901250 T 0030
00901260 T 0031

00901270 T 0032
00901280 T 0036
31 IS 39 LONG, NEXT SEG 8

01000000 T 0061
01000100 T 0061
01000200 T 0061
01000300 T 0061
01000900 T 0061
START OF SEGMENT ***** 32
01001000 T 0000
01001100 T 0000
01001200 T 0003

```

STREAM PROCEDURE LNO(W,T,UC,LL); VALUE T,LL;	01001300 T	0003
BEGIN	01001400 T	0003
LABEL AFT,EVE;	01001500 T	0004
DI := W;	01001600 T	0004
SI := LOC T; SI := SI+7;	01001700 T	0004
IF SC = "A" THEN DS := 14 LIT "GOOD MORNING, "	01001800 T	0004
ELSE	01001900 T	0007
BEGIN SI := LOC T;	01002000 T	0007
IF SC LSS "1" THEN	01002100 T	0007
BEGIN UPS;	01002200 T	0008
IF SC LSS "6" THEN	01002300 T	0009
AFT: DS := 16 LIT "GOOD AFTERNOON, "	01002400 T	0009
ELSE	01002500 T	0012
EVE: DS := 14 LIT "GOOD EVENING, "	01002600 T	0012
END	01002700 T	0015
ELSE	01002800 T	0015
BEGIN	01002900 T	0015
UPS;	01003000 T	0015
IF SC = "2" THEN GO AFT ELSE GO EVE;	01003100 T	0015
END;	01003200 T	0016
END;	01003300 T	0016
SI := UC; SI := SI + 1;	01003400 T	0016
DS := 15 CHR;	01003500 T	0017
DS := 18 LIT " YOU HAVE STATION ";	01003600 T	0017
SI := LOC LL; DS := 2 DEC;	01003700 T	0020
DS := 4 LIT " ";	01003800 T	0020
END OF LNO;	01003900 T	0021
STREAM PROCEDURE PSWRD (W);	01004000 T	0021
BEGIN	01004100 T	0021
LOCAL T;	01004200 T	0022
DI := W;	01004300 T	0022
DS:=17LIT"AND YOUR PASSWORD";	01004400 T	0022
% VOID	01004500 T	0024
END PSWRD;	01004600 T	0024
STREAM PROCEDURE CHRQ(W);	01004700 T	0025
BEGIN	01004800 T	0025
DI := W;	01004900 T	0025
DS := 18 LIT "ENTER CHARGE CODE=";	01005000 T	0025
END CHRQ;	01005100 T	0027
LABEL START,ERR,BAD,EXIT,SCHED,CASE4,STARTANEW;	01005200 T	0028
LABEL WINDUP,PSWD,LOGGON,CHARG,NEXTBLOCK;	01005300 T	0028
INTEGER I;	01005400 T	0028
REAL T,USR,FILE;	01005500 T	0028

```

BOOLEAN MTCH;
CASE DOING OF
BEGIN
  BEGIN % CASE 0 .....
  IF LOGGEDON THEN
    BEGIN
      IF SCHEDULELINE THEN GOERR("HELLO ",40);
      IF FILENAME NEQ 0 THEN
        IF TPCOUNT NEQ 0 OR FILENAME NEQ SOURCEFILE THEN
          IF NOT NOSAVE THEN
            BEGIN
              NOSAVE:=TRUE; GOERR("NO SAVE",48);
            END;
          CLEAR(A[BASE],3); % ZERO OUT 4 WORDS OF STACK
          T:=GETPSWD(WORK[1],A[BASE+1],3,WORKENDADRS); % "HELLO" INPUT
          IF T GTR 0 THEN % GOT SOMETHING AFTER "HELLO"
            COMMENT PROCEDURE "BYE" DOES NOT DISCONNECT WHEN
              A[BASE] GEQ 50, DOES NOT PRINT MESSAGES WHEN
                A[BASE] GEQ 100;
            BEGIN A[BASE]:=124+T;
              BYE(LCW,A[*]); LOGOFF(LCW,A[*]);
              GO TO PSWD;
            END; % IF PARAMETERS TO HELLO
            A[BASE] := 75; % FLAG AS HELLO
            BYE(LCW, A[*]); LOGOFF(LCW,A[*]);
            IAM := VHELLO;
            WANT := WAITFOR := 0;
          END; % IF HELLO
          IOTOT := ETIME := PTIME := INPUTCTR := 0;
          IF SCHEDULELINE THEN
            BEGIN
              A[BASE]:=126; A[BASE+1]:=USERCODE;
              GO TO PSWD;
            END; % IF LOGGEDON
          START: IF NOT LOGGINGON THEN
            BEGIN
          STARTANEW: ANS(WORK);
            TWXOUT(LL, WORK[0], 24, NOCRLF);
            LOGGINGON := TRUE;
            END;
            DOING := 1;
            FILESOK := FALSE;
            WAITFOR := VINPUT;
            A[BASE]:=TEMP:=5;
          END OF CASE 0; % .....
          BEGIN % CASE 1 .....
          CLEAR(A[BASE],3); % ZERO OUT 4 WORDS OF STACK
          T:=GETPSWD(WORK[1],A[BASE+1],3,WORKENDADRS); LOGGINGON:=TRUE;
          IF T = 0 THEN GO TO ERR; % NO INPUT
          IF YOUFINDAVERB (A[BASE+1],VERBTABLE[0],NUMOFVERBS) THEN
            BEGIN IF A[BASE+1].CF NEQ VHELLO THEN
              IF A[BASE+1].CF = VBYE THEN
                BEGIN
                  IAM := VBYE; USERCODE := "GONE ";
                  IOTOT := ETIME := PTIME := INPUTCTR := 0;
                  DOING := A[BASE] := 0;
                  GO EXIT

```

```

01005510 T 0028
01005600 T 0028
01005700 T 0028
01005800 T 0028
01005900 T 0028
01006000 T 0029
01006100 T 0030
01006110 T 0037
01006120 T 0038
01006130 T 0041
01006140 T 0043
01006150 T 0043
01006160 T 0052
01006170 T 0052
01006200 T 0053
01006300 T 0058
01006400 T 0059
01006500 T 0059
01006600 T 0059
01006700 T 0059
01006800 T 0061
01006900 T 0065
01007000 T 0065
01007100 T 0065
01007200 T 0066
01007300 T 0070
01007400 T 0071
01007500 T 0072
01007600 T 0072
01007700 T 0076
01007800 T 0077
01007900 T 0077
01008000 T 0081
01008100 T 0081
01008200 T 0081
01008300 T 0083
01008400 T 0083
01008500 T 0085
01008600 T 0086
01008700 T 0089
01008800 T 0089
01008900 T 0089
01009000 T 0092
01009100 T 0092
01009200 T 0094
01009300 T 0095
01009310 T 0095
01009400 T 0096
01009500 T 0104
01009600 T 0106
01009700 T 0110
01009800 T 0112
01009900 T 0115
01010000 T 0115
01010100 T 0117
01010200 T 0122
01010300 T 0123

```

```

                END % IF BYE
            ELSE
                LITOUT("FUNNY..",1); % IF NOT HELLO
ERR:           MSGPTR := 21;
                GO TO STARTANEW;
                END; % IF A VERB ( OR NO INPUT )
                A[BASE] := 124 + T;
                IF T LSS 2 THEN % NO PASSWORD GIVEN YET
                    BEGIN PSWRD(WORK);
PSWD:          TWXOUT(LL,WORK[0],17,1);
                IF A[BASE] LSS 126 THEN % NO PASSWORD GIVEN YET
                    BLACKOUT;
                END;
                I := ((A[BASE+1].[6:21] + A[BASE+1].[27:21]) MOD 29) + 1;
NEXTBLOCK:    LLDISKREAD:=USERBASE+I;
                DOING := 2;
                WANT := VDISKREAD;
                TEMP:=8; % MUST NOT BE 4 = SSFILE USES AS FLAG
                END CASE 1; % .....
                BEGIN % CASE 2 .....
                FOR T := 0 STEP 2 UNTIL 28 DO
                MATCH(WORK[T],A[BASE+1].[6:42])
                BEGIN
                USR := WORK[T+1].[33:15]; % RELATIVE DISK ADDRESS
                USERLOC:=USERBASE+USR;
                LLDISKREAD := USERLOC;
                WANT := VDISKREAD;
                DOING := 3;
                GO EXIT
                END;
                IF (I:=WORK[1].[18:15]) GTR 0 THEN GO TO NEXTBLOCK; %LINK
                WORK[1] := "*****"; % IF NOT MATCHED
                DOING := 3;
                END CASE 2; % .....
                BEGIN % CASE 3 .....
                RESTRICTEDLANGUAGES:=WORK[8];
                MOVE(2,WORDS,WORK[9],RESTRICTEDVERBS);
                IF SCHEDULELINE THEN
                BEGIN
                LOGON(LCW,A[*]); GO TO SCHED;
                END;
                USERCODE:=WORK[1];
                A[BASE+6]:=WORK[2]; % SAVE PASSWORD
                CHARGE:=WORK[3];
                TIMELIMITS:=WORK[7];
                OPTIONBITS:=WORK[11]; % USERS "SPECIAL" OPTIONS
                MOVE(2,WORDS,WORK[4],A[BASE+4]); % NAME FROM USERS FILE
                IF A[BASE] GTR 125 THEN GO TO CASE4; % HAVE PASSWORD
                % VOID
                DOING := 4;
                WAITFOR := VINPUT;
                END CASE 3; % .....
                BEGIN % CASE 4 .....
                IF A[BASE] LSS 126 THEN % NO PASSWORD GIVEN YET
                A[BASE]:=125+GETPSWD(WORK[1],A[BASE+2],2,WORKENDADRS);
CASE4:        LOGGINGON := FALSE;

```

```

01010400 T 0124
01010500 T 0126
01010600 T 0126
01010700 T 0127
01010800 T 0129
01010900 T 0131
01011000 T 0131
01011100 T 0132
01011200 T 0133
01011300 T 0135
01011400 T 0137
01011500 T 0138
01011600 T 0139
01011700 T 0139
01011710 T 0143
01011800 P 0144
01011900 T 0145
01012000 T 0146
01012100 P 0147
01012200 T 0148
01012300 T 0148
01012400 T 0148
01012500 T 0150
01012600 T 0150
01012700 T 0154
01012710 P 0156
01012720 T 0158
01012800 T 0159
01012900 T 0160
01013000 T 0161
01013100 T 0161
01013110 T 0163
01013200 T 0166
01013300 T 0167
01013400 T 0168
01013500 T 0169
01013600 T 0169
01013700 T 0170
01013800 T 0172
01013810 T 0173
01013820 T 0174
01013830 T 0178
01013900 T 0178
01014000 T 0179
01014100 T 0181
01014110 T 0183
01014200 T 0184
01014210 T 0186
01014300 T 0188
01014400 T 0190
01014500 T 0190
01014600 T 0191
01014700 T 0191
01014800 T 0192
01014900 T 0192
01015000 T 0193
01015100 T 0199

```

```

MATCH(USERCODE,A[BASE+1],[6:42])
MATCH(A[BASE+6],A[BASE+2],[6:42]) % PASSWORD
BEGIN
MATCH(A[BASE],USERCODE) MTCH I= TRUE;
IF RESTART AND NOT MTCH THEN % SCRATCH "OLD" WRK FILES
BEGIN
USR:=A[BASE]; T:=LL&13[33:41:7]; FIL:=MAKEFN("IS",LL);
REMOVEFILE(T,FIL,USR);
REMOVEFILE(T,FIL,[6:36],USR);
REMOVEFILE(T,FIL&"P"[12:42:6],USR);
REMOVEFILE(T,FIL&"T"[12:42:6],USR);
END;
IF NOT RESTART OR RESTART AND NOT MTCH THEN % CHECK TIME
IF NOT TIMECHK(LCW,A) THEN
BEGIN WORK[0]:"$0 "; WORK[0],[12:6]:=LEFTARROW;
TWXOUT(LL,WORK[0],3,-1); % DISCONNECT THE LINE
IAM:=WANT:=WAITFOR:=0; GO TO EXIT;
END;
IF CHARGE = 0 THEN % CHARGE CODE NEEDED
BEGIN IF A[BASE] = 127 THEN % CHARGE ALREADY INPUT
BEGIN CHARGE := A[BASE+3],[6:42];
GO TO LOGGON;
END % IF A[BASE]=127
ELSE % ASK FOR IT
BEGIN
CHRG(WORK); TWXOUT(LL,WORK[0],18,NOCRLF);
WAITFOR:=VINPUT;
DOING I= 5;
GO EXIT;
END; % ASKING FOR CHARGE
END % IF CHARGE = 0
ELSE GO TO LOGGON;
END; % IF MATCHED
BAD: USERCODE := 0;
LITOUT("BADCODE",1);
MSG#TR := 21;
GO TO STARTANEW;
END CASE4; % .....
BEGIN % CASE 5 .....
T:=GETPSWD(WORK[1],CHARGE,1,WORKENDADRS); CHARGE,[1:5]:=0;
IF T = 0 THEN GO TO CHARG; % MUST INPUT SOMETHING
LOGGON: LOGON(LCW,A[*]);
IOTOT := ETIME I= PTIME I= INPUTCTR I= 0;
WAITFOR := IAM I= VLISTNEWS;
FIND(LL&IAM[33:41:7],0,"NEWS ","CANDE ","SITE ");
DOING I= 6;
END CASE5; % .....
BEGIN % CASE 6 .....
IF EVENTS[1] LEQ 0 THEN GO TO WINDUP; % NO FILE
IF EVENTS [3] LSS 0 THEN GO TO WINDUP; % NO RECORDS %0523-
LLDISKREAD:=EVENTS [2],[18:15]; %0523-
WANT := VDISKREAD;
DOINGI:=2; % TRANSFER TO %0523-
IAM:=VSSFILE; % SSFILE %0523-
A [BASE+3]:=LL; %0523-
END CASE6; % .....
BEGIN % CASE 7 .....

```

```

01015200 T 0202
01015300 T 0202
01015400 T 0210
01015500 T 0211
01015600 T 0214
01015605 T 0216
01015610 T 0216
01015615 T 0220
01015620 T 0223
01015625 T 0225
01015630 T 0229
01015635 T 0232
01015700 T 0232
01015800 T 0235
01015900 T 0237
01016000 T 0241
01016100 T 0243
01016200 T 0247
01016300 T 0247
01016400 T 0248
01016500 T 0249
01016600 T 0253
01016700 T 0253
01016800 T 0253
01016900 T 0253
01017000 T 0254
01017100 T 0257
01017200 T 0258
01017300 T 0259
01017400 T 0261
01017500 T 0261
01017600 T 0261
01017700 T 0261
01017800 T 0261
01017900 T 0262
01018000 T 0263
01018100 T 0264
01018200 T 0266
01018300 T 0266
01018400 T 0266
01018500 T 0271
01018600 T 0272
01018700 T 0274
01018800 T 0279
01018900 T 0280
01019000 T 0283
01019100 T 0284
01019200 T 0284
01019300 T 0284
01019350 C 0286
01019400 P 0287
01019500 T 0290
01019550 C 0290
01019600 P 0291
01019650 C 0292
01019700 T 0294
01019800 T 0294

```

```

WINDUP: IF CCLONG THEN CCCOM(LL,USERCODE,1); % SET LONG CARRIAGE
        IF QUICKLOG THEN GO TO SCHED; % DONT PRINT GREETINGS
        WORK[0] := DATE;
        TIMEOFDAY(WORK[1],T:=TIMECONV(LOGONTIME,TRUE));
        TWXOUT(LL,WORK[0],18,1);
        LNO(WORK[0],T,A[BASE+4],LL);
        TWXOUT(LL,WORK[0],52,2);
SCHED:  IF NOT RESTART THEN LINECLEAR(LCW,A[*]);
        WRITEASTERISK:=TRUE; NOSTOP:=FALSE;
        FILENAME := 0;
        DOING := WANT := 0;
        SOURCEFILE := OBJECTFILE := 0;
        IAM := IF RESTART THEN 0 ELSE VSETUPFILES;
        GO EXIT
        END CASE 7; % .....
END ALL CASES;

```

```

EXIT:
END OF HITHERE;

```

```

%*****%
PROCEDURE BREAK(LCW,A); BOOLEAN LCW; ARRAY A[0];
%*****%
        BEGIN
LABEL T;
                WRITEASTERISK := TRUE;
                IF R(ALLMCPMSG) = 0 THEN LINECLEAR(LCW,A[*]) ELSE
                IAM := 0;
        END BREAK;

```

```

%*****%
PROCEDURE WRU(LCW,A); BOOLEAN LCW; ARRAY A[0];
%*****%
        BEGIN
LABEL T;
                WRITEASTERISK := TRUE;
                INITIATEDWHILEBUSY := TRUE;
                BREAKORWRU:=TRUE;% WANT:=WAITFOR:=VLINECLEAR;
                IF REAL(ALLMCPMSG)=0 THEN IAM:=0 ELSE
                MATCH(RUN2,"CANDE ") IAM:=0 ELSE
                WANT:=WAITFOR:=VMCPMSG;
        END WRU;

```

```

01019910 T 0294
01019920 T 0298
01020000 T 0299
01020100 T 0301
01020200 T 0303
01020300 T 0305
01020400 T 0308
01020500 T 0309
01020600 T 0313
01020700 T 0317
01020800 T 0319
01020900 T 0320
01021000 T 0322
01021100 T 0325
01021200 T 0325
01021300 T 0329

```

```

START OF SEGMENT ***** 33
33 IS 9 LONG, NEXT SEG 32
01021400 T 0330
01021500 T 0330
32 IS 334 LONG, NEXT SEG 8

```

```

02000000 T 0061
02000100 T 0061
02000200 T 0061
02000300 T 0061
02000350 T 0061
START OF SEGMENT ***** 34
02000400 T 0000
02000500 T 0002
02000600 T 0005
02000700 T 0007
34 IS 8 LONG, NEXT SEG 8

```

```

03000000 T 0061
03000100 T 0061
03000200 T 0061
03000300 T 0061
03000350 T 0061
START OF SEGMENT ***** 35
03000400 T 0000
03000500 T 0002
03000600 T 0004
03000700 T 0006
03000800 T 0009
03000810 T 0012
03000900 T 0016
35 IS 17 LONG, NEXT SEG 8

```



```

*****%
PROCEDURE LINECLEAR(LCW,A); BOOLEAN LCW; ARRAY A[0];
*****%
  BEGIN
  LABEL EXIT;

  IF RUNNING THEN MATCH(RUN2,"CANDE ") GO EXIT;
  BUSYLINE := FALSE;
  INITIATEDWHILEBUSY := FALSE;
  SAVEDWHILEINITBUSY := FALSE;
  RESTARTNEEDED := FALSE;
  ALLMCPMSG := FALSE;
  DISCONNECTING := FALSE;
  RESTART := FALSE;
  SEQMODE := FALSE;
  HELPTOGGLE := 0;
  DATAENABLED := FALSE;
  BREAKORWRU := FALSE;
  INQUIRY := FALSE; CONTINUEBIT:=FALSE;
  SCWPREV := PREVRCW := LLWORDS; BASE := SBASE;
  WANT := WAITFOR := WAITINGSCH := LLINFO := 0;
  WHILE REAL(THINGSTODO) GTR 0 DO GETSOMETHINGTODO(LCW,A[*]);
  CLEAR(A[SBASE],30); % FOR REPLACIT
  IF EQUATED THEN ENTERESP(=LL,"0,A[*]); % GIVE BACK ESP DISK
  EQUATED := FALSE;
  MSGRTR := 18;
EXIT: IAM:=0;
  END PROCEDURE LINECLEAR;

```

```

04000000 T 0061
04000100 T 0061
04000200 T 0061
04000300 T 0061
04000400 T 0061
START OF SEGMENT ***** 36
04000410 T 0000
04000500 T 0003
04000600 T 0005
04000700 T 0008
04000800 T 0010
04000900 T 0012
04001000 T 0014
04001100 T 0017
04001200 T 0019
04001300 T 0021
04001400 T 0024
04001500 T 0026
04001600 T 0028
04001700 T 0033
04001800 T 0036
04001900 T 0039
04002000 T 0045
04002010 T 0046
04002020 T 0049
04002100 T 0052
04002200 T 0053
04002300 T 0054
36 IS 55 LONG, NEXT SEG 8

```

```

*****%
PROCEDURE INITIALIZECTRANDBASE;
*****%
  BEGIN
  STREAM PROCEDURE USERSDATE(L,DAT);

  BEGIN
  DI:=L; DS:=23LIT"USERS/CANDE FILE DATED ";
  SI:=DAT; DS:=8CHR; DS:=LIT"+";
  END USERSDATE;

```

```

05000000 T 0061
05000100 T 0061
05000200 T 0061
05000300 T 0061
05000310 T 0061
START OF SEGMENT ***** 37
05000315 T 0000
05000320 T 0000
05000325 T 0003
05000330 T 0004

```

```

STREAM PROCEDURE ERR (W);
  BEGIN
  DI:=W; DS:=31LIT"USERS/CANDE HAS MORE THAN 1 ROW"; DS:=LIT"+";
  END OF ERR;

```

```

%0521- 05000332 C 0004
%0521- 05000334 C 0004
%0521- 05000336 C 0005
%0521- 05000338 C 0010

```

```

INTEGER I;
REAL USERFILEHEADER;

```

```

%0521- 05000340 C 0010
05000350 T 0010

```

Moore Business Forms, Inc. sv 14127

```

CLEAR(CTRANDBASE[0],15); % REMOVE
CLEAR(WORK[0],59); % ANY GARBAGE,
MSGFILELOC := EVENTS[2],[20:28];
TANKFILE := ABS(EVENTS[3]);
USERFILEHEADER := EVENTS[2],[5:15];
SYSTEM:=(EVENTS[2],[2:2])x(100);
ROWINUSE := 15;
NOFILES := FALSE;
USERSDATE(L,EVENTS[4]); TWXOUT(0,L[0],32,1); % ACCESS DATE
DISKWAIT(1,WORK[*],30,USERFILEHEADER);
FOR I:=11 STEP 1 UNTIL 29 DO
  IF WORK [I] GTR 0 THEN
    BEGIN
      ERR (L); TWXOUT (0,L [0],32,1);
      I:=0/0; % KILL CANDE
    END;
    USERBASE:=WORK [10];
    IF EVENTS[3] GTR 0 THEN % OLD TANKS EXIST,
  BEGIN
    DISKWAIT(1,WORK[*],30,TANKFILE);
    MOVE(16,WORDS,WORK[10],CTRANDBASE[0]);
    TANKSIZE := WORK[8];
    ROWINUSE := WORK[6],CF;
    ROWCOUNT := WORK[6],FF;
    NOTANKS := FALSE;
  END;
END INITIALIZECTRANDBASE;

```

```

05000400 T 0010
05000500 T 0012
05000600 T 0013
05000700 T 0015
05000800 T 0016
05000860 T 0017
05000900 T 0019
05001000 T 0020
05001010 T 0022
05001020 T 0025
05001025 C 0027
05001030 P 0029
05001035 C 0030
05001040 P 0030
05001045 C 0033
05001050 P 0034
05001055 C 0036
05001100 T 0037
05001200 T 0038
05001300 T 0039
05001400 T 0041
05001500 T 0043
05001600 T 0044
05001700 T 0046
05001750 T 0047
05001800 T 0049
05001900 T 0049

```

37 IS 52 LONG, NEXT SEG 8

```

*****
PROCEDURE DISKCHUNK;
*****
BEGIN
REAL XTRADSK;
BOOLEAN TANKNOTPRESENT;
IF NOFILES THEN % SET UP THE FILE ADDRESSES,
BEGIN
TANKNOTPRESENT := EVENTS[3] LEQ 0;
INITIALIZECTRANDBASE;
END ELSE TANKNOTPRESENT := TRUE;
BEGIN % MUST REALLY BE A NEW CHUNK,
ROWINUSE,[44:4] := ROWINUSE + 1;
IF CTRANDBASE[ROWINUSE] NEQ 0 THEN
BEGIN
XTRADSK := CTRANDBASE[ROWINUSE];
ROWCOUNT := ROWCOUNT -1;
END;
CTRANDBASE[ROWINUSE] := EVENTS[1];
ROWCOUNT := ROWCOUNT + 1;
IF TANKNOTPRESENT THEN DISKWAIT(1,WORK[*],30,TANKFILE);
WORK[6] := ROWINUSE & ROWCOUNT [18:33:15];
MOVE(16,WORDS,CTRANDBASE[0],WORK[10]);

```

```

05002000 T 0061
05002100 T 0061
05002200 T 0061
05002300 T 0061
05002400 T 0061
START OF SEGMENT ***** 38
05002500 T 0000
05002700 T 0000
05002800 T 0000
05002900 T 0001
05003000 T 0002
05003100 T 0003
05003200 T 0004
05003300 T 0005
05003400 T 0006
05003500 T 0008
05003600 T 0009
05003700 T 0010
05003800 T 0011
05003900 T 0012
05004000 T 0012
05004100 T 0013
05004200 T 0015
05004300 T 0017
05004400 T 0019

```

```

TANKSIZE:=WORK[8]; DISKWAIT(0,WORK[*],30,TANKFILE);
END REMEMBERING NEW CHUNK;
DOGSAROUND:=DOGSAROUND OR ROWCOUNT GTR 13;
IF XTRADSK GTR 0 THEN % GIVE IT BACK,
FORGETUSERDISK(XTRADSK,TANKSIZE);%
IAM := 0;
END OF NEW DISK CHUNK;

```

```

05004500 T 0021
05004600 T 0024
05004700 T 0024
05004800 T 0026
05004900 T 0027
05005000 T 0028
05005200 T 0029

```

38 IS 32 LONG, NEXT SEG 8

```

*****
PROCEDURE DISCONNECT(LCW,A); BOOLEAN LCW; ARRAY A[0];
*****
BEGIN
BOOLEAN STREAM PROCEDURE SPECIAL(N,W,T); VALUE N,T;

```

```

06000000 T 0061
06000100 T 0061
06000200 T 0061
06000300 T 0061
06000400 T 0061

```

START OF SEGMENT ***** 39

```

BEGIN
SI := LOC T; SI := SI+1; DI := W;
IF N SC = DC THEN TALLY := 1;
SPECIAL := TALLY;
END SPECIAL;

```

```

06000500 T 0000
06000600 T 0000
06000700 T 0000
06000800 T 0001
06000900 T 0002

```

```

IF NOT BOOLEAN (DOING) THEN
BEGIN
WHILE R(THINGSTODO) GTR 0 DO GETSOMETHINGTODO(LCW,A[*]);
DISCONNECTING := B(DOING := 1);
BREAKORWRU := TRUE;
END;

```

```

06001000 T 0003
06001100 T 0003
06001200 T 0004
06001300 T 0008
06001400 T 0010
06001500 T 0013
06001510 T 0013
06001520 T 0013
06001530 T 0014
06001540 T 0015
06001550 T 0017
06001560 T 0018
06001600 T 0018
06001700 T 0018
06001800 T 0021
06001900 T 0023
06002000 T 0024
06002100 T 0024
06002200 T 0026
06002300 T 0027
06002400 T 0028
06002500 T 0028
06002550 T 0029
06002600 T 0031
06002650 T 0031
06002700 T 0032
06002800 T 0033
06002900 T 0037
06002910 T 0038
06002920 T 0038

```

```

IF DOGLICENSE THEN % TANK FILE CLEAN UP IN PROGRESS
BEGIN
DOGHOUSE[LL]:=0;
DOGCAUGHTER(LCW,A[*]);
IAM := VDISCONNECT;
END;

```

```

IF REASON = VTEACHER THEN
IF SPECIAL(3,WORK[1],"BYE----") THEN
DEVICE := REAL(NOT FALSE);
IF REASON = VMCPMSG THEN
BEGIN
SREG:=BASE+LIBMSGCTR;
MCPMSG(LCW,A[*]);
IAM := VDISCONNECT;
END MCPMSG;
IF (WAITFOR:=REAL(NOT(FALSE))= DEVICE
AND REAL(ALLMCPMSG)=0 )
OR SCHEDULELINE
THEN
BEGIN
LINECLEAR(LCW,A[*]); DISCONNECTING := FALSE;
IF USERCODE NEQ 0 THEN
BEGIN
IF NOT SCHEDULELINE THEN

```

```


```

```


```

```

BEGIN
  A[BASE] := 150; BYE(LCW,A[*]); % SCRATCH WORK FILES
END BYEING OFF;
  LOGOFF(LCW,A[*]);
END LOGGING OFF;
  LCW:=FALSE; CLEAR(A[TBASE],SBASE-TBASE-1);
  IAM := 0; DISCONNECTED(LL);
END IF DONE WITH THIS LINE;
END DISCONNECT;

```

```

06002930 T 0039
06002940 T 0040
06002950 T 0043
06002960 T 0043
06002970 T 0045
06003000 T 0045
06003100 T 0048
06003200 T 0049
06003300 T 0049

```

39 IS 51 LONG, NEXT SEG 8

```

*****
PROCEDURE INPUTDATA (LCW,A);
*****
  BOOLEAN LCW; ARRAY A[0] ;
  BEGIN
LABEL T;
    WECANUSEDATA := NO DOGSAROUND;
    NODATAREQUESTED:= TRUE ;
    DATAOFFSET := EVENTS [1];
    DATAWORDS := EVENTS [2];
    DATADISKLOC := EVENTS [3] & ROWINUSE [36:44:4];
    NODATA:=(DATAWORDS LEQ 0 OR DATAOFFSET GEQ 29);
    IAM := WANT :=0;
  END KLUDGE INPUTDATA;

```

```

07000000 T 0061
07000100 T 0061
07000200 T 0061
07000300 T 0061
07000400 T 0061
07000450 T 0061
07000500 T 0000
07000600 T 0002
07000700 T 0003
07000800 T 0004
07000900 T 0005
07001000 T 0007
07001100 T 0011
07001400 T 0012

```

START OF SEGMENT ***** 40

40 IS 13 LONG, NEXT SEG 8

```

*****
PROCEDURE BUILDADATAEVENT;
*****
  BEGIN
DEFINE INPUTLL=[10:8]#, INPUTCHAR=[33:15]#, INPUTQMARK=[1:1]#;
LABEL EXIT;
REAL COUNT;
BOOLEAN STREAM PROCEDURE FLAGBIT(W);
  BEGIN
    SII=W; IF SB THEN TALLY:=1; FLAGBIT:=TALLY;
  END FLAGBIT;

```

```

07001500 T 0061
07001600 T 0061
07001700 T 0061
07001800 T 0061
07001900 T 0061
07002000 T 0000
07002010 T 0000
07002100 T 0000
07002200 T 0000
07002300 T 0000
07002400 T 0001

```

START OF SEGMENT ***** 41

```

STREAM PROCEDURE INCOMPLETMSG(WORK);
BEGIN DI:=WORK; DS:=27LIT"YOUR REQUEST IS INCOMPLETE,"; END;

```

```

07002410 T 0002
07002420 T 0002

```

IF FLAGBIT(TTYINPUT[DATAOFFSET]) THEN

```

07002500 T 0007

```

```

        BEGIN LL:=0; REASON:=13; GO EXIT END;
    IF DOGSAROUND THEN % BAD NEWS.
BEGIN
    EVENTREASON := REASON := VDOGSCATCHER;
    EVENTLL     := LL     := DOGSLOOSE;
    IF LL GTR 0 THEN GO EXIT;
END DOGSAROUND;
BEGIN
    LL := (ERRSW := TTYINPUT[DATAOFFSET]),INPUTLL;
    EVENTS[3]:= DATADISKLOC;
    EVENTS[2]:= REASON:=((WORK[0]:=ERRSW.INPUTCHAR)+7).[40:5];
    EVENTS[1]:= DATAOFFSET:= DATAOFFSET + 1;
    MOVE (REASON, WORDS, TTYINPUT[DATAOFFSET], WORK[1]);
    WORKADRS := ADDRESS(WORK[1]);
    COUNT := (WORK[0]-1).[39:9];
    WORKENDADRS:=(WORKADRS+COUNT).[40:5]&COUNT[30:45:3];
    NODATA:=(DATAOFFSET := DATAOFFSET + REASON) GEQ 29;
    IF (DATAWORDS:= DATAWORDS - REASON - 1) LSS 1 THEN
        NODATA := TRUE;
    EVENTS[0].[18:15]:= %
        (ERRSW & (REASON:= VINPUT-ERRSW,INPUTQMARK)[3:41:7])
        .[3:15];
        IF REASON = VTEACHER AND CONTINUEBIT THEN % NO DISKREAD
            BEGIN INCOMPLETEMMSG(WORK); TWXOUT(LL,WORK[0],27,1);
            LL:=0; REASON := 13; END; % IF WAITING
        IF REASON=VTEACHER THEN
            IF SEQMODE AND NOT DISCONNECTING THEN REASON:=VINPUT;
        IF LL GTR MAXLINES THEN BEGIN LL:=0; REASON:=13; END;
        END NO DOGS AROUND;
EXIT:  END OF BUILDING A DATA EVENT;

```

```

07002600 T 0009
07002700 T 0012
07002800 T 0012
07002900 T 0012
07003000 T 0015
07003100 T 0019
07003200 T 0020
07003300 T 0020
07003400 T 0020
07003500 T 0022
07003600 T 0023
07003700 T 0026
07003800 T 0030
07003820 T 0032
07003840 T 0034
07003850 T 0036
07003900 T 0039
07004000 T 0042
07004100 T 0044
07004200 T 0046
07004300 T 0047
07004400 T 0050
07004410 T 0051
07004420 T 0053
07004430 T 0057
07004440 T 0058
07004450 T 0059
07004800 T 0063
07004900 T 0066
07005000 T 0066

```

41 IS 70 LONG, NEXT SEG 8

```

*****
PROCEDURE RESTARTIT(LCW,A); BOOLEAN LCW; ARRAY A[0];
*****
    BEGIN
    REAL T;

    LABEL BAD,FINDCHKPT,MAKESURE,CASE8,CASE9,EXIT;
    BOOLEAN STREAM PROCEDURE FINDFLAG(SEGMENT);
        BEGIN LOCAL SV; LABEL EXIT; SI:=SEGMENT;
            30(SV:=SI; IF SB THEN
                BEGIN TALLY:=1; FINDFLAG:=TALLY; JUMP OUT TO EXIT; END;
                SI:=SV; SI:=SI+8);
    EXIT: END STREAM PROCEDURE FINDFLAG;

```

```

09000000 T 0061
09000100 T 0061
09000200 T 0061
09000300 T 0061
09000400 T 0061
START OF SEGMENT ***** 42
09000500 T 0000
09000502 T 0000
09000504 T 0000
09000506 T 0000
09000508 T 0001
09000510 T 0002
09000512 T 0003

```

```

STREAM PROCEDURE FLAGMESSAGE(WORK,FILNAM,USERCODE,DISKADDRESS);
VALUE FILNAM,USERCODE,DISKADDRESS;
    BEGIN
    DI:=WORK; DS:=14LIT"FLAG BIT:FILE ";
    SI:=LOC FILNAM; 2(SI:=SI+1; DS:=7CHR; DS:=LIT"/"); DI:=DI-1;

```

```

09000514 T 0005
09000516 T 0005
09000518 T 0005
09000520 T 0005
09000522 T 0007

```

```

DS:=17LIT" AT DISK ADDRESS "; DS:=8DEC;
DS:=16LIT" FILE DISCARDED.";
END STREAM PROCEDURE FLAGMESSAGE;

```

```

09000524 T 0009
09000526 T 0012
09000528 T 0014

```

```

STREAM PROCEDURE OK(W);
BEGIN
DI := W;
DS:=12 LIT "WORKFILE OK.";
END OK;

```

```

09000600 T 0014
09000700 T 0014
09000800 T 0015
09000900 T 0015
09001000 T 0017

```

```

STREAM PROCEDURE SPEC(WRK,SEQ,FIX); VALUE SEQ,FIX;
BEGIN
DI:=WRK; DS:=20 LIT "INPUT RESTORED THRU ";
FIX(DS:=7 LIT "FIX AT ");
SI:=LOC SEQ; FIX:=DI; DS:=8 DEC;
DI:=FIX; DS:=7 FILL; DI:=FIX; SI:=FIX;
8(IF SC=" " THEN SI:=SI+1 ELSE DS:=CHR);
DS:=16 LIT " ";
END STREAM;

```

```

09001010 T 0017
09001020 T 0017
09001030 T 0018
09001040 T 0021
09001050 T 0023
09001060 T 0024
09001070 T 0025
09001080 T 0027
09001090 T 0029

```

```

CASE DOING OF
BEGIN
BEGIN % CASE 0
IF BOOLEAN(EVENTS[1],[2:1]) THEN
BEGIN
SCHEDULELINE := TRUE;
CHARGE:=EVENTS[3];
USERCODE := EVENTS[1],[6:42];
SCHEDNAME := EVENTS[2],[6:42]; % NAME OF SCHEDULE FILE
DEVICE := TELETYPE;
END
ELSE DEVICE := EVENTS[2];
IF EVENTS[1],[1:2]=1 THEN %SCHEDULE LOG ON (NOT RESTART)
IAM:=VHELLO
ELSE
IF NOTANKS THEN
IAM := VHELLO
ELSE
BEGIN
FILESOK := FALSE; BASE := SBASE;
A[BASE]:=EVENTS[1],[6:42]; % SAVE USERCODE
RESTART := TRUE;
TEMP := DOING := 1;
WANT := VHELLO;
END;
END CASE 0;
BEGIN % CASE 1 GOT HIM LOGGED IN
COMMENT CHECK TIME RESTRICTIONS (NOT CHECKED IN HITHERE ON RESTART);

```

```

09001100 T 0029
09001200 T 0030
09001300 T 0030
09001310 T 0030
09001320 T 0031
09001322 T 0032
09001323 T 0034
09001324 T 0036
09001325 T 0038
09001326 T 0041
09001328 T 0042
09001330 T 0042
09001340 T 0044
09001350 T 0046
09001360 T 0047
09001400 T 0047
09001500 T 0048
09001600 T 0049
09001700 T 0050
09001800 T 0050
09001900 T 0053
09002000 T 0056
09002100 T 0058
09002200 T 0059
09002300 T 0060
09002400 T 0060
09002500 T 0061
09002510 T 0061

```

```

IF R(B(TIMELIMITS,[24:24]) EQV B(R(NOT FALSE).[24:24])) EQL
R(NOT FALSE) OR TIMELIMITS,[24:24] EQL 0 THEN ELSE
TIMERESTRICTBIT := REAL(TRUE);
RESTART := FALSE;
IF USERCODE NEQ A[BASE] THEN
BEGIN
IAM := VSETUPFILES;
DOING := TEMP := 0;
GO EXIT
END;
IF SCHEDULELINE THEN
%TEMPORARY UNTIL WE KNOW WHAT TO DO
GOERR("P*L*O*P",40);
A[BASE+10] := TEN8;
T := MAKEFN("1P",LL);
FIND(LL&IAM[33:41:7],0,T,USERCODE,USERCODE);
DOING := 2;
WAITFOR := IAM;
END CASE 1;
BEGIN % CASE 2
IF EVENTS[1] NEQ 7 THEN
BEGIN
BAD:
FILENAME := DOING := WAITFOR := 0;
IAM := VSETUPFILES;
GO EXIT
END;
A[BASE] := (WRKTBLADR:=EVENTS[4])-1; % ADDR OF FIRST REC
A[BASE+1] := 0;
FINDCHKPT:
LLDISKREAD := A[BASE] + INC(A[BASE+1]);
TEMP := 2;
DOING := 3;
WANT := VDISKREAD;
END CASE 2;
BEGIN % CASE 3
IF FINDFLAG(WORK) THEN
BEGIN
FLAGMESSAGE(WORK,MAKEFN("1P",LL),USERCODE,(A[BASE]+A[BASE+1]));
TWXOUT(0,WORK[0],70,1);
GO TO BAD;
END;
A[BASE+2] := 0;
IF WORK[0] = TEN8 THEN
BEGIN
GOODOBJ := BOOLEAN(WORK[7].[12:1]);
FILENAME := WORK[12];
INCREMENT := WORK[14];
SEQLAST := WORK[15];
SEQIN := WORK[16];
LASTRECORD := WORK[17] + 1; % RECORD COUNT
TABLEINFO := WORK[18];
FILETYPE := WORK[20];
SOURCEFILE := WORK[21];
GO MAKESURE;
END;
IF WORK[10] NEQ TEN8 THEN

```

```

09002511 T 0061
09002512 T 0063
09002513 T 0066
09002600 T 0069
09002700 T 0071
09002800 T 0072
09002900 T 0073
09003000 T 0074
09003100 T 0075
09003200 T 0075
09003210 T 0075
09003220 T 0076
09003230 T 0076
09003300 T 0083
09003400 T 0084
09003500 T 0086
09003600 T 0090
09003700 T 0090
09003800 T 0091
09003900 T 0092
09004000 T 0092
09004100 T 0093
09004200 T 0093
09004300 T 0094
09004400 T 0096
09004500 T 0097
09004600 T 0097
09004700 T 0099
09004800 T 0102
09004900 T 0103
09005000 T 0104
09005100 T 0108
09005200 T 0109
09005300 T 0110
09005400 T 0111
09005500 T 0111
09005510 T 0111
09005515 T 0113
09005520 T 0113
09005525 T 0118
09005530 T 0119
09005535 T 0120
09005600 T 0120
09005700 T 0122
09005800 T 0123
09005810 T 0123
09005900 T 0126
09006000 T 0128
09006100 T 0129
09006200 T 0131
09006250 T 0132
09006300 T 0134
09006400 T 0136
09006500 T 0137
09006600 T 0139
09006700 T 0141
09006800 T 0141

```

```

IF WORK[20] NEQ TEN8 THEN
  IF A[BASE+1] GEQ TBLSZ THEN GO BAD
  ELSE
  BEGIN
    MOVE(10,WORDS,WORK[20],A[BASE]);
    GO FINDCHKPT
  END
  ELSE
  BEGIN
    A[BASE+2] := 20;
    MOVE(10,WORDS,WORK[10],A[BASE]);
    GOODOBJ := BOOLEAN(WORK[27],[12:1]);
  END
  ELSE
  BEGIN
    MOVE(10,WORDS,WORK[0],A[BASE]);
    A[BASE+2] := 10;
    GOODOBJ := BOOLEAN(WORK[17],[12:1]);
    FILENAME := WORK[22];
    INCREMENT := WORK[24];
    SEQLAST := WORK[25];
    SEQIN := WORK[26];
    LASTRECORD := WORK[27] + 1; % RECORD COUNT
    TABLEINFO := WORK[28];
  END;
  LLDISKREAD := A[BASE] + INC(A[BASE+1]);
  TEMP := 3;
  DOING := 4;
  WANT := VDISKREAD;
  END CASE 3;
  BEGIN % CASE 4
    IF A[BASE+2] = 20 THEN
    BEGIN
      FILENAME := WORK[2];
      INCREMENT := WORK[4];
      SEQLAST := WORK[5];
      SEQIN := WORK[6];
      LASTRECORD := WORK[7] + 1; % RECORD COUNT
      TABLEINFO := WORK[8];
      FILETYPE := WORK[10];
      SOURCEFILE := WORK[11];
    END ELSE
    BEGIN
      FILETYPE := WORK [0];
      SOURCEFILE := WORK[1];
    END;
  MAKESURE;
  OBJECTFILE := 0;
  FIND(LL&IAM[33:41:7],0,SOURCEFILE,USERCODE,USERCODE);
  DOING := 5;
  WAITFOR := IAM;
  END CASE 4;
  BEGIN % CASE 5
    IF EVENTS[1] NEQ 7 THEN GO BAD;
    A[BASE] := EVENTS[3]; % SOURCEFILE EOFPTR
    T := MAKEFN("IT",LL);
    FIND(LL&IAM[33:41:7],0,T,USERCODE,USERCODE);

```

```

09906900 T 0142
09007000 T 0143
09007100 T 0145
09007200 T 0145
09007300 T 0146
09007400 T 0148
09007500 T 0148
09007600 T 0150
09007700 T 0150
09007800 T 0150
09007900 T 0152
09007910 T 0154
09008000 T 0157
09008100 T 0157
09008200 T 0157
09008300 T 0158
09008400 T 0160
09008410 T 0162
09008500 T 0165
09008600 T 0166
09008700 T 0168
09008800 T 0169
09008850 T 0171
09008900 T 0173
09009000 T 0174
09009100 T 0174
09009200 T 0179
09009300 T 0180
09009400 T 0180
09009500 T 0181
09009600 T 0182
09009700 T 0182
09009800 T 0183
09009900 T 0184
09010000 T 0185
09010100 T 0187
09010200 T 0188
09010250 T 0190
09010300 T 0192
09010400 T 0193
09010500 T 0195
09010600 T 0196
09010700 T 0196
09010800 T 0197
09010900 T 0198
09011000 T 0200
09011100 T 0200
09011200 T 0200
09011300 T 0201
09011400 T 0205
09011500 T 0206
09011600 T 0206
09011700 T 0207
09011800 T 0207
09011850 T 0208
09011900 T 0210
09012000 T 0211

```



```

WAITFOR := IAM;
TEMP := 1;
DOING := 6;
END CASE 5;
BEGIN % CASE 6
IF EVENTS[1] NEQ 7 THEN GO BAD;
IF FILETYPE NEQ 8 THEN % NOT DATA FILE
IF A[BASE] GEQ 0 THEN % SOURCEFILE NOT EMPTY
BEGIN
LLDISKREAD := EVENTS[4]; % ADDRESS OF "1T" FILE
WANT := VDISKREAD;
DOING := 7;
GO TO EXIT;
END;
GO TO CASE8;
END CASE 6;
BEGIN % CASE 7
IF WORK[1]=TEN8 THEN
BEGIN
A[BASE] := CONCISEBIT;
CONCISEBIT := 1;
WORK[0] := 0;
WORK[1] := (LL+SYSTEM) & 1[8:47:1];
WORK[2] := WORK[4] := ERRSW := USERCODE;
WORK[3] := A[BASE+1] := SOURCEFILE;
A[BASE+2] := FILENAME;
RUN1 := "LOAD ";
RUN2 := "CANDE ";
WANT := VDISPATCH;
DISKWRITE(ESP1);
LITOUT("WAIT" ",NOCRLF);
SOURCEFILE := FILENAME := 0;
TEMP := 3;
DOING := 8;
END ELSE GO CASE8;
END CASE 7;
BEGIN % CASE 8;
CONCISEBIT := A[BASE];
SOURCEFILE := A[BASE+1];
FILENAME := A[BASE+2];
CASE8:
IF GOODOBJ THEN
BEGIN
FIND(LL&IAM[33:41:7],0,SOURCEFILE,[6:36],USERCODE,USERCODE);
DOING := 9; WAITFOR := IAM;
END ELSE GO TO CASE9;
END CASE 8;
BEGIN % CASE 9
IF EVENTS[1]=7 THEN OBJECTFILE:=SOURCEFILE,[6:36] ELSE GOODOBJ:=FALSE;
CASE9:
OK(WORK);
TWXOUT(LL,WORK[0],12,1);
IF TPCOUNT NEQ 0 THEN
BEGIN % TELL USER LINE NUMBER.
T:=TPENTRY;
T:=IF A[T]=TEN8 THEN A[T=1] ELSE A[T];
SPEC(WORK,T,[21:27],REAL(T,[1:2] EQL 3));

```

```

09012100 T 0215
09012110 T 0216
09012120 T 0217
09012130 T 0217
09012140 T 0218
09012150 T 0218
09012160 T 0219
09012170 T 0220
09012180 T 0222
09012190 T 0222
09012200 T 0224
09012210 T 0225
09012215 T 0225
09012220 T 0226
09012221 T 0226
09012230 T 0226
09012240 T 0227
09012250 T 0227
09012260 T 0228
09012270 T 0228
09012280 T 0230
09012290 T 0233
09012300 T 0234
09012310 T 0237
09012320 T 0240
09012330 T 0243
09012340 T 0245
09012350 T 0246
09012360 T 0247
09012370 T 0248
09012380 T 0252
09012390 T 0253
09012400 T 0255
09012410 T 0256
09012420 T 0256
09012430 T 0256
09012440 T 0257
09012450 T 0257
09012460 T 0260
09012470 T 0262
09012480 T 0264
09012510 T 0264
09012520 T 0264
09012530 T 0265
09012540 T 0269
09012550 T 0271
09012560 T 0271
09012570 T 0271
09012580 T 0271
09012590 T 0282
09012600 T 0284
09012700 T 0285
09012710 T 0286
09012720 T 0288
09012730 T 0288
09012740 T 0290
09012750 T 0294

```

```

    TWXOUT(LL,WORK[0],36,2);
END TELLING LINE NUMBER;
LINECLEAR(LCW,A[*]);
IF SOURCEFILE = FILENAME THEN
BEGIN
    T := MAKEFN("IS",LL);
    MAKEFILE(LL&13[33:41:7],1,T,USERCODE,FILETYPE,-2);
END;
FILESOK := TRUE;
END CASE 9;
END ALL CASES;

```

```

EXIT;
END RESTARTIT;

```

```

09012760 T 0297
09012770 T 0299
09012800 T 0299
09012900 T 0300
09013000 T 0302
09013100 T 0302
09013200 T 0304
09013300 T 0308
09013400 T 0308
09013500 T 0310
09013600 T 0310
START OF SEGMENT ***** 43
43 IS 11 LONG, NEXT SEG 42
09013700 T 0312
09013800 T 0312
42 IS 316 LONG, NEXT SEG 8

```

```

*****
PROCEDURE MCPMSG (LCW,A); BOOLEAN LCW; ARRAY A[0];
*****

```

```

BEGIN
LABEL SETT,EXIT;

```

```

REAL T;

```

```

STREAM PROCEDURE MAKEBOJMSG(W,N); VALUE N;
BEGIN
DI:=W; DS:=11 LIT " RUNNING, ";
N(DI:=DI-10; DS:=10 LIT "COMPILING,");
END MAKEBOJMSG;

```

```

14000000 T 0061
14001000 T 0061
14002000 T 0061
14003000 T 0061
14004000 T 0061
START OF SEGMENT ***** 44
14005000 T 0000
14006000 T 0000
14007000 T 0000
14008000 T 0000
14009000 T 0000
14010000 T 0002
14011000 T 0005

```

```

STREAM PROCEDURE SQUASH(A,C); VALUE C;
BEGIN LABEL E,L; LOCAL TOG;
SI:=A; DI:=A; TALLY:=1; TOG:=TALLY; TALLY:=0;
2(36(IF SC = " " THEN
L: BEGIN SI:=SI+1;
IF SC = " " THEN GO TO L;
SI:=SI-1;
END;
IF SC=G THEN
BEGIN
TOG(SI:=SI+19); TOG:=TALLY;
END;
IF SC = LEFTARROW THEN JUMP OUT 2 TO E;
DS:=CHR));
E: 2(DS:=36LIT " ");
END SQUASH;

```

```

14012000 T 0005
14013000 T 0005
14014000 T 0005
14015000 T 0006
14016000 T 0007
14017000 T 0008
14018000 T 0009
14019000 T 0010
14020000 T 0010
14021000 T 0010
14022000 T 0011
14023000 T 0011
14024000 T 0013
14025000 T 0013
14026000 T 0014
14027000 T 0015
14028000 T 0021

```

```

STREAM PROCEDURE EXAMINE(WORK,L);
  BEGIN LOCAL T; LABEL L1,EXIT;
    SI:=WORK; DI:=L; DI:=DI+8;
    6( % 6 WORDS MAXIMUM IN OUTPUT ARRAY L
L1: IF SC=" " THEN
      BEGIN
        SI:=SI+1; GO TO L1;
      END;
    IF SC=LEFTARROW THEN JUMP OUT TO EXIT;
    IF SC=ALPHA THEN
      BEGIN
        DS:=LIT"0"; DS:=CHR;
        6(IF SC=ALPHA THEN DS:=CHR ELSE DS:=LIT" ");
        20(IF SC=ALPHA THEN SI:=SI+1 ELSE JUMP OUT);
        TALLY:=TALLY+1;
      END
    ELSE
      BEGIN
        SI:=SI+1; GO TO L1;
      END);
EXIT: DI:=L; T:=TALLY; SI:=LOC T; DS:=WDS;
      END STREAM PROCEDURE EXAMINE;

```

```

14029000 T 0021
14030000 T 0021
14031000 T 0021
14032000 T 0022
14033000 T 0022
14034000 T 0023
14035000 T 0023
14036000 T 0023
14037000 T 0024
14038000 T 0024
14039000 T 0025
14040000 T 0026
14041000 T 0026
14042000 T 0027
14043000 T 0029
14044000 T 0032
14045000 T 0032
14046000 T 0032
14047000 T 0032
14048000 T 0032
14049000 T 0033
14050000 T 0033
14051000 T 0033
14052000 T 0035

```

```

STREAM PROCEDURE ERRORMESSAGE(WORK);
  BEGIN
    DI:=WORK;
    DS:=47LIT"SYSTEM ERROR;TASK WAS DISCONTINUED-PLEASE WAIT-";
  END;

```

```

14052100 T 0035
14052200 T 0035
14052300 T 0035
14052400 T 0036
14052500 T 0036
14052600 T 0042

```

```

STREAM PROCEDURE LOADCODE (PLACE);
  BEGIN
    DI:=PLACE; DI:=DI+2;
    DS:=6 LIT " PST "; DI:=DI+26;
    DS:=6 LIT " IOT+ ";
  END OF LOADCODE;

```

```

%0519- 14052610 C 0042
%0519- 14052620 C 0042
%0519- 14052630 C 0043
%0519- 14052640 C 0043
%0519- 14052650 C 0044
%0519- 14052660 C 0045

```

```

*.....
IF NOT BOOLEAN(DOING) THEN % FIRST LIBRARY MAINTENANCE CALL
  BEGIN
    SREG:=BASE+LIBMSGCTR=A[BASE];
    DOING := 1;
  END;

```

```

14052700 T 0046
14053000 T 0046
14054000 T 0046
14055000 T 0046
14056000 T 0046
14057000 T 0047
14058000 T 0049
14059000 T 0050
14060000 T 0050

```

```

% EVENTS[1] = 0 FOR EUJ SYNTAX ERRORS,
%      1 FOR EUJ O.K.,
%      2 FOR EUJ DS=ED,
%      3 FOR DS=ED MESSAGE,
%      4 FOR BOJ,
%      5 FOR LIBMTNCE IGNORED,
%      6 FOR LIBMTNCE REMOVED,
%      7 FOR LIBMTNCE CHANGED,
%      8 FOR SECURED FILE,
%      9 FOR ZIP=ERROR,
%     10 FOR NOT ON DISK,
%     11 FOR NOT OBJECT CODE FILE.

IF EVENTS[1] = 4 THEN % BOJ EVENT
BEGIN
IF BOJMSG THEN % INDICATE ACTUAL BOJ
BEGIN
MAKEBOJMSG(WORK[0],COMPILING);
TWXOUT(LL,WORK[0],11,1);
BOJMSG:=FALSE;
END BOJMSG;
IF CONCISE THEN MATCH(RUN2,"CANDE ") ELSE
BEGIN
LITOUT("-BOJ- ",NOCRLF); LITOUT(RUN1,CRLF);
END;
MATCH(RUN1,"PAPER ") MATCH(RUN2,"CANDE ")
BEGIN
READTAPE(LL&13[33:41:7],8,0,USERCODE);
END;
END BOJ ELSE

IF EVENTS[1] GEQ 5 AND EVENTS[1] LEQ 7 THEN % LIBRARY MAINT.
BEGIN
IF LIBMTNCE THEN
BEGIN
A[SRG=LIBMSGCTR]:=EVENTS[1]=5 THEN 4 ELSE 1];
IF LIBMSGCTR:=LIBMSGCTR-1 LEQ 0 THEN IAM := 0;
END ELSE IAM := 0;
END ELSE

IF EVENTS[1] = 3 THEN % DS=ED MESSAGE
BEGIN
TWXOUT(LL,WORK[0],0,1); % CR/LF BEFORE MSG
SQUASH (WORK[0],"!");
IF RUNNING AND NOT BOOLEAN(HELPTOGGLE) AND NOT SCHEDULELINE THEN
BEGIN % TRY RUNNING "HELP" PROGRAM
EXAMINE(WORK,L);
IF (COMPAR(L[1],"EOF ") OR COMPAR(L[1],"PAR ")
AND
(COMPAR(RUN2,"CANDE ") OR COMPAR(RUN2,"DISK "))
AND
L[0] GEQ 5 THEN
BEGIN
SETT: WORK[0]:=L[1]; % MESSAGE TYPE
WORK[1]:=L[4]; % MFID
WORK[2]:=L[5]; % FID
WORK[3]:=RUN1;

```

```

14061000 T 0050
14062000 T 0050
14063000 T 0050
14064000 T 0050
14065000 T 0050
14066000 T 0050
14067000 T 0050
14068000 T 0050
14069000 T 0050
14070000 T 0050
14071000 T 0050
14072000 T 0050
14073000 T 0050
14074000 T 0050
14075000 T 0051
14076000 T 0051
14077000 T 0052
14078000 T 0052
14079000 T 0054
14080000 T 0056
14081000 T 0058
14082000 T 0058
14083000 T 0062
14084000 T 0064
14085000 T 0066
14086000 T 0066
14087000 T 0068
14088000 T 0070
14089000 T 0073
14090000 T 0073
14091000 T 0073
14092000 T 0073
14093000 T 0080
14094000 T 0080
14095000 T 0081
14096000 T 0082
14097000 T 0086
14098000 T 0090
14099000 T 0091
14100000 T 0091
14101000 T 0091
14102000 T 0093
14103000 T 0093
14104000 T 0095
14105000 T 0096
14106000 T 0100
14107000 T 0100
14108000 T 0102
14109000 T 0105
14110000 T 0106
14113000 T 0109
14114000 T 0110
14115000 T 0111
14116000 T 0112
14117000 T 0113
14118000 T 0115
14119000 T 0116

```

```

WORK[4]:=RUN2;
WORK[5]:=LL&FILETYPE(34:42:6);
IF COMPAR(L[4],SOURCEFILE) THEN
  IF COMPAR(L[5],USERCODE) THEN
    WORK[5],[111]:=1; % WORKFILE TOGGLE
  WORK[6]:=USERCODE;
  WORK[7]:=TPCOUNT;
  WORK[8]:=SOURCEFILE;
  MOVE(16,WORDS,CTRANDBASE,WORK[10]);
  DISKWRITE(ESP1);
  HELPTOGGLE:=1;
  NORETRY:=0;
  IF BOOLEAN(UPDATEBIT) THEN % UPDATING WORKFILE
    BEGIN
      T:=MAKEFN("1P",LL);
      IF COMPAR(WORK[1],T) OR WORK[1]=0 THEN NORETRY:=1;
    END;
  END % IF COM11 MESSAGE
ELSE % CHECK TERMINALMESSAGEA ERRORS
  IF (COMPAR(L[1],"FLAG ") OR
    COMPAR(L[1],"INVALID ") OR
    (COMPAR(L[1],"IO ") AND COMPAR(L[2],"ERR ")))
  AND
  (COMPAR(RUN2,"CANDE ") OR COMPAR(RUN2,"DISK ") AND
  SOURCEFILE NEQ 0 THEN
    BEGIN
      L[4]:=L[5]:=0; GO TO SETT;
      END ELSE TWXOUT(LL,WORK[0],72,1);
    END ELSE TWXOUT(LL,WORK[0],72,1);
  END ELSE

IF EVENTS[1] GEQ 8 THEN % MISC. MESSAGES
  BEGIN
  ERRSW := RUN1;
  MSGPTR:=EVENTS[1]+16;
  ALLMCPMSG := FALSE;
  HELPTOGGLE:= RUN1 := RUN2 := IAM := 0;
  NORETRY:=0;
  IF NOT BREAKORWRU THEN IAM:=VERROR;
  GO TO EXIT;
  END;

IF EVENTS[1] LEQ 2 THEN % EOJ FOR THIS GUY,
  BEGIN
  ETIME := ETIME + EVENTS[3],[16:32];
  IOTOT := IOTOT + EVENTS[4],[16:32];
  ERRSW:=RUN1;
  COMMONCELL := WORK[8];
  IF COMPAR(RUN1,"HELP ") AND COMPAR(RUN2,"DISK ") THEN
    BEGIN
    HELPTOGGLE := 0;
    IF BOOLEAN(UPDATEBIT) AND EVENTS[1] = 1 THEN % WORKFILE UPDATE
      BEGIN
      LINECLEAR(LCW,A[*]);
      TWXOUT(LL,LL,0,2);
      IF BOOLEAN(NORETRY) THEN CANCELCKPT(A[*])
      ELSE LITOUT("OK ",CRLF);

```

```

14120000 T 0118
14121000 T 0119
14122000 T 0122
14123000 T 0124
14124000 T 0126
14125000 T 0129
14126000 T 0131
14127000 T 0133
14128000 T 0134
14129000 T 0136
14130000 T 0140
14131000 T 0142
14132000 T 0145
14133000 T 0146
14134000 T 0146
14135000 T 0148
14136000 T 0154
14137000 T 0154
14138000 T 0154
14139000 T 0154
14139010 T 0160
14139020 T 0162
14140000 T 0166
14141000 T 0166
14141100 T 0170
14142000 T 0172
14143000 T 0172
14144000 T 0182
14145000 T 0184
14146000 T 0186
14147000 T 0186
14148000 T 0186
14149000 T 0188
14150000 T 0188
14151000 T 0189
14152000 T 0191
14153000 T 0193
14154000 T 0198
14155000 T 0201
14156000 T 0203
14157000 T 0204
14158000 T 0204
14159000 T 0204
14160000 T 0205
14161000 T 0205
14162000 T 0208
14163000 T 0211
14164000 T 0212
14165000 T 0213
14166000 T 0217
14167000 T 0217
14168000 T 0220
14169000 T 0222
14170000 T 0223
14171000 T 0224
14172000 T 0226
14173000 T 0228

```

```

NORETRY:=0;
IF BOOLEAN(COMMONCELL,[3:1]) THEN
  SOURCEFILE:=MAKEFN("1S",LL);
WRITEASTERISK:=TRUE;
GO TO EXIT;
END; % IF UPDATE WAS IN PROGRESS
NORETRY:=0;
IF BOOLEAN(COMMONCELL,[3:1]) THEN SOURCEFILE:=MAKEFN("1S",LL);
END; % IF EOJ FOR HELP PROGRAM
IF EVENTS[1] NEQ 1 AND NOT BREAKORWRU THEN % BAD
BEGIN
MSGPTR:=IF COMPILING THEN 12 ELSE 28;
IF DOGLICENSE THEN DOGCATCHER(LCW,A[*]);
SCHPREV:=PREVRCW:=LLWORDS;
INITIATEDWHILEBUSY:=FALSE;
END % BAD NEWS
ELSE
BEGIN
MSGPTR:=0;
IF A[BASE+1] := WORK[12] NEQ 0 THEN % CHAIN REQUEST
BEGIN
WANT := VDISPATCH;
IF A[BASE+1],[42:6] NEQ " " THEN GOERR(A[BASE+1],42);
A[BASE+1]:=A[BASE+1],[6:36];
A[BASE+2] := WORK[13];
END
END;
TWXOUT(LL,LL,0,2=CONCISEBIT);
IF (EOJMSG OR EVENTS[1] NEQ 1) AND NOT BOOLEAN(HELPTOGGLE) THEN
BEGIN
WRITEASTERISK := FALSE;
WORK[0]:=(IF EVENTS[1]=1 THEN "END " ELSE "ERR ");
MATCH(RUN2,"CANDE ") WORK[1]:=RUN1 ELSE
BEGIN
WORK[1]:=
IF COMPILING THEN "COMPILE" ELSE
IF RUN1,[12:6]=1 THEN FILENAME ELSE
RUN1;
END MATCH CANDE;
TIMEUSED(WORK[2],TIMECONV(EVENTS[3],[16:32],FALSE));
TIMEUSED(WORK[6],TIMECONV(EVENTS[4],[16:32],FALSE)); %0519=
LOADCODE(WORK[5]); %0519=
BLANK(WORK[1]); SQUASH(WORK[0],""); BLANK(WORK[0]);
TWXOUT(LL,WORK[0],72,2); %0519=
IF SCHEDULE THEN
IF NOT NOSTOP THEN
IF EVENTS[1] NEQ 1 THEN
SCHEDERROR; % TERMINATE SCHEDULE
END IF EOJMSG;
IAM := 0;
END EOJ STUFF;
IF WANT := VDISPATCH THEN % CHAIN REQUEST
BEGIN
RUN1 := A[BASE+1];
RUN2 := A[BASE+2];
END ELSE

```

```

14174000 T 0233
14175000 T 0235
14176000 T 0236
14177000 T 0239
14178000 T 0241
14179000 T 0243
14180000 T 0243
14181000 T 0245
14182000 T 0249
14183000 T 0249
14184000 T 0251
14185000 T 0251
14186000 T 0255
14187000 T 0258
14188000 T 0260
14189000 T 0262
14190000 T 0262
14191000 T 0262
14192000 T 0263
14193000 T 0264
14194000 T 0266
14195000 T 0267
14196000 T 0268
14197000 T 0275
14198000 T 0278
14199000 T 0280
14200000 T 0280
14201000 T 0280
14202000 T 0283
14203000 T 0287
14204000 T 0287
14205000 T 0289
14206000 T 0293
14207000 T 0297
14208000 T 0301
14209000 T 0301
14210000 T 0303
14211000 T 0306
14212000 T 0307
14213000 T 0307
14213100 C 0310
14213200 C 0313
14214000 T 0314
14215000 P 0318
14216000 T 0319
14217000 T 0320
14218000 T 0322
14219000 T 0323
14220000 T 0334
14221000 T 0334
14222000 T 0334
14223000 T 0334
14224000 T 0334
14225000 T 0335
14226000 T 0336
14227000 T 0338
14228000 T 0340

```

```

IF WAITFOR:=IAM = 0 THEN % ALL MESSAGES ARE IN
BEGIN
  ALLMCPMSG := B(LIBMSGCTR:=RUN1:=RUN2:=0);
  IF BOOLEAN(HELPTOGGLE) THEN % TRY AND RECOVER
    IF NOT SCHEDULELINE THEN % PGM,MAY REQUIRE TWX INPUTS
      BEGIN
        RUN1:="HELP ";
        RUN2:="DISK ";
        WANT:=VDISPATCH;
        COMMONVALUE := ESP1;
        ERRORMESSAGE(WORK); TWXOUT(LL,WORK[0],47,CRLF);
      END;
    END IF ALL DONE;
EXIT:
  END OF MCPMSG;

```

```

14229000 T 0340
14230000 T 0340
14231000 T 0341
14232000 T 0342
14233000 T 0347
14234000 T 0348
14235000 T 0350
14236000 T 0350
14237000 T 0351
14238000 T 0353
14239000 T 0353
14240000 T 0354
14241000 T 0357
14242000 T 0357
14243000 T 0357
14244000 T 0358
44 IS 363 LONG, NEXT SEG 8

```

```

*****
PROCEDURE COBLIT(A); ARRAY A[0];
*****
BEGIN
  STREAM PROCEDURE NODISKMESS(WORK,AMOUNT); VALUE AMOUNT;

  BEGIN LOCAL A,B;
    LABEL L;
    DI := LOC A; SI := LOC AMOUNT; DS := 8 DEC;
    DI:=WORK; SI:=LOC A;
    TALLY := 8;
  L: 8(IF SC="0" THEN BEGIN SI := SI+1; TALLY := TALLY+63 END
      ELSE JUMP OUT);
    B := TALLY; DS := B CHR;
    DS:=35LIT" SEGMENTS OF DISK ARE NOT AVAILABLE";
    DS:=8LIT" ";
  END;

  IF LL=0 THEN
  BEGIN WENEEDTOCOOLIT := TRUE;
    INC(NUDCOUNT);
  END ELSE
  BEGIN NODISKMESS(WORK,EVENTS[2]);
    TWXOUT(LL,WORK[0],43,1);
    PREVMSG := MSGPOINTER;
    MSGPOINTER := 47;
  END;
  IAM := WANT := WAITFOR := 0;
END;

```

```

15000000 T 0061
15001000 T 0061
15002000 T 0061
15003000 T 0061
15004000 T 0061
START OF SEGMENT ***** 45
15005000 T 0000
15005500 T 0000
15006000 T 0000
15007000 T 0000
15008000 T 0001
15009000 T 0001
15010000 T 0003
15011000 T 0005
15012000 T 0005
15012100 T 0010
15013000 T 0011

15014000 T 0012
15015000 T 0012
15016000 T 0015
15017000 T 0016
15018000 T 0016
15019000 T 0018
15020000 T 0020
15021000 T 0022
15022000 T 0023
15023000 T 0023
15024000 T 0025
45 IS 26 LONG, NEXT SEG 8

```

```

*****
PROCEDURE MCPOK(A);  ARRAY A[0];
*****
BEGIN
  STREAM PROCEDURE OKMESS(WORK);

  BEGIN DI := WORK;
    DS:=41LIT"DISK SPACE FOUND, YOU ARE RUNNING AGAIN ";
  END;

  IF LL=0 THEN
    IF (DECR(NUDCOUNT))=0 THEN
      BEGIN TIMETOOKLINES := TRUE;
        WENEEDTOCQLIT := FALSE;
      END ELSE
        IF NUDCOUNT LSS 0 THEN NUDCOUNT := 0 ELSE ELSE
          BEGIN OKMESS(WORK);
            TWXOUT(LL,WORK[0],41,CRLF);
            MSG#OINTER := PREVMSG;
            PREVMSG := 0;
          END;
          IAM := 0;
        END;
      END;
    END;
  END;

```

```

16000000 T 0061
16001000 T 0061
16002000 T 0061
16003000 T 0061
16004000 T 0061
START OF SEGMENT ***** 46
16005000 T 0000
16006000 T 0000
16007000 T 0005

16008000 T 0006
16009000 T 0006
16010000 T 0009
16011000 T 0011
16012000 T 0013
16013000 T 0013
16014000 T 0016
16015000 T 0017
16016000 T 0019
16017000 T 0020
16018000 T 0022
16019000 T 0022
16020000 T 0022
46 IS 23 LONG, NEXT SEG 8

```

```

*****
BOOLEAN PROCEDURE MAKEOKEVENT;
*****
BEGIN
  STREAM PROCEDURE OKMESS(WORK);

  BEGIN DI := WORK;
    DS := 28 LIT"SYSTEM OK - YOU MAY PROCEED*";
  END;

  DEFINE A[A1] = LINE[LL,A1]#;
  LABEL EXIT;
  FOR LL := LLLPREV+1 STEP 1 UNTIL MAXLINES DO
    IF NODISKMCP THEN
      BEGIN
        EVENTS[0] := O&LL[33:33:15]&VMCPOK[41:33:7];
        OKMESS(WORK);
        TWXOUT(LL,WORK[0],29,CRLF);
        MAKEOKEVENT := TRUE;
        GO TO EXIT;
      END;
      TIMETOOKLINES := MAKEOKEVENT := FALSE;
    EXIT;
  END;

```

```

16500000 T 0061
16501000 T 0061
16502000 T 0061
16503000 T 0061
16504000 T 0061
START OF SEGMENT ***** 47
16505000 T 0000
16506000 T 0000
16507000 T 0004

16507500 T 0004
16508000 T 0004
16509000 T 0004
16510000 T 0009
16511000 T 0010
16512000 T 0010
16513000 T 0013
16514000 T 0014
16515000 T 0016
16516000 T 0017
16517000 T 0017
16518000 T 0018
16519000 T 0020

```


END;

16920000 T 0021
47 IS 24 LONG, NEXT SEG 8

```

*****
PROCEDURE TEACHER(LCW,A); BOOLEAN LCW; ARRAY A[0];
*****
BEGIN
  REAL T,PTR,INDX,X;

  LABEL MESSAGE, QUIT, EXIT;
  IF SEQMODE OR NOT LOGGEDON THEN GO QUIT;
  IF NOT BOOLEAN(DOING) THEN % CASE 0
    BEGIN
      PTR := WORKADRS; INDX := 1; CLEAR(A[BASE+1],2);
      WHILE T NEQ 4 AND INDX LEQ 2 DO % SCAN INPUT RECORD
        BEGIN
          T+SCAN(PTR,A[BASE +INDX], IF
            INDX= 1 THEN "Z" ELSE "9", WORKENDADRS,X);
          IF A[BASE+1] NEQ QMARK THEN INDX:=INDX+1;
          END; % SCAN
          T := A[BASE+1],[6:42]; % FIRST INPUT AFTER QMK
          MATCH(T,"CONTINU") GO QUIT; % IGNORE "CONTINUE" INPUT
          IF .COMPAR(T,"STATUS ") OR T="+" THEN
            BEGIN IAM := VSTATUSCHECK; GO TO EXIT; END;
          MATCH(T,"DATA ") % SUCCEEDING INPUT GOES TO DATA FILE
            BEGIN
              IF RUNNING OR COMPILING THEN GO TO MESSAGE;
              IF FILENAME = 0 THEN GOERR("NO FILE",1);
              IF FILETYPE NEQ 8 THEN GOERR(T,60); % WRONG TYPE FILE
              DATAENABLED := TRUE; LITOUT("OK ",1); GO QUIT;
            END;
          MATCH(T,"END ") % RETURN TO COMMAND MODE
            BEGIN
              IF RUNNING OR COMPILING THEN GO TO MESSAGE;
              IF NOT DATAENABLED THEN GOERR(T,8);
              DATAENABLED := FALSE; WRITEASTERISK:=TRUE; GO QUIT;
            END;
          IF NOT SCHEDULELINE THEN
            BEGIN
              MATCH(T,"TO ") % SS MESSAGE
                BEGIN
                  IF A[BASE+2],[1:2] GTR 0 THEN GO TO MESSAGE;
                  STICK(PTR,L,WORKENDADRS);
                  SSMSG; LITOUT("S ",1); GO QUIT;
                END;
              MATCH(T,"TAPE ") % SWITCH TO PAPER TAPE MODE
                BEGIN
                  MATCH(RUN2,"CANDE ") GOERR("NO PRGM",8);
                  IF NOT RUNNING THEN GOERR("NOPRGRM",8);
                  IF DEVICE#CONRAC OR DEVICE#BIDS THEN GOERR(T,56);
                  % PUT YOUR FUNNY TERMINALS HERE

                  READTAPE(LL&13[33:41:7],8,0,USERCODE); GO QUIT;

```

```

25000000 T 0061
25000100 T 0061
25000200 T 0061
25000300 T 0061
25000400 T 0061
START OF SEGMENT ***** 48
25000500 T 0000
25000600 T 0000
25000700 T 0002
25000800 T 0003
25000900 T 0003
25001000 T 0007
25001100 T 0009
%0102= 25001200 P 0009
%0102= 25001210 C 0013
25001300 T 0016
25001400 T 0020
25001500 T 0020
25001600 T 0023
25001610 T 0025
25001620 T 0027
25001700 T 0032
25001800 T 0033
25001900 T 0034
25002000 T 0036
25002100 T 0045
25002200 T 0051
25002300 T 0056
25002400 T 0056
25002500 T 0057
25002600 T 0058
25002700 T 0060
25002800 T 0068
25002900 T 0073
25003000 T 0073
25003100 T 0074
25003200 T 0074
25003300 T 0076
%0101= 25003400 P 0076
25003500 T 0079
25003600 T 0081
25003700 T 0085
25003800 T 0085
25003900 T 0086
25003910 T 0087
25003920 T 0097
25004000 T 0104
25004100 T 0111
25004200 T 0111
25004300 T 0111
25004400 T 0111

```

```

        END;
    END; % IF NOT SCHEDULELINE
MESSAGE:
    IF PREVRW = BASE LEQ 35 THEN % NOT ENOUGH ROOM
    IF INITIATEDWHILEBUSY THEN % ORIGINAL REQUEST INCOMPLETE
    BEGIN
        IF RUNNING THEN LITOUT("RUNNING",1) ELSE
        IF COMPILING THEN LITOUT("CMPILNG",1) ELSE
        BEGIN
            L[0] := " STATUS"; BLANK(L[0]); L[1] := "OK      "; BLANK(L[1]);
            TWXOUT(LL,L[0],11,2=CONCISEBIT);
        END;
        GO TO QUIT;
    END;
    IF DATAENABLED THEN MSGPTR := 68; % YOU ARE IN DATA MODE
    LLDISKREAD := MSGPTR DIV 3 + MSGFILELOC;
    DOING := 1;
    WANT := VDISKREAD;
    END % CASE 0
ELSE
    BEGIN
        IF T := MSGPTR MOD 3 NEQ 0 THEN MOVE(10,WORDS,WORK[T*10],WORK[0]);
        TWXOUT(LL,WORK[0],64,2 = CONCISEBIT); INQUIRY := FALSE;
        MSGPTR := REAL(RUNNING OR COMPILING) * MSGPTR;
        IF SCHEDULELINE THEN
            IF NOT NOSTOP THEN
                IF NOT BOOLEAN(SCHEDNAME,[1:1]) THEN
                    BEGIN
                        IAM := VCHANGE;
                        A[BASE] := SCHEDNAME,[6:42];
                        A[BASE+1] := 63; % "ABORTED" FILE TYPE
                        FIND(LL&IAM[33:41:7],0,SCHEDNAME,[6:42],
                            USERCODE,USERCODE);
                        SCHEDNAME := (=SCHEDNAME); % TERMINATING FLAG
                        TEMP := 2; DOING := 1;
                        WAITFOR := IAM;
                        GO TO EXIT;
                    END ELSE SCHEDGOTO(-1);
                END;
            END;
        END;
    END; % CASE 1
EXIT:
    END PROCEDURE TEACHER;

```

```

25004500 T 0114
25004600 T 0114
25004700 T 0114
25004710 T 0115
25004715 T 0116
25004720 T 0117
25004725 T 0118
25004730 T 0120
25004735 T 0124
25004740 T 0126
25004745 T 0130
25004750 T 0133
25004755 T 0133
25004760 T 0136
25004770 T 0136
25004800 T 0138
25004900 T 0141
25005000 T 0141
25005100 T 0142
25005200 T 0142
25005300 T 0142
25005400 T 0143
25005500 T 0148
25005600 T 0153
25005700 T 0157
25005710 T 0157
25005720 T 0159
25005730 T 0161
25005740 T 0161
25005750 T 0162
25005760 T 0165
25005770 T 0166
25005780 T 0166
25005790 T 0172
25005800 T 0173
25005810 T 0175
25005820 T 0176
25005830 T 0176
25005840 T 0180
25005900 T 0184
25006000 T 0184
25006100 T 0184

```

48 IS 187 LONG, NEXT SEG 8

```

*****
PROCEDURE INPUT(LCW,A,TTYINPUT,DKOFFSET,DISKLOC);
*****
VALUE DKOFFSET,DISKLOC;
REAL DKOFFSET,DISKLOC,TTYINPUT;
BOOLEAN LCW;
ARRAY A[0];
BEGIN
REAL STREAM PROCEDURE OUTSUPR(INPUT,OUTPUT); VALUE INPUT;

```

```

26000000 T 0061
26000100 T 0061
26000200 T 0061
26000300 T 0061
26000400 T 0061
26000500 T 0061
26000600 T 0061
26000700 T 0061
26000710 T 0061

```

START OF SEGMENT ***** 49

BEGIN		26000720 T 0000
DI:=LOC INPUT; DS:=7FILL;		26000730 T 0000
SI:=LOC INPUT; DI:=OUTPUT;		26000740 T 0000
B(IF SC=" " THEN SI:=SI+1 ELSE		26000750 T 0001
BEGIN DS:=CHR; TALLY:=TALLY+1; END);		26000760 T 0002
OUTSUPR:=TALLY;		26000770 T 0003
END STREAM PROCEDURE OUTSUPR;		26000780 T 0003
STREAM PROCEDURE PLSWAIT(WORK);		26000790 T 0004
BEGIN		26000792 T 0004
DI:=WORK; DS:=26LIT"PLEASE WAIT - NO USER DISK";		26000794 T 0005
END STREAM PROCEDURE PLSWAIT;		26000795 T 0008
COMMENT	THIS PROCEDURE SCANS THE INPUT USING SCANNER AND	26000800 T 0009
	THEN CONTROLS THE PROCESS OF WHAT IS FOUND.	26000900 T 0009
	IT BUILDS THE TABLES FOR DATA LINES, AND	26001000 T 0009
	CALLS ON THE VERB PROCESSORS FOR COMMAND LINES.	26001100 T 0009
	END COMMENT;	26001200 T 0009
STREAM PROCEDURE FIXERRORMSG(SEQ,L); VALUE SEQ;		26001210 T 0009
BEGIN		26001220 T 0009
DI:=LOC SEQ; DS:=7FILL; SI:=LOC SEQ; DI:=L;		26001230 T 0009
DS:=16LIT"FIX SYNTAX ERR @ ";		26001240 T 0010
B(IF SC=" " THEN SI:=SI+1 ELSE DS:=CHR); DS:=8LIT" ";		26001250 T 0012
END STREAM PROCEDURE FIXERRORMSG;		26001260 T 0015
DEFINE	VERBV = A[SREG],[40:8] #;	26001300 T 0015
	PAR = A[SREG],[25:8] #;	26001400 T 0015
	SWDS = A[BASE],[40:8] #;	26001500 T 0015
	TBLTODK = 2#;	26001600 T 0015
	MOREVERBS = 1 #;	26001700 T 0015
LABEL	CENTRAL,SCAN1,NODISK,EXIT;	26001800 T 0015
REAL	T,E;	26001900 T 0015
DEFINE	TAB[TAB1] = WORK[TAB1] #;	26002000 T 0015
	CENTRAL: CASE DOING OF	26002100 T 0015
	BEGIN	26002200 T 0016
	BEGIN % CASE 0	26002300 T 0016
	IF NOT LOGGEDON THEN	26002400 T 0016
	BEGIN	26002500 T 0017
	IAM := VHELLO; DOING := 1;	26002600 T 0018
	% VOID	26002650 T 0019
	GO EXIT	26002700 T 0019
	END;	26002800 T 0020
	IF SEQMODE THEN % COULD BE LEFTARROW OR DATA LINE.	26002900 T 0020
	BEGIN % CHECK LEFT ARROW.	26003000 T 0021
	IF WORK[0] GTR 1 THEN	26003100 T 0021
	BEGIN % SEQMODE DATA LINE.	26003200 T 0022
	T:=2; A[BASE]:=SEQIN:=SEQLAST:=SEQLAST+INCREMENT;	26003300 T 0023
	IF SCHEDULELINE THEN	26003310 T 0028

```

BEGIN %SEND OUT SEQ NUMBER, THE MCP WILL NOT DO IT
  E := OUTSUPR(DECCONV(SEQLAST+INCREMENT),WORK[0]);
  TWXOUT(LL,WORK[0],E,NOCRLF);
END;
  IF DEVICE=CONRAC OR DEVICE=BIDS THEN
  IF SEQLAST=CHKSEQNMBR THEN OUTPUTNUMBERS(LCW,A[*]);
END ELSE T := 5; % NULL INPUT
END ELSE
  IF DATAENABLED THEN % INPUT TO DATA FILE
  BEGIN
    A[BASE] := LASTRECORD := LASTRECORD + 1;
    T := 2;
  END ELSE
  BEGIN % NOT SEQ MODE.
    SCANNER(WORKADRS,A[*],LCW,WORKENDADRS);
    SREG := BASE + 1;
    T := A[BASE],[3:3];
  END;
  CASE T OF % HANDLE THE INPUT LINE,
  BEGIN
  BEGIN
    % OOPS; SCANNER RETURNED ZERO
    % WHICH MEANS THERE WAS A COMMENT,
    % SO JUST SKIP IT
    IAM:=WANT:=0;
  END SCANNER CASE 0;
  BEGIN
    % SCANNER = 1 : WE HAVE A VERB
  COMMENT VERB CHECK;
  IF CHECKBIT(A[SREG],[9:9] DIV 2,RESTRICTEDVERBS) THEN
  BEGIN ERRSW:=VERBTABLE[A[SREG],[9:9]]; WANT:=VERROR; MSGPTR:=49;
  GO TO EXIT; END; % USER IS NOT PERMITTED TO USE VERB;
    WRITEASTERISK := TRUE;
    TEMP:= 1;
    PARAM := SWDS;
    IF WENEEDTOCOOLIT THEN
    BEGIN DOING := 3;
      GO TO NODISK;
    END;
  SCAN1:
    WANT := VERBV;
    IF VERBV = VSEQ THEN IAM:=0 ELSE
    IF A[BASE],[18:15] GTR 1 THEN DOING := MUREVERBS
    ELSE IAM := 0;
  END SCANNER CASE 1;
  BEGIN
    % SCANNER = 2: WE HAVE A SEQ NUMBER,
    IF FILENAME = 0 OR
    NOT FILESOK THEN GOERR("NO FILE",1);
    E := A[SREG]& DISKLOC[4:36:12]
      & DKOFFSET[16:43:5];
    IF SEQMODE OR DATAENABLED THEN T := 2
    ELSE BEGIN T:=A[SREG+1]; SEQIN:=A[SREG]; END;
    E,[1:2] := T;
    IF T=3 THEN IF FIXERROR(WORKADRS,WORKENDADRS) THEN
    BEGIN
      FIXERRORMSG(DECCONV(A[SREG]),L); TWXOUT(LL,L[0],24,1);
      MSGPTR := 21; IAM := 0; GO TO EXIT;
    END;
  COMMENT NOW PUT PUT ENTRY INTO TABLE;
  NOSAVE := FALSE;

```

```

26003320 T 0028
26003330 T 0029
26003340 T 0033
26003350 T 0035
26003360 T 0035
26003370 T 0037
26003400 T 0041
26003500 T 0043
26003510 T 0043
26003520 T 0045
26003530 T 0045
26003540 T 0048
26003550 T 0049
26003600 T 0049
26003700 T 0049
26003800 T 0052
26003900 T 0053
26004000 T 0054
26004100 T 0054
26004200 T 0055
26004300 T 0055
26004400 T 0055
26004500 T 0055
26004600 T 0055
26004700 T 0056
26004800 T 0057
26004810 T 0057
26004820 T 0057
26004830 T 0060
26004840 T 0064
26004900 T 0065
26005000 T 0067
26005100 T 0068
26005120 T 0069
26005140 T 0070
26005160 T 0071
26005180 T 0072
26005200 T 0072
26005250 T 0073
26005300 T 0076
26005400 T 0079
26005500 T 0081
26005600 T 0081
26005700 T 0081
26005800 T 0082
26005900 T 0090
26006000 T 0091
26006100 T 0093
26006200 T 0095
26006300 T 0099
26006310 T 0101
26006320 T 0103
26006330 T 0104
26006340 T 0108
26006350 T 0110
26006400 T 0110
26006450 T 0110

```

```

GOODOBJ := FALSE; % OBJ DOES NOT MATCH SOURCE
IF DOGHOUSE[LL] LEQ 0 THEN % FIRST DOG,
    DOGHOUSE[LL] := CTRANDBASE[ROWINUSE];
A[TBASE + TPENTRY] := E;
INC (TPCOUNT);
IF TPENTRY LSS 9 THEN
BEGIN INC (TPENTRY);
    A[TPENTRY] := TEN8;
    IAM := DOING := WANT := 0;
END ELSE
BEGIN
    DOING := TBLTODK;
    IF TPREC GTR 0 THEN % GET BACK WHAT WAS WRITTEN BEFORE,
BEGIN
    LLDISKREAD := TPKADR;
    TEMP := 3; WANT := VDISKREAD;
END ELSE GO TO CENTRAL;
END;
END SCANNER CASE 2;
BEGIN % SCANNER = 3; WE HAVE A SPEC CHAR
    IAM := 0;
END SCANNER CASE 3;
BEGIN % SCANNER = 4; ITS AN SS MSG
    WRITEASTERISK := TRUE;
    SSMSG;
    IAM := 0;
END CASE 4;
BEGIN % CASE 5 = NULL INPUT
    WRITEASTERISK := SEQMODE; % RESPOND IF ENDING SEQ. MODE
    SEQMODE := FALSE;
    IAM := 0;
END CASE 5;
END SCANNER CASES;

END CASE 0;
BEGIN % CASE 1: MOREVERBS
NODISK: BEGIN PLSWAIT(WORK);
    TWXOUT(LL,WORK[0],26,1);
    WAITFOR := VMCPK;
    NODISKMCP := TRUE;
    MSGPOINTER := 46;
    GO TO EXIT;
END;
SREG := SREG + (T := PAR + 1);
TEMP := TEMP + T;
PARAM := PARAM - T;
WANT := VERBV;
IF CHECKBIT(A[SREG],[9:9] DIV 2,RESTRICTEDVERBS) THEN
BEGIN
    ERRSW := VERBTABLE[A[SREG],[9:9]);
    WANT := VERROR; MSGPTR := 49;
    GO TO EXIT;
END;
IF SWDS = TEMP + PAR THEN IAM := DOING := 0;
END CASE 1;

```

```

26006500 T 0113
26006600 T 0115
26006700 T 0116
26006800 T 0118
26006900 T 0120
26007000 T 0124
26007100 T 0126
26007200 T 0130
26007300 T 0132
26007400 T 0134
26007500 T 0134
26007600 T 0136
26007700 T 0136
26007800 T 0138
26007900 T 0138
26008000 T 0141
26008100 T 0143
26008200 T 0143
26008300 T 0143
26008400 T 0143
26009200 T 0143
26009300 T 0144
26009400 T 0144
26009500 T 0144
26009600 T 0147
26009700 T 0147
26009800 T 0148
26009810 T 0148
26009820 T 0148
26009830 T 0151
26009840 T 0153
26009850 T 0154
26009900 T 0155
START OF SEGMENT ***** 50
50 IS 7 LONG, NEXT SEG 49
26010000 T 0155
26010100 T 0156
26010110 T 0156
26010120 T 0156
26010130 T 0159
26010140 T 0160
26010150 T 0161
26010160 T 0163
26010170 T 0165
26010180 T 0165
26010200 T 0165
26010300 T 0168
26010400 T 0169
26010500 T 0171
26010510 T 0172
26010520 T 0175
26010530 T 0176
26010540 T 0177
26010550 T 0179
26010560 T 0180
26010600 T 0180
26010700 T 0184

```

Acme Business Forms, Inc. 37 14127

```

BEGIN          % CASE 2: TBLTODK
  CHKPT(A[*]);
  IF TPREC LSS 2 THEN
    INC(TPREC)
  ELSE
  BEGIN
    TPREC := 0;
    INC(TPDKREL);
    IF TPKREL GTR 14 THEN
      BEGIN LITOUT(" (WAIT)",1); WANT:=VUPDATE END;
    END;
    TPENTRY := IAM := 0;
  END CASE 2;
  GO TO SCAN1; % CASE 3
  END ALL CASES;

```

```

EXIT:
  END USER INPUT;

```

```

26010800 T 0185
26010900 T 0185
26011000 T 0186
26011100 T 0188
26011200 T 0188
26011300 T 0192
26011400 T 0192
26011500 T 0195
26011520 T 0199
26011550 T 0200
26011600 T 0202
26011700 T 0202
26011800 T 0205
26011850 T 0206
26011900 T 0208
START OF SEGMENT ***** 51
51 IS 5 LONG, NEXT SEG 49
26012000 T 0208
26012100 T 0209
49 IS 212 LONG, NEXT SEG 8

```

```

*****%
PROCEDURE DISPATCH(LCW,A); BOOLEAN LCW; ARRAY A[0];
*****%
BEGIN % STARTS JOBS
  REAL CHUNKS;

  CLEAR(WORK[0],30);
  WORK[0]:=WORK[14]:=RUN1; % FIRST NAME,
  WORK[1]:=WORK[3] :=RUN2; % SECOND NAME
  WORK[2]:=0&2[8:38:10]; % EXECUTE CODE
  WORK[16]:=IF PROCESSLIMIT GTR 0 THEN PROCESSLIMIT ELSE 54000;
  WORK[17]:=IF IOLIMIT GTR 0 THEN IOLIMIT ELSE 54000;
  IF CORESIZE GTR 0 THEN % CORE ESTIMATE GIVEN
    BEGIN
      CORESIZE:=IF CORESIZE LSS 2048 THEN 32 ELSE CORESIZE DIV 64;
      WORK[20]:=CORESIZE & 1[2:47:1]; % CANT EXPAND CORE
    END;
  WORK[21]:=256; % STACK SIZE
  WORK[24]:=USERCODE; % USER CODE
  IAM:=0; WANT := WAITFOR := VMCPMSG; TEMP := 0;
  MATCH(RUN2,[0:42],"CANDE ") % CANDE JOB TO RUN
  BEGIN
    MATCH(RUN1,"QUIKLIST") CHUNKS:=2 ELSE
    MATCH(RUN1,"PUNCH ") CHUNKS:=2 ELSE
    MATCH(RUN1,"LOAD ") CHUNKS:=3 ELSE
    MATCH(RUN1,"DELETE ") CHUNKS:=3 ELSE
    MATCH(RUN1,"LFILES ") CHUNKS:=3 ELSE
    MATCH(RUN1,"LIST ") CHUNKS:=6 ELSE
    MATCH(RUN1,"PAPER ") CHUNKS:=5 ELSE
    CHUNKS:=4;
    WORK[20]:= (CHUNKS*16) & 1[2:47:1];
    WORK[0].[2:1]:=1; % NOT DS-ABLE
    WORK[1]:=ERRSW;

```

```

27000000 T 0061
27000100 T 0061
27000200 T 0061
27000300 T 0061
27000400 T 0061
START OF SEGMENT ***** 52
27000500 T 0000
27000600 T 0001
27000700 T 0003
27000800 T 0006
27000900 T 0008
27001000 T 0011
27001100 T 0015
27001200 T 0015
27001300 T 0016
27001400 T 0019
27001500 T 0021
27001600 T 0021
27001700 T 0023
27001800 P 0024
27001900 T 0027
27002000 T 0030
27002100 T 0030
27002200 T 0033
27002300 T 0042
27002400 T 0047
27002500 T 0052
27002600 P 0057
27002700 T 0062
27002800 T 0067
27002900 T 0069
27003000 T 0072
27003100 T 0075

```

*0104-

*0510-

```

WORK[18]:=1 & SCHEDULINR[46:47:1]; % PRIORITY
WORK[19]:=ESP1;
MSGPTR:=45;
END
ELSE
BEGIN % MUST BE USER TYPE JOB.
WORK[21] := STACKSIZE; % COMPILER EST. OR SPEC. VALUE
WORK[19]:=COMMONVALUE;
WORK[18] := 4 + REAL(SCHEDULELINE);
EOJMSG := NOT CONCISE; WRITEASTERISK := CONCISE;
IF RUN1.[6:6] = 0 THEN % REQUIRES SHIFTING
    BEGIN
    WORK[0]:=IF RUN1.[12:6]=1 THEN FILENAME ELSE " "&RUN1[6:12:36];
    END SHIFT;
    IF NOT CONCISE AND NOT COMPAR(RUN2,"DISK ") THEN
    LITOUT("RUNNING",3"100002");
    MSGPTR := 44;
    END IF NOT CANDE TYPE JOB;
WORK[2].[33:15]:=WORK[18]; % PRIORITY
RUNNING := TRUE;
STACKSIZE:=PROCESSLIMIT:=IOLIMIT:=CORESIZE:=COMMONVALUE:=0;
IF EQUATED THEN
IF NOT( (COMPAR(RUN2,"CANDE ") ) OR (COMPAR(RUN2,"DISK ") ) ) THEN
    BEGIN
    WORK[13]:=FIRSTESP; % LABEL EQUATION
    ENTERESP(=LL,0,A[*]); % SCRATCH TABLE ENTRIES
    EQUATED := FALSE;
    END;
    RUNJOB(LL&WAITFOR[33:41:7],WORK[0],0);
END OF DISPATCH;

```

```

27003200 T 0076
27003300 T 0079
27003400 T 0080
27003500 T 0081
27003600 T 0081
27003700 T 0081
27003800 T 0082
27003900 T 0083
27004000 T 0084
27004100 T 0087
27004200 T 0093
27004300 T 0094
27004400 T 0095
27004500 T 0100
27004600 T 0100
27004610 T 0104
27004700 T 0105
27004800 T 0106
27004900 T 0106
27005000 T 0109
27005100 T 0111
27005200 T 0114
27005300 T 0114
27005400 T 0119
27005500 T 0119
27005600 T 0121
27005700 T 0123
27005800 T 0126
27005900 T 0126
27006000 T 0128
52 IS 135 LONG, NEXT SEG 8

```

```

*****
PROCEDURE DISKREAD(LCW, A); BOOLEAN LCW; ARRAY A[0];
*****
    BEGIN
    LABEL T;
        IF NOT B(DOING) THEN % DO READ INITIATE.
        BEGIN
            IF PREVRCW = BASE LEQ 33 THEN % AREA IS TOO SMALL.
            BEGIN
                ERRSW := "TOOLONG";
                MSGPOINTER := 9;
                ERROR(SREG,LCW,A[*]);
            END IF NOT ENOUGH ROOM
            ELSE
            BEGIN
                TEMP := 31;
                SREG := (DOING:=1) + SREG;
                WAITFOR := IAM;
                DISKIO(LL&IAM[33:41:7],1,A[0],SREG,30,LLDISKREAD);
            END OF READ INITIATE
            END ELSE

```

```

28000000 T 0061
28000100 T 0061
28000200 T 0061
28000300 T 0061
28000350 T 0061
START OF SEGMENT ***** 53
28000400 T 0000
28000500 T 0000
28000600 T 0001
28000700 T 0002
28000800 T 0002
28000900 T 0003
28001000 T 0004
28001100 T 0006
28001200 T 0006
28001300 T 0006
28001400 T 0009
28001500 T 0009
28001600 T 0011
28001700 T 0012
28001800 T 0015
28001900 T 0015

```

```

BEGIN % READ IS COMPLETE NOW MOVE IT TO WORK,
      MOVE(30,WORDS,A[SREG],WORK[0]);
      IAM := 0;
END IF READ COMPLETE;
END OF DISKREAD;

```

```

28002000 T 0015
28002100 T 0016
28002200 T 0018
28002300 T 0019
28002400 T 0019
53 IS 20 LONG, NEXT SEG 8

```

```

*****
PROCEDURE OOPS (LCW, A); BOOLEAN LCW; ARRAY A[0];
*****

```

```

      BEGIN
LABEL T;

      SREG := BASE + A[BASE];
      ERRSW := A[INC(SREG)];
      MSGPOINTER := 13;
      ERROR (SREG, LCW, A[*]);
END OF OOPS;

```

```

30000000 T 0061
30000100 T 0061
30000200 T 0061
30000300 T 0061
30000350 T 0061
START OF SEGMENT ***** 54
30000400 T 0000
30000500 T 0001
30000600 T 0003
30000700 T 0004
30000800 T 0006
54 IS 7 LONG, NEXT SEG 8

```

```

*****
PROCEDURE DOGCATCHER(LCW,A); BOOLEAN LCW; ARRAY A[0];
*****

```

```

      BEGIN
LABEL T;

      IF NOT BOOLEAN (DOING) THEN
BEGIN % COLLECT TANK RECORDS.
      DOING := TEMP := 1; DOGHOUSE[LL]:=0;
      LITOUT("WAIT) ",NOCRLF);
      IF TPCOUNT NEG 0 THEN WANT := VUPDATE;
END
      ELSE % COLLECTION FINISHED.
BEGIN
      DOGLICENSE := FALSE; IAM := 0;
      DOGCATCHERS:= DOGCATCHERS +1;
      WECANUSEDATA:=DOGCATCHERS LEQ 0;
      LITOUT("PROCEED",CRLF);
END IF;
END DOGCATCHER;

```

```

31000000 T 0061
31000100 T 0061
31000200 T 0061
31000300 T 0061
31000350 T 0061
START OF SEGMENT ***** 55
31000400 T 0000
31000500 T 0000
31000600 T 0001
31000700 T 0003
31000800 T 0004
31000900 T 0007
31001000 T 0007
31001100 T 0007
31001200 T 0010
31001300 T 0013
31001400 T 0014
31001500 T 0016
31001600 T 0017
31001700 T 0017
55 IS 19 LONG, NEXT SEG 8

```

```

*****
PROCEDURE COMPILEIT(LCW,A); BOOLEAN LCW; ARRAY A[0];
*****

```

```

      BEGIN
LABEL T;

```

```

32000000 T 0061
32000100 T 0061
32000200 T 0061
32000300 T 0061
32000350 T 0061
START OF SEGMENT ***** 56

```


%
%
%

GENERATE THE LABEL EQUATIONS FOR THE COMPILER.

CLEAR(WORK[0],59);
WORK[0]:=WORK[14]:=RUN1,[6136]; %FIRST NAME
WORK[1]:=WORK[15]:=USERCODE; %SECND NAME
WORK[3]:=10; WORK[17]:=12; %FILES TYPE
WORK[4] :="CODE "&4[1:43:5];
WORK[18]:=CARD "&4[1:43:5];
DISKWRITE(ESP2); CLEAR(WORK[0],29);

%
%
%

GENERATE THE OBJECT SKELETON ON THE DISK.

WORK[16]:=WORK[17]:=3"7777777777"; %TIME LIMITS
WORK [18]:=5; % PRIORITY %0525=
WORK[20]:=1; %NO CORE EST
WORK[21]:=512; %STACK SIZE
DISKWRITE(ESP1); CLEAR(WORK[0],29);

%
%
%

GENERATE THE COMPILER CALLOUT.

WORK[14]:=ABS(WORK[0]:=RUN2&16[1:43:5]);%COMPILER,
WORK[01]:=IF RUN1.[6112]="1S" THEN
FILENAME ELSE RUN1; %FILENAME,
WORK[02]:=4&4[8:38:10]&ESP1[18:33:15];%OBJ,S
WORK[3]:=DISK "; WORK[13]:=ESP2;%LBL EQN
WORK[18]:=4+SCHEDULINR; % PRIORITY
WORK[19]:=IF COMMONVALUE LSS 0 THEN 1 ELSE 3;
WORK[22]:=8; WORK[24]:=USERCODE; %SAVE&USER
WORK[16] :=
IF PROCESSLIMIT GTR 0 THEN PROCESSLIMIT ELSE 3"7777777777";
WORK[17] :=
IF IOLIMIT GTR 0 THEN IOLIMIT ELSE 3"7777777777";
WORK[21] :=
IF STACKSIZE GTR 0 THEN STACKSIZE ELSE 512;
STACKSIZE:=PROCESSLIMIT:=IOLIMIT:=CORESIZE:=COMMONVALUE:=0;

%
%
%

NOW TELL CANDE & THE MCP.

COMPILING := TRUE; WRITEASTERISK := CONCISE;
BOJMSG := NOT CONCISE; EOJMSG := NOT CONCISE;
MSGPTR := 43; %DCS
IAM:=0; WANT:=WAITFOR:=VMCPMSG; TEMP := 3;
RUNJOB(LL&WAITFOR[33:41:7],WORK[0],0);
ESP1:=GETESPDISK; ESP2:=GETESPDISK;
END OF COMPILEIT;

32000400 T 0000
32000500 T 0000
32000600 T 0000
32000700 T 0000
32000800 T 0001
32000900 T 0004
32001000 T 0006
32001100 T 0009
32001200 T 0011
32001300 T 0013
32001400 T 0018
32001500 T 0018
32001600 T 0018
32001700 T 0018
32001800 P 0020
32001900 T 0022
32002000 T 0023
32002100 T 0024
32002200 T 0029
32002300 T 0029
32002400 T 0029
32002500 T 0029
32002600 T 0033
32002700 T 0035
32002800 T 0037
32002900 T 0040
32003000 T 0043
32003050 T 0045
32003100 T 0049
32003110 T 0051
32003120 T 0052
32003130 T 0054
32003140 T 0055
32003150 T 0058
32003160 T 0058
32003170 T 0061
32003200 T 0061
32003300 T 0064
32003400 T 0064
32003500 T 0064
32003510 T 0069
32003550 T 0076
32003600 T 0077
32003700 T 0080
32003800 T 0082
32003900 T 0085

56 IS 90 LONG, NEXT SEG 8

PROCEDURE SETUPFILES(LCW,A); BOOLEAN LCW; ARRAY A[0];

BEGIN
REAL T, T1, T2;

33000000 T 0061
33000100 T 0061
33000200 T 0061
33000300 T 0061
33000400 T 0061

START OF SEGMENT ***** 57

```

LABEL RETRY, EXIT)

DEFINE
LOOPCOUNT = A[BASE].[18:15]#, % COUNTER FOR "IN-USE" CONDITION
FILECOUNT = A[BASE].[33:15]#, % COUNTER FOR FILES CREATED
STACKSIZE = 5#; % STACK WORDS REQUIRED FOR THIS ROUTINE

```

```

DEFINE
SEGMENTZEROADDRESS(SEGMENTZEROADDRESS1)=
  A[BASE+(SEGMENTZEROADDRESS1*2)-1]#,
HEADERADDRESS(HEADERADDRESS1)=
  A[BASE+(HEADERADDRESS1*2)]#;

```

```

STREAM PROCEDURE ERRORMESSAGE(T,USERCODE,L); VALUE T,USERCODE;
BEGIN
SI:=LOC T; DI:=L;
2(SI:=SI+1; DS:=7CHR; DS:=LIT"/"); DI:=DI-1;
DS:=17LIT" CLEARED(IN USE)+";
END;

```

```

CASE DOING OF
BEGIN
  BEGIN % CASE 0 = INITIALIZATION OR RE-TRY
    FILESON := FALSE;
    A[BASE]:=0; % RESET COUNTERS
  RETRY: % BRANCH POINT FOR ANOTHER ATTEMPT
    CLEAR(A[BASE+1],3);
    FILECOUNT := 0;
    LASTRECORD := 0;
    T := MAKEFN("1T",LL);
    MAKEFILE(LL&IAM[33:41:7],1,T,USERCODE,0,10);
    T := T&"P"[12:42:6];
    MAKEFILE(LL&IAM[33:41:7],1,T,USERCODE,(0&1[41:47:1]),TBSZ);
    % [41:1] IN FILE TYPE SETS "FIXED" BIT IN HEADER
    TABLEINFO := 3"10000";
    TEMP := STACKSIZE;
    DOING := 1;
    WAITFOR := IAM;
    END CASE 0;

  %.....
  BEGIN % CASE 1 = RESULTS OF CREATION REQUEST
    FILECOUNT := FILECOUNT + 1; % COUNT UP NUMBER OF "RESULTS"
    SEGMENTZEROADDRESS(FILECOUNT):=EVENTS[1];
    HEADERADDRESS(FILECOUNT):=EVENTS[2];
    IF FILECOUNT=1 THEN WAITFOR:=IAM % "1T" RESULTS, WAIT FOR "1P"
  ELSE
    BEGIN
      IF ((T1:=SEGMENTZEROADDRESS(1)) LSS 63) OR
        ((T2:=SEGMENTZEROADDRESS(2)) LSS 63) THEN
        BEGIN % INCORRECT RESULT
          IF (T1=0) OR (T2=0) THEN % NO USER DISK CONDITION
            BEGIN
              MSGPOINTER:=46; % NO USER DISK
            END
          END
        END
      END
    END
  END

```

```

33000500 T 0000
33000600 T 0000
33000700 T 0000
33000800 T 0000
33000900 T 0000
33001000 T 0000
33001100 T 0000
33001200 T 0000
33001300 T 0000
33001400 T 0000
33001500 T 0000
33001600 T 0000
33001700 T 0000
33001800 T 0000
33001900 T 0000
33002000 T 0000
33002100 T 0000
33002200 T 0002
33002300 T 0004

33002400 T 0005
33002500 T 0005
33002600 T 0005
33002700 T 0005
33002800 T 0005
33002900 T 0008
33003000 T 0009
33003100 T 0010
33003200 T 0011
33003300 T 0013
33003400 T 0015
33003500 T 0016
33003600 T 0020
33003700 T 0021
33003800 T 0026
33003900 T 0026
33004000 T 0027
33004100 T 0028
33004200 T 0029
33004300 T 0029
33004400 T 0030
33004500 T 0030
33004600 T 0030
33004700 T 0033
33004800 T 0037
33004900 T 0040
33005000 T 0042
33005100 T 0043
33005200 T 0045
33005300 T 0048
33005400 T 0051
33005500 T 0051
33005600 T 0053
33005700 T 0054

```

Moore Business Forms, Inc., NY

```

FILENAME := SOURCEFILE := 0;
IF T1 GEQ 63 THEN % "1T" FILE EXISTS
REMOVEFILE(LL&13[33:41:7],MAKEFN("1T",LL),USERCODE);
IF T2 GEQ 63 THEN
REMOVEFILE(LL&13[33:41:7],MAKEFN("1P",LL),USERCODE);
IAM:=0;
GO TO EXIT;
END; % IF NO USER DISK
IF (T1=1) OR (T2=1) THEN % FILE "IN USE" CONDITION
BEGIN
T := (IF T1=1 THEN 1 ELSE 2); % INDEX VALUE
LLOISKREAD:=HEADERADDRESS(T);
DOING := 2;
WANT := VDISKREAD;
GO TO EXIT;
END;
GOERR("BADFILE",22); % SYSTEM ERROR, FUNNY RESULT
END; % IF THERE WAS AN ERROR IN THE CREATION PROCESS
ACTBASE+10] := TEN8;
WRKTLADR := SEGMENTZEROADDRESS(2); % FIRST RECORD OF "1P" FILE
CANCELCKPT(A[*]);
WORK[0]:=0; WORK[1]:=TEN8;
DISKWRITE(SEGMENTZEROADDRESS(1)); % SET UP "1T" FILE RECORD
FILESOK := TRUE;
IAM:=0;
END; % IF ALL RESULTS WERE IN
END; % CASE 1
*.....
BEGIN % CASE 2 - HEADER IN CORE
LOOPCOUNT := LOOPCOUNT + 1;
IF LOOPCOUNT GTR 2 THEN GOERR("BADFILE",22); % CANT CLEAR IT
WORK[4]:=WORK[4] & 0[1:45:3] & 0[16:28:20];
WORK[9]:=WORK[9] & 0[1:20:28];
T:=(IF SEGMENTZEROADDRESS(1)=1 THEN 1 ELSE 2);
DISKWRITE(HEADERADDRESS(T));
T:=(IF T=1 THEN "1T" ELSE "1P");
T:=MAKEFN(T,LL);
ERRORMESSAGE(T,USERCODE,L);
TWXOUT(0,L[0],32,1);
GO TO RETRY;
END; % CASE 2
END ALL CASES;

EXIT;
END PROCEDURE SETUPFILES;

```

33005800	T	0055	
33005900	T	0057	
33006000	T	0058	
33006100	T	0063	
33006200	T	0063	
33006300	T	0068	
33006400	T	0069	
33006500	T	0069	
33006600	T	0069	
33006700	T	0071	
33006800	T	0072	
33006900	T	0074	
33007000	T	0077	
33007100	T	0078	
33007200	T	0078	
33007300	T	0079	
33007400	T	0079	
33007500	T	0085	
33007600	T	0085	
33007700	T	0086	
33007800	T	0089	
33007900	T	0091	
33008000	T	0093	
33008100	T	0098	
33008200	T	0100	
33008300	T	0101	
33008400	T	0101	
33008500	T	0102	
33008600	T	0102	
33008700	T	0102	
33008800	T	0105	
33008900	T	0115	
33009000	T	0118	
33009100	T	0121	
33009200	T	0125	
33009300	T	0130	
33009400	T	0132	
33009500	T	0134	
33009600	T	0136	
33009700	T	0137	
33009800	T	0138	
33009900	T	0138	
START OF SEGMENT *****			58
58 IS	4 LONG,	NEXT SEG	57
33010000	T	0139	
33010100	T	0140	
57 IS	143 LONG,	NEXT SEG	8

```

*****%
PROCEDURE ERROR(S,LCW,A); VALUE S; REAL S; BOOLEAN LCW; ARRAY A[0];
*****%
BEGIN
REAL T;

```

34000000	T	0061	
34000100	T	0061	
34000200	T	0061	
34000300	T	0061	
34000400	T	0061	
START OF SEGMENT *****			59

```

REAL STREAM PROCEDURE MKER(ERRSW,L,CHRTOG,STRTOG,MINTOG,PLSTOG);
VALUE ERRSW,CHRTOG,STRTOG,MINTOG,PLSTOG;
BEGIN LABEL EXIT;
  SI:=LOC ERRSW; DI:=L; DS:=5LIT"ERR "; TALLY:=5;
  CHRTOG(SI:=SI+7; DS:=1CHR; TALLY:=TALLY+1; JUMP OUT TO EXIT);
  STRTOG(SI:=SI+1; DS:=7CHR; TALLY:=TALLY+7; JUMP OUT TO EXIT);
  MINTOG(DS:=LIT"-"; TALLY:=TALLY+1);
  PLSTOG(DS:=LIT"+"; TALLY:=TALLY+1);
  MINTOG:=DI; DS:=8 DEC; DI:=MINTOG; DS:=7FILL;
  SI:=MINTOG; DI:=MINTOG;
  B(IF SC=" " THEN SI:=SI+1 ELSE
    BEGIN DS:=CHR; TALLY:=TALLY+1; END);
EXIT:
  MKER := TALLY;
END STREAM PROCEDURE MKER;

```

```

34000500 T 0000
34000600 T 0000
34000700 T 0000
34000800 T 0000
34000900 T 0001
34001000 T 0004
34001100 T 0006
34001200 T 0008
34001300 T 0010
34001400 T 0011
34001500 T 0012
34001600 T 0013
34001700 T 0014
34001800 T 0014
34001900 T 0015

```

```

%.....
T:=MKER(ERRSW,[6:42],L,(ERRSW,[1:2]=1),(ERRSW,[6:12] GTR 0),
(ERRSW,[1:2]=2),(ERRSW,[1:2]=3));
TWXOUT(LL,L[0],T,1);
WRITEASTERISK := FALSE;
T := MSGPTR;
LINECLEAR(LCW,A[*]);
MSGPTR := T;
IF SCHEDULELINE OR HELPFUL THEN
  BEGIN % TERMINATE SCHEDULE I/P & EXPLAIN WHY
    IAM := VTEACHER; DOING := 1; % DONT SCAN INPUT
    LLDISKREAD := MSGPTR DIV 3 + MSGFILELOC;
    WANT := VDISKREAD;
    WRITEASTERISK := TRUE;
  END;
END ERROR PROCEDURE;

```

```

34002000 T 0016
34002100 T 0016
34002200 T 0022
34002300 T 0025
34002400 T 0027
34002500 T 0029
34002600 T 0030
34002700 T 0032
34002800 T 0033
34002900 T 0035
34003000 T 0035
34003100 T 0037
34003200 T 0039
34003300 T 0040
34003400 T 0042
34003500 T 0042

```

59 IS 45 LONG, NEXT SEG 8

```

*****
PROCEDURE CLOSEWORKTABLE (LCW, A); BOOLEAN LCW; ARRAY A[0];
*****
BEGIN
  LABEL CASE1, EXIT;
  CASE DOING OF
  BEGIN
  BEGIN % CASE 0
    IF GOODOBJ THEN
      GOODOBJ := TPCOUNT = 0;
    IF NOT FILESOK THEN GOERR("NO FILE",1);
    IF TPENTRY LSS 9 THEN A[TBASE+TPENTRY] := TEN8;
    IF TPREC GTR 0 THEN
      BEGIN
        LLDISKREAD := TPKADR;
        DOING := 1;

```

```

35000000 T 0061
35000100 T 0061
35000200 T 0061
35000300 T 0061
35000400 T 0061
START OF SEGMENT ***** 60
35000500 T 0000
35000600 T 0000
35000700 T 0000
35000800 T 0000
35000900 T 0001
35001000 T 0005
35001100 T 0013
35001200 T 0017
35001300 T 0019
35001400 T 0019
35001500 T 0022

```

```

        WANT := VDISKREAD;
    END ELSE GO TO CASE1
    END CASE 0;
CASE1:
    BEGIN                % CASE 1
    CHKPT(A[*]);
    % VOID
    IAM := 0;
    END CASE 1;
    END ALL CASES;

EXIT:
    END CLOSEWORKTABLE;

```

```

35001600 T 0023
35001700 T 0023
35001800 T 0023
35001900 T 0024
35002000 T 0025
35002100 T 0025
35002200 T 0026
35002300 T 0026
35002400 T 0027
35002500 T 0027
START OF SEGMENT ***** 61
61 IS 3 LONG, NEXT SEG 60
35002600 T 0028
35002700 T 0028
60 IS 30 LONG, NEXT SEG 8

```

```

*****
PROCEDURE CHNGPSWD(LCW,A); BOOLEAN LCW; ARRAY A[0];
*****
BEGIN
LABEL ERR,EXIT;

STREAM PROCEDURE OLDPSWD(W);
    BEGIN DI:=W; DS:=30LIT"PLEASE ENTER YOUR OLD PASSWORD"; END;

```

```

36000000 T 0061
36000100 T 0061
36000200 T 0061
36000300 T 0061
36000400 T 0061
START OF SEGMENT ***** 62
36000500 T 0000
36000600 T 0000

```

```

STREAM PROCEDURE NEWPSWD(W);
    BEGIN DI:=W; DS:=28LIT"NEXT ENTER YOUR NEW PASSWORD"; END;

```

```

36000700 T 0004
36000800 T 0004

```

```

STREAM PROCEDURE VFYPSWD(W);
    BEGIN DI:=W; DS:=31LIT"PLEASE REPEAT YOUR NEW PASSWORD"; END;

```

```

36000900 T 0009
36001000 T 0009

```

```

STREAM PROCEDURE SCANPSWD(W,S,EADRS); VALUE EADRS;
    BEGIN LOCAL SV,DV;
    SI:=W;
    DI:=S; DS:=8LIT"0 "; DI:=DI-7;
    7(IF SC = LEFTARROW THEN
        BEGIN
            SV:=SI; SI:=LOC SV; SI:=SI+5;
            DV:=DI; DI:=LOC EADRS; DI:=DI+5;
            IF 3SC=DC THEN JUMP OUT;
            SI:=SV; DI:=DV;
        END;
    IF SC="" THEN JUMP OUT;
    IF SC="," THEN JUMP OUT; DS:=CHR);
    END STREAM PROCEDURE SCANPSWD;

```

```

36001010 T 0014
36001020 T 0014
36001030 T 0015
36001040 T 0015
36001050 T 0017
36001060 T 0017
36001070 T 0017
36001080 T 0019
36001090 T 0019
36001100 T 0020
36001110 T 0021
36001115 T 0021
36001120 T 0022
36001130 T 0024

```

```

*.....
CASE DOING OF
BEGIN
  BEGIN % CASE 0 = REQUESTING CURRENT PASSWORD
  IF SCHEDULELINE THEN GOERR("CHANGE ",40);
  OLDPSWD(WORK); TWXOUT(LL,WORK[0],30,1);
  BLACKOUT;
  CLEAR(A[BASE],2); TEMP:=3; DOING:=1;
  WAITFOR := VINPUT;
  END CASE 0;
*.....
  BEGIN % CASE 1 = SCANNING FOR CURRENT PASSWORD
  SCANPSWD(WORK[1],A[BASE],WORKENDADRS);
  MATCH(A[BASE],[6:42]," ") GO ERR;
  NEWPSWD(WORK); TWXOUT(LL,WORK[0],28,1);
  BLACKOUT;
  DOING := 2;
  WAITFOR := VINPUT;
  END CASE 1;
*.....
  BEGIN % CASE 2 = SCANNING FOR NEW PASSWORD
  SCANPSWD(WORK[1],A[BASE+1],WORKENDADRS);
  MATCH(A[BASE+1],[6:42]," ") GO ERR;
  VFYPSWD(WORK); TWXOUT(LL,WORK[0],31,1);
  BLACKOUT;
  DOING := 3;
  WAITFOR := VINPUT;
  END CASE 2;
*.....
  BEGIN % CASE 3 = SCANNING FOR VERIFICATION
  SCANPSWD(WORK[1],A[BASE+2],WORKENDADRS);
  MATCH(A[BASE+1],A[BASE+2]) ELSE GO ERR;
  LLDISKREAD := USERLOC;
  WANT := VDISKREAD;
  DOING := 4;
  END CASE 3;
*.....
  BEGIN % CASE 4 = USERS/CANDE RECORD IN CORE
  MATCH(A[BASE],[6:42],WORK[2]) ELSE GO ERR;
  WORK[2]:=A[BASE+1],[6:42];
  DISKWRITE(USERLOC);
  LLDISKREAD:=USERBASE; % USERS/CANDE RECORD ZERO %0521=
  WANT := VDISKREAD;
  DOING := 5;
  END CASE 4;
*.....
  BEGIN % CASE 5 = PUT NEW DATE IN USERS/CANDE FILE
  WORK[2]:=DATE;
  DISKWRITE (USERBASE); %0521=
  IAM := 0;
  END CASE 5;
END ALL CASES;

```

```

36001140 T 0024
36001200 T 0024
36001300 T 0025
36001400 T 0025
36001500 T 0025
36001600 T 0033
36001700 T 0035
36001800 T 0036
36001900 T 0039
36002000 T 0039
36002100 T 0040
36002200 T 0040
36002300 T 0040
36002400 T 0042
36002500 T 0046
36002600 T 0048
36002700 T 0049
36002800 T 0050
36002900 T 0050
36003000 T 0051
36003100 T 0051
36003200 T 0051
36003300 T 0054
36003400 T 0058
36003500 T 0060
36003600 T 0061
36003700 T 0062
36003800 T 0062
36003900 T 0063
36004000 T 0063
36004100 T 0063
36004200 T 0066
36004300 T 0069
36004400 T 0071
36004500 T 0071
36004600 T 0072
36004700 T 0073
36004800 T 0073
36004900 T 0073
36005000 T 0076
36005100 T 0080
36005200 P 0083
36005300 T 0084
36005400 T 0085
36005500 T 0086
36005600 T 0086
36005700 T 0086
36005800 T 0086
36005900 P 0088
36006000 T 0091
36006100 T 0092
36006200 T 0092

```

```

GO TO EXIT;
ERR;
LITOUT("BADCODE",1); MSGPTR:=21; LINECLEAR(LCW,A[*]);
EXIT;
END PROCEDURE CHANGEPASSWORD;

```

```

63 IS      7 LONG, NEXT SEG  62
36006300 T  0093
36006400 T  0095
36006500 T  0095
36006600 T  0099
36006700 T  0099
62 IS     101 LONG, NEXT SEG  8

```

```

*****
PROCEDURE THYME(LCW,A);
*****
BOOLEAN LCW;
ARRAY A[0];
BEGIN
  LABEL DUMMY;

  WORK[0] := "USER IS"; BLANK(WORK[0]);
  WORK[1] := USERCODE; BLANK(WORK[1]);
  WORK[2] := DECCONV(LL); WORK[2] := WORK[2] & "LINE "[6:18:30];
  BLANK(WORK[2]);
  TWXOUT(LL,WORK[0],24,CRLF);
  WORK[0] := "TIME IS"; BLANK(WORK[0]);
  TIMEOFDAY(WORK[1],TIMECONV(TIME(1),TRUE));
  TWXOUT(LL,WORK[0],18,CRLF);
  WORK[0] := "C&E USE"; BLANK(WORK[0]);
  TIMEUSED(WORK[1],TIMECONV(PTIME,FALSE));
  TWXOUT(LL,WORK[0],33,1);
  WORK[0] := "EXECUTE"; BLANK(WORK[0]);
  TIMEUSED(WORK[1],TIMECONV(ETIME,FALSE));
  TWXOUT(LL,WORK[0],33,1);
  WORK[0] := "IO TIME"; BLANK(WORK[0]);
  TIMEUSED(WORK[1],TIMECONV(IOTOT,FALSE));
  TWXOUT(LL,WORK[0],33,1);
  WORK[0] := DATE;
  TWXOUT(LL,WORK[0],8,2);
  IAM := 0;
END OF THYME;

```

```

37001000 T  0061
37001100 T  0061
37001200 T  0061
37001300 T  0061
37001400 T  0061
37001500 T  0061
37001550 T  0061
START OF SEGMENT ***** 64
37001600 T  0000
37001700 T  0002
37001710 T  0004
37001720 T  0009
37001800 T  0010
37001900 T  0012
37002000 T  0014
37002100 T  0017
37002200 T  0018
37002300 T  0021
37002400 T  0023
37002500 T  0025
37002600 T  0027
37002700 T  0030
37002800 T  0031
37002900 T  0034
37003000 T  0036
37003100 T  0038
37003200 T  0039
37003300 T  0041
37003400 T  0042
64 IS     49 LONG, NEXT SEG  8

```

```

*****
PROCEDURE CARRIAGE(LCW,A); BOOLEAN LCW; ARRAY A[0];
*****
BEGIN
  REAL T;

  LABEL ERR,EXIT;
  CASE DOING OF
  BEGIN
    BEGIN % CASE 1 SYNTAX CHECK
      IF DEVICE NEQ TELETYPE THEN GOERR("CC ",56);
      TEMP:=1; WRITEASTERISK:=TRUE;

```

```

38000000 T  0061
38000100 T  0061
38000200 T  0061
38000300 T  0061
38000400 T  0061
START OF SEGMENT ***** 65
38000500 T  0000
38000600 T  0000
38000700 T  0000
38000710 T  0000
38000720 T  0000
38000800 T  0008

```

```

IF A[SREG]=BASE-A[BASE].[18:15] GEQ 1 THEN
  BEGIN
  MATCH(A[SREG]=SREG+1).[6:42],"LONG  ") T:=1 ELSE
  MATCH(A[SREG].[6:42],"SHORT  ") T:=0 ELSE GO TO ERR;
  END
ELSE
  BEGIN
  A[SREG]:="CC  ";
  GOERR(A[SREG],8);
  END;
ERR:
IF (T=1 AND CCLONG) OR (T=0 AND NOT CCLONG) THEN
  BEGIN
  IAM:=0; GO TO EXIT;
  END;
A[BASE]:=T; TEMP:=1;
LLDISKREAD := USERLOC; % USERS/CANDE RECORD LOCATION
WANT:=VDISKREAD;
DOING := 1;
END; % CASE 0
*.....
BEGIN % CASE 1 - USERS/CANDE RECORD IN CORE
CCLONGBIT := REAL(A[BASE]=1);
WORK[11]:=OPTIONBITS; % UPDATE RECORD
DISKWRITE(USERLOC); % REPLACE USERS/CANDE RECORD
CCOM(LL,USERCODE,A[BASE]);
IAM := 0;
END CASE 1;
END ALL CASES;

EXIT:
END PROCEDURE CARRIAGE CONTROL;

```

```

38000900 T 0011
38001000 T 0014
38001100 T 0014
38001200 T 0019
38001300 T 0026
38001400 T 0026
38001500 T 0026
38001600 T 0028
38001700 T 0029
38001800 T 0036
38001900 T 0036
38002000 T 0040
38002100 T 0041
38002200 T 0042
38002300 T 0042
38002400 T 0044
38002500 T 0045
38002600 T 0046
38002700 T 0047
38002800 T 0047
38002900 T 0047
38003000 T 0047
38003100 T 0051
38003200 T 0052
38003300 T 0056
38003400 T 0057
38003500 T 0058
38003600 T 0059
START OF SEGMENT ***** 66
66 IS 3 LONG, NEXT SEG 65
38003700 T 0059
38003800 T 0060
65 IS 63 LONG, NEXT SEG 8

```

```

*****
PROCEDURE CHNGNAME(LCW,A); BOOLEAN LCW; ARRAY A[0];
*****
BEGIN
STREAM PROCEDURE NEWNAME(A);
  BEGIN
  DI:=A; DS:=20LIT"ENTER YOUR NEW NAME=";
  END STREAM;

```

```

39000000 T 0061
39000100 T 0061
39000200 T 0061
39000300 T 0061
39000400 T 0061
START OF SEGMENT ***** 67
39000500 T 0000
39000600 T 0000
39000700 T 0003

```

```

STREAM PROCEDURE SAVENAME(F,T,EADRS); VALUE EADRS;
  BEGIN
  LOCAL SV,DV; LABEL L;
  SI:=F; DI:=T; DS:=LIT"0"; DS:=15LIT" "; DI:=DI-15;
  L: IF SC=" " THEN BEGIN SI:=SI+1; GO L END;
  15(IF SC=LEFTARROW THEN
  BEGIN

```

```

39000800 T 0003
39000900 T 0003
39001000 T 0004
39001100 T 0004
39001200 T 0007
39001300 T 0009
39001310 T 0010

```



```

SV:=SI; SII=LOC SV; SII=SI+5;
DV:=DI; DII=LOC EADRS; DII=DI+5;
IF 3SC=DC THEN JUMP OUT;
SII=SV; DII=DV;
END;
DS:=CHR);
END STREAM;

```

```

39001320 T 0010
39001330 T 0011
39001340 T 0012
39001350 T 0013
39001360 T 0013
39001370 T 0013
39001400 T 0015

```

```

LABEL EXIT;
IF SCHEDULELINE THEN GOERR("CHANGE ",40);
CASE DOING OF
BEGIN
BEGIN % CASE 0.
NEWNAME(WORK);
TWXOUT(LL,WORK[0],20,NOCRLF);
DOING:=1; WAITFOR:=VINPUT;
END CASE 0;
BEGIN % CASE 1.
SAVENAME(WORK[1],A[BASE],WORKENDADRS);
DOING:=TEMP:=2;
LLDISKREAD:=USERLOC;
WANT:=VDISKREAD;
END CASE 1;
BEGIN % CASE 2.
MOVE(2,WORDS,A[BASE],WORK[4]);
DISKWRITE(USERLOC);
IAM := 0;
END CASE 2;
END ALL CASES;

```

```

39001500 T 0015
39001600 T 0015
39001700 T 0023
39001800 T 0023
39001900 T 0023
39002000 T 0023
39002100 T 0024
39002200 T 0026
39002300 T 0028
39002400 T 0028
39002500 T 0028
39002600 T 0030
39002700 T 0032
39002800 T 0033
39002900 T 0034
39003000 T 0034
39003100 T 0034
39003200 T 0037
39003300 T 0040
39003400 T 0041
39003500 T 0041

```

```

START OF SEGMENT ***** 68
68 IS 4 LONG, NEXT SEG 67
39003600 T 0042
67 IS 45 LONG, NEXT SEG 8

```

```
EXIT:END CHANGE NAME;
```

```

*****
PROCEDURE MONITORR(LCW,A); BOOLEAN LCW; ARRAY A[0];
*****
BEGIN
REAL PC,WORD;
LABEL EXIT;
TEMP := 3;
CASE DOING OF
BEGIN
%.....
BEGIN % CASE 0 - SYNTAX CHECK
PC := A[SREG:=BASE-A[BASE] ],[18:15]; % PARAMETER COUNT
WORD := A[SREG:=SREG+1]; % FIRST PARAMETER
IF PC NEQ 1 THEN % WRONG NUMBER OF PARAMETERS
BEGIN
IF PC=0 THEN GUERR("FILENAM",6); % FILE NAME REQUIRED

```

```

40000000 T 0061
40000100 T 0061
40000200 T 0061
40000300 T 0061
40000400 T 0061
40000500 T 0000
40000600 T 0000
40000700 T 0000
40000800 T 0001
40000900 T 0001
40001000 T 0001
40001100 T 0001
40001200 T 0004
40001300 T 0006
40001400 T 0007
40001500 T 0007

```

```
START OF SEGMENT ***** 69
```

Mobile Business Terminal, Inc. sv 1472

```

GOERR(WORD,8); % ILLEGAL PARAMETER
END;
IF WORD,[3:3]=0 THEN GOERR(WORD,8); % NOT A FILE NAME
IF WORD,[3:3] GTR 6 THEN GOERR(WORD,42); % 6 CHARACTERS MAX
IF FINDAWORD(WORD,RESWRDTABLE[1],NUMOFRESWRDS) NEQ 0 THEN
GOERR(WORD,29);
A[BASE] := WORD := WORD,[6:42]; % STORE FILE NAME
A[BASE+1] := A[BASE+2] := 0; % RESET EVENTS COUNTER AND ERR. LOC.
FIND(LL&IAM[33:41:7],0,WORD,USERCODE,USERCODE);
FIND(LL&IAM[33:41:7],0,WORD,[6:36],USERCODE,USERCODE);
DOING := 1; WAITFOR := IAM;
END; % CASE 0
*.....
BEGIN % CASE 1 = RESULTS OF DIRECTORY SEARCH
IF EVENTS[1] GEQ 0 THEN % FILE WAS IN DIRECTORY
IF A[BASE+1]=0 THEN % THIS IS FIRST FILE FOUND
IF A[BASE+2]=0 THEN A[BASE+1]:=A[BASE] % SOURCE VERSION
ELSE A[BASE+1]:=A[BASE],[6:36]; % OBJECT VERSION
IF A[BASE+2]:=A[BASE+2]+1 LSS 2 THEN WAITFOR:=IAM ELSE
IF A[BASE+1] NEQ 0 THEN GOERR(A[BASE+1],3) ELSE
BEGIN
MAKEFILE(LL&IAM[33:41:7],1,A[BASE],USERCODE,8,200);
DOING:=2; WAITFOR:=IAM;
END;
END; % CASE 1
*.....
BEGIN % CASE 2 = ADJUST NEW FILE HEADER
IF EVENTS[1] = 0 THEN GOERR("NO DISK",46);
LLDISKREAD:=A[BASE+1]:=EVENTS[2]; % HEADER ADDRESS
DOING := 3; WANT := VDISKREAD;
END; % CASE 2
*.....
BEGIN % CASE 3 = HEADER IN CORE
WORK[0]:=10 & 30[30:36:12] & 300[15:33:15] & 10[1:34:14];
WORK[3],[2:10] := 7; % SAVE FACTOR
WORK[7] := -1; % EOF POINTER
WORK[9],[43:5] := 20; % NUMBER OF ROWS
DISKWRITE(A[BASE+1]); % REPLACE HEADER
LLDISKREAD:=USERLOC; % USERS/CANDE RECORD ADDRESS
DOING := 4; WANT := VDISKREAD;
END; % CASE 3
*.....
BEGIN % CASE 4 = USERS/CANDE RECORD IN CORE
MONITORBIT := TRUE;
WORK[12] := A[BASE]; % MONITOR FILE NAME
DISKWRITE(USERLOC); % REPLACE RECORD
IAM := 0;
END; % CASE 4
END ALL CASES;

```

```

40001600 T 0015
40001700 T 0019
40001800 T 0019
40001900 T 0025
40002000 T 0032
40002100 T 0034
40002200 T 0039
40002300 T 0041
40002400 T 0045
40002500 T 0048
40002600 T 0053
40002700 T 0054
40002800 T 0055
40002900 T 0055
40003000 T 0055
40003100 T 0056
40003200 T 0058
40003300 T 0061
40003400 T 0065
40003500 T 0070
40003600 T 0078
40003700 T 0078
40003800 T 0082
40003900 T 0083
40004000 T 0083
40004100 T 0084
40004200 T 0084
40004300 T 0084
40004400 T 0092
40004500 T 0095
40004600 T 0096
40004700 T 0097
40004800 T 0097
40004900 T 0097
40005000 T 0101
40005100 T 0103
40005200 T 0105
40005300 T 0107
40005400 T 0111
40005500 T 0113
40005600 T 0114
40005700 T 0115
40005800 T 0115
40005900 T 0115
40006100 T 0117
40006200 T 0119
40006300 T 0122
40006400 T 0123
40006500 T 0123

```

```

START OF SEGMENT ***** 70
70 IS 6 LONG, NEXT SEG 69
40006600 T 0124
40006700 T 0125
69 IS 128 LONG, NEXT SEG 8

```

```

EXIT;
END PROCEDURE MONITRR;

```

```

*****
PROCEDURE SETIT(LCW,A); BOOLEAN LCW; ARRAY A[0];
*****
BEGIN
REAL NC, PTR, WORD, MAXPTR, TYPE, SETBIT, X;

LABEL CASE2, EXIT;
DEFINE ERR(ERR1, ERR2, ERR3)=
  BEGIN
  ERRSW := ERR1 & ERR2[1:43:5]; MSGPTR := ERR3;
  IAM := VERROR; GO TO EXIT;
  END#;
*.....
CASE DOING OF
  BEGIN
  *.....
  BEGIN % CASE 0 - SYNTAX ANALYSIS
  PTR:=WORKADRS; MAXPTR:=WORKENDADRS; NC:=0; % SETUP
  WHILE (TYPE:=SCAN(PTR,WORD,"9",MAXPTR,X)) NEQ 5 DO
  BEGIN
  IF TYPE NEQ 1 THEN % NOT ALPHA STRING
  BEGIN
  ERR(WORD,WORD,[1:5],8); % INVALID IF NOT SPO AND NUMBER
  END;
  NC := NC + 1; WORD,[1:5] := 0;
  MATCH(WORD,"SET ") SETBIT := 1
  ELSE MATCH(WORD,"RESET ") SETBIT := 0
  ELSE MATCH(WORD,"CONCISE") CONCISEBIT := SETBIT
  ELSE MATCH(WORD,"HELPFUL") HELPFULBIT := SETBIT
  ELSE MATCH(WORD,"QUICKLO") QUICKLOGBIT := SETBIT
  ELSE MATCH(WORD,"QUICKBY") QUICKBYEBIT := SETBIT
  ELSE MATCH(WORD,"NOSTOP ")
  BEGIN NOSTOP:=BOOLEAN(SETBIT); IAM:=0; GO EXIT; END
  ELSE MATCH(WORD,"ALLOWMS") ALLOWMSGBIT:=SETBIT
  ELSE MATCH(WORD,"FREEFIL") UNLOCKTOGGLE:=SETBIT
  %
  %
  ELSE MATCH(WORD,"MONITOR") GO TO CASE2 % SET/RESET MONITOR FILE
  ELSE ERR(WORD,7,8);
  END WHILE STATEMENT;
  IF NC LSS 2 THEN ERR("INCMPLT",7,8);
  IF SCHEDULELINE THEN
  BEGIN
  IAM:=0; WRITEASTERISK:=TRUE;
  END ELSE
  BEGIN
  LLDISKREAD := USERLOC; % USERS/CANDE RECORD ADDRESS
  WANT := VDISKREAD;
  DOING := 1;
  END;
  END CASE 0;
  *.....
  BEGIN % CASE 1 - USERS/CANDE RECORD IN CORE
  WORK[11] := OPTIONBITS; % PUT NEW SETTINGS IN RECORD
  DISKWRITE(USERLOC); % REPLACE USERS/CANDE RECORD
  IAM := 0;
  WRITEASTERISK := TRUE;

```

```

41000000 T 0061
41000100 T 0061
41000200 T 0061
41000300 T 0061
41000400 T 0061
41000500 T 0000
41000700 T 0000
41000800 T 0000
41000900 T 0000
41001000 T 0000
41001100 T 0000
41004700 T 0000
41004900 T 0000
41005000 T 0000
41005100 T 0000
41005200 T 0000
41005300 T 0000
41005400 T 0003
41005500 T 0006
41005600 T 0006
41005610 T 0007
41005700 T 0007
41005710 T 0012
41005730 T 0012
41005800 T 0015
41005900 T 0017
41006000 T 0022
41006100 T 0028
41006200 T 0034
41006300 T 0040
41006400 T 0046
41006410 T 0047
41006500 T 0056
41006600 T 0059
41006700 T 0066
41006800 T 0066
41006900 T 0066
41007100 T 0070
41007200 T 0077
41007300 T 0077
41007310 T 0084
41007320 T 0084
41007330 T 0085
41007340 T 0088
41007350 T 0088
41007400 T 0088
41007500 T 0090
41007600 T 0091
41007650 T 0091
41007700 T 0091
41007800 T 0092
41007900 T 0092
41008000 T 0092
41008100 T 0093
41008200 T 0097
41008300 T 0098

```

START OF SEGMENT ***** 71

```

END CASE 1;
*.....
BEGIN % CASE 2 = MONITOR SET/RESET REQUEST
CASE2: IF NOT BOOLEAN(SETBIT) THEN % RESET MONITOR
      BEGIN
        MONITORBIT := FALSE;
        IAM := 0;
        WRITEASTERISK := TRUE;
        GO TO EXIT;
      END;
      LLDISKREAD := USERLOC; % USERS/CANDE RECORD ADDRESS
      WANT := VDISKREAD;
      DOING := 3;
      END CASE 2;
*.....
BEGIN % CASE 3 = USERS/CANDE RECORD IN CORE
IF WORK[12]=0 THEN ERR("MON,FIL",7,2); % NO FILE NAME LISTED
A[BASE] := WORK[12]; % SAVE MONITOR FILE NAME
FIND(LL&IAM[33:41:7],0,A[BASE],USERCODE,USERCODE); % SEARCH
TEMP := 2; DOING := 4; WAITFOR := IAM;
      END CASE 3;
*.....
BEGIN % CASE 4 = RESULTS OF DIRECTORY SEARCH
IF EVENTS[1] LEQ 0 THEN ERR(A[BASE],6,2); % NOT ON DISK
MONITORBIT := TRUE;
LITOUT(A[BASE],NOCRLF); LITOUT(" MONITR",CRLF);
IAM := 0;
      END CASE 4;
*.....
END ALL CASES;

```

```

EXIT;
END PROCEDURE SETIT;

```

```

41008400 T 0100
41008500 T 0100
41008600 T 0100
41008700 T 0100
41008800 T 0101
41008900 T 0102
41009000 T 0104
41009100 T 0105
41009200 T 0107
41009300 T 0107
41009400 T 0107
41009500 T 0109
41009600 T 0110
41009700 T 0110
41009800 T 0111
41009900 T 0111
41010000 T 0111
41010100 T 0118
41010200 T 0119
41010300 T 0123
41010400 T 0125
41010500 T 0126
41010600 T 0126
41010700 T 0126
41010800 T 0132
41010900 T 0134
41011000 T 0136
41011100 T 0137
41011200 T 0138
41023300 T 0138

```

```

START OF SEGMENT ***** 72
72 IS 6 LONG, NEXT SEG 71
41023400 T 0138
41023500 T 0139
71 IS 145 LONG, NEXT SEG 8

```

```

*****
PROCEDURE SSFILE(LCW,A); BOOLEAN LCW; ARRAY A[0];
*****
BEGIN % PRINT CONTENTS OF SPECIFIED FILE ON TERMINALS
REAL I,LIN,PC,REKS,LINMAX,WORD,MAXSREG;

DEFINE SPOCNTRL = A [BASE+2] #;
LABEL EXIT;
CASE DOING OF
  BEGIN
    BEGIN % CASE 0 = SYNTAX ANALYSIS
      IF LL NEQ 0 THEN GUERR("SSFILE ",13);
      SREG := BASE - A[BASE];
      PC := A[SREG],[18:15]; MAXSREG := SREG + PC;
      A[BASE+3] := 999; % FLAG FOR LINE NUMBER INPUT
      FOR SREG := SREG+1 STEP 1 UNTIL MAXSREG DO
        BEGIN % PICK UP THE PARAMETERS
          WORD := A[SREG];

```

```

42000000 T 0061
42000100 T 0061
42000200 T 0061
42000300 T 0061
42000400 T 0061
START OF SEGMENT ***** 73
%0523=
42000450 C 0000
42000500 T 0000
42000600 T 0000
42000700 T 0000
42000800 T 0000
42000900 T 0000
42001000 T 0008
42001100 T 0009
42001200 T 0012
42001300 T 0014
42001400 T 0018
42001500 T 0018

```

IF WORD.[1:5]=0 THEN % LINE NUMBER GIVEN	42001600 T 0019
BEGIN	42001700 T 0020
IF WORD GTR MAXLINES THEN GOERR("NOT ON ",8);	42001800 T 0021
IF LINE[WORD,UCLOC]=0 THEN GOERR("NOT ON ",8);	42001900 T 0028
A[BASE+3] := WORD;	42002000 T 0036
END	42002100 T 0037
ELSE IF WORD.[3:3] GTR 0 THEN % FILE NAME	42002200 T 0037
BEGIN	42002300 T 0039
IF (REKS:=REKS+1) GTR 2 THEN GOERR(WORD,8);	42002400 T 0040
A[BASE+REKS] := WORD.[6:42];	42002500 T 0046
END	42002600 T 0049
ELSE IF WORD EQL "/" THEN	42002700 P 0049
IF SREG + 2 LEQ MAXSREG THEN FIXUSER(I,A,EXIT,FALSE) ELSE	42002710 C 0050
ELSE	42002780 C 0055
GOERR (WORD,8);	42002785 C 0056
END FOR STATEMENT;	42002800 T 0061
IF REKS NEQ 2 THEN GOERR("FIL NAM",8);	42002900 T 0062
FIND(LL&IAM[33:41:7],0,A[BASE+1],A[BASE+2],"SITE ");	42003000 T 0069
TEMP := 4; DOING := 1; WAITFOR := IAM;	42003100 T 0073
END CASE 0;	42003200 T 0076
*.....	42003300 T 0076
BEGIN % CASE 1 - RESULTS OF DIRECTORY SEARCH	42003400 T 0076
IF EVENTS[1] LSS 0 THEN % NOT ON DISK	42003500 T 0076
BEGIN	42003600 T 0077
WORK[0]:=A[BASE+1]; WORK[1]:=A[BASE+2];	42003700 T 0078
WORK[2]:= " NOT ON"; WORK[3]:= "DISK ";	42003800 T 0082
FOR I:=0,1,2,3 DO BLANK(WORK[I]);	42003900 T 0084
TWXOUT(LL,WORK[0],32,1);	42004000 T 0098
IAM := 0;	42004100 T 0099
GO EXIT;	42004200 T 0100
END IF NO FILE;	42004300 T 0101
LLDISKREAD:=EVENTS[2],[18:15]; % HEADER ADDRESS	42004400 T 0101
WANT := VDISKREAD;	42004500 T 0103
DOING := 2;	42004600 T 0104
END CASE 1;	42004700 T 0104
*.....	42004800 T 0105
BEGIN % CASE 2 - HEADER IN CORE	42004900 T 0105
IF (SPOCNTRL:=WORK[7]+1) GTR (I:=WORK[8]*3) THEN SPOCNTRL:=I;	42005000 T 0105
IF (A[BASE]:=WORK[10])=0 OR SPOCNTRL=0 THEN % NO RECORDS	42005100 T 0111
BEGIN IAM:=0; GO EXIT; END;	42005200 T 0115
A[BASE+1]:=0; % INITIALIZE SEGMENT POINTER	42005300 T 0117
LLDISKREAD := A[BASE]; % ADDRESS OF FIRST SEGMENT	42005400 T 0119
WANT := VDISKREAD;	42005500 T 0120
DOING := 3;	42005600 T 0121
END CASE 2;	42005700 T 0122
*.....	42005800 T 0122
BEGIN % CASE 3 - READ DISKFILE	42005900 T 0122
A[BASE+1] := A[BASE+1] + 1; % INCREMENT SEGMENT COUNTER	42006000 T 0122
REKS:=IF (SPOCNTRL:=SPOCNTRL-3) GEQ 0 THEN 3 ELSE 3+SPOCNTRL;	42006100 T 0125
LIN := 1; LINMAX := MAXLINES;	42006200 T 0132
IF A[BASE+3] LSS 999 THEN LIN:=LINMAX:=A[BASE+3]; % NUMBER GIVEN	42006300 T 0133
FOR LIN:=LIN STEP 1 UNTIL LINMAX DO IF LINE[LIN,UCLOC] NEQ 0 THEN	42006400 T 0137
FOR I:=1 STEP 1 UNTIL REKS DO	42006500 T 0140
IF I=1 THEN TWXOUT(LIN,WORK[0],72,1) ELSE	42006600 T 0142
IF I=2 THEN TWXOUT(LIN,WORK[10],72,1) ELSE	42006700 T 0145
IF I=3 THEN TWXOUT(LIN,WORK[20],72,1);	42006800 T 0148
IF SPOCNTRL LEQ 0 THEN % END OF FILE OR DISK ROW	42006900 T 0156

```

BEGIN
IF TEMP NEQ 4 THEN % RETURN TO "HITHERE"
  BEGIN
  IAM:=VHELLO; DOING:=7;
  END
ELSE
  IAM:=0;
  GO TO EXIT;
END;
LLDISKREAD := A[BASE] + A[BASE+1]; % NEXT SEGMENT ADDRESS
WANT := VDISKREAD;
END CASE 3;
END ALL CASES;

```

```

%0523= 42007000 P 0158
%0523= 42007010 C 0158
%0523= 42007020 C 0159
%0523= 42007030 C 0159
%0523= 42007040 C 0161
%0523= 42007050 C 0161
%0523= 42007060 C 0161
%0523= 42007070 C 0162
%0523= 42007080 C 0163
42007100 T 0163
42007200 T 0165
42007300 T 0166
42007400 T 0167

```

```

START OF SEGMENT ***** 74
74 IS 5 LONG, NEXT SEG 73
42007500 T 0167
42007600 T 0168
73 IS 174 LONG, NEXT SEG 8

```

```

EXIT;
END PROCEDURE SSFILE;

```

```

*****
PROCEDURE EQUATE(LCW,A); BOOLEAN LCW; ARRAY A[0];
*****
BEGIN
REAL I,K,WORD,UNIT,TYPE,NAM,NAM1,NAM2,VERBV,MAXSREG,EUVAL;

BOOLEAN DSK,BKUP,EQLTOG,FILTOG,LIBTOG,CHECKTYPE,QUOT,Q1,Q2;
BOOLEAN FORMTOG,NOLABEL,FASTOG,SLOWTOG,EUTOG;
LABEL ERR,CHECK,EXIT;
DEFINE ENTRY = A[BASE+1]#, THISESP = A[BASE+2]#;
%.....
STREAM PROCEDURE FILEID(ADRS,ARAY,MAX); VALUE ADRS,MAX;
%.....
BEGIN
SI:=ADRS; DI:=ARAY; DI:=DI+1;
MAX(IF SC=ALPHA THEN BEGIN DS:=CHR; TALLY:=TALLY+1; END
ELSE DS:=LIT" ");
ADRS:=TALLY; SI:=LOC ADRS; SI:=SI+7; DI:=ARAY; DS:=CHR;
END STREAM PROCEDURE FILEID;

```

```

43000000 T 0061
43000100 T 0061
43000200 T 0061
43000300 T 0061
43000400 T 0061
START OF SEGMENT ***** 75
43000500 T 0000
43000510 T 0000
43000600 T 0000
43000700 T 0000
43000800 T 0000
43000900 T 0000
43001000 T 0000
43001100 T 0000
43001200 T 0000
43001300 T 0000
43001310 T 0003
43001400 T 0004
43001500 T 0005

```

```

CASE DOING OF
BEGIN
BEGIN % CASE 0 .....
LABEL WAIT;

IF NOT CONTINUEBIT THEN % FIRST CALL ON THE ROUTINE
  BEGIN
  CONTINUEBIT := TRUE;
  FIRSTESP:=THISESP:=ESP1; % SAVE FIRST ESP RECORD ADDRESS
  ENTRY:=0; TEMP:=3; CLEAR(L[0],29); L[14]:=3"14";
  SREG I= BASE = A[BASE]; MAXSREG I= SREG + A[SREG],[18;15];
  END % IF FIRST CALL

```

```

43001600 T 0005
43001700 T 0006
43001800 T 0006
43001810 T 0006
START OF SEGMENT ***** 76
43001900 T 0000
43002000 T 0001
43002100 T 0001
43002200 T 0003
43002300 T 0008
43002400 T 0013
43002500 T 0016

```

Micro Business Forms, Inc. sv 1422

```

ELSE
  BEGIN
    BASE := BASE + 3; % SHIFT UP TO SAVE STACK VALUES
    SCANNER(WORKADRS,A[*],LCW,WORKENDADRS);
    BASE := (I:=BASE) - 3; % RE-ADJUST BASE FOR THIS PROCEDURE
    IF BOOLEAN(A[I],[17:1]) THEN % ERROR DURING SCAN
      BEGIN
        WORD := A[I+2]; MSGPTR := 65; GO TO ERR;
      END;
    IF (TYPE:=A[I],[3:3]) NEQ 1 THEN % NOT A VERB
      BEGIN
        IF TYPE=0 OR TYPE=5 THEN GO WAIT; % COMMENT OR NULL
        IF TYPE=4 THEN % SSMSG
          BEGIN
            BASE:=I; SSMSG; BASE:=I-3; LITOUT(" " ,1); GO WAIT;
          END;
        WORD:=IF TYPE=2 THEN A[I+1] ELSE A[I+2]; % NUMBER OR SPEC CHR
        MSGPTR := 65; GO TO ERR;
        END; % IF NOT A VERB
        IF (VERBV:=A[I+1],[40:8])=VEXECUTE THEN
          MATCH(VERBTABLE[A[I+1],[9:9]],[6:42],"COMPILE") ELSE
            BEGIN
              PARAM := A[I+1],[18:15]+1; TEMP := 1;
              MOVE(PARAM+1,WORDS,A[I],A[BASE]); % SHIFT LEFT FOR "EXECUTE"
              WANT := VERBV; IAM := 0; CONTINUEBIT:=FALSE;
              DOING := 0;
              GO TO EXIT;
            END;
          IF VERBV NEQ VEQUATE THEN % WRONG VERB
            BEGIN
              WORD := VERBTABLE[A[I+1],[9:9]]; MSGPTR := 65;
              GO TO ERR;
            END;
          IF ENTRY=0 THEN % STARTING NEW SEGMENT
            BEGIN
              DISKWAIT(1,L,30,THISESP);
              L[29]:=ESP; % ESTABLISH LINK TO NEXT SEGMENT
              DISKWAIT(0,L,30,THISESP);
              CLEAR(L[0],29); L[14]:=3"14";
              THISESP:= ESP; % ADDRESS OF NEXT SEGMENT TO BE USED
            END % IF ENTRY = 0
          ELSE DISKWAIT(1,L,30,THISESP); % RECOVER PREVIOUS VALUES
          SREG := I+1; MAXSREG := SREG+A[SREG],[18:15];
          END; % IF NOT THE FIRST CALL
          %.....SYNTAX CHECK.....
          UNIT := 12; MSGPTR := 8; FILTOG:=TRUE; EUVAL:=1; % INITIALIZE VALUES
          FOR SREG := SREG + 1 STEP 1 UNTIL MAXSREG DO
            BEGIN % EXAMINE INPUT STRING
              LABEL ALPH;
              %.....NUMERIC.....
              WORD := A[SREG];
              IF QUOT THEN % QUOTED STRING
                BEGIN
                  IF QUOT AND (EQLTOG OR LIBTOG) THEN % ALLOW QUOTED STRINGS
                    BEGIN
                      FILEID(SCANPTR[SREG-SBASE],WORD,7);

```

43002600	T	0016
43002700	T	0016
43002800	T	0017
43002900	T	0018
43003000	T	0020
43003100	T	0022
43003200	T	0023
43003300	T	0024
43003400	T	0029
43003500	T	0029
43003600	T	0031
43003700	T	0031
43003800	T	0034
43003900	T	0034
43004000	T	0035
43004100	T	0041
43004200	T	0041
43004300	T	0045
43004400	T	0048
43004500	T	0048
43004510	T	0051
43004600	T	0056
43004700	T	0058
43004800	T	0061
43004900	T	0064
43005000	T	0068
43005100	T	0068
43005200	T	0071
43005300	T	0071
43005400	T	0071
43005500	T	0072
43005600	T	0075
43005700	T	0078
43005800	T	0078
43005900	T	0079
43006000	T	0080
43006100	T	0082
43006200	T	0084
43006300	T	0087
43006400	T	0089
43006500	T	0091
43006600	T	0091
43006700	T	0094
43006800	T	0098
43006900	T	0098
43007000	T	0098
43007100	T	0102
43007200	T	0106
43007210	T	0106
43007300	T	0000
43007400	T	0000
43007410	T	0001
43007500	T	0001
43007600	T	0001
43007605	T	0003
43007610	T	0003

START OF SEGMENT ***** 77

```

MSGPTR := 66;
I := SCANPTR[SREG=SBASE]; SREG := SREG + 1;
WHILE SREG LSS MAXSREG AND A[SREG],[6:42] NEQ "" DO
SREG := SREG + 1;
K := SCANPTR[SREG=SBASE];
IF A[SREG],[6:42] NEQ "" OR
(K,[33:15]=I,[33:15])x8+(K,[30:3]=I,[30:3]) GTR 7 THEN
GO TO ERR;
GO TO ALPH;
END ELSE GO TO ERR;
END % IF QUOT
%.....
ELSE IF WORD,[1:5]=0 THEN % UNSIGNED NUMBER
BEGIN
IF EUTOG THEN
BEGIN
IF (EUVAL:=WORD) GTR 19 THEN GO TO ERR;
END
ELSE GO TO ERR;
END % IF UNSIGNED NUMBER
%.....ALPHA.....
ELSE IF WORD,[3:3] GTR 0 THEN % ALPHA STRING
ALPH: BEGIN
WORD := (NAM:=WORD),[6:42];
IF FILTOG THEN % LOOKING FOR INTERNAL NAME
BEGIN
FILEID(SCANPTR[SREG=SBASE],L[ENTRY+4],63); % INTERNAL NAME
FILTOG := FALSE;
END % IF FILTOG
ELSE IF EQLTOG THEN % "=" PRECEDED THIS WORD
BEGIN
L[ENTRY+1]:=WORD;
NAM2:=NAM; L[ENTRY]:=0; % ASSUME ENTRY IS FID
EQLTOG := NOT(CHECKTYPE:=TRUE);
IF QUOT THEN
BEGIN Q2:=TRUE; QUOT:=FALSE; END;
END % IF EQLTOG
ELSE IF LIBTOG THEN % "/" PRECEDED THIS WORD
BEGIN
L[ENTRY]:=L[ENTRY+1]; NAM1:=NAM2; Q1:=Q2;
L[ENTRY+1]:=WORD; NAM2:=NAM;
LIBTOG := FALSE;
IF QUOT THEN
BEGIN Q2:=TRUE; QUOT:=FALSE; END;
END % IF LIBTOG
ELSE IF CHECKTYPE THEN % CHECK HARDWARE TYPE
BEGIN
MATCH(WORD,"BACKUP ") BKUP:=TRUE
ELSE MATCH(WORD,"BACK ")
BEGIN
IF SREG LSS MAXSREG THEN
IF A[SREG+1],[3:3]=2 THEN
MATCH(A[SREG+1],[6:42],"UP ")
BEGIN
BKUP:=TRUE; SREG:=SREG+1;
END;
END % IF "BACK UP"

```

```

43007615 T 0005
43007620 T 0006
43007625 T 0009
43007630 T 0013
43007635 T 0015
43007640 T 0017
43007645 T 0019
43007650 T 0024
43007655 T 0027
43007660 T 0027
43007662 T 0030
43007664 T 0030
43007666 T 0030
43007668 T 0032
43007670 T 0032
43007672 T 0032
43007674 T 0033
43007676 T 0037
43007678 T 0037
43007700 T 0040
43007800 T 0040
43007900 T 0040
43008000 T 0041
43008100 T 0042
43008200 T 0045
43008300 T 0045
43008400 T 0046
43008500 T 0049
43008600 T 0050
43008700 T 0050
43008800 T 0051
43008900 T 0051
43009000 T 0054
43009100 T 0057
43009110 T 0058
43009120 T 0058
43009200 T 0060
43009300 T 0060
43009400 T 0061
43009500 T 0062
43009600 T 0067
43009700 T 0070
43009710 T 0071
43009720 T 0071
43009800 T 0073
43009900 T 0073
43010000 T 0074
43010100 T 0074
43010200 T 0076
43010300 T 0077
43010400 T 0081
43010500 T 0081
43010600 T 0084
43010700 T 0084
43010800 T 0088
43010900 T 0090
43011000 T 0090

```



```

ELSE MATCH(WORD,"REMOTE ") UNIT:=14
ELSE MATCH(WORD,"DISK ")
  BEGIN
  UNIT:=IF BKUP THEN IF UNIT=0 THEN 22 ELSE 15 ELSE 12;
  BKUP := FALSE;
  END
  ELSE MATCH(WORD,"SERIAL ") UNIT := 12
ELSE MATCH(WORD,[6:30],"PRINT") UNIT := 17
ELSE MATCH(WORD,"SPECIAL") UNIT := 3
  ELSE MATCH(WORD,"PUNCH ") UNIT := 0
  ELSE MATCH(WORD,"CARD ") UNIT := 0
  ELSE MATCH(WORD,"RANDOM ") UNIT := 10
  ELSE MATCH(WORD,"UPDATE ") UNIT := 13
  ELSE MATCH(WORD,"PAPER ") UNIT := 7
  ELSE MATCH(WORD,"SPO ") UNIT := 11
  ELSE MATCH(WORD,"FAST ")
    IF SLOWTOG THEN GO ERR ELSE FASTOG:=TRUE
  ELSE MATCH(WORD,"SLOW ")
    IF FASTOG THEN GO ERR ELSE SLOWTOG:=TRUE
  ELSE MATCH(WORD,"EU ") EUTOG:=TRUE
  ELSE MATCH(WORD,"TAPE ")
    BEGIN
    IF UNIT NEQ 7 THEN % PAPER TAPE
    UNIT:=IF BKUP THEN IF UNIT=0 THEN 20 ELSE 6 ELSE 2;
    BKUP := FALSE;
    END % IF TAPE
  ELSE MATCH(WORD,"FORM ") FORMTOG:=TRUE
  ELSE MATCH(WORD,"NOLABEL") NOLABEL:=TRUE
  ELSE MATCH(WORD,"COPY ")
    BEGIN
    IF UNIT=0 THEN UNIT:=22 ELSE % PUNCH TO PUD
    IF UNIT=2 THEN UNIT:=6 ELSE % TAPE TO PBT
    IF UNIT=17 OR UNIT=12 THEN UNIT:=15; % PRINT TO PBD
    IF SREG LSS MAXSREG THEN
    IF (WORD=A[SREG+1]),[1:5]=0 THEN
    IF WORD LEQ 256 THEN
    BEGIN
    L[ENTRY+3],[15:8] := WORD-REAL(WORD NEQ 0);
    SREG:=SREG+1;
    END
    ELSE GO ERR;
    END
    ELSE GO ERR;
    END % IF CHECKTYPE
  ELSE GO ERR;
  END % IF ALPHA STRING
ELSE
%.....SPECIAL CHR.....
  BEGIN
  IF WORD = "=" THEN EQTOG := NOT EUTOG
  ELSE IF WORD="/" THEN LIBTOG := TRUE
  ELSE IF WORD="'" THEN QUOT:=TRUE
  ELSE GO ERR;
  END; %IF SPECIAL CHARACTER
END WHILE STATEMENT;
%.....END SCAN.....

```

```

43011100 T 0090
43011200 T 0096
43011300 T 0096
43011400 T 0101
43011500 T 0105
43011600 T 0106
43011700 T 0106
43011800 P 0110
43011810 C 0115
43011900 T 0120
43012000 T 0125
43012100 T 0130
43012200 T 0135
43012300 T 0140
43012400 T 0145
43012410 T 0150
43012420 T 0150
43012430 T 0160
43012440 T 0160
43012450 T 0168
43012500 T 0171
43012600 T 0172
43012700 T 0176
43012800 T 0176
43012900 T 0181
43013000 T 0182
43013002 T 0182
43013004 T 0186
43013010 T 0191
43013015 T 0191
43013020 T 0196
43013025 T 0198
43013030 P 0202
43013035 T 0205
43013040 T 0206
43013045 T 0209
43013050 T 0210
43013055 T 0211
43013060 T 0215
43013065 T 0217
43013070 T 0217
43013075 T 0219
43013100 T 0219
43013200 T 0222
43013300 T 0222
43013400 T 0225
43013500 T 0225
43013600 T 0225
43013700 T 0225
43013800 T 0225
43013900 T 0227
43013910 T 0230
43014000 T 0232
43014100 T 0235
43014200 T 0235
43014300 T 0107

```

```

IF NOT CHECKTYPE THEN % INCOMPLETE REQUEST
  BEGIN
  WORD:="INCMPLT"; MSGPTR := 66; GO ERR;
  END;
IF NOLABEL THEN
  UNIT:=IF UNIT=2 THEN 9 ELSE IF UNIT=7 THEN 8 ELSE UNIT;
  UNIT:=UNIT & REAL(FORMTOG)[42;47;1];
  L[ENTRY+3],[42;6]:=UNIT; L[ENTRY+12]:=0;
  IF DSK:=(UNIT=10 OR UNIT=12 OR UNIT=13) THEN % DISK
  BEGIN
  IF L[ENTRY]=0 THEN % MUST HAVE BOTH NAMES FOR DISK
  BEGIN
  L[ENTRY]:=L[ENTRY+1]; NAM1:=NAM2; Q1:=Q2;
  L[ENTRY+1]:=NAM2:=USERCODE;
  END;
  IF NOT Q1 THEN % NOT QUOTED STRING
  BEGIN
  IF (WORD:=L[ENTRY]),[42;6] NEQ " " THEN
  IF WORD,[6;6] NEQ 0 THEN
  BEGIN
  MSGPTR := 42; GO TO ERR;
  END;
  IF FINDAWORD(NAM1,RESWRDTABLE[1],NUMOFRESWRDS) NEQ 0 THEN
  BEGIN MSGPTR := 29; GO TO ERR; END;
  END;
  IF EUVAL GEQ 0 THEN L[ENTRY+12],[18;5]:=EUVAL+1 ELSE
  IF (FASTOG OR SLOWTOG) THEN
  L[ENTRY+12],[16;2] := 1 + REAL(SLOWTOG);
  END; % IF DISK
  DISKWAIT(0,L,30,THISESP); % WRITE OUT THIS SEGMENT
  IF ENTRY=0 THEN % STARTING NEW SEGMENT
  BEGIN
  ENTERESP(LL,THISESP,A[+]); % ENTER SEGMENT ADDRESS IN TABLE
  IF BOOLEAN(ESPFLAG) THEN % NO ROOM IN ESPTABLE FOR THIS ENTRY
  BEGIN
  WORD:="NO ROOM"; MSGPTR := 64; GO ERR;
  END;
  THISESP := ESP1; ESP1 := GETESPDISK; EQUATED := TRUE;
  END; % IF STARTING NEQ SEGMENT
  ENTRY := IF ENTRY=0 THEN 14 ELSE 0; % 2 EQUATIONS PER SEGMENT
  IF DSK THEN MATCH(NAM2,[6;42],USERCODE) ELSE % OUTSIDE FILE
  BEGIN
  FIND(LL&IAM[33;41;7],0,NAM1,NAM2,USERCODE);
  A[BASE]:=NAM1;
  DOING := 1; WAITFOR := IAM; GO TO EXIT;
  END;
WAIT:
GO TO CHECK;
END; % CASE 0

%.....
BEGIN % CASE 1 = RESULTS OF DIRECTORY SEARCH
IF EVENTS[1] LSS 0 THEN % FILE NOT IN DIRECTORY
  BEGIN
  WORD:=A[BASE]; MSGPTR := 67; GO TO ERR;
  END;
CHECK:

```

```

43014400 T 0107
43014500 T 0108
43014600 T 0108
43014700 T 0112
43014710 T 0112
43014720 T 0113
43014730 T 0118
43014800 T 0120
43014900 T 0126
43015000 T 0129
43015100 T 0130
43015200 T 0131
43015300 T 0132
43015400 T 0137
43015500 T 0140
43015510 T 0140
43015520 T 0141
43015600 T 0141
43015700 T 0144
43015800 T 0146
43015900 T 0146
43016000 T 0150
43016200 T 0150
43016300 T 0152
43016310 T 0156
43016320 T 0156
43016330 T 0161
43016340 T 0163
43016400 T 0168
43016500 T 0168
43016600 T 0171
43016700 T 0172
43016800 T 0173
43016900 T 0175
43017000 T 0176
43017100 T 0177
43017200 T 0181
43017300 T 0181
43017400 T 0187
43017500 T 0187
43017600 T 0191
43017700 T 0195
43017800 T 0198
43017900 T 0201
43018000 T 0202
43018100 T 0206
43018200 T 0206
43018500 T 0207
43018600 T 0209
76 IS 210 LONG, NEXT SEG 75
43018700 T 0008
43018800 T 0008
43018900 T 0008
43019000 T 0009
43019100 T 0010
43019200 T 0012
43019250 T 0012

```

```

DOING:=0; WAITFOR:=VINPUT;
IF SCHEDULELINE THEN REQUESTIP(LL) ELSE
  BEGIN
  I:=0; K:=REAL(THINGSLINK);
  WHILE I LSS REAL(THINGSTODO) DO
  BEGIN
  I:=I+1;
  IF TOBEDONE[K,16],[18:7]=VINPUT THEN
  BEGIN
  TOBEDONE[K,16],[18:7]=VEQUATE;
  IAM := WANT := WAITFOR := 0;
  GO TO EXIT;
  END;
  K:=TOBEDONE[K,0],[3:6];
  END;
  END;
GO TO EXIT;
END; % CASE 1
END ALL CASES;

```

```

ERR:
ERRSW := WORD;
IF EQUATED THEN ENTERESP(-LL,"0,A[*]); % GIVE BACK ESP DISK
IAM := VERROR;
EQUATED := FALSE; CONTINUEBIT := FALSE;
EXIT:
END PROCEDURE EQUATE;

```

```

*****
PROCEDURE STATUSCHECK(LCW,A); BOOLEAN LCW; ARRAY A[0];
*****
BEGIN
REAL CPU,IO,SEQ;

LABEL NOTRUNNING,EXIT;
*.....
STREAM PROCEDURE MESSAGE(WORK,CPU,IO,CMP,R27); VALUE CPU,IO,CMP,R27;
*.....
  BEGIN LOCAL SV1,SV2;
  DI:=WORK; DS:=8LIT" "; SI:=WORK; DS:=9WDS; % BLANK ARRAY
  DI:=WORK; DS:=4LIT"CPU="; SV1:=DI; SI:=LOC CPU;
  2(DS:=2CHR; DS:=6LIT" HRS, ";
  DS:=2CHR; DS:=6LIT" MIN, ";
  DS:=2CHR; DS:=LIT","; DS:=CHR; DS:=6LIT" SEC,X");
  DI:=SV1; DS:=18FILL; SV2:=DI; SI:=SV2; DI:=SV1;
  26(IF SC="" THEN JUMP OUT ELSE DS:=CHR);
  SI:=LOC IO; DS:=4LIT" IO="; SV1:=DI; DI:=DI-4;
  CMP(DS:=10LIT" AT SEQ # "; SV1:=DI; SI:=LOC R27;
  DS:=8DEC; DS:=LIT"X"; DI:=SV1; DS:=7FILL; SV2:=DI;
  SI:=SV2; DI:=SV1; 8(IF SC="" THEN JUMP OUT ELSE DS:=CHR));
  DS:=25LIT" ";
END STREAM PROCEDURE MESSAGE;

```

```

43019300 T 0013
43019310 T 0014
43019320 T 0019
43019330 T 0019
43019340 T 0021
43019350 T 0023
43019360 T 0023
43019370 T 0025
43019380 T 0027
43019390 T 0027
43019400 T 0031
43019410 T 0032
43019420 T 0033
43019430 T 0033
43019440 T 0035
43019450 T 0036
43019460 T 0036
43019470 T 0036
43019500 T 0037
START OF SEGMENT ***** 78
78 IS 3 LONG, NEXT SEG 75
43019600 T 0037
43019700 T 0038
43019800 T 0038
43019900 T 0042
43020000 T 0043
43020100 T 0047
43020200 T 0048
75 IS 59 LONG, NEXT SEG 8
44000000 T 0061
44000100 T 0061
44000200 T 0061
44000300 T 0061
44000400 T 0061
START OF SEGMENT ***** 79
44000500 T 0000
44000600 T 0000
44000700 T 0000
44000800 T 0000
44000900 T 0000
44001000 T 0000
44001100 T 0002
44001200 T 0003
44001300 T 0005
44001400 T 0006
44001500 T 0008
44001600 T 0009
44001700 T 0012
44001800 T 0013
44001900 T 0016
44002000 T 0018
44002100 T 0021
44002200 T 0024

```

```

CASE DOING OF
BEGIN
  BEGIN % CASE 0 - INITIAL CHECK
  IF RUNNING OR COMPILING THEN
  IF RUN1 NEQ 0 AND RUN2 NEQ 0 THEN
  BEGIN
  A[BASE+3]:=REAL(COMPILING);
  IF RUNNING THEN
  BEGIN
  IF (A[BASE]:=RUN1).[6:6]=0 THEN % REQUIRES SHIFTING
  BEGIN
  A[BASE]:=IF RUN1.[12:6]="1" THEN FILENAME ELSE
  " "&RUN1[6:12:36];
  END;
  A[BASE+1]:=MATCH(RUN2,"CANDE ") USERCODE ELSE RUN2;
  A[BASE+2]:=USERCODE;
  END % IF RUNNING
  ELSE
  BEGIN
  A[BASE]:=RUN2; % COMPILER NAME
  A[BASE+1]:=IF RUN1.[6:12]="1S" THEN FILENAME ELSE RUN1;
  A[BASE+2]:=USERCODE; % NEGATIVE IF COMPILING
  END; % IF COMPILING
  STATUS(LL&IAM[33:41:7],A[BASE],A[BASE+1],A[BASE+2]);
  WAITFOR := IAM;
  TEMP := 4;
  DOING := 1;
  GO TO EXIT;
  END;
NOTRUNNING:
  WORK[0]:="NO JOBS"; BLANK(WORK[0]);
  WORK[1]:="RUNNING"; BLANK(WORK[1]);
  TWXOUT(LL,WORK[0],16,1);
  INQUIRY:=FALSE;
  IAM := 0;
  END; % CASE 0
  *.....
  BEGIN % CASE 1 - RESULTS OF INQUIRY
  IF EVENTS[1] LSS 0 THEN GO TO NOTRUNNING;
  CPU:=TIMECONV(EVENTS[1],FALSE);
  IO:=IF EVENTS[2]=REAL(NOT FALSE) THEN 0 ELSE
  TIMECONV(EVENTS[2],FALSE);
  IF BOOLEAN(A[BASE+3]) THEN % COMPILING
  IF (SEQ:=EVENTS [3]) = - USERCODE THEN A [BASE+3]:=0;
  MESSAGE(WORK,CPU,IO,A[BASE+3],SEQ);
  TWXOUT(LL,WORK[0],80,1);
  INQUIRY:=FALSE;
  IAM := 0;
  END CASE 2;
END ALL CASES;

```

```

44002300 T 0025
44002400 T 0025
44002500 T 0025
44002600 T 0025
44002700 T 0027
44002800 T 0030
44002900 T 0030
44003000 T 0033
44003100 T 0033
44003200 T 0034
44003300 T 0036
44003400 T 0037
44003500 T 0040
44003600 T 0042
44003700 T 0042
44003800 T 0048
44003900 T 0050
44004000 T 0050
44004100 T 0050
44004200 T 0052
44004300 T 0053
44004400 T 0058
44004500 T 0060
44004600 T 0060
44004700 T 0065
44004800 T 0066
44004900 T 0067
44005000 T 0068
44005100 T 0068
44005200 T 0068
44005300 T 0069
44005400 T 0071
44005500 T 0073
44005600 T 0075
44005700 T 0077
44005800 T 0078
44005900 T 0078
44006000 T 0078
44006100 T 0078
44006200 T 0080
44006300 T 0082
44006400 T 0084
44006500 T 0086
44006600 P 0087
44006700 T 0092
44006800 T 0095
44006900 T 0097
44007000 T 0099
44007100 T 0100
44007200 T 0100

```

```

START OF SEGMENT ***** 80
80 IS 3 LONG, NEXT SEG 79
44007300 T 0101

```

EXIT:

END PROCEDURE STATUSCHECK;

44007400 T 0101
79 IS 106 LONG, NEXT SEG 8

PROCEDURE SSMSG;
BEGIN
DEFINE A[A1]=LINE[LL,A1];

STREAM PROCEDURE FRM(F,T,UC,LL); VALUE UC,LL;
BEGIN
SI := LOC UC; UPS;
DI := T;
DS := 8 LIT "** FROM ";
DS := 7 CHR;
DS := 2 LIT "(";
SI := LOC LL;
DS := 2 DEC;
DS := 5 LIT ")"
SI := F;
DS := 15 WDS;
END FRM;

50000000 T 0061
50000100 T 0061
50000200 T 0061
50000300 T 0061
START OF SEGMENT ***** 81
50000400 T 0000
50000500 T 0000
50000600 T 0000
50000700 T 0000
50000800 T 0000
50000900 T 0002
50001000 T 0002
50001100 T 0002
50001200 T 0003
50001300 T 0003
50001400 T 0004
50001500 T 0004
50001600 T 0004

PROCEDURE SPOUT(LCW,A,T); VALUE T; BOOLEAN LCW; ARRAY A[0];
REAL T;

BEGIN
LABEL XIT;

STREAM PROCEDURE BZY(W,UC,LL); VALUE UC,LL;
BEGIN
DI := W;
SI := LOC UC;
UPS;
DS := 7 CHR;
DS := LIT "(";
SI := LOC LL;
DS := 2 DEC;
DS := 6 LIT ")" BUSY";
END BZY;

50001700 T 0005
50001800 T 0005
50001900 T 0005
50002000 T 0005
START OF SEGMENT ***** 82
50002100 T 0000
50002200 T 0000
50002300 T 0000
50002400 T 0000
50002500 T 0000
50002600 T 0000
50002700 T 0001
50002800 T 0001
50002900 T 0001
50003000 T 0002
50003100 T 0003

IF IDLELINE OR ALLOWMSG THEN TWXOUT(LL,WORK[0],144,2)

ELSE

BEGIN

IF RUNNING THEN MATCH("CANDE ",RUN2) ELSE

BEGIN

BZY(WORK[20],USERCODE,LL);

TWXOUT(T,WORK[20],16,1);

GO XIT

END;

50003200 T 0003
50003300 T 0008
50003400 T 0008
50003500 T 0009
50003600 T 0012
50003700 T 0014
50003800 T 0015
50003900 T 0017
50004000 T 0018

```

EVENTS[0] := O&LL[25:40:8]&VSSMSG[18:41:7];
MOVE (4,WORDS,WORK[10],EVENTS[1]);
REMEMBERTHIS(LCW);
END;
XIT;
END SPOUT;

```

```

50004100 T 0018
50004200 T 0021
50004300 T 0023
50004400 T 0024
50004500 T 0024
50004600 T 0025
82 IS 26 LONG, NEXT SEG 81

```

```

LABEL ERR, EXIT;
BOOLEAN GOTONE;
REAL T,WORD;
*****: START HERE
FRM(L,WORK,USERCODE,LL);
T := LL;
IF (WORD:=A[BASE+2]).[3:3] = 0 THEN
BEGIN
IF LL := WORD GTR MAXLINES THEN GO ERR;
IF LOGGEDON THEN SPOUT(LCW,A[*],T) ELSE GO ERR;
GO EXIT
END;
IF T = 0 THEN
MATCH(WORD,[6:42], "ALL ")
BEGIN
FOR LL := 1 STEP 1 UNTIL MAXLINES DO
IF USERCODE NEQ 0 THEN
BEGIN
SPOUT(LCW,A[*],T);
GOTONE := TRUE;
END;
IF NOT GOTONE THEN GO ERR; GO EXIT
END;
FOR LL := 0 STEP 1 UNTIL MAXLINES DO
IF LOGGEDON THEN
MATCH(WORD,[6:42], LINE[LL, UCLOC])
BEGIN SPOUT(LCW,A[*],T);
GOTONE := TRUE;
END;
IF GOTONE THEN GO EXIT;
ERR:
LL := T;
LITOUT("NOT ON ",1);
WRITEASTERISK := FALSE;
EXIT;
LL := T;
END SSMSG;

```

```

50004700 T 0005
50004800 T 0005
50004900 T 0005
50005000 T 0005
50005100 T 0005
50005200 T 0008
50005300 T 0008
50005400 T 0012
50005500 T 0012
50005600 T 0014
50005700 T 0020
50005800 T 0020
50005900 T 0020
50006000 T 0021
50006100 T 0021
50006200 T 0025
50006300 T 0027
50006400 T 0028
50006500 T 0029
50006600 T 0033
50006700 T 0034
50006800 T 0036
50006900 T 0037
50007000 T 0037
50007100 T 0039
50007200 T 0040
50007300 T 0040
50007400 T 0049
50007500 T 0050
50007600 T 0052
50007700 T 0053
50007800 T 0054
50007900 T 0054
50008000 T 0055
50008100 T 0058
50008200 T 0059
50008300 T 0059
81 IS 63 LONG, NEXT SEG 8

```

```

*****
PROCEDURE PDIT (LCW, A); BOOLEAN LCW; ARRAY A[0];
*****
BEGIN

```

```

51000000 T 0061
51000100 T 0061
51000200 T 0061
51000300 T 0061

```

```

LABEL GETIT, EXIT;
INTEGER I;
REAL STREAM PROCEDURE FORM(T,F);
BEGIN
  SI I= F; DI I= T; UPS;
  IF SC LSS "0" THEN
    BEGIN
      DS=LIT" "; DS:=7CHR; TALLY:=1;
    END ELSE
      IF SC EQL "0" THEN % OBJECTFILE.
    BEGIN UPS;
      IF SC LSS "0" THEN % NOT WRKFILE
        BEGIN DS:=2LIT" *"; DS:=6CHR; TALLY:=1; END;
    END IF OBJECT FILE;
      FORM:=TALLY;
    END OF FORM;

```

```

51000400 T 0061
START OF SEGMENT ***** 83
51000500 T 0000
51000600 T 0000
51000700 T 0000
51000800 T 0000
51000900 T 0000
51000950 T 0001
51001000 T 0001
51001010 T 0002
51001020 T 0003
51001030 T 0003
51001040 T 0004
51001050 T 0004
51001060 T 0006
51001090 T 0006
51001100 T 0006

```

```

CASE DOING OF
BEGIN
  BEGIN % CASE 0.
    A[BASE]:=MODULUS; DOING:=1;
    A[BASE+1] := 0; % USED FOR ADDR SECOND HALF.
    A[BASE+2] := 0; % USED FOR ADDR LINK.
    SREG := BASE+3; % BUFFER POINTER.
    TEMP := 11; % ALLOW 8 WDS BUFFER.
  END CASE 0;
  BEGIN % CASE 1. (FIRST HALF OF BLOCK)
    A[BASE] :=(I:=A[BASE])=1;
    IF I=0 THEN GO EXIT;
    LLDISKREAD:=A[BASE+1]; I:=SCRAMBLE(I,USERCODE);
    WANT :=VDISKREAD;
    DOING:=2;
  END CASE 1;
  BEGIN % CASE 2. (FIRST HALF IN).
    A[BASE+2]:=WORK[2],FF; % ADDR LINK
    A[BASE+1]:=A[BASE+1]+1; % ADDR SECOND HALF.
    GO GETIT;
  END CASE 2;
  BEGIN % CASE 3. (ANY HALF IN).
GETIT:
  FOR I:=0 STEP 3 UNTIL 27 DO
    IF WORK[I+1]=USERCODE THEN
      BEGIN
        SREG:=FORM(A[SREG],WORK[I])+SREG;
        IF SREG=BASE GTR 10 THEN
          BEGIN % PRINT THE LINE.
            SREG:=BASE+3;
            MOVE(8,WORDS,A[SREG],L[0]);
            TWXOUT(LL,L[0],64,1);
          END PRINTING;
        END FOR;
        IF LLDISKREAD:=A[BASE+1] GTR 1000 THEN
          BEGIN % GET SECOND HALF BLOCK.

```

```

51001200 T 0007
51001300 T 0008
51001400 T 0008
51001500 T 0008
51001600 T 0010
51001700 T 0012
51001800 T 0014
51001900 T 0015
51002000 T 0016
51002100 T 0016
51002200 T 0016
51002300 T 0019
51002400 T 0020
51002500 T 0030
51002600 T 0031
51002700 T 0032
51002800 T 0032
51002900 T 0032
51003000 T 0035
51003100 T 0038
51003200 T 0038
51003300 T 0039
51003400 T 0039
51003500 T 0040
51003600 T 0041
51003700 T 0042
51003800 T 0043
51003900 T 0046
51004000 T 0047
51004100 T 0048
51004200 T 0049
51004300 T 0051
51004400 T 0053
51004500 T 0053
51004600 T 0055
51004700 T 0058

```

```

        A[BASE+1]:=0;
        DOING:=3;
        WANT:=VDISKREAD;
    END ELSE
    IF LLDISKREAD:=A[BASE+2] GTR 1000 THEN
    BEGIN
        A[BASE+2]:=0;
        A[BASE+1]:=LLDISKREAD;
        DOING:=2;
        WANT:=VDISKREAD;
    END ELSE DOING:=1;
    END CASE 3;
    BEGIN % CASE 4, (WRAPUP),
EXIT:  IF SREG=BASE GTR 3 THEN
    BEGIN
        I:=SREG-BASE-3; SREG:=BASE+3;
        MOVE(I,WORDS,A[SREG],L[0]);
        TWXOUT(LL,L[0],[I*8],CRLF);
    END IF LEFTOVERS;
        IAM:=0;
    END CASE 4;
    END ALL CASES;

    END OF PDIT;

```

```

51004800 T 0058
51004900 T 0060
51005000 T 0061
51005100 T 0061
51005200 T 0061
51005300 T 0064
51005400 T 0065
51005500 T 0067
51005600 T 0069
51005700 T 0069
51005800 T 0070
51005900 T 0071
51006000 T 0072
51006100 T 0072
51006200 T 0074
51006300 T 0074
51006400 T 0077
51006500 T 0080
51006600 T 0082
51006700 T 0082
51006800 T 0083
51006900 T 0083
START OF SEGMENT ***** 84
84 IS 6 LONG, NEXT SEG 83
51007000 T 0084
83 IS 87 LONG, NEXT SEG 8

```

```

*****
PROCEDURE COPYIT(LCW,A); BOOLEAN LCW; ARRAY A[*];
*****
BEGIN
COMMENT HANDLES CANDE VERBS;
    APPEND, COPY, DELETE, MERGE, PUNCH, RESEQ, RMERGE;
REAL INPCOUNT, OUTCOUNT, PREVNUMBER, WORD, UNIT, UNITNUMBER,

FILNAM1, FILNAM2, FILNAM3, SAVEWORD, NPARAMS, VERB;
LABEL EXIT,ERROR;
BOOLEAN TOGWORD,WINDUP;
DEFINE
REVERSETOG      = TOGWORD.[01:1]#, RESEQVERB      = TOGWORD.[02:1]#,
RESEQPARAM      = TOGWORD.[03:1]#, HAVEMINUS      = TOGWORD.[04:1]#,
HARDCOPY        = TOGWORD.[05:1]#, NOTWORKFILE    = TOGWORD.[06:1]#,
HAVEINPUTFILE   = TOGWORD.[07:1]#, LIBTOG         = TOGWORD.[08:1]#,
SAVETO          = TOGWORD.[09:1]#, HAVERESEQRANGE = TOGWORD.[10:1]#,
HAVEINCREMENT   = TOGWORD.[11:1]#, HAVENEWBASE    = TOGWORD.[12:1]#,
HAVEOUTPUTFILE  = TOGWORD.[13:1]#, LASTPARAM      = TOGWORD.[14:1]#;
CASE DOING OF
    BEGIN
        BEGIN % CASE 0
        LABEL RESTART;

        NPARAMS := (A[SREG]-BASE-A[BASE]),FF; % PARAMETER COUNT
        VERB := IF A[BASE]=0 THEN "COPY" ELSE
        VERBTABLE[A[SREG],[9:9]].[6:42]; % GET VERB NAME FROM TABLE

```

```

52000000 T 0061
52000100 T 0061
52000200 T 0061
52000300 T 0061
52000400 T 0061
52000500 T 0061
52000600 T 0061
START OF SEGMENT ***** 85
52000700 T 0000
52000800 T 0000
52000900 T 0000
52001000 T 0000
52001100 T 0000
52001200 T 0000
52001300 T 0000
52001400 T 0000
52001500 T 0000
52001600 T 0000
52001700 T 0000
52001800 T 0000
52001900 T 0000
52002000 T 0000
52002010 T 0000
START OF SEGMENT ***** 86
52002100 T 0000
52002200 T 0004
52002300 T 0006

```


"delete"

"punch"

"merge"

"reseq"

```

MATCH(VERB,"DELETE ") % SEE IF REFERS TO WORK FILE FIRST
IF NPARAMS=0 OR NPARAMS=1 AND A[SREG+1],[6142]="ALL " THEN
  BEGIN % SCRATCH THE WORKFILE
  IF FILENAME=0 THEN GOERR("NO FILE",1);
  DOING := OBJECTFILE := 0; GOODOBJ := FALSE;
  SOURCEFILE := MAKEFN("1S",LL); % "1S" FILE NAME
  MAKEFILE(LL&13[3314117],1,SOURCEFILE,USERCODE,FILETYPE,0);
  IAM := VSETUPFILES; WRITEASTERISK := TRUE; GO TO EXIT;
END; % IF DELETING THE WORKFILE
MATCH(VERB,"PUNCH ") % PAPER TAPE
BEGIN
  IF SCHEDULELINE THEN GOERR("PUNCH ",40);
  IF DEVICE=CONRAC OR DEVICE=BIDS THEN GOERR("PUNCH ",56);
  % PUT YOUR FUNNY TERMINALS HERE

  HARDCOPY := TRUE;
END; % IF PUNCH VERB
IF FILETYPE=8 THEN % TYPE DATA FILE
  BEGIN
  MATCH(VERB,"MERGE ") GOERR("MERGE ",57);
  MATCH(VERB,"RMERGE ") GOERR("RMERGE ",57);
END; % IF DATA FILE
IF TPCOUNT GTR 0 THEN % WORK FILE NEEDS UPDATING
  BEGIN
  LITOUT("WAIT, ",NOCRLF); TEMP:=1; WANT:=VUPDATE;
  GO TO EXIT;
END;
MATCH(VERB,"RMERGE ") % REVERSE MERGE
BEGIN
  REVERSETOG := TRUE; VERB := "MERGE ";
END;
CLEAR(WORK[0],29); % ZERO OUT THE "WORK" ARRAY
MATCH(VERB,"RESEQ ") % SET DEFAULT OPTIONS
BEGIN
  RESEQVERB := TRUE;
  WORK[5] := -1; % DEFAULT LOWER BOUND FOR RESEQ
  WORK[6] := 100000000; % DEFAULT UPPER BOUND FOR RESEQ
  WORK[7] := WORK[8] := 100; % DEFAULT BASE AND INCR. FOR RESEQ,
END;
WORK[9] := MAKEFN("1S",LL); % DEFAULT OUTPUT FILE NAME
FILNAM1 := WORK[10] := SOURCEFILE; % DEFAULT SOURCE FILE NAME
FILNAM2 := WORK[2] := USERCODE;
INPCOUNT := 0; OUTCOUNT := 20; PREVIOUSNUMBER := -1;
RESTART:
%.....
WHILE (INPCOUNT LSS NPARAMS) OR WINDUP DO % PICK UP ALL PARAMS
%.....
  BEGIN
  LABEL UNSIGNED,NEGATIVE;

  LABEL SUFFIX;
  LABEL NUMUSER;
  IF WINDUP THEN % FINISHING UP LAST PARAMETER
    BEGIN
    WINDUP:=FALSE; GO TO UNSIGNED;
    END;

```

52002400	T	0010
52002500	T	0011
52002600	T	0017
52002650	C	0018
52002700	T	0026
52002800	T	0031
52002900	T	0033
52003000	T	0039
52003100	T	0044
52003200	T	0044
52003300	T	0046
52003400	T	0046
52003500	T	0054
52003600	T	0065
52003700	T	0065
52003800	T	0065
52003900	T	0065
52004000	T	0066
52004100	T	0066
52004200	T	0068
52004300	T	0068
52004400	T	0077
52004500	T	0086
52004600	T	0086
52004700	T	0088
52004800	T	0088
52004900	T	0091
52005000	T	0093
52005100	T	0093
52005200	T	0095
52005300	T	0095
52005400	T	0098
52005500	T	0098
52005600	T	0099
52005700	T	0100
52005800	T	0101
52005900	T	0103
52006000	T	0104
52006100	T	0105
52006200	T	0108
52006300	T	0120
52006400	T	0122
52006500	T	0124
52006600	T	0127
52006610	T	0129
52006700	T	0130
52006800	T	0130
52006900	T	0131
52007000	T	0131
52007010	T	0131
52007012	C	0000
52007015	C	0000
52007020	T	0000
52007030	T	0000
52007040	T	0000
52007050	T	0002

START OF SEGMENT ***** 87

%0407=
%0501=

Moore Business Forms, Inc. 14727

```

SREG := SREG + 1; % ADVANCE "STACK" POINTER
INPCOUNT := INPCOUNT + 1; % COUNT THIS PARAMETER
WORD := A[SREG]; % TRANSFER PARAMETER
%.....
IF WORD.[3:3] GTR 0 THEN % PARAMETER IS A LETTER STRING
%.....
BEGIN
  WORD.[1:5] := 0; % ZERO OUT "CODE" PORTION
  MATCH(WORD,"END ") % CONVERT TO EOFMARK
  BEGIN
    WORD := 100000000;
    GO TO UNSIGNED;
  END;
  MATCH(WORD,"PRINTER") UNIT := 2 ELSE
  MATCH(WORD,"PUNCH ") UNIT := 3 ELSE
  MATCH(WORD,"TAPE ") UNIT := 4 ELSE
  MATCH(WORD,"LPA ") UNIT := 5 ELSE
  MATCH(WORD,"LPB ") UNIT := 6 ELSE
  UNIT := 0;
  IF UNIT GTR 0 THEN % HARD COPY REQUESTED
  BEGIN
    MATCH(VERB,"COPY ") ELSE GO ERROR;
    IF HARDCOPY THEN GO ERROR; % REDUNDANT
    IF NOT HAVEMINUS THEN GO ERROR; % MISSING "TO"
    IF HAVEOUTPUTFILE THEN GO ERROR;
    UNITNUMBER := UNIT; % SAVE UNIT NUMBER
    HARDCOPY := TRUE;
    HAVEMINUS := FALSE;
  END % IF UNIT GTR 0
  ELSE MATCH(WORD,"RESEQ ")
  IF LIBTOG THEN GO SUFFIX ELSE %"RESEQ" IS FILE=NAME%0407=
  BEGIN
    IF RESEQVERB OR RESEQPARAM THEN GO ERROR;
    RESEQPARAM := TRUE;
    WORK[5] := -1; % DEFAULT LOWER BOUND FOR RESEQ
    WORK[6] := 100000000; % DEFAULT UPPER BOUND FOR RESEQ
    WORK[7] := WORK[8] := 100; % DEFAULT BASE AND INCR.
    PREVNUMBER := -1; % RESET COMPARISON VALUE
    IF SAVETOG THEN % NUMBER STORED IN SAVEDWORD
    BEGIN
      IF OUTCOUNT := OUTCOUNT+1 GTR 29 THEN
        GOERR("TOOMANY",30);
        WORK[OUTCOUNT] := SAVEDWORD;
        SAVETOG := FALSE;
      END; % IF SAVETOG;
    END % IF "RESEQ"
  ELSE MATCH(WORD,"TO ")
  BEGIN
    IF HAVEMINUS THEN GO ERROR; % REDUNDANT
    HAVEMINUS := TRUE;
  END % IF "TO"
  ELSE MATCH(WORD,"LIBRARY")
  BEGIN
    IF LIBTOG THEN GO ERROR; % REDUNDANT
    IF NOT HAVEINPUTFILE THEN GOERR(WORD,55); % NO FIRST NAME
    LIBTOG := TRUE;

```

NUMUSER:

%0501=

```

52007100 T 0002
52007200 T 0003
52007300 T 0004
52007400 T 0006
52007500 T 0006
52007600 T 0007
52007700 T 0007
52007800 T 0007
52007900 T 0009
52008000 T 0011
52008100 T 0011
52008200 T 0012
52008300 T 0015
52008400 T 0015
52008500 T 0017
52008600 T 0022
52008610 C 0027
52008620 C 0032
52008700 T 0037
52008800 T 0040
52008900 T 0041
52009000 T 0042
52009100 T 0048
52009200 T 0051
52009300 T 0055
52009400 T 0059
52009500 T 0059
52009600 T 0061
52009700 T 0063
52009800 T 0063
52009850 C 0063
52009900 T 0066
52010000 T 0067
52010100 T 0071
52010200 T 0073
52010300 T 0075
52010400 T 0076
52010500 T 0078
52010600 T 0079
52010700 T 0080
52010800 T 0080
52010900 T 0082
52011000 T 0089
52011100 T 0091
52011200 T 0092
52011300 T 0092
52011400 T 0092
52011500 T 0092
52011600 T 0099
52011700 T 0102
52011800 T 0104
52011900 T 0104
52012000 T 0104
52012100 T 0108
52012200 T 0111
52012300 T 0119
52012305 C 0121

```

```

IF INPCOUNT + 2 LEQ NPARAMS THEN                                %0501=
    FIXUSER (INPCOUNT,A,EXIT,TRUE);                             %0501=
END % IF "LIBRARY"
ELSE IF LIBTOG THEN % SHOULD BE SECOND NAME OF FILE
    BEGIN
        MATCH(VERB,"DELETE ") GOERR(WORD,39); % OUTSIDE FILE
        MATCH(VERB,"RESEQ ") GOERR(WORD,39); % OUTSIDE FILE
        IF HAVEOUTPUTFILE THEN GOERR(WORD,39);
        FILNAM2 := WORD; % SAVE SECOND NAME OF FILE
        LIBTOG := FALSE;
        END % IF LIBTOG
    ELSE IF HAVEINPUTFILE THEN % MIGHT BE OUTPUT FILE
        BEGIN
            MATCH(VERB,"COPY ") ELSE GO ERROR;
            IF NOT HAVEMINUS THEN GO ERROR ELSE HAVEMINUS:=FALSE;
            IF HARDCOPY THEN GO ERROR;
            IF HAVEOUTPUTFILE THEN GO ERROR;
            IF (FILNAM3:=WORD),[42:6] NEQ " " THEN GOERR(WORD,42);
            HAVEOUTPUTFILE := TRUE;
            NOTWORKFILE := TRUE;
            END % IF HAVEINPUTFILE
        ELSE
            BEGIN % SHOULD BE INPUT FILE NAME
                IF HAVEMINUS THEN GO ERROR;
                IF COMPAR(VERB,"DELETE ") OR COMPAR(VERB,"RESEQ ") THEN
                    BEGIN
                        IF COMPAR(WORD,FILENAME) THEN GOERR("WRKFILE",62);
                        NOTWORKFILE := TRUE;
                    END;
                FILNAM1 := WORD;
                HAVEINPUTFILE := TRUE;
            END; % IF FILENAME
        END % IF PARAMETER IS A "STRING"
    ELSE
        %.....
        IF WORD,[1:2]=1 THEN % PARAMETER IS A "SPECIAL" CHARACTER
            %.....
            BEGIN
                IF WORD,[42:6]="/" THEN
                    BEGIN
                        IF LIBTOG THEN GO ERROR; LIBTOG:=TRUE;
                        GO TO NUMUSER;
                    END
                ELSE IF WORD,[42:6]="=" THEN
                    BEGIN
                        IF HAVEMINUS THEN GO ERROR; HAVEMINUS:=TRUE;
                    END
                ELSE GO ERROR; % FUNNY CHARACTER
            END % IF A SPECIAL CHARACTER
        ELSE
            %.....
            IF WORD,[1:5]=0 THEN % AN UNSIGNED NUMBER
                %.....
                BEGIN
                    IF HAVEMINUS THEN % SHOULD BE NEGATIVE NUMBER

```

SUFFIX:

%0407=

%0501=

if this is fixed
look
potential new error

COPY 100 - ABC TO XYZ

UNSIGNED:

```

52012310 C 0122
52012320 C 0123
52012400 T 0126
52012500 T 0126
52012600 T 0129
52012650 C 0130
52012700 T 0131
52012800 T 0139
52012900 T 0148
52013000 T 0156
52013100 T 0157
52013200 T 0159
52013300 T 0159
52013400 T 0162
52013500 T 0163
52013600 T 0169
52013700 T 0175
52013800 T 0178
52013900 T 0182
52014000 T 0191
52014100 T 0193
52014200 T 0194
52014300 T 0194
52014400 T 0194
52014500 T 0195
52014510 T 0198
52014520 T 0202
52014600 T 0202
52014700 T 0212
52014800 T 0213
52014810 T 0213
52014900 T 0214
52015000 T 0216
52015100 T 0216
52015200 T 0216
52015300 T 0216
52015400 T 0216
52015500 T 0221
52015600 T 0221
52015700 T 0221
52015800 T 0223
52015900 T 0223
52015950 C 0228
52016000 T 0229
52016100 T 0229
52016200 T 0231
52016300 T 0231
52016400 T 0236
52016500 T 0236
52016600 T 0239
52016700 T 0239
52016800 T 0239
52016900 T 0239
52017000 T 0241
52017100 T 0241
52017200 T 0241
52017300 T 0242

```

```

BEGIN
WORD := "WORD; HAVEMINUS := FALSE; GO TO NEGATIVE;
END;
IF SAVETOG THEN % ALREADY HAVE A NUMBER STORED IN "SAVEWORD"
BEGIN
IF RESEQPARAM THEN
BEGIN
IF HAVERESEQRANGE THEN GOERR(WORD,59);
IF LASTPARAM THEN WORK[7]:=SAVEWORD
ELSE GOERR(WORD,59);
END % IF RESEQPARAM
ELSE IF RESEQVERB THEN % BASE OR RANGE L.B.
BEGIN
IF HAVERESEQRANGE AND HAVENEWBASE THEN GOERR(WORD,59);
IF HAVERESEQRANGE OR LASTPARAM THEN
BEGIN
WORK[7] := SAVEWORD;
HAVENEWBASE := TRUE;
END
ELSE
BEGIN
WORK[5] := WORK[6] := SAVEWORD; % RESEQ ONE RECORD
HAVERESEQRANGE := TRUE;
END;
END % IF RESEQVERB
ELSE
BEGIN
OUTCOUNT := OUTCOUNT + 1;
IF OUTCOUNT GTR 29 THEN GOERR("TOOMANY",30);
WORK[OUTCOUNT] := SAVEWORD;
END;
END; % IF SAVETOG
SAVEWORD := WORD; % SAVE THIS NUMBER
SAVETOG := TRUE;
END % IF PARAMETER IS AN UNSIGNED NUMBER
ELSE
%.....
IF WORD.[1:2]=2 THEN % PARAMETER IS A NEGATIVE NUMBER
%.....
BEGIN
NEGATIVE:
IF NOT SAVETOG THEN GOERR(WORD,11); % MISSING LOWER BOUND
IF PREVNUMBER GEQ SAVEWORD THEN GOERR(SAVEWORD,58);
IF SAVEWORD GEQ ABS(WORD) THEN GOERR(WORD,58);
IF RESEQVERB OR RESEQPARAM THEN
BEGIN
IF HAVERESEQRANGE THEN GO ERROR;
WORK[5] := SAVEWORD; % LOWER BOUND FOR RESEQ
WORK[7] := SAVEWORD; % BASE FOR RESEQ;
SAVETOG := FALSE;
WORK[6] := ABS(WORD); % UPPER BOUND FOR RESEQ
HAVERESEQRANGE := TRUE;
END % IF RESEQVERB OR RESEQPARAM
ELSE
BEGIN
IF OUTCOUNT:=OUTCOUNT+1 GTR 28 THEN GOERR("TOOMANY",30);
WORK[OUTCOUNT] := SAVEWORD;

```

```

52017400 T 0242
52017500 T 0243
52017600 T 0246
52017700 T 0246
52017800 T 0247
52017900 T 0247
52018000 T 0248
52018100 T 0249
52018200 T 0257
52018300 T 0258
52018400 T 0266
52018500 T 0266
52018600 T 0268
52018700 T 0268
52018800 T 0277
52018900 T 0279
52019000 T 0279
52019100 T 0281
52019200 T 0282
52019300 T 0282
52019400 T 0282
52019500 T 0283
52019600 T 0285
52019700 T 0287
52019800 T 0287
52019900 T 0287
52020000 T 0287
52020100 T 0287
52020200 T 0289
52020300 T 0297
52020400 T 0298
52020500 T 0298
52020600 T 0298
52020700 T 0299
52020800 T 0300
52020900 T 0300
52021000 T 0300
52021100 T 0300
52021200 T 0304
52021300 T 0304
52021400 T 0304
52021500 T 0305
52021600 T 0313
52021700 T 0321
52021800 T 0329
52021900 T 0331
52022000 T 0331
52022100 T 0335
52022200 T 0336
52022300 T 0337
52022400 T 0339
52022500 T 0341
52022600 T 0342
52022700 T 0342
52022800 T 0342
52022900 T 0343
52023000 T 0352

```

```

SAVETOG :=FALSE;
PREVNUMBER := ABS(WORK[OUTCOUNT:=OUTCOUNT+1]:=WORD);
END; % IF NOT A RESEQ PARAMETER
END % IF PARAMETER IS A NEGATIVE NUMBER
ELSE
%.....
IF WORD.[1:2]=3 THEN % PARAMETER IS A "+" NUMBER
%.....
BEGIN
IF NOT(RESEQVERB OR RESEQPARAM) THEN GO ERROR;
IF HAVEINCREMENT THEN GO ERROR; % REDUNDANT
IF SAVETOG THEN % HAVE AN UNSIGNED NUMBER IN "SAVEWORD"
BEGIN
IF RESEQPARAM AND HAVERESEQRANGE THEN GUERR(SAVEWORD,59);
WORK[7] := SAVEWORD; % NEW BASE
SAVETOG := FALSE;
HAVENEWBASE := TRUE;
END; % IF SAVETOG
WORK[8] := WORD.[6:42]; % RESEQ INCREMENT
HAVEINCREMENT := TRUE;
END; % IF PARAMETER IS A "+" NUMBER
%.....
END WHILE STATEMENT; % NO MORE PARAMETERS
%.....
IF SAVETOG THEN IF NOT LASTPARAM THEN % NUMBER LEFT IN SAVEWORD
BEGIN LASTPARAM:=WINDUP:=TRUE; GO TO RESTART; END;
IF RESEQVERB OR RESEQPARAM THEN IF WORK[5]=0 THEN WORK[5]:=-1;
IF HARDCOPY THEN % OUTPUT TO PERIPHERAL UNIT
BEGIN
WORK[9] := 0 & UNITNUMBER[6:42:6] &
(IF HAVEINPUTFILE THEN FILNAM1 ELSE FILENAME)[12:6:36];
END; % IF HARDCOPY ("PUNCH" VERB ASSUMES REMOTE OUTPUT) %0405-
IF NOTWORKFILE THEN IF HARDCOPY THEN ELSE %0405-
WORK[9]:=IF HAVEOUTPUTFILE THEN FILNAM3 ELSE FILNAM1
ELSE
BEGIN % USING THE WORKFILE
IF FILENAME=0 THEN GOERR("NO FILE",1);
IF NOT HAVEINPUTFILE THEN % CHECK VERBS REQUIRING FILES
BEGIN
MATCH(VERB,"MERGE ") GOERR(VERB,6);
MATCH(VERB,"APPEND ") GOERR(VERB,6);
END;
IF RESEQVERB AND FILETYPE=8 THEN GOERR("RESEQ ",57);
IF RESEQVERB AND FILETYPE=1 THEN % RESEQ BASIC FILE
IF HAVENEWBASE AND HAVERESEQRANGE THEN GOERR("BASIC ",15);
END; % IF USING THE WORKFILE
WORK[3] := FILNAM1;
WORK[4] := FILNAM2;
OUTCOUNT := OUTCOUNT - 20; % ACTUAL PARAMETER COUNT
IF RESEQVERB THEN
IF HAVERESEQRANGE AND HAVENEWBASE THEN % CALL MERGE, NOT RESEQ
BEGIN
WORK[11]:=WORK[5]; % DELETION RANGE
WORK[21]:=IF WORK[5]=-1 THEN 0 ELSE WORK[5];
WORK[22] := -WORK[12] := WORK[6];
VERB := "MERGE "; WORK[10] := FILNAM1;

```

52023100	T	0353
52023200	T	0355
52023300	T	0358
52023400	T	0358
52023500	T	0358
52023600	T	0358
52023700	T	0358
52023800	T	0361
52023900	T	0361
52024000	T	0361
52024100	T	0366
52024200	T	0370
52024300	T	0370
52024400	T	0371
52024500	T	0380
52024600	T	0381
52024700	T	0383
52024800	T	0385
52024900	T	0385
52025000	T	0387
52025100	T	0389
52025200	T	0389
52025300	T	0389
87 IS	390 LONG,	NEXT SEG 86
52025400	T	0133
52025500	T	0133
52025600	T	0135
52025610	T	0139
52025700	T	0144
52025800	T	0145
52025900	T	0145
52026000	T	0147
52026100	P	0151
52026200	P	0151
52026300	T	0154
52026400	T	0157
52026500	T	0157
52026600	T	0158
52026700	T	0167
52026800	T	0168
52026900	T	0168
52027000	T	0177
52027100	T	0186
52027200	T	0186
52027210	T	0195
52027220	T	0198
52027300	T	0207
52027400	T	0207
52027500	T	0209
52027600	T	0210
52027700	T	0211
52027800	T	0212
52027900	T	0214
52028000	T	0215
52028010	T	0216
52028100	T	0220
52028200	T	0223

WHILE on line 52006800

Master Records File, 14021

```

    OUTCOUNT := 2;
    REVERSETOG := TRUE;
    END;
WORK[1] := (LL+SYSTEM)&OUTCOUNT[27:42:6]&R(FILETYPE=8)[3:47:1]
&R(REVERSETOG)[5:47:1]&R(NOT HAVEINPUTFILE)[6:47:1]
& (CONCISEBIT)[8:47:1];
RUN1 := VERB; RUN2 := "CANDE "; ERRSW := USERCODE; TEMP := 4;
LITOUT("WAIT. ",NDCRLF); A[BASE+3] := REAL(TOGWORD);
IF HAVEINPUTFILE OR HAVEOUTPUTFILE THEN % SEARCH DIRECTORY
    BEGIN
    IF HAVEINPUTFILE THEN
        BEGIN
        A[BASE] := 2; A[BASE+1] := FILNAM1;
        FIND(LL&IAM[33:41:7],0,FILNAM1,FILNAM2,USERCODE);
        END;
    IF HAVEOUTPUTFILE THEN
        BEGIN
        A[BASE] := 1+2*REAL(HAVEINPUTFILE);
        A[BASE+2] := FILNAM3;
        END;
    DOING := 1;
    WAITFOR := IAM;
    END % IF INPUT/OUTPUT FILES SPECIFIED
ELSE
    BEGIN % REFERS TO WORK FILE
    IF RESEQVERB AND FILETYPE=1 THEN RUN1 := "RESEQB ";
    % VOID
    % VOID
    % VOID
    WANT := VDISPATCH;
    A[BASE] := RUN1;
    DOING := 2;
    END;
    EOJMSG := NOT CONCISE;
    WRITEASTERISK := CONCISE;
    DISKWRITE(ESP1);
    END; % CASE 0
%.....
    BEGIN % CASE 1
    LABEL BUMMY;
    TOGWORD := BOOLEAN(A[BASE+3]); % RECOVER TOGGLES
    IF A[BASE] NEQ 1 THEN % INPUT FILE SEARCH
        BEGIN
        IF EVENTS[1] LEQ 0 THEN GOERR(A[BASE+1],2);
        IF EVENTS[2].[36:6] = 8 THEN
            BEGIN
            MATCH(RUN1,"RESEQ ") GOERR(A[BASE+1],57);
            MATCH(RUN1,"MERGE ") GOERR(A[BASE+1],57);
            END;
        IF EVENTS[2].[36:6]=1 AND RESEQVERB THEN % BASIC RESEQ
            BEGIN
            IF HAVERESEQRANGE AND HAVENEWBASE THEN GOERR("BASIC ",15);
            RUN1 := "RESEQB ";
            END;
        IF A[BASE]=3 THEN % WAITFOR OUTPUT FILE SEARCH

```

52028300	T	0225		
52028400	T	0226		
52028500	T	0227		
52028600	T	0227		
52028700	T	0232		
52028710	T	0235		
52028800	T	0238		
52028900	T	0244		
52029000	T	0247		
52029100	T	0249		
52029200	T	0249		
52029300	T	0250		
52029400	T	0250		
52029500	T	0254		
52029600	T	0258		
52029700	T	0258		
52029800	T	0259		
52029900	T	0260		
52030000	T	0263		
52030100	T	0274		
52030200	T	0274		
52030300	T	0275		
52030400	T	0275		
52030500	T	0275		
52030600	T	0275		
52030700	T	0276		
52030800	T	0281		
52030900	T	0281		
52031000	T	0281		
52031100	T	0281		
52031200	T	0281		
52031300	T	0284		
52031400	T	0285		
52031500	T	0285		
52031600	T	0288		
52031700	T	0292		
52031800	T	0296		
			86 IS 298 LONG, NEXT SEG	85
52031900	T	0002		
52032000	T	0002		
52032010	T	0002		
			START OF SEGMENT *****	88
52032100	T	0000		
52032200	T	0002		
52032300	T	0003		
52032400	T	0004		
52032500	T	0013		
52032600	T	0015		
52032700	T	0015		
52032800	T	0026		
52032900	T	0037		
52033000	T	0037		
52033010	T	0039		
52033020	T	0040		
52033030	T	0049		
52033040	T	0050		
52033100	T	0050		

```

      BEGIN
      FIND(LL&IAM[33:41:7],0,A[BASE+2],USERCODE,USERCODE);
      A[BASE] := 1; WAITFOR := IAM; GO TO EXIT;
      END; % IF WAITING FOR OUTPUT FILE
    END % IF A[BASE] NEQ 1
  ELSE
    BEGIN % CHECKING OUTPUT FILE
      IF EVENTS[1] GTR 0 THEN GOERR(A[BASE+2],3);
      END; % IF A[BASE] = 1;
      ERRSW := USERCODE;
      WANT := VDISPATCH;
      A[BASE] := RUN1;
      DOING := 2;
      END CASE 1;
  *.....
  BEGIN % CASE 2 - ADJUST AFTER CANDE JOB EOJ
  LABEL BUMMY;

  IF EVENTS[1]=1 THEN % EOJ OK
  BEGIN
    VERB:=A[BASE]; TOGWORD := BOOLEAN(A[BASE+3]);
    IF NOT ( HARDCOPY OR NOTWORKFILE) THEN % WORK FILE ALTERATION
    BEGIN
      IF FILETYPE=8 THEN % TYPE DATA FILE
      MATCH(VERB,"APPEND ") LASTRECORD :=
      LASTRECORD + COMMONCELL.[21:27] ELSE
      MATCH(VERB,"COPY  ") LASTRECORD:=COMMONCELL.[21:27];
      SOURCEFILE:=MAKEFN("IS",LL); GOODOBJ:=FALSE; NOSAVE:=FALSE;
      END;
    END; % IF EOJ OK
    IAM:=0; IF BREAKORWRU THEN LINECLEAR(LCW,A[*]);
    END; % CASE 2

  END ALL CASES;

  GO TO EXIT;
  ERROR: GOERR(WORD,8);
  EXIT;
  END PROCEDURE COPYIT;

```

```

52033200 T 0052
52033300 T 0052
52033400 T 0058
52033500 T 0063
52033600 T 0063
52033700 T 0063
52033800 T 0063
52033900 T 0068
52034000 T 0077
52034100 T 0077
52034200 T 0079
52034300 T 0079
52034400 T 0082
52034500 T 0083
88 IS 84 LONG, NEXT SEG 85
52034600 T 0003
52034700 T 0003
52034710 T 0003
START OF SEGMENT ***** 89
52034800 T 0000
52034810 T 0001
52034820 T 0001
52034830 T 0005
52034840 T 0007
52034850 T 0007
52034860 T 0009
52034870 T 0012
52034880 T 0015
52034890 T 0023
52034900 T 0030
52035000 T 0030
52035100 T 0030
52035200 T 0033
89 IS 36 LONG, NEXT SEG 85
52035300 T 0004
START OF SEGMENT ***** 90
90 IS 4 LONG, NEXT SEG 85
52035400 T 0005
52035500 T 0005
52035600 T 0011
52035700 T 0011
85 IS 19 LONG, NEXT SEG 8
*****%
53000000 T 0061
53000100 T 0061
53000200 T 0061
53000300 T 0061
53000400 T 0061
START OF SEGMENT ***** 91
53000450 T 0000
53000500 T 0000
53000550 T 0000
53000600 T 0001
53000700 T 0002

```

```

*****%
PROCEDURE SEQ(LCW,A); BOOLEAN LCW; ARRAY A[0];
*****%
  BEGIN
  REAL STREAM PROCEDURE OUTSUPR(INPUT,OUTPUT); VALUE INPUT;

  BEGIN
  DI:=LOC INPUT; DS:=7FILL;
  SI:=LOC INPUT; DI:=OUTPUT;
  B(IF SC=" " THEN SI:=SI+1 ELSE
  BEGIN DS:=CHR; TALLY:=TALLY+1; END);

```

```

OUTSUPR:=TALLY;
END STREAM PROCEDURE OUTSUPR;

```

```

53000800 T 0003
53000900 T 0003

```

```

PROCEDURE SEQCOM(LL,SEQWORD); VALUE LL, SEQWORD; REAL LL, SEQWORD;
COMMUNICATE (-14);

```

```

53001000 T 0004
53001100 T 0004

```

```

REAL PC, I, WORD, SEQNO, INCR, SEQOUT;
LABEL ERR, EXIT;
ERRSW := "SEQ "; MSGPTR := 8; SEQNO:=INCR:=1;
IF FILENAME = 0 THEN GOERR("NO FILE",1);
IF DEVICE = TC500 THEN GOERR("SEQUENC",56);
IF FILETYPE=8 THEN GOERR("DATA ",57);
IF PC := A[SREG:=BASE=A[BASE]],FF GEQ 3 THEN
  BEGIN MSGPTR := 16; GO ERR; END;
WHILE I LSS PC DO
  BEGIN
    I := I + 1;
    IF (WORD := A[INC(SREG)]),[2:4] = 0 THEN
      SEQNO := WORD,[21:27]
    ELSE IF WORD,[1:2] = 3 THEN
      INCR := WORD,[21:27]
    ELSE GO ERR;
  END
  WHILE I;
  IF SEQNO = 0 OR INCR GTR 500000 OR INCR = 0 THEN GO ERR;
  IF INCR NEQ -1 THEN
    BEGIN INCREMENT := INCR; SEQLAST := SEQIN; END;
  IF SEQNO NEQ -1 THEN SEQOUT := SEQNO
    ELSE SEQOUT := SEQLAST + INCREMENT;
  SEQLAST := SEQOUT - INCREMENT;
  IF SCHEDULELINE THEN
  BEGIN %SEND OUT SEQ NUMBER FOR SCHEDULE LINES
    I := OUTSUPR(DECONV(SEQOUT),WORK[0]);
    TWXOUT(LL,WORK[0],I,NOCRLF);
  END
  ELSE IF DEVICE=CONRAC OR DEVICE=BIDS THEN % SEND SEQUENCE
    OUTPUTNUMBERS(LCW,A[*])
  ELSE SEQCOM(LL,SEQOUT&INCREMENT[2:29:19]);
% VOID
WRITEASTERISK := FALSE;
SEQMODE := TRUE; IAM := 0; GO EXIT;
ERR: ERROR(SREG,LCW,A[*]);
EXIT:END SEQ VERB;

```

```

53001200 T 0006
53001300 T 0006
53001400 T 0006
53001500 T 0009
53001520 T 0018
53001540 T 0025
53001600 T 0032
53001700 T 0035
53001800 T 0037
53001900 T 0039
53002000 T 0039
53002100 T 0040
53002200 T 0043
53002300 T 0044
53002400 T 0047
53002500 T 0047
53002600 T 0048
53002700 T 0049
53002800 T 0052
53002900 T 0053
53003000 T 0057
53003100 T 0058
53003150 T 0062
53003200 T 0064
53003250 T 0065
53003300 T 0066
53003350 T 0069
53003400 T 0071
53003410 T 0071
53003420 T 0075
53003500 T 0077
53003600 T 0080
53003700 T 0080
53003800 T 0082
53003900 T 0086
53004000 T 0088

```

91 IS 92 LONG, NEXT SEG 8

```

*****
PROCEDURE BYE (LCW,A); BOOLEAN LCW; ARRAY A[0];
*****
BEGIN

```

```

54000000 T 0061
54000100 T 0061
54000200 T 0061
54000300 T 0061

```


LABEL SKIP, EXIT;

REAL T;

```

      IF A[BASE] LSS 100 THEN
      BEGIN
        % NOT A HELLO[NEWUSER] CALL
        IF USERCODE=IAM:=0 THEN GO EXIT;
        IF A [BASE] LSS 50 THEN % CHECK FOLLOWING IF NOT "HELLO"
      BEGIN
        IF T:=((A [SREG]=BASE-A [BASE]),FF) > 1 THEN
          GOERR ("TOOMANY",16); % TOO MANY PARAMETERS
        IF T = 1 THEN
          IF T:=(A [INC (SREG)],[6:42])="BYE " THEN%FORCED BYE
          ELSE GOERR (T,8) % ILLEGAL PARAMETER
        ELSE
          IF FILENAME NEQ 0 THEN
            IF TPCOUNT NEQ 0 OR FILENAME NEQ SOURCEFILE THEN
              IF NOT NUSAVE THEN
      BEGIN
        NOSAVE:=TRUE; GOERR("NO SAVE",48);
      END;
      END;
      IF QUICKBYE THEN
      BEGIN
        LITOUT("BYE ",1); GO SKIP;
      END;
      WORK[0] := "ON FOR "; BLANK(WORK[0]);
      TIMEUSED(WORK[1],TIMECONV(TIME(1)-LOGONTIME,FALSE));
      IF NOT SCHEDULELINE THEN
        TWXOUT(LL,WORK[0],33,CRLF);
      WORK[0] := "C&E USE"; BLANK(WORK[0]);
      TIMEUSED(WORK[1],CHARGE:=TIMECONV(PTIME,FALSE));
      TWXOUT(LL,WORK[0],33,CRLF);
      WORK[0] := "EXECUTE"; BLANK(WORK[0]);
      TIMEUSED(WORK[1],TIMECONV(ETIME,FALSE));
      TWXOUT(LL,WORK[0],33,CRLF);
      WORK[0] := "IO TIME"; BLANK(WORK[0]);
      TIMEUSED(WORK[1],TIMECONV(IOTOT,FALSE));
      TWXOUT(LL,WORK[0],33,CRLF);
      WORK[0] := "OFF AT "; BLANK(WORK[0]);
      TIMEOFDAY(WORK[1],TIMECONV(TIME(1),TRUE));
      IF NOT SCHEDULELINE THEN
        TWXOUT(LL,WORK[0],18,CRLF);
      WORK[0] := "GOODBYE"; BLANK(WORK[0]);
      WORK[1] := USERCODE; BLANK(WORK[1]);
      TWXOUT(LL,WORK[0],16,CRLF);
      WORK[0]:=DATE;

      % VOID
      TWXOUT(LL,WORK[0],8,11);
      WRITEASTERISK:=FALSE; WORK[0]:="$ ";
      SKIP:
      IF A[BASE] LSS 50 THEN % NOT HELLO
      BEGIN
        TWXOUT(LL,WORK[0],3,-1); % DISCONNECT THE LINE
        IF NOT SCHEDULELINE THEN GO EXIT; % REM.WK.FIL IN DISCNCT
      END;
      END;
      CHARGE := TIMECONV(PTIME,FALSE);
      FILESAWAY (TRUE,USERCODE);

```

```

54000400 T 0061
START OF SEGMENT ***** 92
54000500 T 0000
54000550 T 0000
54000560 T 0001
54000600 T 0001
54000601 C 0003
54000602 C 0004
54000603 C 0005
54000604 C 0008
54000605 C 0015
54000606 C 0015
54000607 C 0020
54000608 C 0026
54000610 T 0026
54000620 T 0028
54000630 T 0031
54000640 T 0033
54000650 T 0033
54000660 T 0042
54000665 C 0042
54000670 T 0042
54000675 T 0043
54000680 T 0043
54000690 T 0046
54000700 T 0046
54000800 T 0048
54000890 T 0051
54000900 T 0052
54001000 T 0055
54001050 T 0057
54001100 T 0060
54001150 T 0062
54001200 T 0064
54001250 T 0067
54001260 T 0069
54001270 T 0071
54001280 T 0073
54001300 T 0075
54001400 T 0077
54001490 T 0080
54001500 T 0081
54001600 T 0083
54001700 T 0085
54001800 T 0088
54001810 T 0090
54001820 T 0091
54001830 T 0091
54001900 T 0093
54002000 T 0096
54002050 T 0097
54002100 T 0098
54002110 T 0100
54002120 T 0101
54002150 T 0101
54002175 T 0101
54002200 P 0103

```

Moore Business Forms, Inc. 1473

```

IF SCHEDULELINE THEN DISCONNECT(LCW,A);
FILESOK := B(FILENAME:=0); TPCOUNT := 0;
EXIT: END BYE;

```

```

54002950 T 0104
54003000 T 0107
54003100 T 0113
92 IS 124 LONG, NEXT SEG 8

```

```

*****
PROCEDURE MAWNEWFILE(A); ARRAY A[0];
*****

```

```

BEGIN
LABEL NEXT,ERR,EXIT;

```

```

REAL T,MAXSREG;

```

```

DEFINE
NAMEWORD      = A[BASE]#,      % FILE NAME
TYPEWORD      = A[BASE+1]#,    % FILE TYPE
SEARCHCOUNT  = A[BASE+2]#,    % NUMBER OF DIRECTORY SEARCHES
SAVESIZE      = 3#,           % NUMBER OF WORDS USED ABOVE

```

```

DEFINE ERROR(ERROR1)=
BEGIN
MSGPTR:=ERROR1; GO TO ERR;
END#;

```

```

STREAM PROCEDURE MSG(AREA,TOG); VALUE TOG;
BEGIN
LABEL XT;
DI:=AREA;
TOG(DS:=15LIT" DUPLICATE NAME"; JUMP OUT TO XT);
DS:=15LIT" CREATED ";
XT: END STREAM PROCEDURE MSG;

```

```

CASE DOING OF
BEGIN
BEGIN % CASE 0 = SYNTAX CHECK
IF FILENAME NEQ 0 THEN
IF TPCOUNT NEQ 0 OR FILENAME NEQ SOURCEFILE THEN
IF NOT NOSAVE THEN
BEGIN
NOSAVE := TRUE; GOERR("NO SAVE",48);
END;
MAXSREG := A[SREG:=BASE-A[BASE]],[18:15] + SREG;
CLEAR(A[BASE],SAVESIZE); % ZERO OUT "STACK" WORDS
FOR SREG:=SREG+1 STEP 1 UNTIL MAXSREG DO
BEGIN
IF A[SREG],[3:3] NEQ 0 THEN % ALPHABETIC STRING
BEGIN
IF NAMEWORD=0 THEN NAMEWORD:=A[SREG] ELSE
IF TYPEWORD=0 THEN TYPEWORD:=A[SREG] ELSE
ERROR(8);

```

```

55000000 T 0061
55001000 T 0061
55002000 T 0061
55003000 T 0061
55004000 T 0061
START OF SEGMENT ***** 93
55005000 T 0000
55006000 T 0000
55007000 T 0000
55008000 T 0000
55009000 T 0000
55010000 T 0000
55011000 T 0000
55012000 T 0000
55013000 T 0000
55014000 T 0000
55015000 T 0000
55016000 T 0000
55017000 T 0000
55018000 T 0000
55019000 T 0000
55020000 T 0000
55021000 T 0000
55022000 T 0000
55023000 T 0004
55024000 T 0006
55025000 T 0007
55026000 T 0007
55027000 T 0008
55028000 T 0008
55029000 T 0008
55030000 T 0009
55031000 T 0013
55032000 T 0015
55033000 T 0015
55034000 T 0024
55035000 T 0024
55036000 T 0027
55037000 T 0028
55038000 T 0033
55039000 T 0033
55040000 T 0034
55041000 T 0035
55042000 T 0038
55043000 T 0042

```

14121 Moore Business Forms, Inc. NY

```

        END % IF ALPHABETIC STRING
    ELSE IF A[SREG]="!" THEN
        BEGIN
            IF TYPEWORD=0 AND SREG LSS MAXSREG THEN
                IF A[SREG:SREG+1],[3:3] GTR 0 THEN
                    BEGIN
                        IF (T:=FINDALLETTER(A[SREG],TYPETABLE[1],NUMOFTYPES))#0 THEN
                            BEGIN
                                TYPEWORD:= T&1[1:47:1]; % MARK ALREADY CHECKED
                                GO TO NEXT;
                            END; % IF IN THE TYPETABLE
                        END; % IF ALPHABETIC STRING
                    ERROR(8);
                END % IF "!"
            ELSE ERROR(8); % NO OTHER INPUTS ALLOWED
        NEXT;
        END SREG LOOP;

        IF NAMEWORD=0 THEN GOERR("NO NAME",6); % NO FILE NAME GIVEN
        IF NAMEWORD,[3:3] GTR 6 THEN GOERR(NAMEWORD,42);
        IF FINDAWORD(NAMEWORD,RESWRDTABLE[1],NUMOFRESWRDS) NEQ 0 THEN
            GOERR(NAMEWORD,29);
        IF TYPEWORD = 0 THEN TYPEWORD :=7 ELSE % DEFAULT IS SEQUENCE
            BEGIN
                IF NOT BOOLEAN(TYPEWORD,[1:1]) THEN % NOT PREVIOUSLY LOCATED
                    BEGIN
                        IF (T:=FINDAWORD(TYPEWORD,TYPETABLE[1],NUMOFTYPES))=0 THEN
                            GOERR(TYPEWORD,20);
                        END;
                        TYPEWORD := ABS(T);
                    IF CHECKBIT(TYPEWORD=1,RESTRICTEDLANGUAGES) THEN
                        GOERR(TYPETABLE[TYPEWORD],[6:42], 49);
                    END; % CHECKING FILE TYPE

                NAMEWORD,[1:5] :=0;
                GENFILINE(NAMEWORD,TYPETABLE[TYPEWORD]);
                FIND(LL&IAM[33:41:7],0,NAMEWORD,USERCODE,USERCODE); % SRC
                FIND(LL&IAM[33:41:7],0,NAMEWORD,[6:36],USERCODE,USERCODE); % OBJ
                FILESOK := FALSE;
                TEMP := SAVESIZE;
                DOING := 1;
                WAITFOR := IAM;
            END; % CASE 0
        *.....
        BEGIN % CASE 1 - RESULTS OF DIRECTORY SEARCH
        IF EVENTS[1] GEQ 0 THEN NAMEWORD,[1:1] :=1; % ERROR FLAG
        IF (SEARCHCOUNT:=SEARCHCOUNT+1)=2 THEN
            BEGIN
                MSG(WORK,NAMEWORD,[1:1]); % BUILD MESSAGE FOR USER.
                TWXOUT(LL,WORK[0],15,1);
                IF BOOLEAN(NAMEWORD,[1:1]) THEN % ERROR
                    BEGIN
                        LINECLEAR(LCW,A[+]);
                        MSGPTR := 3;
                        FILESOK:=TRUE;
                        WRITEASTERISK := TRUE;
                        IF SCHEDULELINE THEN

```

```

55044000 T 0044
55045000 T 0044
55046000 T 0046
55047000 T 0046
55048000 T 0049
55049000 T 0052
55050000 T 0052
55051000 T 0055
55052000 T 0056
55053000 T 0058
55054000 T 0059
55055000 T 0059
55056000 T 0059
55057000 T 0061
55058000 T 0061
55059000 T 0063
55060000 T 0064
55061000 T 0064
55062000 T 0064
55063000 T 0072
55064000 T 0078
55065000 T 0081
55066000 T 0086
55067000 T 0090
55068000 T 0090
55069000 T 0092
55070000 T 0092
55071000 T 0096
55072000 T 0101
55073000 T 0101
55073100 T 0103
55073200 T 0106
55074000 T 0114
55075000 T 0114
55076000 T 0114
55077000 T 0116
55078000 T 0118
55079000 T 0122
55080000 T 0127
55081000 T 0129
55082000 T 0130
55083000 T 0131
55084000 T 0132
55085000 T 0132
55086000 T 0132
55087000 T 0132
55088000 T 0136
55089000 T 0140
55090000 T 0140
55091000 T 0142
55092000 T 0144
55093000 T 0145
55094000 T 0145
55095000 T 0149
55095500 C 0151
55096000 T 0153
55097000 T 0156

```

X0103=

```

        IF NOT NOSTOP THEN SCHEDERROR; % TERMINATE SCHEDULE
        GO TO EXIT;
    END;
    FILENAME := NAMEWORD.[6:42];
    FILETYPE := TYPEWORD;
    LASTRECORD := 0;
    SOURCEFILE := MAKEFN("1S",LL);
    MAKEFILE(LL&13[33:41:7],1,SOURCEFILE,USERCODE,FILETYPE,-2);
    GOODOBJ := FALSE;
    NOSAVE := FALSE;
    OBJECTFILE := 0;
    SEQLAST := SEQIN := 0;
    INCREMENT := 100;
    IAM := VSETUPFILES;
    DOING := 0;
    END % IF ALL SEARCHES DONE
ELSE
    BEGIN
        WAITFOR := IAM;
        GO TO EXIT;
    END;
END CASE 1;
END ALL CASES;

```

```

GO TO EXIT;
ERR: GOERR(A[SREG],8);
EXIT;
    END PROCEDURE MAKENEWFILE;

```

```

%*****%
PROCEDURE LISTIT (LCW,A);    BOOLEAN LCW;    ARRAY A[0];
%*****%
BEGIN
    LABEL PRNT, CASES, CASE4, ERR, EXIT;

    LABEL NUMUSER;
    REAL PREVW, WORD, TYPE, CHAR, PC, I, WI;
    BOOLEAN LIB, SQUASH, CHANGES, NUMBERED, DISPLAY;
    STREAM PROCEDURE NOCHGMSG(WORK);
        BEGIN
            DI:=WORK; DS:=29LIT"NO CHANGES SINCE LAST UPDATE.";
        END;

```

```

CASE DOING OF
    BEGIN
        %.....
        BEGIN % DOING = 0

```

```

55098000 T 0157
55099000 T 0169
55100000 T 0170
55101000 T 0170
55102000 T 0173
55103000 T 0175
55104000 T 0176
55105000 T 0178
55106000 T 0182
55107000 T 0185
55108000 T 0187
55109000 T 0188
55110000 T 0191
55111000 T 0192
55112000 T 0193
55113000 T 0193
55114000 T 0193
55115000 T 0193
55116000 T 0194
55117000 T 0195
55118000 T 0195
55119000 T 0195
55120000 T 0196
START OF SEGMENT ***** 94
94 IS 3 LONG, NEXT SEG 93
55121000 T 0196
55122000 T 0197
55123000 T 0197
55124000 T 0201
55125000 T 0201
55126000 T 0202
93 IS 205 LONG, NEXT SEG 8
56000000 T 0061
56000100 T 0061
56000200 T 0061
56000300 T 0061
56000400 T 0061
START OF SEGMENT ***** 95
%0501- 56000450 C 0000
56000500 T 0000
56000600 T 0000
56000700 T 0000
56000800 T 0000
56000900 T 0000
56001000 T 0004
56001100 T 0004
56001200 T 0005
56001300 T 0005
56001400 T 0005

```

```

PREVW := -1; % ALLOW 0 AS PARAMETER.
WORK[2]:=WORK[4]:=USERCODE; WORK[3]:=SOURCEFILE;
PC := A[ SREG := BASE = A[BASE]].FF;
MATCH(VERBTABLE[A[SREG],[9:9]],[6:42],"DISPLAY")
  BEGIN DISPLAY := TRUE; GO TO PRNT; END;
IF IAM = VPRINT THEN
  BEGIN
PRNT:   IAM := VLIST;
        A[BASE] := 0;
        WRITEASTERISK := TRUE;
        END;
WI:=21;
WHILE I LSS PC DO % PICK UP ALL PARAMETERS
  BEGIN INC(I);
        WORD := A[INC(SREG)];
        IF WORD.[2:4] = 0 THEN TYPE := 0 % NUMBER OR -NUMBER.
  ELSE
        IF WORD.[1:2] = 3 THEN TYPE := 1 % +NUMBER.
  ELSE
        IF WORD.[1:2] = 1 THEN TYPE := 2 % SPECIAL CHARACTER.
  ELSE
        TYPE := 3; % ALPHA WORD.
CASES:  CASE TYPE OF
        BEGIN
%.....
        BEGIN % TYPE = 0.
          IF PREVW GEQ PREVW:=ABS(WORD) THEN GO ERR;
          WORK[WI]:=IF CHAR="-" THEN CHAR:=-ABS(WORD) ELSE WORD;
          IF INC(WI) GTR 30 THEN GO ERR;
          END CASE 0;
%.....
GO TO ERR; % TYPE = 1
%.....
        BEGIN % TYPE = 2.
          IF LIB := (CHAR := WORD.CF) = "/" THEN %0501=
NUMUSER: %0501=
          IF I + 2 LEQ PC THEN FIXUSER (I,A,EXIT,TRUE) ELSE%0501=
          ELSE %0501=
          IF CHAR = "*" THEN SQUASH := TRUE
        ELSE
          IF CHAR = "#" THEN NUMBERED := TRUE
        ELSE
          IF CHAR = "$" THEN CHANGES := TRUE;
          END CASE 2;
%.....
        BEGIN % TYPE = 3, ALPHA WORD.
          WORD.[1:5]:= 0;
          MATCH(WORD, "END ")
          BEGIN
            A[SREG] := WORD := TEN8;
            TYPE := 0;
            GO TO CASES;
          END;
          MATCH (WORD, "FILES ") % LIST FILES
          BEGIN
            IAM := VLFILES;
            COMMONCELL := PC-I;

```

```

56001500 T 0005
56001600 T 0006
56001700 T 0010
56001710 T 0013
56001720 T 0013
56001800 T 0021
56001900 T 0021
56002000 T 0022
56002100 T 0023
56002200 T 0025
56002300 T 0027
56002400 T 0027
56002500 T 0028
56002600 T 0029
56002700 T 0030
56002800 T 0032
56002900 T 0034
56003000 T 0035
56003100 T 0037
56003200 T 0038
56003300 T 0040
56003400 T 0041
56003500 T 0042
56003600 T 0043
56003700 T 0043
56003800 T 0043
56003900 T 0043
56004000 T 0045
56004100 T 0050
56004200 T 0052
56004300 T 0052
56004400 T 0052
56004500 T 0053
56004600 T 0053
56004700 P 0053
56004705 C 0055
56004710 C 0056
56004790 C 0060
56004800 T 0061
56004900 T 0063
56005000 T 0063
56005100 T 0065
56005200 T 0066
56005300 T 0068
56005400 T 0069
56005500 T 0069
56005600 T 0069
56005700 T 0071
56005800 T 0071
56005900 T 0073
56006000 T 0074
56006100 T 0075
56006200 T 0078
56006300 T 0078
56006400 T 0079
56006500 T 0080
56006600 T 0080

```

```

        GO EXIT;
        END;
        MATCH(WORD,"TO      ") CHAR := "="
    ELSE
        MATCH(WORD,"CHANGES") CHANGES:=TRUE
    ELSE
        MATCH(WORD,"SQUASHE") SQUASH :=TRUE
    ELSE
        MATCH(WORD,"NUMBERE") NUMBERED :=TRUE
    ELSE
        MATCH(WORD,"LIBRARY") BEGIN LIB:=TRUE;GO NUMUSER END ELSE
        BEGIN % NOT SPECIAL WORD = MUST BE FILE NAME.
        WORK[3 + REAL(LIB)] := WORD;
        LIB := FALSE;
        WORK[2] := 0;
        END HANDLING FILE NAME;
        END OF CASE 3;
%.....
END ALL CASES;

END ALL PARAMETERS;
IF WORK[3] = 0 THEN GOERR("NOFILE ", 1);
IF (PC:=WI-21)=0 THEN
    BEGIN
    IF TPCOUNT = 0 THEN % UPDATE NOT REQUIRED
        BEGIN
        IF CHANGES THEN % NO CHANGES IF TPCOUNT=0
            BEGIN
            NOCHGMSG(WORK);
            TWXOUT(LL,WORK[0],29,1);
            IAM := 0;
            GO TO EXIT;
            END;
        END % IF TPCOUNT = 0
        ELSE IF NOT CHANGES THEN % LIST PRGM, NEEDS PARAMS
            BEGIN
            WORK[21] := 0;
            WORK[22] := TEN8 & 3[1:46:2];
            PC := 2;
            END;
        END % IF PC = 0
        MOVE (16,WORDS,CTRANDBASE[0],WORK[5]);
        DOGHOU[LL] := WORK[0] := 0;
        WORK[1] := (LL+SYSTEM) & PC[27:42:6] & R(CHANGES)[1:47:1]
            & R(SQUASH)[2:47:1] & R(FILETYPE=8)[3:47:1]
            & R(NUMBERED)[4:47:1] & R(WORK[2] NEQ 0)[6:47:1]
            & R(DISPLAY)[7:47:1] & R(MONITORBIT)[9:47:1];
        IF WORK[2] NEQ 0 AND TPCOUNT GTR 0 THEN WORK[4] := USERLOC;
        DISKWRITE(ESP1);
        RUN1 := IF TPCOUNT=0 OR WORK[2]=0 THEN "QUIKLST" ELSE "LIST ";
        RUN2 := "CANDE "; ERRSW := USERCODE;
        IF WORK[2] NEQ 0 THEN % MEANS LIST THE WORKFILE.
            BEGIN
            WANT := VCLOSEWORKTABLE;
            ERRSW := FILENAME;
            DOING := 1;

```

56006700	T	0082
56006800	T	0084
56006900	T	0084
56007000	T	0086
56007100	T	0086
56007200	T	0091
56007300	T	0091
56007400	T	0096
56007500	T	0096
56007600	T	0101
56007650	C	0101
56007700	T	0109
56007800	T	0109
56007900	T	0111
56008000	T	0112
56008100	T	0113
56008200	T	0113
56008300	T	0113
56008400	T	0113
56008500	T	0114
56008600	T	0114
56008700	T	0122
56008800	T	0123
56008900	T	0124
56009000	T	0125
56009100	T	0126
56009200	T	0126
56009300	T	0127
56009400	T	0128
56009500	T	0129
56009600	T	0130
56009700	T	0131
56009800	T	0131
56009900	T	0131
56010000	T	0132
56010100	T	0132
56010200	T	0133
56010300	T	0136
56010400	T	0136
56010500	T	0136
56010600	T	0136
56010700	T	0139
56010800	T	0141
56010900	T	0144
56011000	T	0147
56011010	T	0150
56011020	T	0153
56011100	T	0157
56011200	T	0161
56011300	T	0166
56011400	T	0168
56011500	T	0169
56011600	T	0170
56011700	T	0171
56011800	T	0172

START OF SEGMENT ***** 96
96 IS 5 LONG, NEXT SEG 95

Moore Business Forms, Inc. sv 1487

```

IF TPCOUNT NEQ 0 THEN UPDATEBIT := 1; % UPDATE IN PROGRESS
IF A[BASE] NEQ 0 THEN % PRINT HEADING
BEGIN
  TWXOUT (LL,WORK[0],0,2);
  GENFILINE (FILENAME, TYPETABLE[FILETYPE]);
  GO TO CASE4;
END;
END % IF WORKFILE
ELSE
BEGIN % FILE NAME GIVEN
  FIND(LL&IAM[33:41:7],0,WORK[3],WORK[4],USERCODE);
  A[BASE+1] := WORK[3]; % SAVE FILE NAME
  TEMP := DOING := 3;
  WAITFOR := IAM;
  GO TO EXIT;
END;
END CASE 0;
%.....
BEGIN % DOING = 1
  WANT := VDISPATCH; ERRSW := USERCODE;
  DOING := 2;
END CASE 1;
%.....
BEGIN % DOING = 2
  IF BOOLEAN(UPDATEBIT) AND EVENTS[1] = 1 THEN % UPDATE OK
  BEGIN
    CANCELCKPT(A[*]); SOURCEFILE := MAKEFN("1S",LL);
  END;
  IAM:=0; IF BREAKORWRU THEN LINECLEAR(LCW,A[*]);
END CASE 2;
%.....
BEGIN % DOING = 3
  IF EVENTS[1] LEQ 0 THEN GOERR(A[BASE+1], 2);
  IAM := VDISPATCH; ERRSW := USERCODE;
  IF A[BASE] NEQ 0 THEN % PRINT HEADING
  BEGIN
    TWXOUT(LL,WORK[0],0,2);
    I := FNONLY(L[0],PC,ERRSW:=A[BASE+1],1);
    TWXOUT(LL,L[0],I,NOCRLF);
    GO TO CASE4;
  END;
END CASE 3;
%.....
BEGIN % DOING = 4 (DUMMY CASE)
CASE4:
  EOJMSG:=TRUE; WRITEASTERISK:=FALSE;
  TIMEOFDAY(WORK[1],TIMECONV(TIME(1),TRUE));
  WORK[0]:=DATE; TWXOUT(LL,WORK[0],18,2);
END CASE 4;
%.....
END ALL CASES;

GO EXIT;
ERR: GOERR(WORD,8);
EXIT;
END OF LISTIT;

```

```

56011900 T 0172
56012000 T 0177
56012100 T 0178
56012200 T 0178
56012300 T 0180
56012400 T 0182
56012500 T 0187
56012600 T 0187
56012700 T 0187
56012800 T 0187
56012900 T 0187
56013000 T 0191
56013100 T 0193
56013200 T 0194
56013300 T 0195
56013400 T 0196
56013500 T 0196
56013600 T 0196
56013700 T 0196
56013800 T 0196
56013900 T 0198
56014000 T 0199
56014100 T 0199
56014200 T 0199
56014210 T 0199
56014220 T 0201
56014300 T 0202
56014310 T 0205
56014400 T 0205
56014500 T 0209
56014600 T 0209
56014700 T 0209
56014800 T 0209
56014900 T 0216
56015000 T 0218
56015100 T 0219
56015200 T 0219
56015300 T 0221
56015400 T 0225
56015500 T 0227
56015600 T 0229
56015700 T 0229
56015800 T 0229
56015900 T 0229
56016000 T 0229
56016100 T 0230
56016200 T 0234
56016300 T 0237
56016400 T 0240
56016500 T 0240
56016600 T 0240
START OF SEGMENT ***** 97
97 IS 6 LONG, NEXT SEG 95
56016700 T 0241
56016800 T 0241
56016900 T 0246
56017000 T 0247

```

```

*****
PROCEDURE EXECUTE(LCW,A); BOOLEAN LCW; ARRAY A[0];
*****
BEGIN
  % HANDLER FOR "RUN","EXECUTE" AND "COMPILE" REQUESTS

  REAL T,MAXSREG;

  LABEL CASE1,CHKTYPE,RUNN,ERR,EXIT;
  DEFINE
  VERBWORD           = A[BASE]#,           % VERB "NAME"
  NAMEWORD(NAMEWORD1) = A[BASE+1+NAMEWORD1]#, % MFID,FID
  TYPEWORD           = A[BASE+3]#,         % FILE TYPE
  COMMONWORD         = A[BASE+4]#,         % COMMON VALUE
  STACKWORD          = A[BASE+5]#,         % STACK SIZE
  COREWORD           = A[BASE+6]#,         % CORE LIMIT
  PROCESSWORD        = A[BASE+7]#,         % PROCESS LIMIT
  IOWORD             = A[BASE+8]#,         % I/O LIMIT
  SEARCHTYPE         = A[BASE+9].[33:15]#, % TYPE FROM HEADER
  SEARCHCOUNT       = A[BASE+9].[18:15]#, % NUMBER OF SEARCHES
  SPECNAME           = A[BASE+9].[17:01]#, % NO "0" IN MFID %0502=
  SEARCHTOGGLE       = A[BASE+9].[01:01]#, % TOGGLE FOR CASE2
  COMPNAME           = A[BASE+10]#,        % SPECIAL COMPILER %0514=
  SAVESIZE           = 11#,               % WORDS USED ABOVE %0514=

  DEFINE TRANSFEREQUATIONVALUES =
  BEGIN
    COMMONVALUE := COMMONWORD;
    STACKSIZE   := STACKWORD;
    CORESIZE    := COREWORD;
    PROCESSLIMIT := PROCESSWORD;
    IOLIMIT     := IOWORD;
  END#;

  DEFINE ERROR(ERROR1) =
  BEGIN
    MSGPTR:=ERROR1; GO TO ERR;
  END#;

CASE DOING OF
BEGIN
  BEGIN % CASE 0 = SYNTAX CHECK
  LABEL TYP;

  LABEL LIBNAME;
  REAL WORD;
  BOOLEAN WITHTOGGLE,LIBTOGGLE;
  BOOLEAN OBJECTTOGGLE;
  BOOLEAN PROCEDURE FINDNUMBER(A,MAXSREG); VALUE MAXSREG;
  REAL MAXSREG; ARRAY A[0];

```

START OF SEGMENT ***** 98

START OF SEGMENT ***** 99

%0501=

%0502=

57000000 T 0061
57001000 T 0061
57002000 T 0061
57003000 T 0061
57004000 T 0061
57005000 T 0061
57006000 T 0061
57007000 T 0061
57008000 T 0000
57009000 T 0000
57010000 T 0000
57011000 T 0000
57012000 T 0000
57013000 T 0000
57014000 T 0000
57015000 T 0000
57016000 T 0000
57017000 T 0000
57018000 T 0000
57019000 T 0000
57019500 C 0000
57020000 T 0000
57020500 C 0000
57021000 P 0000
57022000 T 0000
57023000 T 0000
57024000 T 0000
57025000 T 0000
57026000 T 0000
57027000 T 0000
57028000 T 0000
57029000 T 0000
57030000 T 0000
57031000 T 0000
57032000 T 0000
57033000 T 0000
57034000 T 0000
57035000 T 0000
57036000 T 0000
57037000 T 0000
57038000 T 0000
57039000 T 0000
57040000 T 0000
57041000 T 0000
57041500 C 0000
57042000 T 0000
57043000 T 0000
57043500 C 0000
57044000 T 0000
57045000 T 0000


```

BEGIN
% "SREG" WILL POINT TO THE LOCATION OF THE NUMBER IF THERE IS ONE
% "FINDNUMBER" WILL BE TRUE IF A NUMBER IS LOCATED
% "=" CHARACTERS ARE SKIPPED (OPTIONAL CHARACTERS)
FINDNUMBER:=FALSE; % MARK AS ERROR AT START
FOR SREG:=SREG+1 STEP 1 UNTIL MAXSREG DO
  BEGIN
    IF A[SREG].[1:5]=0 THEN % NUMERIC INPUT
      BEGIN
        MAXSREG:=0; FINDNUMBER:=TRUE;
      END
    ELSE IF NOT(A[SREG].[1:2]=1 AND A[SREG]="=") THEN MAXSREG:=0;
  END;
SREG:=SREG-1; % BACK UP FROM END OF "FOR" STATEMENT
END PROCEDURE FINDNUMBER;

```

```

57046000 T 0000
57047000 T 0000
57048000 T 0000
57049000 T 0000
57050000 T 0000
57051000 T 0000
57052000 T 0005
57053000 T 0005
57054000 T 0006
57055000 T 0007
57056000 T 0008
57057000 T 0008
57058000 T 0013
57059000 T 0013
57060000 T 0015

```

```

CLEAR(A[BASE+1],SAVESIZE-1); % ZERO OUT "STACK" WORDS
MAXSREG := A[SREG:=BASE-A[BASE]].[18:15] + SREG;
VERBWORD:= VERBTABLE[A[SREG].[9:9]].[6:42]; % VERB NAME FROM TABLE
FOR SREG := SREG+1 STEP 1 UNTIL MAXSREG DO
  BEGIN % EXAMINE THE PARAMETERS
    IF A[SREG].[3:3] GTR 0 THEN % ALPHABETIC STRING
      BEGIN
        WORD := A[SREG].[6:42];
        IF WITHTOGGLE THEN % "WITH" PRECEDED THIS WORD
          BEGIN
            MATCH(WORD,"LISTING")
            BEGIN
              MATCH(VERBWORD,"COMPILE") COMMONWORD:=-1 ELSE ERROR(8);
            END
          ELSE
            BEGIN % CHECK LABEL EQUATION VALUES
              IF NOT FINDNUMBER(A,MAXSREG) THEN ERROR(8);
              MATCH(WORD,"STACK ") STACKWORD := A[SREG].[36:12] ELSE
              MATCH(WORD,"CORE ") COREWORD := A[SREG].[33:15] ELSE
              MATCH(WORD,"PROCESS") PROCESSWORD:= A[SREG]*60 ELSE
              MATCH(WORD,"IO ") IOWORD := A[SREG]*60 ELSE
              MATCH(WORD,"COMMON ") COMMONWORD := A[SREG] ELSE
              GOERR(WORD,8); % NONE OF THE ABOVE
            END; % CHECKING LABEL EQUATION VALUES
          END % IF WITHTOGGLE
        ELSE MATCH(WORD,"WITH ") WITHTOGGLE := TRUE
        ELSE MATCH(WORD,"LIBRARY")
          BEGIN
LIBNAME:
            IF LIBTOGGLE OR NAMEWORD[0]=0 THEN ERROR(8);
            LIBTOGGLE := TRUE;
            IF SREG + 2 LEQ MAXSREG THEN FIXUSER (T,A,EXIT,FALSE);%0501=
          END
        ELSE MATCH (WORD,"OBJECT ") %0502=
          BEGIN %0502=
            MATCH (VERBWORD,"EXECUTE") %0502=
            IF NAMEWORD [0] ≠ 0 THEN ERROR (8) %MFID ALREADY GIVEN

```

```

57061000 T 0017
57062000 T 0017
57063000 T 0020
57064000 T 0023
57065000 T 0027
57066000 T 0031
57067000 T 0031
57068000 T 0033
57069000 T 0033
57070000 T 0035
57071000 T 0036
57072000 T 0036
57073000 T 0036
57074000 T 0038
57075000 T 0049
57076000 T 0049
57077000 T 0049
57078000 T 0050
57079000 T 0055
57080000 T 0060
57081000 T 0066
57082000 T 0072
57083000 T 0078
57084000 T 0084
57085000 T 0092
57086000 T 0092
57087000 T 0092
57088000 T 0095
57089000 T 0095
57089500 C 0099
57090000 T 0099
57091000 T 0105
57091100 C 0106
57092000 T 0111
57092100 C 0111
57092200 C 0114
57092300 C 0115
57092400 C 0116

```

```

ELSE OBJECTTOGGLE := TRUE                                *0502= 57092500 C 0123
ELSE ERROR (8)                                           *0502= 57092600 C 0126
END                                                       *0502= 57092700 C 0130
ELSE MATCH (WORD,"SPECIAL") IF TYPEWORD EQL 0 THEN      *0514= 57092800 C 0130
    TYPEWORD:=99 ELSE ERROR (8)                          *0514= 57092900 C 0134
ELSE
    BEGIN % CHECK FOR FILE TYPES,FILE NAMES
    IF T:=FINDAWORD(A[SREG],TYPETABLE[1],NUMOFTYPES)≠0 THEN 57093000 T 0142
        BEGIN % WORD IS A FILETYPE
        IF TYPEWORD NEQ 0 THEN ERROR(8); % DUP TYPE      57094000 T 0142
        IF NOT BOOLEAN(T,[1:1]) THEN GOERR("TYPE ",19); 57095000 T 0143
        TYPEWORD:=ABS(T);
        IF CHECKBIT(TYPEWORD-1,RESTRICTEDLANGUAGES) THEN 57096000 T 0145
            ERROR(49); % USER IS NOT PERMITTED THIS LANGUAGE
        END % IF FILE TYPE
        ELSE
        IF TYPEWORD EQL 99 THEN IF COMPNAME EQL 0 THEN    *0514= 57097000 T 0146
            COMPNAME := WORD ELSE ERROR (8) ELSE        *0514= 57098000 T 0152
            BEGIN % CHECK FILE NAMES
            IF NAMEWORD[REAL(LIBTOGGLE)] NEQ 0 THEN ERROR(8);
            IF NOT LIBTOGGLE THEN % MFID, 6 CHRS,MAXIMUM 57099000 T 0160
            IF NOT OBJECTTOGGLE THEN
                IF WORD,[42:6] NEQ " " THEN ERROR(42); % 6 CHRS MAX *0502= 57100000 T 0162
                NAMEWORD[REAL(LIBTOGGLE)]:=WORD;
            END % FILE NAME CHECK
            END % FILE TYPE OR FILE NAME
            END % ALPHABETIC STRING
        ELSE IF A[SREG]="/" THEN
            GO TO LIBNAME
        ELSE IF A[SREG]=":" THEN % COMPILER NAME ABBREVIATION *0501= 57101000 T 0165
            BEGIN
            IF SREG=MAXSREG THEN ERROR(8) ELSE WORD:=(A[SREG]:SREG+1));
            IF T:=FINDALETTER(WORD,TYPETABLE[1],NUMOFTYPES)≠0 THEN GO TYP;
            ERROR(20); % NOT FILE TYPE;
            END
            ELSE ERROR(8); % NO OTHER ENTRIES ACCEPTABLE
            END SREG LOOP;
        TEMP := SAVESIZE; % SAVE SPACE IN "STACK" FOR VARIABLES
        IF OBJECTTOGGLE THEN
            IF NAMEWORD [0] = 0 THEN GOERR ("INCOMPL",6) *0502= 57102000 T 0169
            ELSE SPECNAME := 1;
            IF TYPEWORD EQL 99 THEN
                IF COMPNAME EQL 0 THEN GOERR ("INCOMPL",6); *0502= 57103000 T 0169
            IF NAMEWORD[0]=0 THEN % DEFAULT IS THE WORK FILE
                IF FILENAME=0 THEN GOERR("NO FILE",1);
            IF TPCOUNT NEQ 0 THEN % WORK FILE HAS BEEN ALTERED
                BEGIN
                WANT := VUPDATE;
                LITOUT("WAIT, ",NOCRLF);
                DOING := 1;
                GO TO EXIT;
                END;
            GO TO CASE1; % NO UPDATE REQUIRED
            END CASE 0;
        *0502= 57104000 T 0180
        *0502= 57105000 T 0181
        *0502= 57106000 T 0187
        *0502= 57106500 C 0187
        *0502= 57107000 T 0188
        *0502= 57108000 T 0194
        *0502= 57109000 T 0196
        *0502= 57110000 T 0196
        *0502= 57111000 T 0196
        *0502= 57112000 T 0196
        *0501= 57113000 P 0198
        *0501= 57117000 T 0198
        *0502= 57118000 T 0199
        *0502= 57119000 T 0200
        *0502= 57120000 T 0207
        *0502= 57121000 T 0210
        *0502= 57122000 T 0214
        *0502= 57123000 T 0214
        *0502= 57124000 T 0218
        *0502= 57125000 T 0218
        *0502= 57126000 T 0218
        *0502= 57126100 C 0219
        *0502= 57126200 C 0219
        *0502= 57126300 C 0228
        *0514= 57126400 C 0234
        *0514= 57126500 C 0235
        *0514= 57127000 T 0244
        *0514= 57128000 T 0246
        *0514= 57129000 T 0254
        *0514= 57130000 T 0256
        *0514= 57131000 T 0256
        *0514= 57132000 T 0257
        *0514= 57133000 T 0258
        *0514= 57134000 T 0259
        *0514= 57135000 T 0261
        *0514= 57136000 T 0261
        *0514= 57137000 T 0263
        *0514= 57138000 T 0002
    
```

x.....

```

BEGIN % CASE 1 - WORKFILE UPDATED
CASE1:
MATCH(VERBWORD,"COMPILE") % COMPILE ONLY
BEGIN
IF NAMEWORD[0]#0 THEN % COMPILE THE WORKFILE
BEGIN
NAMEWORD[0]:=SOURCEFILE;
NAMEWORD[1]:=USERCODE;
IF TYPEWORD=0 THEN % FILE TYPE NOT SPECIFIED
BEGIN
TYPEWORD := FILETYPE; % USE WORK FILE TYPE
CHKTYPE: IF TYPEWORD NEQ 99 THEN %0514=
BEGIN %0514=
IF NOT BOOLEAN ((T:=TYPETABLE [TYPEWORD]).[1:1]) THEN %0514=
GOERR("TYPE ",19); % NOT COMPILER TYPE
IF CHECKBIT(TYPEWORD=1,RESTRICTEDLANGUAGES) THEN
GOERR(T.[6:42],49);
END; %0514=
END;
END % IF WORKFILE OR BRANCH FROM BELOW
ELSE IF SEARCHCOUNT=0 THEN % DIRECTORY SEARCH NOT EXECUTED
BEGIN
IF NAMEWORD[1] NEQ 0 THEN % CHECK OUTSIDE FILE
BEGIN
MATCH(NAMEWORD[1],USERCODE) ELSE GOERR(NAMEWORD[1],8);
END
ELSE NAMEWORD[1]:=USERCODE;
FIND(LL&IAM[33:41:7],0,NAMEWORD[0],NAMEWORD[1],USERCODE);
SEARCHTOGGLE:=1; % STOP PROCESSING ON SEARCH ERROR
SEARCHCOUNT := 1;
DOING := 2;
WAITFOR := IAM;
GO TO EXIT;
END % IF DIRECTORY SEARCH WAS NOT EXECUTED
ELSE
BEGIN % DIRECTORY SEARCH FINISHED, FILE IS IN DIRECTORY
IF TYPEWORD = 0 THEN TYPEWORD := SEARCHTYPE; % FROM HEADER
GO TO CHKTYPE;
END; % IF SEARCH WAS EXECUTED SUCCESSFULLY

% CALL OUT THE COMPILER

RUN1 := NAMEWORD[0];
RUN2:=IF COMPNAME EQL 0 THEN TYPETABLE [TYPEWORD].[6:42] ELSE
COMPNAME; %0514=
TRANSFEREQUATIONVALUES; % LABEL EQUATION ENTRIES
WANT := VCOMPILEIT;
DOING := 3; % RETURN AFTER COMPILING
GO TO EXIT;
END; % IF COMPILING

% "RUN" OR "EXECUTE" VERB

IF NAMEWORD[0]#0 THEN % EXECUTE THE WORK FILE
BEGIN
NAMEWORD[0]:=SOURCEFILE;
NAMEWORD[1]:=USERCODE;

```

```

57139000 T 0002
57140000 T 0002
57141000 T 0003
57142000 T 0004
57143000 T 0005
57144000 T 0007
57145000 T 0007
57146000 T 0010
57147000 T 0012
57148000 T 0014
57149000 T 0014
57150000 P 0016
57150100 C 0018
57150200 C 0019
57151000 T 0021
57151100 T 0029
57151200 T 0032
57151300 C 0038
57152000 T 0038
57153000 T 0038
57154000 T 0038
57155000 T 0040
57156000 T 0041
57157000 T 0043
57158000 T 0043
57159000 T 0053
57160000 T 0053
57161000 T 0056
57162000 T 0062
57163000 T 0065
57164000 T 0068
57165000 T 0068
57166000 T 0069
57167000 T 0070
57168000 T 0070
57169000 T 0070
57170000 T 0070
57171000 T 0075
57172000 T 0076
57173000 T 0076
57174000 T 0076
57175000 T 0076
57176000 T 0076
57177000 P 0078
57177050 C 0084
57178000 T 0085
57179000 T 0093
57180000 T 0093
57181000 T 0094
57182000 T 0095
57183000 T 0095
57184000 T 0095
57185000 T 0095
57186000 T 0095
57187000 T 0097
57188000 T 0097
57189000 T 0100

```

```

IF TYPEWORD=0 THEN TYPEWORD:=FILETYPE; % USE WORK FILE TYPE
IF NOT GOODOBJ THEN % NOT THE CORRECT OBJECT FILE
  BEGIN
  MATCH(VERBWORD,"EXECUTE") GOERR("NO OBJ.",17);
  SEARCHCOUNT:=3; % AVOID DIRECTORY SEARCH AFTER COMPILE
  GO TO CHKTYPE; % COMPILE THE SOURCE
  END; % IF NOT A GOOD OBJECT FILE
NAMEWORD[0]:=" "&OBJECTFILE[6:12:36]; % IN CASE SOURCE REMOVED
GO TO RUNN; % OBJECT FILE IS OK
END % IF THE WORKFILE
ELSE IF SEARCHCOUNT=0 THEN % SEARCH FOR OBJECT FILE FIRST
  BEGIN
  IF NAMEWORD[1]=0 THEN NAMEWORD[1]:=USERCODE; % FID NOT SPECIFIED
  FIND (LL&IAM [33:41:7],0,IF SPECNAME = 1 THEN NAMEWORD [0] ELSE
    NAMEWORD [0],[6:36],NAMEWORD [1],USERCODE); %0502=
  SEARCHCOUNT := 1;
  DOING := 2;
  WAITFOR := IAM;
  GO TO EXIT;
  END % INITIATING SEARCH FOR OBJECT
ELSE IF SEARCHCOUNT=1 THEN % RESULTS OF OBJECT SEARCH
  BEGIN
  IF EVENTS[1] LEQ 0 THEN % NO OBJECT IN DIRECTORY
    BEGIN
    MATCH(NAMEWORD[1],USERCODE) ELSE GOERR("NO OBJ.",17);
    MATCH(VERBWORD, "EXECUTE") GOERR("NO OBJ.",17);
    FIND(LL&IAM[33:41:7],0,NAMEWORD[0],NAMEWORD[1],USERCODE);
    SEARCHCOUNT := 2;
    SEARCHTOGGLE := 1; % STOP PROCESSING IF NO SOURCE
    DOING := 2;
    WAITFOR := IAM;
    GO TO EXIT;
    END; % IF OBJECT VERSION NOT ON DISK
  GO TO RUNN; % OBJECT FILE ON DISK
  END % RESULTS OF OBJECT SEARCH
ELSE IF SEARCHCOUNT=2 THEN % SOURCE IS IN DIRECTORY
  BEGIN
  IF TYPEWORD=0 THEN TYPEWORD:=SEARCHTYPE;
  GO TO CHKTYPE; % COMPILE THE SOURCE
  END; % IF SOURCE WAS PRESENT IN DIRECTORY
RUNN:
  RUN1 := IF SPECNAME = 1 THEN NAMEWORD [0] ELSE NAMEWORD [0],[6:36];
  RUN2:=NAMEWORD[1];
  TRANSFEREQUATIONVALUES; % LABEL EQUATION VALUES
  WANT := VDISPATCH;
  IAM := 0;
  GO TO EXIT;
  END CASE 1;
%.....
  BEGIN % CASE 2 = RESULTS OF DIRECTORY SEARCH
  IF BOOLEAN(SEARCHTOGGLE) THEN % STOP PROCESSING ON NEGATIVE SEARCH
    IF EVENTS[1] NEQ 7 THEN GOERR(NAMEWORD[0],2); % NOT IN LIBRARY
  IF (T:=EVENTS[2],[36:6]) GTR NUMOFTYPES THEN T:=0;
  SEARCHTYPE:=T;
  GO TO CASE1;
  END CASE 2;
%.....

```

```

57190000 T 0102
57191000 T 0106
57192000 T 0107
57192100 T 0108
57193000 T 0117
57194000 T 0119
57195000 T 0120
57196000 T 0120
57197000 T 0123
57198000 T 0124
57199000 T 0124
57200000 T 0126
57201000 T 0127
57202000 P 0132
57202500 C 0132
57203000 T 0143
57204000 T 0145
57205000 T 0146
57206000 T 0147
57207000 T 0147
57208000 T 0147
57209000 T 0150
57210000 T 0151
57211000 T 0152
57212000 T 0152
57212100 T 0162
57213000 T 0171
57214000 T 0177
57215000 T 0179
57216000 T 0182
57217000 T 0183
57218000 T 0184
57219000 T 0184
57220000 T 0184
57221000 T 0185
57222000 T 0185
57223000 T 0187
57224000 T 0188
57225000 T 0193
57226000 T 0193
57227000 T 0193
57228000 P 0194
57229000 T 0201
57230000 T 0204
57231000 T 0211
57232000 T 0212
57233000 T 0213
57234000 T 0213
57235000 T 0214
57236000 T 0214
57237000 T 0214
57238000 T 0215
57239000 T 0223
57240000 T 0226
57241000 T 0229
57242000 T 0229
57243000 T 0230

```

```

BEGIN % CASE 3 = AFTER COMPILING
MATCH(NAWORD[0],SOURCEFILE)
BEGIN % COMPILED THE WORKFILE
OBJECTFILE:=SOURCEFILE.[6:36];
GOODOBJ := TRUE;
CANCELCKPT(A[*]);
END;
MATCH(VERBWORD,"COMPILE") ELSE GO TO RUNN;
IAM := 0;
END CASE 3;
END ALL CASES;

```

```

GO TO EXIT;
ERR: ERRSW:=A[SREG]; WANT:=WAITFUR:=0; IAM:=VERROR;
EXIT;
END PROCEDURE EXECUTE;

```

```

%*****%
PROCEDURE LOADIT(LCW,A); BOOLEAN LCW; ARRAY A[0];
%*****%
BEGIN
DEFINE NAMEWORD      = A[BASE] #,
        TYPEWORD     = A[BASE+1]#,
        RECORDCOUNT = A[BASE+2]#,
        OBJECTNAME    = A[BASE+3]#,
        DISKTGGLE     = A[BASE+4]#,
        SAVESIZE      = 5#;
DEFINE ERR(ERR1,ERR2)=
BEGIN
ERRSW:=ERR1; MSGPTR:=ERR2; GO TO ERROR;
END#;
LABEL CASE3, ERROR, EXIT;
REAL T;
CASE DOING OF
BEGIN
BEGIN % CASE 0 = SYNTAX CHECK
IF FILENAME NEQ 0 THEN
IF TPCOUNT NEQ 0 OR FILENAME NEQ SOURCEFILE THEN
IF NOT NOSAVE THEN
BEGIN
NOSAVE:=TRUE; ERR("NO SAVE",48);
END;
SREG := BASE - A[BASE];
CLEAR(A[BASE],SAVESIZE-1);
TEMP := SAVESIZE;

```

57244000	T	0230	
57245000	T	0230	
57246000	T	0230	
57247000	T	0233	
57248000	T	0235	
57249000	T	0237	
57250000	T	0239	
57251000	T	0239	
57252000	T	0241	
57253000	T	0242	
57254000	T	0242	
START OF SEGMENT ***** 100			
100 IS	5 LONG,	NEXT SEG	98
57255000	T	0243	
57256000	T	0243	
57257000	T	0245	
57258000	T	0245	
57259000	T	0248	
57260000	T	0248	
57261000	T	0248	
98 IS	254 LONG,	NEXT SEG	8

58000000	T	0061	
58001000	T	0061	
58002000	T	0061	
58003000	T	0061	
58004000	T	0061	
START OF SEGMENT ***** 101			
58005000	T	0000	
58006000	T	0000	
58007000	T	0000	
58008000	T	0000	
58009000	T	0000	
58010000	T	0000	
58011000	T	0000	
58012000	T	0000	
58013000	T	0000	
58014000	T	0000	
58015000	T	0000	
58016000	T	0000	
58017000	T	0000	
58018000	T	0000	
58019000	T	0000	
58020000	T	0000	
58021000	T	0000	
58022000	T	0000	
58023000	T	0001	
58024000	T	0005	
58025000	T	0006	
58026000	T	0007	
58027000	T	0013	
58028000	T	0013	
58029000	T	0014	
58030000	T	0016	

```

IF A[SREG].[18:15] NEQ 1 THEN ERR("NAME ",6);
IF (T:=A[SREG]+SREG+1).[3:3]=0 THEN ERR("NAME ",6);
IF T GTR 6 THEN ERR(0,42);
IF FINDAWORD(A[SREG],RESWRDTABLE[1],NUMOFRESWRDS)≠0 THEN ERR(0,8);
NAMEWORD := A[SREG].[6:42];
FIND(LL&IAM[33:41:7],0,NAMEWORD,USERCODE,USERCODE);
DOING := 1;
WAITFOR := IAM;
END CASE 0;
*.....
BEGIN % CASE 1 - RESULTS OF SEARCH FOR SOURCE FILE
IF EVENTS[1] NEQ 7 THEN ERR(NAMEWORD,2);
IF (T:=EVENTS[2].[36:6]) LEQ NUMOFTYPES THEN TYPEWORD:=T;
RECORDCOUNT := EVENTS[3] + 1; % RECORD COUNT
IF NOT CONCISE THEN GENFILINE(NAMEWORD,TYPETABLE[TYPEWORD]);
FIND(LL&IAM[33:41:7],0,NAMEWORD,[6:36],USERCODE,USERCODE);
WAITFOR := IAM;
DOING := 2;
END CASE 1;
*.....
BEGIN % CASE 2 - RESULTS OF SEARCH FOR OBJECT FILE OR SETUPFILES
IF NOT BOOLEAN(DISKTOGGLE) THEN
  IF EVENTS[1]=7 THEN OBJECTNAME := NAMEWORD.[6:36];
  IF FILEOK THEN CANCELCKPT(A[*]) ELSE
    BEGIN
      IF NOT BOOLEAN(DISKTOGGLE) THEN % NO RETRY FOR DISK
        BEGIN
          DISKTOGGLE := 1;
          WANT := VSETUPFILES;
          GO EXIT;
        END
      ELSE
        % VOID A CARD
        ERR("NO DISK",46); % NO USER DISK
    END;
IF NOT CONCISE THEN LITOUT("LOADING",NOCRLF);
IF TYPEWORD = 8 THEN % TYPE DATA FILE
  BEGIN
    IF NOT CONCISE THEN
      BEGIN
        TWXOUT(LL,L[0],0,2);
        L[0] := "ND LOAD" & "E"[1:43:5];
        TWXOUT(LL,L[0],8,1);
      END;
    WRITEASTERISK := TRUE;
    GO TO CASE3;
  END;
WORK[3] := NAMEWORD;
WORK[0] := 0;
WORK[1] := (LL+SYSTEM) & CONCISEBIT[8:47:1];
WORK[2] := WORK[4] := ERRSW := USERCODE;
RUN1 := "LOAD ";
RUN2 := "CANDE ";
WANT := VDISPATCH;
DISKWRITE(ESP1);
WRITEASTERISK := CONCISE;
EQJMSG := NOT CONCISE;

```

```

58031000 T 0017
58032000 T 0023
58033000 T 0030
58034000 T 0033
58035000 T 0039
58036000 T 0041
58037000 T 0045
58038000 T 0046
58039000 T 0047
58040000 T 0047
58041000 T 0047
58042000 T 0047
58043000 T 0052
58044000 T 0056
58045000 T 0058
58046000 T 0062
58047000 T 0067
58048000 T 0068
58049000 T 0068
58050000 T 0069
58051000 T 0069
58052000 T 0069
58053000 T 0070
58054000 T 0075
58055000 T 0077
58056000 T 0078
58057000 T 0079
58058000 T 0079
58059000 T 0081
58060000 P 0082
58061000 T 0082
58062000 T 0082
58063000 P 0082
58064000 T 0082
58067000 T 0087
58068000 T 0087
58069000 T 0089
58070000 T 0091
58071000 T 0091
58072000 T 0093
58073000 T 0093
58074000 T 0095
58075000 T 0097
58076000 T 0099
58077000 T 0099
58078000 T 0101
58079000 T 0105
58080000 T 0105
58081000 T 0106
58082000 T 0107
58083000 T 0111
58084000 T 0114
58085000 T 0115
58086000 T 0116
58087000 T 0117
58088000 T 0121
58089000 T 0124

```

```

%0406=
%0406=

```

1410T
 Moore Business Forms, Inc.

```

GOODOBJ:=B(SOURCEFILE:=FILENAME:=FILETYPE:=LASTRECORD:=0);
DOING := 3;
GO TO EXIT;
END CASE 2;
%.....
BEGIN % CASE 3 = RETURN AFTER LOAD OR BRANCH ON DATA FILE
IF COMMONCELL.[2:1]=0 THEN % LOAD WAS OK
BEGIN
CASE3:
FILENAME := SOURCEFILE := NAMEWORD;
FILETYPE := TYPEWORD;
GOODOBJ := (OBJECTFILE:=OBJECTNAME) NEQ 0;
LASTRECORD := RECORDCOUNT;
NOSAVE := FALSE;
SEQLAST := SEQIN := 0;
INCREMENT := 100;
T:=MAKEFN("1S",LL);
MAKEFILE(LL&13[33:41:7],1,T,USERCODE,FILETYPE,0);
CANCELCKPT(A[*]); % SAVE THE NAME OF THE WORK FILE
END;
IAM := 0;
END CASE 3;
END ALL CASES;

```

```

GO TO EXIT;
ERROR:
IF ERRSW=0 THEN ERRSW:=A[SREG];
IAM := VERROR;
WANT := WAITFOR := 0;
EXIT:
END PROCEDURE LOADIT;

```

```

%*****
PROCEDURE SAVER(LCW,A); BOOLEAN LCW; ARRAY A[0];
%*****
BEGIN
LABEL ERR,CASE1,FINISHED,EXIT;
REAL T,F;
DEFINE DEFAULTSAVEFACTOR = 7#;
CASE DOING OF
BEGIN
BEGIN % CASE 0 = SYNTAX ANALYSIS
IF FILENAME=0 THEN % NO WORK FILE
BEGIN
ERRSW:="NO FILE"; MSGPTR := 1; GO TO ERR;
END;
IF (T:=A[SREG:=BASE=A[BASE]].[18:15]) GTR 1 THEN % TOO MANY PARAMS
BEGIN
ERRSW:= A[SREG+2]; MSGPTR := 4; GO TO ERR;
END;
IF T = 0 THEN A[BASE] + =1 ELSE

```

58090000	T	0127		
58091000	T	0133		
58092000	T	0134		
58093000	T	0137		
58094000	T	0137		
58095000	T	0137		
58095100	T	0137		
58095200	T	0139		
58096000	T	0139		
58097000	T	0140		
58098000	T	0142		
58099000	T	0144		
58100000	T	0149		
58101000	T	0151		
58102000	T	0153		
58103000	T	0155		
58104000	T	0156		
58105000	T	0158		
58105010	T	0162		
58105100	T	0163		
58106000	T	0163		
58107000	T	0164		
58108000	T	0164		
START OF SEGMENT ***** 102				
102 IS	5 LONG,	NEXT SEG	101	
58109000	T	0165		
58110000	T	0165		
58111000	T	0166		
58112000	T	0168		
58113000	T	0169		
58114000	T	0170		
58115000	T	0171		
101 IS	174 LONG,	NEXT SEG	8	
59000000	T	0061		
59000100	T	0061		
59000200	T	0061		
59000300	T	0061		
59000400	T	0061		
START OF SEGMENT ***** 103				
59000500	T	0000		
59000600	T	0000		
59000700	T	0000		
59000800	T	0000		
59000900	T	0000		
59001000	T	0000		
59001100	T	0001		
59001200	T	0002		
59001300	T	0006		
59001400	T	0006		
59001500	T	0009		
59001600	T	0010		
59001700	T	0013		
59001800	P	0013		

%0105=

```

IF (A[BASE]≠A[SREG+1]),[1:5] NEQ 0 THEN % NOT UNSIGNED NUMBER
  BEGIN
  ERRSW:=A[BASE]; MSGPTR:=4; GO TO ERR;
  END;
TEMP := 1;
IF TRCBUNT = 0 THEN GO TO CASE1; % WORK FILE ALREADY UPDATED
WANT := VUPDATE;
DOING := 1;
LITOUT("WAIT= ",NOCRLF);
END; % CASE 0
*.....
CASE1: BEGIN % CASE 1 - WORKFILE UPDATED
  IF NOT CONCISE THEN GENFILINE(FILENAME,TYPETABLE[FILETYPE]);
  MATCH(FILENAME,SOURCEFILE) % FILE ALREADY SAVED
  BEGIN
  FIND(LL&IAM[33:41:7],0,FILENAME,USERCODE,USERCODE);
  DOING := 3;
  WAITFOR := IAM;
  END
  ELSE
  BEGIN
  T:=LL&VMCPMSG[33:41:7];
  REPLACE(T,SOURCEFILE,USERCODE,FILENAME,USERCODE,
  (F + IF A[BASE] < 0 THEN 0 ELSE A[BASE]) & %0105=
  FILETYPE[32:42:6] & UNLOCKTOGGLE [31:47:1]); %0105=
  IF GOODOBJ THEN
  REPLACE(T,SOURCEFILE,[6:36],USERCODE,
  FILENAME,[6:36],USERCODE,
  F&FILETYPE[32:42:6]&UNLOCKTOGGLE[31:47:1])
  ELSE REMOVEFILE(T,FILENAME,[6:36],USERCODE);
  WANT:=WAITFOR:=VMCPMSG;
  DOING := PARAM := LIBMSGCTR := 2;
  LIBMTNCE := TRUE;
  END;
  END; % CASE 1
*.....
  BEGIN % CASE 2 - RESULTS OF FILE REPLACE
  IF A[BASE+1] NEQ 7 THEN % NOT REPLACED = PROBABLY FILE IN USE
  BEGIN
  ERRSW:="IGNORED"; MSGPTR := 22; GO TO ERR;
  END;
  T:=MAKEFN("IS",LL);
  MAKEFILE(LL&13[33:41:7],1,T,USERCODE,FILETYPE,0);
  SOURCEFILE:=FILENAME;
  IF GOODOBJ THEN OBJECTFILE:=FILENAME.[6:36];
  IF FILESOK THEN CANCELCKPT(A[*]);
  FINISHED:
  LITOUT("SAVED. ",2=CONCISEBIT);
  WRITEASTERISK := FALSE;
  IAM := 0;
  END; % CASE 2
*.....
  BEGIN % CASE 3 - RESULTS OF DIRECTORY SEARCH
  IF EVENTS[1] LEQ 0 THEN
  BEGIN
  ERRSW := FILENAME; MSGPTR := 67; GO TO ERR;

```

```

59001900 T 0016
59002000 T 0019
59002100 T 0020
59002200 T 0022
59002300 T 0022
59002400 T 0023
59002500 T 0025
59002600 T 0026
59002700 T 0027
59002800 T 0028
59002900 T 0028
59003000 T 0028
59003100 T 0028
59003200 T 0029
59003300 T 0032
59003400 T 0034
59003500 T 0035
59003600 T 0039
59003700 T 0039
59003800 T 0040
59003900 T 0040
59004000 T 0040
59004100 T 0043
59004300 T 0044
59004400 P 0044
59004450 C 0044
59004500 T 0054
59004600 T 0054
59004700 T 0054
59004710 T 0054
59004800 T 0054
59004900 T 0066
59005000 T 0067
59005100 T 0069
59005200 T 0072
59005300 T 0072
59005400 T 0072
59005500 T 0072
59005600 T 0072
59005700 T 0074
59005800 T 0074
59005900 T 0078
59006000 T 0078
59006100 T 0079
59006200 T 0083
59006300 T 0084
59006400 T 0088
59006500 T 0090
59006600 T 0091
59006700 T 0093
59006800 T 0095
59006900 T 0096
59007000 T 0096
59007100 T 0096
59007200 T 0096
59007300 T 0097
59007400 T 0098

```



```

END;
IF EVENTS[2],[36:6] NEQ FILETYPE OR A[BASE] GEQ 0 OR
BOOLEAN(UNLUCKTOGGLE) THEN % CHANGE HEADER
BEGIN
LLDISKREAD := A[BASE+1] := EVENTS[2],[18:15]; % HEADER ADDRESS
WANT := VDISKREAD;
TEMP := 2;
DOING := 4;
END
ELSE GO TO FINISHED;
END; % CASE 3
*.....
BEGIN % CASE 4 - HEADER IN CORE
WORK[4],[36:6]:=FILETYPE;
IF A[BASE] GEQ 0 THEN WORK[3],[2:10]:=A[BASE]; % SAVE FACTOR
IF BOOLEAN(UNLUCKTOGGLE) THEN
IF WORK[5],[1:1]=0 THEN % NOT GUARDED
WORK[5] := WORK[6] := 12;
DISKWRITE(A[BASE+1]);
IF GOODOBJ AND (A[BASE] GEQ 0 OR BOOLEAN(UNLUCKTOGGLE)) THEN
BEGIN % CHANGE OBJECT HEADER
FINB(LL&IAM[33:41:7],0,FILENAME,[6:36],USERCODE,USERCODE);
DOING := 5;
WAITFOR := IAM;
END
ELSE GO TO FINISHED;
END; % CASE 4
*.....
BEGIN % CASE 5 - RESULTS OF OBJECT DIRECTORY SEARCH
IF EVENTS[1] LEQ 0 THEN GO TO FINISHED;
LLDISKREAD := A[BASE+1] := EVENTS[2],[18:15];
WANT := VDISKREAD;
DOING := 6;
END; % CASE 5;
*.....
BEGIN % CASE 6 - OBJECT HEADER IN CORE
IF A[BASE] > 0 THEN WORK[3],[2:10] ← A[BASE]; %0105=
IF BOOLEAN(UNLUCKTOGGLE) THEN
IF WORK[5],[1:1]=0 THEN % NOT GUARDED
WORK[5] := WORK[6] := 12;
DISKWRITE(A[BASE+1]);
GO TO FINISHED;
END; % CASE 6
END ALL CASES;
GO TO EXIT;
ERR;
WANT:=WAITFOR:=0; IAM:=VERROR;
EXIT;
END PROCEDURE SAVER;

```

59007500	T	0102
59007600	T	0102
59007610	T	0105
59007700	T	0106
59007800	T	0106
59007900	T	0110
59008000	T	0111
59008100	T	0112
59008200	T	0112
59008300	T	0112
59008400	T	0112
59008500	T	0113
59008600	T	0113
59008700	T	0113
59008800	T	0116
59008810	T	0120
59008820	T	0121
59008830	T	0123
59008900	T	0126
59009000	T	0130
59009100	T	0133
59009200	T	0133
59009300	T	0138
59009400	T	0139
59009500	T	0139
59009600	T	0139
59009700	T	0139
59009800	T	0140
59009900	T	0140
59010000	T	0140
59010100	T	0141
59010200	T	0145
59010300	T	0146
59010400	T	0147
59010500	T	0147
59010600	T	0147
59010700	P	0147
59010710	T	0151
59010720	T	0152
59010730	T	0154
59010800	T	0157
59010900	T	0161
59011000	T	0162
59011100	T	0162
START OF SEGMENT ***** 104		
104 IS	8 LONG,	NEXT SEG 103
59011200	T	0163
59011300	T	0163
59011400	T	0164
59011500	T	0166
59011600	T	0166
103 IS	169 LONG,	NEXT SEG 8
*****X		
61000000	T	0061

Micro Business Computer, Inc.


```

X:=MAKEFN ("IS",LL);
REMOVEFILE (LL & 13 [33:41:7],X,[6:36],USERCODE);
IAM:=0;
GO EXIT;
END
ELSE
GOERR ("NO FILE",26)
ELSE
GO TO WRKFILE;
DOING:=1; WANT:=WAITFOR:=VMCPMSG;
LIBMTNCE:=TRUE; A[INC(SREG)]:=WC:=PARAM:=LIBMSGCTR:=WC-1;
FOR I:=1 STEP 1 UNTIL WC DO
REMOVEFILE(LL&WAITFOR[33:41:7],A[BASE+I],
USERCODE);
EXIT;
END OF REMOVE PROCEDURE;

```

```

%0527= 61003230 P 0096
%0527= 61003233 C 0097
%0527= 61003236 C 0101
%0527= 61003240 P 0102
%0527= 61003243 C 0102
%0527= 61003246 C 0102
%0527= 61003250 P 0102
%0527= 61003253 C 0109
%0527= 61003260 P 0109
%0527= 61003900 T 0109
%0527= 61004000 T 0111
%0527= 61004100 T 0118
%0527= 61004200 T 0119
%0527= 61004300 T 0119
%0527= 61004400 T 0125
%0527= 61004500 T 0126

```

105 IS 130 LONG, NEXT SEG 8

```

*****
PROCEDURE CHANGE(LCW,A); BOOLEAN LCW; ARRAY A[];
BEGIN
***** START OF CHANGE CODE
LABEL EXIT;

REAL T;
TEMP:=6;
CASE DOING OF
BEGIN
BEGIN % CASE 0
CASE CHANGEWHICH(A[+]) OF
BEGIN
BEGIN
IAM := VERROR;
GO EXIT
END;
BEGIN % WHICH=1==WORKFILE TYPE
IF FILENAME = 0 THEN GOERR("NO FILE",1);
IF FILETYPE=8 AND A[BASE+1] NEQ 8 THEN GOERR("DATA ",61);
FILETYPE := A[BASE+1];
% VOID A CARD
FIND (LL&IAM [33:41:7],0,MAKEFN ("IS",LL),USERCODE,USERCODE);
TEMP := 3;
DOING := 5;
WAITFOR := IAM;
END;
BEGIN % WHICH=2==FILENAME TYPE
FIND (LL&IAM [33:41:7],0,A [BASE],A [BASE+5],USERCODE);
DOING := 1;
WAITFOR := IAM;
END;
BEGIN % WHICH=3==NAME TO NAME
FIND (LL&IAM [33:41:7],0,A [BASE+1],A [BASE+6],USERCODE);
DOING := 3; WAITFOR := IAM;
END;

```

```

62000000 T 0061
62000100 T 0061
62000200 T 0061
62000300 T 0061
62000400 T 0061
START OF SEGMENT ***** 106
62000410 T 0000
%0518= 62000500 P 0000
62000600 T 0000
62000700 T 0001
62000800 T 0001
62000900 T 0001
62001000 T 0002
62001100 T 0003
62001200 T 0003
62001300 T 0004
62001400 T 0004
62001500 T 0005
62001600 T 0005
62001650 T 0012
62001700 T 0021
%0518= 62001800 P 0023
%0518= 62001900 P 0023
62002000 T 0027
62002100 T 0028
62002200 T 0029
62002300 T 0029
62002400 T 0030
%0518= 62002500 P 0030
62002600 T 0034
62002700 T 0035
62002800 T 0036
62002900 T 0036
%0518= 62003000 P 0036
62003100 T 0041
62003200 T 0043

```



DOING=0

```

BEGIN % WHICH=4--LOCK
GOERR("LOCK ",15);
COMMENT
PUT
LOCK
STUFF
HERE;
END;
BEGIN % WHICH=5--PASSWORD
IF LL=0 THEN GOERR("PASWORD",49);
IAM := VCHNGPSWD;
DOING := 0;
GO EXIT
END;
BEGIN % WHICH=6 -- CHANGE NAME...
IF LL=0 THEN GOERR("NAME ",49);
IAM:=VCHNGNAME; GO EXIT;
END CASE 6;
BEGIN % WHICH = 7, CHANGE SAVE "FACTOR"
IF A [BASE+1] = 2 THEN IF A [BASE],[42:6] EQL " " THEN
A [BASE]:=A [BASE],[6:36] ELSE
GOERR (A [BASE],42);
FIND (LL&IAM [33:41:7],0,A [BASE],A [BASE+5],USERCODE);
DOING := 7;
WAITFOR := IAM;
END;
END WHICH CASES;

```

DOING=1

DOING=2

```

END CASE 0;
BEGIN % CASE 1- GET HEADER(CHANGE TYPE)
T:=IF SCHEDULELINE AND BOOLEAN (SCHEDNAME,[1:1]) THEN 0 ELSE 3;
IF EVENTS[1] LSS T THEN GOERR(A[BASE],2);
LLDISKREAD := A[BASE+2] := EVENTS[2],FF;
WANT := VDISKREAD;
DOING:=2;
END CASE 1;
BEGIN % CASE 2
WORK[4],[36:6] := A[BASE+1];
DISKWRITE(A[BASE+2]);
IF SCHEDULELINE THEN
IF BOOLEAN(SCHEDNAME,[1:1]) THEN % TERMINATING SCHEDULE
BEGIN
IAM := WANT := WAITFOR := 0;
SCHEDGOTO(-1);
GO TO EXIT;
END;
IF A [BASE+5] EQL USERCODE THEN
MATCH(A[BASE],FILENAME) IF FILETYPE NEQ 8 THEN
BEGIN FILETYPE := A[BASE+1];
MATCH(FILENAME,SOURCEFILE)
ELSE
BEGIN FIND(LL&IAM[33:41:7],0,SOURCEFILE,USERCODE,USERCODE);
DOING := 5; WAITFOR := IAM;
GO EXIT
END;
END; IAM := 0;

```

62003300	T	0043
62003400	T	0043
62003500	T	0050
62003600	T	0050
62003700	T	0050
62003800	T	0050
62003900	T	0050
62004000	T	0050
62004100	T	0050
62004110	T	0050
62004200	T	0058
62004300	T	0058
62004400	T	0059
62004500	T	0060
62004520	T	0060
62004530	T	0060
62004540	T	0068
62004560	T	0069
62004562	T	0069
62004563	C	0069
62004564	P	0073
62004565	C	0075
62004566	P	0081
62004568	T	0085
62004570	T	0086
62004572	T	0087
62004600	T	0087
START OF SEGMENT ***** 107		
107 IS 9 LONG, NEXT SEG 106		
62004700	T	0088
62004800	T	0088
62004900	P	0088
62004910	T	0092
62005000	T	0098
62005100	T	0102
62005200	P	0103
62005300	T	0104
62005400	T	0104
62005500	T	0104
62005600	T	0107
62005610	T	0111
62005620	T	0112
62005630	T	0114
62005640	T	0114
62005650	T	0116
62005660	T	0119
62005670	T	0120
62005680	C	0120
62005700	T	0122
62005800	T	0126
62005900	T	0128
62006000	T	0128
62006100	T	0131
62006200	T	0135
62006300	T	0137
62006400	T	0137
62006500	T	0137

%0518-
%0518-
%0518-
%0518-

%0518-

%0518-

CHANGEFILE
defined on line
00041100

DOING=3

```
END CASE 2;
BEGIN % CASE 3 = CHANGING NAMES
  IF EVENTS[1] GEQ 0 THEN GOERR(A[BASE+1],3);
  A[BASE+4] := (A[BASE+2] := A[BASE+1]),[6:36];
  A[BASE+3] := (A[BASE+1] := A[BASE]),[6:36];
  TI=LL & VMCPMSG [33:41:7];
  CHANGEFILE (T,A [BASE+1],A [BASE+5],A [BASE+2],A [BASE+6]);
  IF A [BASE],[42:6] EQL " " THEN
  BEGIN
    PARAM:=LIBMSGCTR:=2;
    CHANGEFILE (T,A [BASE+3],A [BASE+5],A [BASE+4],A [BASE+6]);
  END
  ELSE
    PARAM:=LIBMSGCTR:=1;
  % VOID A CARD the card was: TEMP:=3;
  WANT := WAITFOR := VMCPMSG;
  LIBMTCN := TRUE;
  DOING := 4;
END CASE 3;
```

DOING=4

```
BEGIN % CASE 4
  IF A [BASE+6] NEQ 7 THEN GOERR (- A [BASE+6],2);
  IF A [BASE+5] EQL USERCODE THEN
    MATCH(FILENAME,A[BASE+1])
  BEGIN
    FILENAME := A[BASE+2];
    MATCH(A[BASE+1],SOURCEFILE)
  BEGIN SOURCEFILE := A[BASE+2];
  % VOID A CARD
    MAKEFILE (LL&13[33:41:7],1,MAKEFN ("1S",LL),USERCODE,FILETYPE,0); %518
    IF GOODOBJ THEN OBJECTFILE := A[BASE+4];
  END;
  END;
  IAM := 0; ← means "I am finished"
END CASE 4;
```

DOING=5

```
BEGIN % CASE 5 = CHANGING SOURCEFILE TYPE
  IF EVENTS[1] NEQ 7 THEN GOERR("ARRGH ",24);
  LLDISKREAD := A[BASE+2] := EVENTS[2],FF;
  WANT := VDISKREAD;
  DOING:=6;
END CASE 5;
```

DOING=6

```
BEGIN % CASE 6 = FINISH SOURCEFILE TYPE
  WORK[4],[36:6] := A[BASE+1];
  DISKWRITE(A[BASE+2]);
  GOODOBJ:=FALSE;
  LLDISKREAD:=TPDKADR; % GET EOF POINTER RECORD IN "1P" FILE
  WANT:=VDISKREAD;
  DOING:=11;
END CASE 6;
```

DOING=7

```
BEGIN % CASE 7 = CHANGING SAVE FACTOR
  IF EVENTS[1] NEQ 7 THEN GOERR(A[BASE],2);
  LLDISKREAD := A[BASE+3] := EVENTS[2],FF; % SAVE ADDRESS
  WANT := VDISKREAD;
  DOING := 8;
END; % CASE 7
```

DOING=8

```
BEGIN % CASE 8 = HEADER NOW IN CORE
  WORK[3],[2:10] := A[BASE+2]; % REPLACE SAVE FACTOR
  DISKWRITE(A[BASE+3]); % REPLACE HEADER
```

- 62006600 T 0138
- 62006700 T 0138
- 62006800 T 0138
- 62006900 T 0145
- 62007000 T 0150
- 62007100 P 0154
- 62007200 P 0155
- 62007210 C 0161
- 62007220 C 0162
- 62007230 C 0163
- 62007300 P 0165
- 62007400 P 0170
- 62007410 C 0170
- 62007420 C 0170
- 62007500 P 0172
- 62007600 T 0172
- 62007700 T 0174
- 62007900 T 0176
- 62008000 T 0177
- 62008100 T 0177
- 62008200 P 0177
- 62008250 C 0185
- 62008300 T 0186
- 62008400 T 0186
- 62008500 T 0190
- 62008600 T 0192
- 62008700 T 0192
- 62008800 P 0197
- 62008900 P 0197
- 62009000 T 0201
- 62009100 T 0205
- 62009200 T 0205
- 62009300 T 0205
- 62009400 T 0205
- 62009500 T 0206
- 62009600 T 0206
- 62009700 T 0214
- 62009800 T 0217
- 62009900 P 0218
- 62010000 T 0219
- 62010100 T 0219
- 62010200 T 0219
- 62010300 T 0223
- 62010350 C 0227
- 62010375 C 0229
- 62010400 P 0232
- 62010450 C 0232
- 62010500 T 0233
- 62010600 T 0234
- 62010700 T 0234
- 62010800 T 0240
- 62010900 T 0244
- 62011000 T 0244
- 62011100 T 0245
- 62011200 T 0246
- 62011300 T 0246
- 62011400 T 0249

DOING=8

DOING=9

DOING=10

DOING=11

DOING=12

```

IF A [BASE+1] GTR 0 OR A [BASE].[42:6] NEQ " " THEN
  BEGIN IAM := 0; GO TO EXIT;
  END;
  FIND (LL&IAM [33:41:7],0,A [BASE].[6:36],A [BASE+5],USERCODE);
  WAITFOR := IAM;
  DOING := 9;
  END; % CASE 8
  BEGIN % CASE 9
  IF EVENTS[1] NEQ 7 THEN % NO OBJECT VERSION
  BEGIN LITOUT ("ERR: ",NOCRLF);
  LITOUT (0 & "*" [6:42:6] & A [BASE] [12:6:36],CRLF);
  IAM:=0; GO TO EXIT;
  END;
  LLDISKREAD := A[BASE+3] := EVENTS[2].FF; % SAVE ADDRESS
  WANT := VDISKREAD;
  DOING := 10;
  END; % CASE 9;
  BEGIN % CASE 10 = HEADER NOW IN CORE
  WORK[3].[2:10] := A[BASE+2]; % REPLACE SAVE FACTOR
  DISKWRITE(A[BASE+3]); % REPLACE HEADER
  IAM := 0;
  END; % CASE 10
  BEGIN % CASE 11 = "1P" EOF POINTER IN CORE
  IF WORK [0]=TEN8 THEN % EOF POINTER
  BEGIN
  WORK [20]=FILETYPE; % NEW FILE TYPE
  DISKWRITE (TPDKADR); % REPLACE EOF POINTER
  IAM:=0;
  END
  ELSE
  BEGIN % RATS = FILE TYPE IN NEXT SEGMENT
  IF WORK [10]=TEN8 THEN
  A [BASE]:=0
  ELSE
  IF WORK [20]=TEN8 THEN
  A [BASE]:=10
  ELSE
  BEGIN IAM:=0; GO EXIT END;
  DOING:=12;
  LLDISKREAD:=TPDKADR+1;
  WANT:=VDISKREAD;
  TEMP:=1;
  END;
  END; % CASE 11
  BEGIN % CASE 12 = FINALLY GOT FILETYPE IN "1P" EOF RECORD
  WORK [A [BASE]]:=FILETYPE;
  DISKWRITE (TPDKADR+1);
  IAM:=0;
  END; % CASE 12
  END ALL CASES;

```

```

%0518- 62011500 P 0253
62011600 T 0256
62011700 T 0258
62011800 P 0258
62012000 T 0263
62012100 T 0264
62012200 T 0264
62012300 T 0265
62012400 T 0265
%0518- 62012500 P 0266
%0518- 62012550 C 0267
%0518- 62012600 P 0271
%0518- 62012650 C 0275
62012700 T 0275
62012800 T 0278
62012900 T 0279
62013000 T 0280
62013100 T 0280
62013200 T 0280
62013300 T 0284
62013400 T 0288
62013500 T 0288
%0518- 62013502 C 0289
%0518- 62013504 C 0289
%0518- 62013506 C 0290
%0518- 62013508 C 0290
%0518- 62013510 C 0292
%0518- 62013512 C 0297
%0518- 62013514 C 0297
%0518- 62013516 C 0297
%0518- 62013518 C 0297
%0518- 62013520 C 0300
%0518- 62013522 C 0301
%0518- 62013524 C 0302
%0518- 62013526 C 0302
%0518- 62013528 C 0306
%0518- 62013530 C 0307
%0518- 62013532 C 0307
%0518- 62013534 C 0311
%0518- 62013536 C 0312
%0518- 62013538 C 0315
%0518- 62013540 C 0316
%0518- 62013542 C 0316
%0518- 62013544 C 0316
%0518- 62013546 C 0317
%0518- 62013548 C 0317
%0518- 62013550 C 0319
%0518- 62013552 C 0324
%0518- 62013554 C 0325
62013600 T 0325

```

```

START OF SEGMENT ***** 108
108 IS 14 LONG, NEXT SEG 106
62013700 T 0326
62013800 T 0326
106 IS 329 LONG, NEXT SEG 8

```

```

EXIT;
END CHANGE VERB;

```

```

*****
PROCEDURE  RENAMEIT (LCW, A);  BOOLEAN LCW;  ARRAY A[0];
*****
      BEGIN
LABEL      EXIT;

          CASE DOING OF
      BEGIN
      BEGIN  % CASE 0
          IF FILENAME = 0 THEN GOERR ("NOFILE ", 1);
          SREG := BASE = A[BASE];
          IF A[SREG].FF=0 THEN GOERR("FILENAM",6);
          IF A[INC(SREG)].[3:3] = 0 THEN GOERR(A[SREG],8);
          IF A[SREG].[3:3]=7 THEN GOERR(A[SREG],42);
          IF FINDAWORD(A[SREG], RESWRDTABLE[1],
              NUMOFRESWRDS) NEQ 0 THEN GOERR(A[SREG],29);
          FIND (LL&IAM[33:4:7],0,A[SREG].[6:42],
              USERCODE, USERCODE);
          DOING := 1;
          WAITFOR := IAM;
      END CASE 0;
      BEGIN  % CASE 1
          IF EVENTS[1] GEQ 0 THEN GOERR(A[SREG],3);
          MATCH(FILENAME,SOURCEFILE)
      BEGIN
          WANT := VCOPY;
      END;
          FILENAME := A[SREG].[6:42]&" "[42:42:6];
          IAM := 0;
      END CASE 1;
      END ALL CASES;

EXIT:
      END OF RENAMEIT;

```

```

63000000 T 0061
63000100 T 0061
63000200 T 0061
63000300 T 0061
63000400 T 0061
START OF SEGMENT ***** 109
63000500 T 0000
63000600 T 0000
63000700 T 0000
63000800 T 0000
63000900 T 0008
63001000 T 0009
63001100 T 0018
63001150 T 0025
63001200 T 0032
63001300 T 0033
63001400 T 0040
63001500 T 0040
63001600 T 0045
63001700 T 0046
63001800 T 0046
63001900 T 0047
63002000 T 0047
63002100 T 0053
63002200 T 0053
63002300 T 0056
63002400 T 0056
63002500 T 0056
63002600 T 0060
63002700 T 0061
63002800 T 0061
START OF SEGMENT ***** 110
110 IS 3 LONG, NEXT SEG 109
63002900 T 0062
63003000 T 0063
109 IS 64 LONG, NEXT SEG 8

```

```

*****
PROCEDURE WHASIT(LCW,A);  BOOLEAN LCW;  ARRAY A[0];
*****
      BEGIN
LABEL CASE1,EXIT;

REAL TM,TP,DATE;
BOOLEAN OBJ,SLASH;  INTEGER I,PC;  REAL WORD;
*****
STREAM PROCEDURE MSG(L,FILNAM,WRKTOG,TYPE,NREKS,DATE,TM,SAVEF);
*****
VALUE FILNAM,WRKTOG,TYPE,NREKS,DATE,TM,SAVEF;
      BEGIN LOCAL SV; LABEL EXIT;
          DI:=L; DS:=8LIT" "; SI:=L; DS:=9WDS;
          DI:=L; DS:=5LIT"FILE "; SI:=LOC FILNAM; SI:=SI+1;
          7(IF SC=" " THEN JUMP OUT ELSE DS:=CHR);
          WRKTOG(DS:=11LIT" (WORKFILE)");

```

```

64000000 T 0061
64000100 T 0061
64000200 T 0061
64000300 T 0061
64000400 T 0061
START OF SEGMENT ***** 111
64000500 T 0000
64000550 C 0000
64000600 T 0000
64000700 T 0000
64000800 T 0000
64000900 T 0000
64001000 T 0000
64001100 T 0000
64001200 T 0002
64001300 T 0003
64001400 T 0006

```

```

DSI:=7LIT", TYPE "; SII:=LOC TYPE; SII:=SII+1;
7(IF SC=" " THEN JUMP OUT ELSE DSI:=CHR);
DSI:=2LIT", "; SVI:=DI;
SII:=LOC NREKS; DSI:=8DEC; DI:=SV; DSI:=7FILL; SII:=SV; DI:=SV;
8(IF SC=" " THEN SII:=SII+1 ELSE DSI:=CHR);
DSI:=8LIT" RECORDS";
WRKTOG(JUMP OUT TO EXIT);
DSI:=10LIT", CREATED "; SII:=LOC DATE; DSI:=8CHR;
SII:=LOC TM; DSI:=2LIT" ("; SVI:=DI; DSI:=4DEC;
DI:=SV; DSI:=3FILL; SII:=SV; DI:=SV;
4(IF SC=" " THEN SII:=SII+1 ELSE DSI:=CHR); DSI:=2LIT" ) ";
DSI:=3LIT"SF="; SVI:=DI;
SII:=LOC SAVEF; DSI:=4DEC; DI:=SV; DSI:=3FILL; SII:=SV; DI:=SV;
4(IF SC=" " THEN SII:=SII+1 ELSE DSI:=CHR);
EXIT; DSI:=8LIT" ";
END STREAM PROCEDURE MSG;

```

```

64001500 T 0009
64001600 T 0010
64001700 T 0013
64001800 T 0013
64001900 T 0015
64002000 T 0017
64002100 T 0018
64002200 T 0020
64002300 T 0022
64002400 T 0023
64002500 T 0024
64002600 T 0026
64002700 T 0027
64002800 T 0029
64002900 T 0030
64003000 T 0032

```

```

*.....
CASE DOING OF
BEGIN
  BEGIN % CASE 0
  SREG := BASE - A[BASE];
  TEMP := 2;
  IF A[BASE] := A[SREG], FF = 0 THEN % NO PARAMS, WANTS WRKFIL
  BEGIN
    IF FILENAME = 0 THEN GOERR("NO FILE",1); % NO WORKFILE
    IF TPCOUNT GTR 0 THEN % UPDATE REQUIRED
    BEGIN
      WANT := VUPDATE;
      LITOUT("WAIT. ",1);
      DOING := 1;
      GO TO EXIT;
    END % IF WORKFILE NEEDS UPDATE
    ELSE A[BASE+1] := SOURCEFILE;
    A [BASE+2]:=USERCODE;
    END % IF WORKFILE REQUESTED
  ELSE
  BEGIN
    SLASH:=OBJ:=BOOLEAN (A [BASE+2]:=A [BASE+1]:=I:=0);
    PC:=A [BASE];
    WHILE I LSS PC DO
    BEGIN
      INC (I);
      WORD:=A [INC (SREG)];
      IF WORD,[3:3] GTR 0 THEN % LETTER STRING
      BEGIN
        WORD:=WORD,[6:42];
        MATCH (WORD,"OBJECT ")
        OBJ:=TRUE
      ELSE
        IF A [BASE+1] EQL 0 THEN
          A [BASE+1]:=WORD
        ELSE
          IF A [BASE+2] EQL 0 THEN

```

```

64003100 T 0032
64004400 T 0032
64004500 T 0033
64004600 T 0033
64004700 T 0033
64004800 T 0035
64004900 T 0036
64005000 T 0038
64005100 T 0039
64005200 T 0047
64005300 T 0048
64005400 T 0049
64005500 T 0049
64005600 T 0050
64005700 T 0051
64005800 T 0053
64005900 T 0053
64005950 C 0055
64006000 T 0057
64006010 C 0057
64006020 C 0057
64006030 C 0058
64006040 C 0062
64006050 C 0063
64006060 C 0065
64006070 C 0065
64006080 C 0066
64006090 C 0068
64006100 P 0069
64006110 C 0070
64006120 C 0072
64006130 C 0073
64006140 C 0074
64006150 C 0074
64006160 C 0078
64006170 C 0080
64006180 C 0080

```


Moore Business Forms, Inc. 14127

```

IF SLASH THEN
  BEGIN
    A [BASE+2]:=WORD;
    SLASH:=FALSE;
  END
ELSE
  GOERR (WORD,8)
ELSE
  GOERR (WORD,8)
END
ELSE
IF WORD.[1:2] EQL 1 THEN % SPECIAL CHARACTER
IF WORD EQL "/" THEN
IF SLASH THEN
  GOERR (WORD,8)
ELSE
  BEGIN
    SLASH:=TRUE;
    IF I + 2 LEQ PC THEN
      FIXUSER (I,A,EXIT,TRUE);
    END
  ELSE
    GOERR (WORD,8)
  ELSE
    BEGIN
      WORD:=DECCONV (WORD);
      GOERR (WORD,8);
    END;
  END;
IF SLASH THEN GOERR ("INCMPLT",8);
IF OBJ THEN
IF A [BASE+1] EQL 0 THEN
IF GOODOBJ THEN
  BEGIN
    A [BASE+1]:=SOURCEFILE.[6:36];
    A [BASE]:=0;
  END
ELSE
  GOERR ("NO FILE",1)
ELSE
  A [BASE+1]:=A [BASE+1].[6:36];
IF A [BASE+2] EQL 0 THEN
  A [BASE+2]:=USERCODE;
END;
GO TO CASE1;
END; % CASE 0
*.....
BEGIN % CASE 1
A[BASE+1] := SOURCEFILE;
A [BASE+2]:=USERCODE;
CASE1:
FIND (LL&IAM (33:41:7),0,A [BASE+1],A [BASE+2],USERCODE);
WAITFOR := IAM;
DOING := 2;
END; % CASE 1
*.....
BEGIN % CASE 2

```

%0517-	64006190	C	0082
%0517-	64006200	P	0083
%0517-	64006210	C	0084
%0517-	64006220	C	0085
%0517-	64006230	C	0086
%0517-	64006240	C	0086
%0517-	64006250	C	0086
%0517-	64006260	C	0091
%0517-	64006270	C	0091
%0517-	64006280	C	0096
%0517-	64006290	C	0096
%0517-	64006300	P	0096
%0517-	64006310	C	0098
%0517-	64006320	C	0099
%0517-	64006330	C	0100
%0517-	64006340	C	0105
%0517-	64006350	C	0105
%0517-	64006360	C	0105
%0517-	64006370	C	0106
%0517-	64006380	C	0107
%0517-	64006390	C	0111
%0517-	64006400	P	0111
%0517-	64006410	C	0111
%0517-	64006420	C	0116
%0517-	64006430	C	0116
%0517-	64006440	C	0116
%0517-	64006450	C	0118
%0517-	64006460	C	0123
%0517-	64006470	C	0123
%0517-	64006480	C	0123
%0517-	64006490	C	0130
%0517-	64006500	P	0130
%0517-	64006510	C	0132
%0517-	64006520	C	0133
%0517-	64006530	C	0134
%0517-	64006540	C	0136
%0517-	64006550	C	0137
%0517-	64006560	C	0137
%0517-	64006570	C	0137
%0517-	64006580	C	0144
%0517-	64006590	C	0144
%0517-	64006600	P	0147
%0517-	64006610	C	0149
%0517-	64006620	C	0151
%0517-	64006630	C	0151
%0517-	64006640	C	0152
%0517-	64006700	T	0152
%0517-	64006800	T	0152
%0517-	64006900	T	0152
%0517-	64006950	C	0154
%0517-	64007000	T	0156
%0517-	64007100	P	0157
%0517-	64007200	T	0162
%0517-	64007300	T	0162
%0517-	64007400	T	0163
%0517-	64007500	T	0164
%0517-	64007600	T	0164

```

IF A [BASE+1],[6:6] EQL 0 THEN A [BASE+1],[6:6]:="*";
IF EVENTS [1] LEQ 0 THEN
  BEGIN
    GOERR (A [BASE+1],2);
  END;
LLDISKREAD := EVENTS[2].FF; % HEADER ADDRESS
WANT := VDISKREAD;
DOING := 3;
END; % CASE 2
%.....
BEGIN % CASE 3; % HEADER NOW IN CORE
DATE := DATECONV(WORK[3],[30:18]);
IF TP:=WORK[4],[36:6] GTR NUMOFTYPES THEN TP:=0;
TM:=WORK[1],[25:23]; % CREATION TIME
TM:=TM DIV 216000*100+TM DIV 3600 MOD 60 DIV 1;
IF A[BASE] = 0 THEN % WANTS WORKFILE SPECS
MSG (L,IF A [BASE+1],[6:6] EQL "*" THEN 0 & "*" [6:42:6] &
  FILENAME [12:6:36] ELSE FILENAME,TRUE,TYPETABLE [FILETYPE],
  WORK [7]+1,DATE,TM,WORK [3],[2:10])
ELSE % WANTS NAMED FILE SPECS
MSG(L,A[BASE+1],FALSE,TYPETABLE[TP],WORK[7]+1,
  DATE,TM,WORK[3],[2:10]);
TWXOUT(LL,L[0],80,1);
IAM := 0;
END CASE 3;
END ALL CASES;

```

```

%0517= 64007650 C 0164
%0517= 64007700 P 0169
%0517= 64007710 C 0170
%0517= 64007730 C 0171
%0517= 64007740 C 0176
64007800 T 0176
64007900 T 0178
64008000 T 0179
64008100 T 0180
64008200 T 0180
64008300 T 0180
64008400 T 0180
64008500 T 0182
64008600 T 0185
64008700 T 0187
64008800 T 0191
%0517= 64008900 P 0192
64008950 C 0197
%0517= 64009000 P 0200
64009100 T 0203
64009200 T 0203
64009300 T 0209
64009400 T 0211
64009500 T 0213
64009600 T 0214
64009700 T 0214

```

```

START OF SEGMENT ***** 112
112 IS 5 LONG, NEXT SEG 111
64009800 T 0215
64009900 T 0215
111 IS 221 LONG, NEXT SEG 8

```

```

EXIT;
END PROCEDURE WHATSIT;

```

```

%*****
PROCEDURE GUARDIT(LCW,A); BOOLEAN LCW; ARRAY A[0];
%*****
BEGIN
  LABEL T;
  IF TPCOUNT NEQ 0 THEN
    BEGIN
      LITOUT("WAIT, ",NOCRLF);
      WANT := VUPDATE;
    END ELSE
    BEGIN
      RUN1 := "GUARD ";
      RUN2 := "DISK ";
      IAM := VDISPATCH;
      LITOUT("WAIT ",NOCRLF);
    END;
  END GUARDIT;

```

```

67000000 T 0061
67000100 T 0061
67000200 T 0061
67000300 T 0061
67000350 T 0061
START OF SEGMENT ***** 113
67000400 T 0000
67000500 T 0001
67000600 T 0002
67000700 T 0003
67000800 T 0003
67000900 T 0003
67001000 T 0007
67001100 T 0008
67001200 T 0009
67001210 T 0010
67001300 T 0011
67001400 T 0011
113 IS 16 LONG, NEXT SEG 8

```

```

*****
PROCEDURE LOCKIT(LCW,A); BOOLEAN LCW; ARRAY A[0];
*****
BEGIN
REAL I,PC,IMAX,WORD;

                                START OF SEGMENT ***** 114
BOOLEAN LOK,UNLK,PUB,WORKFILE,NAM1,NAM2,SRC,OBJ,WTH;
                                68000000 T 0061
BOOLEAN SLASH,NAM3,NAM4; INTEGER FORGETSET;                                68000100 T 0061
LABEL RESTRT,SELECT,SECUR,CHECK,ERR1,ERR2,EXIT;                                68000200 T 0061
LABEL CHECKUSER, SELECTIT;                                68000300 T 0061
CASE DOING OF                                68000400 T 0061
  BEGIN
    BEGIN % CASE 0 = SYNTAX ANALYSIS
      PC := A[SREG:=BASE-A[BASE]],[18:15]; % PARAMETER COUNT                                68000500 T 0000
      I := VERBTABLE[A[SREG],[9:9],[6:42]]; % VERB NAME                                68000550 C 0000
      MATCH(I,"UNLOCK ") UNLK := TRUE ELSE                                68000600 T 0000
      MATCH(I,"PUBLIC ") PUB := TRUE ELSE LOK := TRUE;                                68000650 C 0000
      MATCH(FILENAME,0) ELSE % GOOD WORKFILE                                68000700 T 0000
        BEGIN
          A [BASE+1]:=SOURCEFILE; % DEFAULT FILE NAME                                68000800 T 0000
          WORKFILE := TRUE;                                68000900 T 0000
        END;                                68001000 T 0000
      IF PC=0 THEN % NO PARAMETERS                                68001100 T 0003
        BEGIN
          IF NOT WORKFILE THEN GOERR("NOFILE ",1); % NO WORK FILE                                68001110 T 0006
          NAM1:=TRUE; A[BASE]:=0; % ASSUME WORK FILE NAME                                68001120 T 0009
          GO TO SELECT; % NO FURTHER SYNTAX ANALYSIS                                68001200 T 0016
        END;                                68001300 T 0019
        A[BASE]:= (I:=SREG+1)&(IMAX:=SREG+PC)[18:33:15]&                                68001400 P 0019
        R(PUB)[15:47:1]&R(LOK)[16:47:1]&R(UNLK)[17:47:1];                                68001500 T 0021
      RESTRT:                                68001600 T 0022
        FOR I:=I STEP 1 UNTIL IMAX DO                                68001700 T 0022
          BEGIN
            IF A [I],[3:3] = 0 THEN % NOT A LETTER STRING                                68001800 T 0023
              IF A [I] = "/" THEN                                68001900 T 0023
                IF SLASH THEN GO ERR1                                68002000 T 0030
                ELSE IF WTH THEN IF NAM2 THEN                                68002100 T 0032
                  BEGIN                                68002200 T 0032
                    SLASH:=TRUE;                                68002300 T 0032
                    IF I + 2 LEQ IMAX THEN                                68002310 T 0035
                      BEGIN SREG:=I;                                68002320 T 0039
                        FIXUSER(I,A,EXIT,TRUE) END;                                68002400 T 0040
                      END                                68002500 T 0041
                    ELSE GO ERR1                                68002600 P 0041
                  ELSE IF NAM1 THEN GO CHECKUSER                                68002610 C 0042
                  ELSE GO ERR1                                68002620 C 0044
                  ELSE GO ERR1                                68002621 C 0044
                  ELSE                                68002623 C 0046
                    BEGIN                                68002625 C 0047
                      SLASH:=TRUE;                                68002627 C 0047
                      IF I + 2 LEQ IMAX THEN                                68002628 C 0047
                        BEGIN SREG:=I;                                68002629 C 0049
                          FIXUSER(I,A,EXIT,TRUE) END;                                68002630 C 0050
                        END                                68002635 C 0053
                      ELSE GO ERR1                                68002640 C 0053
                    ELSE IF NAM1 THEN GO CHECKUSER                                68002650 C 0053
                    ELSE GO ERR1                                68002660 C 0054
                  ELSE                                68002670 C 0055
                    GO ERR1                                68002680 C 0055
                  ELSE                                68002685 C 0055
                    BEGIN                                68002690 C 0055
                      WORD := A[I],[6:42];                                68002700 T 0055
                      MATCH(WORD,"SOURCE ")                                68002800 T 0057
                        BEGIN                                68002900 T 0057
                          IF SRC THEN GO TO ERR1;                                68003000 P 0059
                          IF WTH                                68003050 C 0060
                          OR OBJ THEN BEGIN FORGETSET:=2; GO SELECTIT END;                                68003100 P 0060
                        END
                    END
          END
        END
      END
    END
  END

```

SRC:=TRUE;	%0516-	68003150	C	0065
END		68003200	T	0065
ELSE MATCH(WORD,"OBJECT ")		68003300	T	0065
BEGIN		68003400	T	0065
IF OBJ THEN GO TO ERR1;	%0516-	68003500	P	0068
IF WTH	%0516-	68003550	C	0069
OR SRC THEN BEGIN FORGETSET:=1; GO SELECTIT END;	%0516-	68003600	P	0069
OBJ:=TRUE;	%0516-	68003650	C	0073
END		68003700	T	0073
ELSE MATCH(WORD,"WITH ")		68003800	T	0073
BEGIN		68003900	T	0073
IF UNLK OR PUB OR WTH THEN GO ERR1; % INVALID PARAMETER		68004000	T	0076
IF NOT(NAM1 OR WORKFILE) THEN GOERR("NOFILE ",1);		68004100	T	0078
NAM1 := WTH := TRUE; % LOCK THE WORK FILE IF NAME NOT GIVEN		68004200	T	0087
END		68004300	T	0088
ELSE IF WTH THEN % LOOKING FOR GUARD FILE NAME		68004400	T	0088
BEGIN		68004500	T	0089
IF SLASH THEN IF NAM4 THEN GO SELECTIT	%0516-	68004510	C	0089
ELSE BEGIN A [BASE+7]:=WORD;	%0516-	68004520	C	0090
SLASH:=FALSE;	%0516-	68004525	C	0093
NAM4:=TRUE END	%0516-	68004530	C	0093
ELSE	%0516-	68004540	C	0094
IF NAM2 THEN GO SELECTIT ELSE BEGIN	%0516-	68004600	P	0094
A[BASE+2] := WORD; % SAVE GUARD FILE NAME		68004700	T	0096
NAM2 := TRUE;		68004800	T	0097
END	%0516-	68004850	C	0098
END		68004900	T	0098
ELSE IF NOT NAM1 THEN % NAME OF FILE TO BE LOCKED		68005000	T	0098
BEGIN		68005100	T	0099
IF OBJ THEN	%0516-	68005110	C	0100
IF WORD,[42:6] NEQ " " THEN	%0516-	68005120	C	0100
GOERR (WORD,42);	%0516-	68005130	C	0102
A[BASE+1] := WORD;		68005200	T	0107
NAM1 := TRUE;		68005300	T	0108
END		68005400	T	0109
ELSE		68005500	T	0109
BEGIN		68005510	T	0109
IF SLASH THEN IF NAM3 THEN GO SELECTIT	%0516-	68005511	C	0110
ELSE BEGIN A [BASE+6]:=WORD;	%0516-	68005513	C	0111
SLASH:=FALSE;	%0516-	68005514	C	0113
NAM3:=TRUE END	%0516-	68005515	C	0114
ELSE BEGIN	%0516-	68005517	C	0115
SELECTIT:	%0516-	68005518	C	0115
A[BASE],[33:15]:=I; GO TO SELECT;		68005520	T	0116
END;		68005530	T	0118
END;	%0516-	68005535	C	0118
END;	%0516-	68005540	C	0118
END I LOOP;		68005600	T	0118
A[BASE]:=0; % END OF PARAMETER LIST		68005610	T	0120
SELECT:		68005700	T	0122
IF SLASH OR (NOT NAM1) OR (WTH AND NOT NAM2) THEN	%0516-	68005800	P	0122
GOERR ("INCMPLT",6);	%0516-	68005850	C	0124
IF UNLK THEN A[BASE+3]:=A[BASE+4];:=3"14" ELSE		68005900	T	0131
IF LOK THEN A[BASE+3]:=A[BASE+4];:= 0 ELSE		68005910	T	0135
BEGIN		68005920	T	0139
A[BASE+3]:=3"14"; A[BASE+4]:=0;		68005930	T	0140
END;		68005940	T	0143

1470
Michele Brindley, Esq., B.A., LL.M.

```
A [BASE+5]:=REAL (SRC) & REAL (OBJ)[46:47:1] & FORGETSET[44:46:2];
IF NOT NAM3 THEN A [BASE+6]:=USERCODE;
IF WTH THEN % CHECK FOR GUARD FILE
BEGIN
  A [BASE+3]:= " A [BASE+2];
  A [BASE+4]:=IF NAM4 THEN A [BASE+7] ELSE USERCODE;
  FIND (LL&IAM [33:7],0,A [BASE+2],A [BASE+4],USERCODE);
  TEMP:=7;
  DOING := 1;
  WAITFOR := IAM;
  GO TO EXIT;
  END; % IF "WTH"
SECUR:
IF NOT OBJ THEN SECURE (A[BASE+1],A[BASE+6],A[BASE+3],A[BASE+4]);
IF A [BASE+1],[42:6] EQL " " THEN
IF NOT SRC THEN SECURE (A [BASE+1],[6:36],A [BASE+6],A [BASE+3],
  A [BASE+4]); % OBJECT
TEMP:=6;
PARAM:=IF SRC OR OBJ OR A [BASE+1],[42:6] NEQ " " THEN 1 ELSE 2;
A [BASE+5],[42:2]:=PARAM;
LIBMSGCTR := PARAM;
LIBMTNCE := TRUE;
DOING := 2;
WANT := WAITFOR := VMCPMSG;
GO TO EXIT;
END CASE 0;
*.....
BEGIN % CASE 1 - RESULTS OF DIRECTORY SEARCH
IF EVENTS [1] LSS 0 THEN % NO GUARD FILE
BEGIN
  WORD:=A [BASE+2];
  MSGPTR := 36;
  GO ERR2;
  END;
IF EVENTS[2],[36:6] NEQ 9 THEN % NOT "LOCK" FILE
BEGIN
  WORD:=A [BASE+2];
  MSGPTR := 63;
  GO ERR2;
  END;
OBJ := BOOLEAN(A[BASE+5],[46:1]);
SRC := BOOLEAN(A[BASE+5],[47:1]);
GO TO SECUR;
END CASE 1;
*.....
BEGIN % CASE 2 - RESULTS OF LIBRARY MAINTENANCE
WORD:=A [BASE+5],[42:2] + 5;
FOR I:=6 STEP 1 UNTIL WORD DO
IF A [BASE+I] NEQ 7 THEN % SECURITY MAINTENANCE IGNORED
BEGIN
  LITOUT ("ERR: ",NOCRLF);
  MSGPTR := 36;
  IF A [BASE+I],[6:6] EQL 0 THEN
  BEGIN
  A [BASE+I],[6:6]:="*";
  IF A [BASE+I],[12:6] LEQ 9 THEN % IE "01S010"
  A [BASE+I],[12:36]:=FILENAME,[6:36];
```

```
68005950 C 0143
68005960 C 0146
68006000 T 0150
68006100 T 0150
68006210 C 0151
68006220 C 0153
68006300 P 0158
68006400 P 0163
68006500 T 0163
68006600 T 0164
68006700 T 0165
68006800 T 0165
68006900 T 0165
68007000 P 0166
68007050 C 0173
68007100 P 0175
68007150 C 0176
68007200 P 0184
68007300 P 0184
68007350 C 0189
68007400 T 0192
68007500 T 0194
68007600 T 0196
68007700 T 0197
68007800 T 0198
68007900 T 0198
68008000 T 0199
68008100 T 0199
68008200 P 0199
68008300 T 0200
68008400 P 0200
68008500 T 0202
68008600 T 0203
68008700 T 0204
68008800 T 0204
68008900 T 0205
68009000 P 0206
68009100 T 0207
68009200 T 0208
68009300 T 0209
68009400 T 0209
68009500 T 0211
68009800 T 0213
68009900 T 0213
68010000 T 0214
68010100 T 0214
68010110 C 0214
68010120 C 0216
68010200 P 0218
68010300 T 0219
68010400 P 0220
68010500 T 0221
68010510 C 0222
68010520 C 0224
68010530 C 0224
68010540 C 0227
68010550 C 0229
```

```

                END;
                LITOUT (" A [BASE+I],CRLF);
                END;
CHECK:
IF A[BASE]#0 THEN IAM:=0 ELSE
BEGIN
I:=A[BASE],[33:15]; IMAX:=A[BASE],[18:15];
PUB :=B(A[BASE],[15:1]); LOK:=B(A[BASE],[16:1]);
UNLK:=B(A[BASE],[17:1]);
IF B (FORGETSET:=A [BASE+5],[44:2]) THEN SRC:=FALSE
ELSE SRC:=B (A [BASE+5]);
IF FORGETSET = 2 THEN OBJ:=FALSE
ELSE OBJ:=B (A [BASE+5],[46:1]);
GO TO RESTRT;
END;
GO TO EXIT;
END CASE 2;
END ALL CASES;

```

```

ERR1:
ERRSW := A[I];
MSGPTR := 8;
IAM := VERROR;
GO TO EXIT;
ERR2:
LITOUT("ERR: ",NOCRLF);
LITOUT(WORD,CRLF);
GO TO CHECK; % CHECK FOR MORE PARAMETERS
EXIT;
END PROCEDURE LOCKIT;

```

```

*****
PROCEDURE TPEIT(LCW,A); BOOLEAN LCW; ARRAY A[0];
*****
BEGIN
REAL T,WORD;

BOOLEAN SQ;
LABEL EXIT;
IF SCHEDULELINE THEN GOERR("TAPE ",40);
IF DEVICE=CONRAC OR DEVICE=BIDS THEN GOERR("TAPE ",56);
CASE DOING OF
BEGIN
%.....
BEGIN % CASE 0
IF FILENAME = 0 THEN GOERR("NO FILE",1);
A[BASE+2] := A[BASE+1] := 100; % SEQ, PARAMETERS
IF T := A[SREG:=BASE-A[BASE]].FF NEQ 0 THEN % PARAMETERS GIVEN
BEGIN
BO BEGIN
IF (WORD:=A[INC(SREG)]),[3:3] NEQ 0 THEN %CHR, STRING
MATCH(WORD,[6:42],"SEQ ") SQ := TRUE

```

```

%0516- 68010560 C 0234
%0516- 68010600 P 0234
68010700 T 0236
68010710 T 0238
68010800 T 0239
68010810 T 0241
68010820 T 0244
68010830 T 0247
68010840 T 0250
%0516- 68010842 C 0251
%0516- 68010844 C 0254
%0516- 68010846 C 0257
%0516- 68010848 C 0258
68010850 T 0261
68010860 T 0262
68010900 T 0262
68011000 T 0262
68011100 T 0263
START OF SEGMENT ***** 115
115 IS 4 LONG, NEXT SEG 114
68011200 T 0264
68011300 T 0264
68011400 T 0265
68011500 T 0266
68011600 T 0267
68011700 T 0267
68011800 T 0268
68011900 T 0269
68012000 T 0270
68012100 T 0273
68012200 T 0273
114 IS 282 LONG, NEXT SEG 8

69000000 T 0061
69000100 T 0061
69000200 T 0061
69000300 T 0061
69000400 T 0061
START OF SEGMENT ***** 116
69000500 T 0000
69000600 T 0000
69000700 T 0000
69000800 T 0007
69000900 T 0016
69001000 T 0016
69001100 T 0016
69001200 T 0016
69001300 T 0016
69001400 T 0024
69001500 T 0027
69001600 T 0030
69001700 T 0031
69001800 T 0032
69001900 T 0035

```

```

ELSE GOERR(WORD,8)
ELSE IF WORD.[1:2] = 0 THEN A[BASE+1] := WORD.[21:27]
ELSE IF WORD.[1:2] = 3 THEN A[BASE+2] := WORD.[30:18]
ELSE GOERR(" " & WORD[6:42:6],8);
END UNTIL DECR(T) LEQ 0;
IF NOT SQ THEN GOERR("WHAT, ",34);
END;
WORK[0] := 0;
WORK[1] := (LL+SYSTEM)&(REAL(FILETYPE=8))[3:47:11];
WORK[2] := USERCODE;
WORK[3] := SOURCEFILE;
WORK[7] := IF SQ THEN A[BASE+1] ELSE 0; % BASE FOR SEQ;
WORK[8] := A[BASE+2]; % INCREMENT FOR SEQ;
DISKWRITE(ESP1);
RUN1 := "PAPER ";
RUN2 := "CANDE ";
ERRSW := USERCODE;
WANT := VDISPATCH;
DOING := 1;
END; % CASE 0
*.....
BEGIN % CASE 1
IF EVENTS[1] = 1 THEN % EQJ OK
BEGIN
SOURCEFILE := MAKEFN("IS",LL);
GOODOBJ := FALSE;
END;
IAM := MSGPTR := 0;
WRITEASTERISK := TRUE;
% VOID
END; % CASE 1
END ALL CASES;

EXIT;
END PROCEDURE TAPEIT;

```

```

69002000 T 0038
69002100 T 0039
69002200 T 0048
69002300 T 0053
69002400 T 0062
69002500 T 0064
69002600 T 0071
69002700 T 0071
69002800 T 0072
69002900 T 0075
69003000 T 0077
69003100 T 0078
69003200 T 0082
69003300 T 0084
69003400 T 0087
69003500 T 0089
69003600 T 0090
69003700 T 0091
69003800 T 0092
69003900 T 0092
69004000 T 0093
69004100 T 0093
69004200 T 0093
69004300 T 0094
69004400 T 0094
69004500 T 0096
69004600 T 0099
69004700 T 0099
69004800 T 0100
69004900 T 0103
69005000 T 0103
69005100 T 0103
START OF SEGMENT ***** 117
117 IS 3 LONG, NEXT SEG 116
69005200 T 0104
69005300 T 0104
116 IS 109 LONG, NEXT SEG 8

```

```

*****
PROCEDURE CHARGEIT(LCW,A); BOOLEAN LCW; ARRAY A[0];
*****
BEGIN
% VOID
% VOID
LABEL EXIT;
REAL T;
IF SCHEDULELINE THEN GOERR("CHARGE ",40);
IF A[SREG]=BASE-A[BASE].FF NEQ 1 THEN
GOERR("NO CODE",7);
T := IF A[INC(SREG)].[3:3] = 0 THEN
DECCONV(A[SREG].[21:27]) ELSE A[SREG].[6:42];
CHARGE := TIMECONV(PTIME,FALSE);
LOGOFF(LCW,A[*]);

```

```

71000000 T 0061
71000100 T 0061
71000200 T 0061
71000300 T 0061
71000400 T 0061
71000500 T 0061
71000600 T 0061
START OF SEGMENT ***** 118
71000700 T 0000
71000750 T 0000
71000800 T 0007
71000900 T 0010
71001000 T 0016
71001100 T 0018
71001150 T 0023
71001160 T 0026

```

```

LOGONTIME := TIME(1);
CHARGE := T;
IOTOT:=ETIME:=PTIME:=INPUTCTR:=0;
LOGON(LCW,A[*]);
WRITEASTERISK := TRUE;
IAM := 0;
EXIT;
END CHARGEIT;

```

```

71001170 T 0027
71001180 T 0029
71001190 T 0030
71001200 T 0035
71001300 T 0036
71001400 T 0039
71001500 T 0039
71001600 T 0040
118 IS 43 LONG, NEXT SEG 8

```

```

*****
PROCEDURE SCHEDULE(LCW,A);  BOOLEAN LCW;  ARRAY A[0];
*****
BEGIN
LABEL  ERR, EXIT;

BOOLEAN  AFTER, LIB, TOIT;
REAL     WORD, TYPE, PC, I;
        CASE DOING OF
BEGIN    # CASES.
BEGIN    % CASE 0
        IF TPCOUNT NEG 0 THEN    % NEED MERGE.
        BEGIN
        LITOUT("WAIT, ",NOCRLF);
        TEMP:=1; WANT:=VUPDATE;
        END ELSE
        BEGIN
        CLEAR(WORK[0],29);
        WORK[1] := LL&VMCPMSG[33:41:7];
        WORK[5]:=SOURCEFILE;
        WORK[4]:=WORK[6]:=USERCODE;
        WORK[7] := MAKEFN("9S",LL);
        WORK[9]:=CHARGE;
        PC:=A[SREG:=BASE=A[BASE]],FF;
        WHILE I LSS PC DO    % PICK UP ALL PARAMETERS.
        BEGIN
        INC(I);  WORD:=A[INC(SREG)];
        TYPE := IF WORD.[2:4] = 0 THEN 0    % N OR "N.
        ELSE    IF WORD.[1:2] = 3 THEN 1    % +N.
        ELSE    IF WORD.[1:2] = 1 THEN 2    % SPECIAL CHAR.
        ELSE 3;    % ALPHA WORD.
        CASE TYPE OF
        BEGIN
        BEGIN    % PARAM IS N, -N
        IF NOT AFTER THEN GOERR(DECCONV(ABS(WORD)),8);
        IF WORD LSS 800 OR WORD GTR 2400 OR
        DECCONV(WORD).[36:6] GTR 5 THEN GOERR("TIME ",41);
        WORK[8] := WORD;  AFTER := FALSE;
        END N CASE;
        GOERR(DECCONV(WORD),8);  % PARAM IS +N
        BEGIN    % CASE 2.  (SPECIAL CHAR)
        IF LIB := WORD.CF = "/" THEN    %0501=
        IF TOIT THEN GOERR ("NO USER",39) ELSE    %0501=
        IF I + 2 LEQ PC THEN FIXUSER (I,A,EXIT,TRUE) ELSE

```

```

72000000 T 0061
72000100 T 0061
72000200 T 0061
72000300 T 0061
72000400 T 0061
START OF SEGMENT ***** 119
72000500 T 0000
72000600 T 0000
72000700 T 0000
72000800 T 0000
72000900 T 0000
72001000 T 0000
72001100 T 0002
72001200 T 0002
72001300 T 0003
72001400 T 0005
72001500 T 0005
72001600 T 0008
72001700 T 0009
72001800 T 0011
72001900 T 0013
72002000 T 0015
72002050 T 0017
72002100 T 0019
72002200 T 0022
72002300 T 0023
72002400 T 0023
72002500 T 0026
72002600 T 0028
72002700 T 0031
72002800 T 0033
72002900 T 0034
72003000 T 0035
72003100 T 0035
72003200 T 0035
72003300 T 0042
72003400 T 0044
72003500 T 0054
72003600 T 0056
72003700 T 0056
72003800 T 0062
72003805 C 0062
72003810 C 0064
72003820 C 0071

```



```

ELSE
  GOERR (WORD,8);
END CASE 2;
BEGIN
  % CASE 3. (ALPHA WORD)
  WORD := WORD,[6:42];
  MATCH(WORD,"TO ") AFTER:=LIB:=NOT (TOIT:=TRUE)
ELSE
  MATCH(WORD,"AFTER ") LIB:=TOIT:=NOT (AFTER:=TRUE)
ELSE
  MATCH(WORD,"CHARGE ")
  BEGIN INC(I); WORD:=A[INC(SREG)];
  WORK[9]:=IF WORD,[3:3] GTR 0 THEN % ALPHA
  WORD,[6:42] ELSE DECCONV(WORD,[6:42]); END
ELSE %
  IF TOIT THEN WORK[3] := WORD
ELSE IF LIB THEN WORK[6] := WORD
ELSE
  WORK[5] := WORD;
END CASE 3;
END ALL CASES;

END ALL PARAMS;
IF WORK[5] = 0 THEN GOERR("NO FILE",1);
IF WORK[3] = 0 THEN GOERR("NO FILE",38);
IF WORK[3],[42:6] ≠ " " THEN GOERR(WORK[3],42);
FIND(LL&IAM[33:41:7],0,A[BASE]:=WORK[5],WORK[6],USERCODE);
A[BASE+1] := WORK[3]; DISKWRITE(ESP1);
TEMP := 2; DOING := 1; WAITFOR := IAM;
END IF TPCOUNT;
END DOING 0;
BEGIN % DOING = 1, RETURN AFTER FIND ON INPUT FILE.
IF EVENTS[1] LSS 0 THEN GOERR(A[BASE],2); % DOES NOT EXIST.
IF EVENTS[1] = 0 THEN GOERR(A[BASE],24); % SECURED FILE.
FIND(LL&IAM[33:41:7],0,A[BASE+1],USERCODE,USERCODE);
DOING := 2; WAITFOR := IAM;
END DOING 1;
BEGIN % DOING = 2, RETURN AFTER FIND ON OUTPUT FILE.
IF EVENTS[1] GEQ 0 THEN GOERR(A[BASE+1],3);
LITOUT("WAIT. ",NUCRLF);
EQJMSG := NOT CONCISE; WRITEASTERISK := CONCISE;
RUN1 := "SCHEDUL"; RUN2 := "CANDE "; IAM := VDISPATCH;
ERRSW:=USERCODE; % USED BY DISPATCH.
END DOING 2;
END ALL CASES DOING OF;

```

%0501=	72003900	P	0076
%0501=	72003905	C	0076
	72004000	T	0081
	72004100	T	0082
	72004200	T	0082
	72004300	T	0084
	72004400	T	0086
	72004500	T	0088
	72004600	T	0092
	72004620	T	0094
	72004640	T	0094
	72004645	T	0101
	72004650	T	0103
	72004660	T	0108
	72004700	T	0108
	72004800	T	0111
	72004900	T	0113
	72005000	T	0116
	72005100	T	0116
START OF SEGMENT ***** 120			
120 IS	5 LONG,	NEXT SEG	119
	72005200	T	0117
	72005300	T	0117
	72005400	T	0125
	72005450	T	0132
	72005500	T	0138
	72005600	T	0143
	72005700	T	0149
	72005800	T	0151
	72005900	T	0151
	72006000	T	0152
	72006100	T	0152
	72006200	T	0158
	72006300	T	0164
	72006400	T	0169
	72006500	T	0170
	72006600	T	0171
	72006700	T	0171
	72006750	T	0177
	72006760	T	0178
	72006800	T	0185
	72006850	T	0188
	72006900	T	0189
	72007000	T	0189
START OF SEGMENT ***** 121			
121 IS	4 LONG,	NEXT SEG	119
	72007100	T	0190
119 IS	201 LONG,	NEXT SEG	8

EXIT: END OF SCHEDULE;

```

*****!
PROCEDURE SCHEDSTATUS (LCW,A); BOOLEAN LCW; ARRAY A[0];
*****!!
BEGIN
REAL T;

```

73000000	T	0061
73000100	T	0061
73000200	T	0061
73000300	T	0061
73000400	T	0061

Moore Business Forms, Inc. 27

START OF SEGMENT ***** 122

```

LABEL EXIT;
%.....
REAL STREAM PROCEDURE STMESS(W,N,E1,E2); VALUE E1,E2;
%.....
  BEGIN LOCAL DV; LABEL ERR,SCH,RUN,EXT,DUN;
  DI:=W; CI:=CI+E1; GO ERR; GO ERR; GO SCH; GO RUN; GO DUN;
  DS:=8LIT"ABORTED."; TALLY:=8; GO EXT;
DUN: DS:=5LIT"DONE."; TALLY:=5; GO TO EXT;
SCH: DS:=10LIT"SCHEDULED."; TALLY:=10; GO TO EXT;
RUN: DS:=8LIT"RUNNING("; TALLY:=9; DV:=DI;
  DI:=LOC E2; DS:=7FILL; DI:=DV; SI:=LOC E2;
  8(IF SC="" THEN SI:=SI+1 ELSE
  BEGIN DS:=CHR; TALLY:=TALLY+1; END);
  DS:=LIT")"; GO TO EXT;
ERR: DS:=4LIT"ERR!"; SI:=N; SI:=SI+1; DS:=7CHR; TALLY:=11;
EXT: STMESS := TALLY;
  END STREAM PROCEDURE STMESS;

```

```

73002100 T 0017
73002200 T 0018
73002300 T 0018
73002400 T 0018
73002500 T 0022
73002600 T 0031
73002700 T 0033
73002800 T 0033
73002900 T 0046
73003000 T 0048
73003100 T 0048
73003200 T 0048
73003300 T 0048
73003400 T 0049
73003500 T 0050
73003600 T 0052
73003700 T 0052
73003800 T 0054
73003900 T 0059
73004000 T 0061
73004100 T 0061
73004200 T 0062

```

```

CASE DOING OF
BEGIN
  BEGIN % CASE 0
  IF (T:=A[SREG]=BASE=A[BASE]),FF NEQ 1 OR
  A[INC(SREG)],[3:3] = 0 THEN GOERR("NAME ",6);
  A[SREG],[1:5] := 0;
  MATCH(VERBTABLE[T],[9:9]],[6:42],"STATUS ")
  SCHEDSTAT(A[SREG]) ELSE SCHEDSTOP(A[SREG]);
  WAITFOR := IAM; DOING := 1;
  END CASE 0;
%.....
  BEGIN % CASE 1
  IF EVENTS[1] LEQ 1 THEN
  BEGIN
  MSGPOINTER := EVENTS[1] + 36;
  END;
  WRITEASTERISK:=FALSE;
  T:=STMESS(WORK[0],A[SREG],EVENTS[1],DEC CONV(EVENTS[2]));
  TWXOUT(LL,WORK[0],T,1);
  IAM := 0;
  END CASE 1;
  END ALL CASES;

```

START OF SEGMENT ***** 123

```

EXIT;
  END OF SCHEDSTAT;

```

```

123 IS 3 LONG, NEXT SEG 122
73004300 T 0062
73004400 T 0063
122 IS 66 LONG, NEXT SEG 8

```

```

%*****
PROCEDURE LISTFILE(LCW,A); BOOLEAN LCW; ARRAY A[0];
%*****
BEGIN
74000000 T 0061
74000100 T 0061
74000200 T 0061
74000300 T 0061

```

DEFINE

UNK = SPECS,[1:1]#, BAS = SPECS,[2:1]#, ALG = SPECS,[3:1]#,
COB = SPECS,[4:1]#, FTR = SPECS,[5:1]#, TSP = SPECS,[6:1]#,
XAL = SPECS,[7:1]#, SEQ = SPECS,[8:1]#, DAT = SPECS,[9:1]#,
LOK = SPECS,[10:1]#, INF = SPECS,[11:1]#, C68 = SPECS,[12:1]#;

DEFINE

SRC = SPECS,[47:1]#, OBJ = SPECS,[46:1]#, LKD = SPECS,[45:1]#,
UNL = SPECS,[44:1]#, SOL = SPECS,[43:1]#, PUB = SPECS,[42:1]#;

BOOLEAN FROM, TOO, SPECS, LITERAL, HAVELITERAL, HAVEDOUTPUT,

HAVEUSER;

REAL I, PC, WORD, NAMECOUNT, TYPE;

LABEL RUNIT, GETOBJECT, ERR, EXIT;

%.....

STREAM PROCEDURE HEADING(WORK);

%.....

BEGIN

DI:=WORK; DS:=39LIT"NAME TYPE RECS SEGS CREATED ";

DS:=41LIT"ACCESSED W/R W/B S=F LOCKD BY ";

END STREAM PROCEDURE HEADING;

%.....

STREAM PROCEDURE OUTFORMAT

(WORK,FILNAM,FILTYPR,RECS,SEGS,CDAT,ADAT,WPR,WPB,SAVFAC,SECNAM,AFLG);

%.....

VALUE FILNAM,FILTYPR,RECS,SEGS,CDAT,ADAT,WPR,WPB,SAVFAC,SECNAM,AFLG;

BEGIN LOCAL DV;

DI:=WORK; SI:=LOC FILNAM;

2(SI:=SI+1; DS:=7CHR; DS:=LIT" "); % FILE NAME AND TYPE

2(DS:=5DEC; DV:=DI; DI:=DI-5; DS:=4FILL; DI:=DV); % RECS AND SEGS

DS:=2LIT" "; DS:=8CHR; % CREATION DATE

DS:=3LIT" "; AFLG(DI:=DI-3; DS:=3LIT" * ");

DS:=8CHR; % ACCESSED DATE

3(DS:=5DEC; DV:=DI; DI:=DI-5; DS:=4FILL; DI:=DV);

SI:=SI+1; DS:=2LIT" "; DS:=7CHR; DS:=9LIT" "; % SECURITY NAME

END STREAM PROCEDURE OUTFORMAT;

%.....

PROCEDURE PRINTIT(FILNAM,FILTYPR);VALUE FILNAM,FILTYPR;REAL FILNAM,FILTYPR;

%.....

BEGIN REAL I, SECNAM; LABEL SUMM;

SECNAM := IF BOOLEAN(WORK[5],[1:1]) THEN WORK[5],[6:42] % LOCKED

ELSE IF WORK[5]=12 AND WORK[6] NEQ 12 THEN "PUBLIC "

ELSE IF WORK[5]=12 AND WORK[6]=12 THEN "UNLOCKD"

ELSE " ";

FOR I:=10 STEP 1 UNTIL 29 DO IF WORK[I]=0 THEN GO TO SUMM;

SUMM:

OUTFORMAT(L,FILNAM,FILTYPR,(WORK[7]+1),((I-10)*WORK[8]),

DATECONV(WORK[3],[30:18]),DATECONV(WORK[3],[12:18]),

WORK[0],[1:14],WORK[0],[15:15],WORK[3],[2:10],SECNAM,FALSE);

74000400 T 0061
START OF SEGMENT ***** 124

74000500 T 0000
74000600 T 0000
74000700 T 0000
74000800 T 0000
74000900 T 0000
74001000 T 0000
74001100 T 0000
74001200 T 0000
74001300 T 0000
74001400 T 0000
74001500 T 0000
74002900 T 0000
74003000 T 0000
74003100 T 0000
74003200 T 0000
74003300 T 0000
74003400 T 0005
74003500 T 0011

74003600 T 0011
74003700 T 0011
74003800 T 0011
74003900 T 0011
74004000 T 0011
74004100 T 0011
74004200 T 0012
74004300 T 0012
74004400 T 0014
74004500 T 0015
74004600 T 0016
74004700 T 0019
74004800 T 0019
74004900 T 0021
74005000 T 0024

74005100 T 0024
74005200 T 0024
74005300 T 0024
74005400 T 0024
START OF SEGMENT ***** 125

74005500 T 0000
74005600 T 0001
74005610 T 0006
74005620 T 0010
74005700 T 0011
74005710 T 0019
74005800 T 0020
74005900 T 0024
74006000 T 0027

```
TWXOUT(LL,L[0],72,1);
END PROCEDURE PRINTIT;
```

```
74006100 T 0030
74006200 T 0032
125 IS 35 LONG, NEXT SEG 124
```

```
*.....
REAL STREAM PROCEDURE CHK(FIELD); VALUE FIELD;
*.....
BEGIN % COUNT NUMBER OF BITS ON IN FIELD
SI:=LOC FIELD; SI:=SI+5;
18(IF SB THEN TALLY:=TALLY+1; SKIP SB);
CHK:=TALLY;
END STREAM PROCEDURE CHK;
```

```
74006300 T 0024
74006400 T 0024
74006500 T 0024
74006600 T 0024
74006700 T 0025
74006800 T 0025
74006900 T 0027
74007000 T 0027
```

```
*.....
ALPHA PROCEDURE GETTYPE (FILETYPE); VALUE FILETYPE; INTEGER FILETYPE;
BEGIN
IF FILETYPE GTR NUMOFTYPES THEN FILETYPE:=0; %0506=
GETTYPE:=IF BOOLEAN (WORK [4],[10;1]) THEN "OBJ( ) " & %0506=
TYPETABLE [FILETYPE] [30;6;6] ELSE TYPETABLE [FILETYPE]; %0506=
END OF GETTYPE; %0506=
```

```
74007010 C 0028
74007020 C 0028
74007030 C 0028
74007040 C 0028
74007050 C 0031
74007060 C 0032
74007070 C 0035
```

```
*.....
BOOLEAN STREAM PROCEDURE MASK(BIT); VALUE BIT;
BEGIN
DI:=LOC MASK; BIT(SKIP DB); DS:=SET;
END MASK;
```

```
74007100 T 0038
74007110 T 0038
74007120 T 0038
74007130 T 0039
74007140 T 0041
```

```
*.....
CASE DOING OF
BEGIN
BEGIN % CASE 0 = SYNTAX ANALYSIS
LABEL DUMMY;

PC := COMMONCELL; % TRANSFERRED FROM PROCEDURE "LISTIT"
CLEAR(WORK,29); % ZERO OUT WORK ARRAY
WORK[2]:=WORK[4]:=USERCODE; WORK[6]:= "TELETYP"; % DEFAULT OPTIONS
NAMECOUNT := 20; % POINTER FOR FILE NAMES
WHILE I LSS PC DO
BEGIN
I := I + 1;
IF (WORD:=A[SREG:=SREG+1]),[1;2]=1 THEN
IF WORD="/" THEN WORD:="FROM " & 4[3;45;3];
IF WORD,[3;3]=0 THEN GO ERR ELSE WORD,[1;5]=0;
MATCH(WORD,"LITERAL")
IF HAVELITERAL THEN GO ERR ELSE LITERAL:=TRUE
```

```
74007150 T 0042
74007200 T 0042
74007300 T 0042
74007400 T 0042
74007410 T 0042
START OF SEGMENT ***** 126
74007500 T 0000
74007600 T 0001
74007700 T 0002
74007800 T 0006
74007900 T 0006
74008000 T 0008
74008100 T 0008
74008200 T 0009
74008300 T 0012
74008400 T 0016
74008500 T 0024
74008600 T 0024
```

```

ELSE MATCH(WORD,"TO ")
  IF HAVEOUTPUT THEN GO ERR ELSE TOO:=TRUE
ELSE MATCH(WORD,"FROM ")
  IF HAVEUSER THEN GO ERR ELSE
    BEGIN
      FROM:=TRUE;
      IF I + 2 LEQ PC THEN FIXUSER (I,A,EXIT,TRUE);
    END
ELSE IF TOO THEN
  IF HAVEOUTPUT THEN GO ERR ELSE
    BEGIN WORK[6]:=WORD; TOO:=NOT(HAVEOUTPUT:=TRUE); END
ELSE IF FROM THEN
  IF HAVEUSER THEN GO ERR ELSE
    BEGIN WORK[4]:=WORD; FROM:=NOT(HAVEUSER:=TRUE); END
ELSE IF LITERAL THEN
  IF HAVELITERAL THEN GO ERR ELSE
    BEGIN WORK[5]:=WORD; LITERAL:=NOT(HAVELITERAL:=TRUE); END
ELSE MATCH(WORD,"PRINTER")
  IF HAVEOUTPUT THEN GO ERR ELSE
    BEGIN WORK[6]:=WORD; HAVEOUTPUT :=TRUE; END
ELSE IF TYPE:=FINDAWORD(WORD&7[1:43:5],RESWRDTABLE[0],18) # 0 THEN
  SPECS:=SPECS OR MASK(TYPE+29xREAL(TYPE GTR 12))
ELSE % ASSUME IT IS A FILE NAME
  BEGIN
    IF (NAMECOUNT:=NAMECOUNT+1) GTR 29 THEN GOERR("TOOMANY",30);
    WORK[NAMECOUNT] := WORD; % SAVE FILE NAME
  END;
END WHILE STATEMENT;
IF FROM THEN GOERR("FROM ",8);
IF TOO THEN GOERR("TO ",8);
IF LITERAL THEN GOERR("LITERAL",8);
IF CHK(SPECS,[1:12]) GTR 1 THEN GOERR("FILTYPE",31);
IF CHK(SPECS,[42:44]) GTR 1 THEN GOERR("SECURTY",8);
IF CHK(SPECS,[46:48]) GTR 1 THEN GOERR("SRC=OBJ",8);
TEMR := 7;
IF (NAMECOUNT:=NAMECOUNT-21) = 0 THEN % ONE FILENAME SPECIFIED
IF (PREVRCW=BASE) GTR 41 THEN % ENOUGH ROOM IN STACK
  BEGIN % LET CANDE DO THE I/O
    A[BASE] := 7; % LOOP COUNTER
    A[BASE+1] := WORK[21]; % SAVE FILE NAME IN STACK
    A[BASE+7] := WORK[4]; % SAVE USERNAME
    FIND(LL&IAM[33:41:7],0,WORK[21],WORK[4],USERCODE); % SRC
    FIND(LL&IAM[33:41:7],0,WORK[21],[6:36],WORK[4],USERCODE); % OBJ
    DOING := 2;
    WAITFOR := IAM;
    GO TO EXIT;
  END;
WORK[3] := REAL(SPECS);
WORK[7] := NAMECOUNT;
DISKWRITE(ESP1);
IF HAVEOUTPUT THEN % OUTPUT MEDIA SPECIFIED
MATCH(WORK[6],"PRINTER") ELSE
MATCH(WORK[6],"TELETYPE") ELSE
  BEGIN % DISK FILE NAME
    FIND(LL&IAM[33:41:7],0,WORK[6],USERCODE,USERCODE);
    A[BASE+1] := WORK[6]; % SAVE OUTPUT FILE NAME
    DOING := 1;

```

```

74008700 T 0032
74008800 T 0032
74008900 T 0040
74009000 P 0040
74009005 C 0046
74009010 C 0048
74009020 C 0048
74009095 C 0053
74009100 T 0053
74009200 T 0054
74009300 T 0057
74009400 T 0061
74009500 T 0061
74009600 T 0065
74009700 T 0068
74009800 T 0069
74009900 T 0072
74010000 T 0076
74010100 T 0076
74010200 T 0081
74010300 T 0085
74010400 T 0088
74011900 T 0091
74012000 T 0093
74012100 T 0093
74012200 T 0102
74012300 T 0103
74012400 T 0103
74012500 T 0105
74012600 T 0112
74012700 T 0119
74012800 T 0126
74012900 T 0135
74013000 T 0144
74013100 T 0153
74013200 T 0153
74013300 T 0155
74013400 T 0157
74013500 T 0157
74013600 T 0159
74013700 T 0161
74013800 T 0163
74013900 T 0167
74014000 T 0171
74014100 T 0172
74014200 T 0173
74014300 T 0175
74014400 T 0175
74014500 T 0176
74014600 T 0177
74014700 T 0181
74014800 T 0181
74014900 T 0184
74015000 T 0194
74015100 T 0196
74015200 T 0200
74015300 T 0202

```

Academy Business Systems, Inc. by 11/27/77

```

        WAITFOR I= IAM;
        GO TO EXIT;
    END;
GO RUNIT; % TO PRINTER OR TELETYPE
END; % CASE 0

*.....
BEGIN % CASE 1 = SEARCH FOR OUTPUT FILE ON DISK
IF EVENTS[1] GEQ 0 THEN GOERR(A[BASE+1],3); % ALREADY PRESENT
RUNIT;
    RUN1 := "LFILES "; RUN2 := "CANDE "; ERRSW := USERCODE;
    IAM := VDISPATCH;
    EDJMSG := NOT CONCISE;
    WRITEASTERISK := CONCISE;
    GO TO EXIT;
END; % CASE 1

*.....
BEGIN % CASE 2 = SEARCH FOR FILE NAME
I := A[BASE] := A[BASE] - 2;
A[BASE+I] := EVENTS[1]; A[BASE+I+1] := EVENTS[2],[18:15]; % HDR ADRS,
IF I GTR 3 THEN % WAIT FOR OBJECT FILE RESULT
    BEGIN
        WAITFOR I= IAM;
        GO TO EXIT;
    END;
IF A[BASE+3] LEQ 0 AND A[BASE+5] LEQ 0 THEN GOERR(A[BASE+1],2);
TIMEOFDAY(WORK[2],TIMECONV(TIME(1),TRUE));
WORK[0] := DATE; WORK[1] := A[BASE+7]; BLANK(WORK[1]); % USERNAME
TWXOUT(LL,WORK[0],26,1);
HEADING(WORK); TWXOUT(LL,WORK[0],72,1);
WRITEASTERISK := TRUE;
IF A[BASE+5] LEQ 0 THEN GO GETOBJECT; % SOURCE VERSION NOT ON DISK
LLDISKREAD := A[BASE+6];
WANT := VDISKREAD;
DOING := 3;
GO EXIT;
END; % CASE 2

*.....
BEGIN % CASE 3 = SOURCE FILE HEADER IN CORE
PRINTIT (A [BASE + 1],GETTYPE (WORK [4],[36:6])); %0506=
IF A[BASE+3] LEQ 0 THEN
    BEGIN
        IAM := 0; GO TO EXIT;
    END;
GETOBJECT;
LLDISKREAD := A[BASE+4];
WANT := VDISKREAD;
DOING := 4;
GO TO EXIT;
END; % CASE 3

*.....
BEGIN % CASE 4 = OBJECT HEADER IN CORE
PRINTIT (A [BASE + 1],GETTYPE (WORK [4],[36:6])); %0506=
IAM := 0;
GO TO EXIT;
END CASE 4;
END ALL CASES;

```

```

74015400 T 0202
74015500 T 0203
74015600 T 0205
74015700 T 0205
74015800 T 0208
126 IS 209 LONG, NEXT SEG 124
74015900 T 0044
74016000 T 0044
74016100 T 0044
74016200 T 0051
74016300 T 0052
74016400 T 0055
74016500 T 0056
74016600 T 0059
74016700 T 0062
74016800 T 0065
74016900 T 0065
74017000 T 0065
74017100 T 0065
74017200 T 0068
74017300 T 0073
74017400 T 0074
74017500 T 0074
74017600 T 0075
74017700 T 0075
74017800 T 0075
74017900 T 0084
74018000 T 0087
74018100 T 0091
74018200 T 0093
74018300 T 0096
74018400 T 0098
74018500 T 0100
74018600 T 0102
74018700 T 0103
74018800 T 0103
74018900 T 0104
74019000 T 0104
74019100 T 0104
74019300 P 0104
74019400 T 0107
74019500 T 0109
74019600 T 0109
74019700 T 0111
74019800 T 0111
74019900 T 0111
74020000 T 0113
74020100 T 0113
74020200 T 0114
74020300 T 0115
74020400 T 0115
74020500 T 0115
74020800 P 0115
74020900 T 0118
74021000 T 0119
74021100 T 0119
74021200 T 0120

```

```
ERR: ERRSW := A[REG]; MSGPTR := 8; IAM := VERROR;
EXIT;
END PROCEDURE LISTFILES;
```

```
START OF SEGMENT ***** 127
127 IS 6 LONG, NEXT SEG 124
74021300 T 0120
74021400 T 0124
74021500 T 0124
124 IS 131 LONG, NEXT SEG 8
```

```
*****
PROCEDURE UPDATE(LCW,A); BOOLEAN LCW; ARRAY A[0];
*****
BEGIN
```

```
75000000 T 0061
75000100 T 0061
75000200 T 0061
75000300 T 0061
%0405= 75000350 P 0061
START OF SEGMENT ***** 128
```

```
    LABEL EXIT;
    CASE DOING OF
    BEGIN
    IF FILENAME=0 THEN GOERR ("NO FILE",1) ELSE % CASE 0
    IF TPCOUNT=0 THEN IAM:=0 ELSE % CASE 0.
    BEGIN % CASE 0
    WORK[0] := DOGHOUSE[LL] := 0;
    WORK[1] := (LL+SYSTEM)&R(FILETYPE=8)[3:47:1]
    &R(MONITORBIT)[9:47:1];
    WORK[2]:=USERCODE; WORK[4]:=USERLOC;
    WORK[3] := SOURCEFILE;
    MOVE(16,WORDS,CTRANDBASE[0],WORK[5]);
    ERRSW := USERCODE;
    RUN1 := "LIST "; RUN2 := "CANDE ";
    WANT := VCLOSEWORKTABLE; DOING := 1;
    DISKWRITE(ESP1);
    END CASE 0;
    BEGIN % CASE 1
    WANT := VDISPATCH; DOING := 2; ERRSW := USERCODE;
    UPDATEBIT := 1; % MARK UPDATE IN PROGRESS
    END CASE 1;
    BEGIN % CASE 2
    IF EVENTS[1] = 1 THEN % EOJ OK
    BEGIN
    SOURCEFILE := MAKEFN("IS",LL);
    CANCELCKPT(A[*]);
    END;
    IAM:=0; IF BREAKORWRU AND NOT DOGLICENSE THEN LINECLEAR(LCW,A[*]);
    END CASE 2;
    END ALL CASES;
```

```
%0405= 75000400 T 0000
75000500 T 0000
75000550 C 0000
75000600 T 0008
75000700 T 0011
75000800 T 0011
75000900 T 0014
75000910 T 0017
75001000 T 0019
75001100 T 0022
75001200 T 0023
75001300 T 0025
75001400 T 0026
75001500 T 0029
75001600 T 0030
75001700 T 0034
75001800 T 0034
75001900 T 0034
75001910 T 0037
75002000 T 0039
75002100 T 0040
75002200 T 0040
75002210 T 0041
75002220 T 0041
75002300 T 0043
75002310 T 0045
75002320 T 0045
75002400 T 0050
75002500 T 0050
```

```
EXIT;
END CLEANWORKFILE;
```

```
START OF SEGMENT ***** 129
129 IS 4 LONG, NEXT SEG 128
%0405= 75002550 C 0051
75002600 T 0051
128 IS 34 LONG, NEXT SEG 8
```

```
*****
REAL PROCEDURE SPECSCAN
(IADR,OADR,SIZ,VOCAB,VSIZE,WHICH,FTOG,EADRS);
```

```
76000000 T 0061
76000100 T 0061
76000110 T 0061
```

```

VALUE VSIZE,FTOG,EADRS;
REAL IADR,OADR,SIZ,VSIZE,WHICH,FTOG,EADRS;
ARRAY VOCAB[0];
BEGIN
LABEL L;

REAL STREAM PROCEDURE SPCSCAN
(IADR,OADR,SIZ,VOCAB,VSIZE,WHICH,FTOG,EADRS);
VALUE VSIZE,FTOG,EADRS;
BEGIN LOCAL T,SV,DV,COMA; LABEL NEXT,LETTER,NUMBR,FIIL,EXIT;
COMMENT SPECIAL SCANNER FOR "FIND/REPLACE" INSTRUCTIONS.
RETURNS A VALUE ( TO SPCSCAN ) OF:
1 = "INSTRUCTION"
2 = DELIMITED STRING
3 = SEQUENCE NUMBER,
4 = SEQUENCE NUMBER FOLLOWED BY "-" ( SEQUENCE RANGE ),
5 = END OF RECORD WITH CONTINUATION CHARACTER (",")
6 = END OF STATEMENT
7 = FILE NAME
10= ERROR (MISSING STRING DELIMETER OR STRING TOO LONG)
11= ERROR (FIND/REPLACE "INSTRUCTION" TOO LONG)
12= ERROR (FIND/REPLACE "INSTRUCTION" NOT RECOGNIZED)
13= ERROR (SEQUENCE NUMBER TOO LONG)
14=MISSING FILE NAME
POSITION OF "INSTRUCTION" RETURNED TO "WHICH":
0 = "FIND"
1 = "REPLACE"
2 = "FIRST"
3 = "LITERAL"
4 = "WITH"
5 = "ARROW"
6 = "PRINT"
7 = "END"
8 = "SEQUENCE"
9 = "TEXT"
10= "FILE"
11= "SITE"
END OF COMMENT;
SI:=IADR; SI:=SI+5; SII:=SC; TALLY:=6; COMA:=TALLY;
NEXT: IF SC=" " THEN BEGIN SI:=SI+1; GO NEXT; END; % SKIP OVER BLANKS
IF SC=LEFTARROW THEN % END OF RECORD
BEGIN
SV:=SI; SI:=LOC SV; SII:=SI+5;
DVI:=DI; DI:=LOC EADRS; DI:=DI+5;
IF 3 SC=DC THEN % END OF RECORD
BEGIN
TALLY:=COMA; SPCSCAN:=TALLY; GO TO EXIT;
END;
SI:=SV; DI:=DV;
END;
IF SC="," THEN % COULD BE A CONTINUATION CHARACTER
BEGIN TALLY:=5; COMA:=TALLY; SI:=SI+1; GO TO NEXT; END;
TALLY:=6; COMA:=TALLY; % NOT A CONTINUATION
FTOG(JUMP OUT TO FIIL); % JUMP TO FILENAME SCANNER
IF SC=ALPHA THEN % COULD BE ALPHABET, DIGIT OR "QMARK"
BEGIN IF SC LSS "0" THEN GO TO LETTER;
IF SC GTR "9" THEN ELSE GO TO NUMBR;

```

```

76000200 T 0061
76000300 T 0061
76000310 T 0061
76000320 T 0061
76000330 T 0061
START OF SEGMENT ***** 130
76000340 T 0000
76000350 T 0000
76000360 T 0000
76000400 T 0000
76000500 T 0000
76000600 T 0000
76000700 T 0000
76000800 T 0000
76000900 T 0000
76001000 T 0000
76001100 T 0000
76001200 T 0000
76001210 T 0000
76001300 T 0000
76001400 T 0000
76001500 T 0000
76001600 T 0000
76001610 T 0000
76001700 T 0000
76001800 T 0000
76001900 T 0000
76002000 T 0000
76002100 T 0000
76002200 T 0000
76002300 T 0000
76002400 T 0000
76002500 T 0000
76002600 T 0000
76002700 T 0000
76002710 T 0000
76002720 T 0000
76002800 T 0000
76002900 T 0000
76003000 T 0001
76003100 T 0003
76003200 T 0004
76003210 T 0004
76003220 T 0005
76003230 T 0006
76003240 T 0006
76003250 T 0006
76003260 T 0008
76003270 T 0008
76003280 T 0008
76003300 T 0008
76003400 T 0009
76003500 T 0010
76003550 T 0011
76003600 T 0013
76003700 T 0013
76003800 T 0014

```



```

END IF ALPHA;
*.....
% MUST BE A DELIMITER
DI:=LOC T; DI:=DI+7; DS:=CHR; % SAVE THE DELIMITER
SV:=SI; SI:=LOC SV; DI:=QADR; DS:=WDS; % STARTING ADDRESS
SI:=SV; TALLY:=10; SPCSCAN:=TALLY; TALLY:=0;
63(IF SC=LEFTARROW THEN
  BEGIN
    SV:=SI; SI:=LOC SV; SI:=SI+5; DI:=LOC EADRS; DI:=DI+5;
    IF 3SC=DC THEN JUMP OUT TO EXIT ELSE SI:=SV;
  END;
IF SC=T THEN JUMP OUT; % 2ND DELIMITER
  TALLY:=TALLY+1; SI:=SI+1);
IF TOGGLE THEN ELSE % TOO LONG
  BEGIN TALLY:=10; SPCSCAN:=TALLY; GO EXIT; END;
SI:=SI+1; SV:=SI; % SKIP OVER 2ND DELIMITER, SAVE ADDRESS
T:=TALLY; SI:=LOC T; DI:=SIZ; DS:=WDS; % SIZE
SI:=LOC SV; DI:=IADR; DS:=WDS; % NEXT ADDRESS
TALLY:=2; SPCSCAN:=TALLY; % CODE FOR DELIMITED STRING
GO TO EXIT;
*.....
LETTER: % SHOULD BE AN "INSTRUCTION"
SV:=SI; TALLY:=0; % SAVE STARTING ADDRESS
SI:=LOC SV; DI:=QADR; DS:=WDS; SI:=SV;
9(IF SC LEQ "9" THEN ELSE JUMP OUT;
  IF SC=ALPHA THEN ELSE JUMP OUT; SI:=SI+1; TALLY:=TALLY+1);
IF TOGGLE THEN % TOO LONG
  BEGIN TALLY:=11; SPCSCAN:=TALLY; GO EXIT; END;
DV:=SI; SI:=LOC DV; DI:=IADR; DS:=WDS; % NEXT ADDRESS
T:=TALLY; SI:=LOC T; SI:=SI+7; % SIZE OF "INSTRUCTION"
IF SC LSS "3" THEN % 3 CHARACTERS MINIMUM
  BEGIN TALLY:=12; SPCSCAN:=TALLY; GO EXIT; END;
% NOW CHECK INPUT STRING AGAINST VOCABULARY
DI:=VOCAB; DV:=DI; TALLY:=0;
VSIZE(SI:=SV; IF T SC = DC THEN JUMP OUT;
  DI:=DV; DI:=DI+8; DV:=DI; TALLY:=TALLY+1);
IF TOGGLE THEN ELSE % NO MATCH
  BEGIN TALLY:=12; SPCSCAN:=TALLY; GO EXIT; END;
T:=TALLY; SI:=LOC T; DI:=WHICH; DS:=WDS;
TALLY:=1; SPCSCAN:=TALLY; % CODE FOR "INSTRUCTION";
GO TO EXIT;
*.....
NUMB: % SHOULD BE A SEQUENCE NUMBER
SV:=SI; SI:=LOC SV; DI:=QADR; DS:=WDS; % STARTING ADDRESS
SI:=SV; TALLY:=0;
9(IF SC GTR "9" THEN JUMP OUT;
  IF SC LSS "0" THEN JUMP OUT; SI:=SI+1; TALLY:=TALLY+1);
IF TOGGLE THEN ELSE % TOO LONG
  BEGIN TALLY:=13; SPCSCAN:=TALLY; GO EXIT; END;
T:=TALLY; 63(IF SC=" " THEN SI:=SI+1 ELSE JUMP OUT);
IF SC="*" THEN % SEQUENCE RANGE SPECIFIED
  BEGIN SI:=SI+1; TALLY:=4; END ELSE TALLY:=3;
SV:=SI; SI:=LOC SV; DI:=IADR; DS:=WDS; % NEXT ADDRESS
SI:=LOC T; DI:=SIZ; DS:=WDS; % SIZE OF FIELD
SPCSCAN:=TALLY; GO TO EXIT;
*.....
FIIL: % LOOKING FOR A FILE NAME

```

```

76003900 T 0015
76004000 T 0015
76004100 T 0015
76004200 T 0015
76004300 T 0016
76004400 T 0017
76004500 T 0018
76004510 T 0019
76004520 T 0019
76004530 T 0021
76004550 T 0022
76004600 T 0022
76004700 T 0024
76004800 T 0025
76004900 T 0025
76005000 T 0026
76005100 T 0026
76005200 T 0027
76005300 T 0028
76005400 T 0029
76005500 T 0029
76005600 T 0029
76005700 T 0030
76005800 T 0030
76005900 T 0031
76006000 T 0033
76006100 T 0035
76006200 T 0035
76006300 T 0036
76006400 T 0037
76006500 T 0038
76006600 T 0038
76006700 T 0040
76006800 T 0040
76006900 T 0040
76007000 T 0043
76007100 T 0045
76007200 T 0045
76007300 T 0046
76007400 T 0047
76007500 T 0047
76007600 T 0048
76007700 T 0048
76007800 T 0048
76007900 T 0049
76008000 T 0049
76008100 T 0050
76008200 T 0053
76008300 T 0053
76008400 T 0054
76008500 T 0057
76008600 T 0057
76008700 T 0059
76008800 T 0060
76008900 T 0060
76008905 T 0061
76008910 T 0061

```

```

SV:=SI; SI:=LOC SV; DI:=OADR; DS:=WDS; % START OF STRING
SI:=SV; TALLY:=0;
IF SC=ALPHA THEN ELSE % MISSING FILE NAME
  BEGIN TALLY:=14; SPCSCAN:=TALLY; GO TO EXIT; END;
7(IF SC=ALPHA THEN ELSE JUMP OUT; SI:=SI+1; TALLY:=TALLY+1);
IF SC="/" THEN % SECOND NAME GIVEN
  BEGIN SI:=SI+1; TALLY:=TALLY+1; % SKIP OVER "/"
  7(IF SC=ALPHA THEN ELSE JUMP OUT; SI:=SI+1; TALLY:=TALLY+1);
  END;
SV:=SI; SI:=LOC SV; DI:=IADR; DS:=WDS; % NEXT ADDRESS
T:=TALLY; SI:=LOC T; DI:=SIZ; DS:=WDS; % SIZE OF FILE NAME
TALLY:=7; SPCSCAN:=TALLY; GO EXIT; % CODE FOR FILE NAME
EXIT;
END STREAM PROCEDURE SPCSCAN;

```

```

76008915 T 0062
76008920 T 0063
76008925 T 0063
76008930 T 0064
76008935 T 0065
76008945 T 0068
76008950 T 0068
76008960 T 0069
76008965 T 0072
76008970 T 0072
76008975 T 0073
76008980 T 0074
76009000 T 0074
76009100 T 0074

```

```

SPCSCAN:=SPCSCAN
(IADR,OADR,SIZ,VOCAB,VSIZ,WHICH,FTOG,EADR);
END PROCEDURE SPCSCAN;

```

```

76009110 T 0076
76009120 T 0076
76009130 T 0081

```

130 IS 84 LONG, NEXT SEG 8

```

*****
PROCEDURE REPLACEIT(LCW,STACK); BOOLEAN LCW; ARRAY STACK[0];
*****
BEGIN
COMMENT ROUTINE TO CHECK SYNTAX AND PACK INFORMATION INTO STACK.
STACK CODE:
  STACK[BASE+1],[01:01] (1 = "FILE " INSTRUCTION RECEIVED)
  STACK[BASE+1],[02:01] (1 = INPUT FILE SPECIFIED )
  STACK[BASE+1],[03:01] (1 = OUTPUT FILE SPECIFIED )
  STACK[BASE+1],[04:01] (1 = "SITE " INSTRUCTION RECEIVED)
  STACK[BASE+1],[05:01] (1 = "FIND " INSTRUCTION RECEIVED)
  STACK[BASE+1],[06:01] (1 = "REPLACE " INSTRUCTION RECEIVED)
  STACK[BASE+1],[07:01] (1 = "FIRST " INSTRUCTION RECEIVED)
  STACK[BASE+1],[08:01] (1 = "LITERAL " INSTRUCTION RECEIVED)
  STACK[BASE+1],[11:01] (1 = "PRINT " INSTRUCTION RECEIVED)
  STACK[BASE+1],[12:01] (1 = "SEQUENCE" INSTRUCTION RECEIVED)
  STACK[BASE+1],[17:01] (1 = "TEXT " INSTRUCTION RECEIVED)
  STACK[BASE+1],[18:01] (1 = REPLACE OR FIND STATEMENT COMPLETE)
  STACK[BASE+1],[19:01] (1 = FIRST REPLACE INSTRUCTION RECVD)
  STACK[BASE+1],[20:04] ( STRING COUNT )
  STACK[BASE+1],[24:24] ( TOTAL SIZE OF STATEMENT ( NO. OF CHARACTERS
4 CHARACTER CODE PRECEEDS EACH "STRING" IN STACK:
  1ST CHARACTER = TYPE OF STRING ( 2, 3 OR 4 )
  2ND CHARACTER = 1 IF "FIRST" ONLY, 0 IF ALL ( FOR REPLACE )
  3RD CHARACTER = 1 IF "LITERAL" COMPARISON, 0 IF NOT
  4TH CHARACTER = SIZE ( NO OF CHARACTERS IN STRING ),
  THE ROUTINE USES MESSAGE/CANDE RECORDS MSGVAL THRU MSGVAL+6
END OF COMMENT;
DEFINE WRD = STACK[BASE+1] #,
      FTOG = WRD,[01:01] #,

```

```

76009200 T 0061
76009300 T 0061
76009400 T 0061
76009500 T 0061
76009600 T 0061
76009700 T 0061
76009710 T 0061
76009720 T 0061
76009730 T 0061
76009740 T 0061
76009800 T 0061
76009900 T 0061
76010000 T 0061
76010100 T 0061
76010200 T 0061
76010300 T 0061
76010400 T 0061
76010500 T 0061
76010600 T 0061
76010700 T 0061
76010800 T 0061
76010900 T 0061
76011000 T 0061
76011100 T 0061
76011200 T 0061
76011300 T 0061
76011400 T 0061
76011500 T 0061
76011600 T 0061
76011610 T 0000

```

START OF SEGMENT ***** 131

INFIL = WRD,[02:01] #;	76011620 T 0000
OUTFIL = WRD,[03:01] #;	76011630 T 0000
SITE = WRD,[04:01] #;	76011640 T 0000
FND = WRD,[05:01] #;	76011700 T 0000
REP = WRD,[06:01] #;	76011800 T 0000
FRST = WRD,[07:01] #;	76011900 T 0000
LIT = WRD,[08:01] #;	76012000 T 0000
PRNT = WRD,[11:01] #;	76012100 T 0000
SEQ = WRD,[12:01] #;	76012200 T 0000
TEXT = WRD,[17:01] #;	76012300 T 0000
OK = WRD,[18:01] #;	76012400 T 0000
FREP = WRD,[19:01] #;	76012500 T 0000
STRGCNT = WRD,[20:04] #;	76012600 T 0000
TSIZ = WRD,[24:24] #;	76012700 T 0000
PREV = STACK[BASE] #;	76012800 T 0000
ARRAY VOCAB[0:18];	76012900 T 0000
DEFINE MSGVAL = 50 #, A = STACK #;	76013000 T 0001
INTEGER X;	76013100 T 0001
REAL IADR, OADR, AROADR, ENDADR, TYPE, SIZ, VSIZ, WHICH;	76013200 T 0001
LABEL EXIT, CASE1;	76013300 T 0001
*****	76013400 T 0001
REAL STREAM PROCEDURE ERRVALUE(OADR); VALUE OADR;	76013500 T 0001
*****	76013600 T 0001
BEGIN % EXTRACT STRING FOR ERROR MESSAGE	76013700 T 0001
SI:=OADR; DI:=LOC ERRVALUE; DS:=8LIT"0"; DI:=DI-7;	76013800 T 0003
?(IF SC=ALPHA THEN DS:=CHR ELSE JUMP OUT);	76013900 T 0005
END STREAM PROCEDURE ERRVALUE;	76014000 T 0007
*****	76014100 T 0008
REAL STREAM PROCEDURE CNVERT(OADR, SIZ); VALUE OADR, SIZ;	76014200 T 0008
*****	76014300 T 0008
BEGIN SI:=OADR; DI:=LOC CNVERT; DS:=SIZ OCT;	76014400 T 0008
END STREAM PROCEDURE CNVERT;	76014500 T 0009
*****	76014600 T 0010
STREAM PROCEDURE CLEARSTACK(STACK, TSIZ); VALUE TSIZ;	76014700 T 0010
*****	76014800 T 0010
BEGIN % ZERO OUT THE STACK TO AVOID POSSIBLE FLAG BIT ERRORS;	76014900 T 0010
LOCAL T1, T2;	76015000 T 0010
SI:=LOC TSIZ; SI:=SI+6;	76015100 T 0010
DI:=LOC T1; DI:=DI+7; DS:=CHR; DI:=LOC T2; DI:=DI+7; DS:=CHR;	76015200 T 0010
DI:=STACK; % SHOULD POINT TO STACK[BASE+2]	76015300 T 0012
T1(2(DS:=32LIT"0")); T2(DS:=LIT"0");	76015400 T 0012
END STREAM PROCEDURE CLEARSTACK;	76015500 T 0020
*****	76015600 T 0020
STREAM PROCEDURE SETL(L, AROADRS, ENDADRS);	76015700 T 0020
*****	76015800 T 0020

```

BEGIN LOCAL SV;
DI:=L; DS:=8LIT"9"; DS:=LIT LEFTARROW;
DS:=7LIT"0"; DS:=LIT "x";
DS:=7LIT"0"; DS:=LIT "=";
DS:=7LIT"0"; DS:=LIT "<";
DS:=7LIT"0"; DS:=LIT ">";
DS:=7LIT"0"; DS:=LIT "≤";
DS:=7LIT"0"; DS:=LIT "≥";
SI:=L; SV:=SI; SI:=LOC SV; DI:=ENDADRS; DS:=WDS;
SI:=SV; SI:=SI+8; SV:=SI; SI:=LOC SV; DI:=AROADRS; DS:=WDS;
END STREAM PROCEDURE SETL;

```

```

76015900 T 0020
76016000 T 0021
76016010 T 0023
76016020 T 0024
76016030 T 0026
76016040 T 0028
76016050 T 0030
76016060 T 0031
76016100 T 0033
76016200 T 0034
76016300 T 0036

```

```

*****
STREAM PROCEDURE SAVEIT(STACK,OLDSIZ,OADR,ADDSIZ,TYPE,NEWSIZ);
*****
VALUE OLDSIZ,OADR,ADDSIZ,TYPE,NEWSIZ;
COMMENT SAVE STRING AND PREFIX IN STACK;
BEGIN LOCAL T1,T2,LT;
SI:=LOC OLDSIZ; SI:=SI+6; % CHARACTERS TO SKIP OVER
DI:=LOC T1; DI:=DI+7; DS:=CHR; DI:=LOC T2; DI:=DI+7; DS:=CHR;
DI:=STACK; DI:=DI+16; % SKIP OVER 1ST 2 WORDS
T1(2(DI:=DI+32)); DI:=DI+T2; % SKIP OVER PREVIOUS STRINGS
SI:=LOC TYPE; SI:=SI+7; DS:=CHR; % STRING TYPE ( 2,3,4 OR 7 )
SI:=STACK; SI:=SI+9;
SKIP SB; IF SB THEN DS:=LIT"1" ELSE DS:=LIT"0"; % "FIRST" CODE
LT:=DI; SKIP SB; IF SB THEN DS:=LIT"1" ELSE DS:=LIT"0"; % "LITERAL"
SI:=LOC ADDSIZ; SI:=SI+7; DS:=CHR; % SIZE OF STRING
SI:=OADR; DS:=ADDSIZ CHR; % TRANSFER STRING
SI:=LOC NEWSIZ; SI:=SI+4; DI:=STACK; DI:=DI+12; DS:=4 CHR;
% CHECK FOR ALL BLANKS IN TYPE 1 STRINGS
SI:=LOC TYPE; SI:=SI+7; IF SC="1" THEN
BEGIN SI:=OADR; ADDSIZ(IF SC=" " THEN SI:=SI+1 ELSE JUMP OUT);
IF TOGGLE THEN % ALL BLANKS, MUST BE IN "LITERAL" MODE
BEGIN DI:=LT; DS:=LIT"1"; END;
END;
END STREAM PROCEDURE SAVEIT;

```

```

76016400 T 0036
76016500 T 0036
76016600 T 0036
76016700 T 0036
76016800 T 0036
76016900 T 0036
76017000 T 0037
76017100 T 0037
76017200 T 0039
76017300 T 0039
76017400 T 0042
76017500 T 0042
76017600 T 0043
76017700 T 0045
76017800 T 0047
76017900 T 0048
76018000 T 0049
76018010 T 0050
76018020 T 0050
76018030 T 0051
76018040 T 0055
76018050 T 0055
76018060 T 0056
76018100 T 0056

```

```

*****
STREAM PROCEDURE SETVOCAB(VOCAB);
*****
BEGIN DI:=VOCAB; DS:=32LIT"FIND REPLACE FIRST LITERAL ";
DS:=48LIT"WITH ARROW PRINT END SEQUENCETEXT ";
DS:=16LIT"FILE SITE ";
DS:=48LIT"NEQ EQL LSS GTR LEQ GEQ ";
END STREAM PROCEDURE SETVOCAB;

```

```

76018200 T 0056
76018300 T 0056
76018400 T 0056
76018500 T 0056
76018600 T 0061
76018610 T 0067
76018620 T 0070
76018700 T 0076

```

```

*****
CASE DOING OF

```

```

76018800 T 0076
76019100 T 0076

```

```

BEGIN
BEGIN % CASE 0
LABEL LOOP,STRING,ERROR;

IF NOT CONTINUEBIT THEN % FIRST CALL ON THE ROUTINE
BEGIN STACK[BASE+1]=0; CONTINUEBIT=TRUE; PREV=-1; END;
IF PREVRCH = BASE LEQ 34 THEN % NOT ENOUGH ROOM
BEGIN CLEARSTACK(STACK[BASE+2],TSIZ);
GOERR("NO ROOM",MSGVAL);
END;
IADR:=ADDRESS(WORK[1]); % ADDRESS OF INPUT RECORD
SETVOCAB(VOCAB); VSIZ:=18; % SET UP VOCABULARY
SETL(L,AROADR,ENDADR); % LOCATION OF 8 NINES AND LEFT ARROW
LOOP: % START OF SYNTAX CHECK
TYPE:=SPECSCAN(IADR,OADR,SIZ,VOCAB,VSIZ,WHICH,FTOG,WORKEADRS);%SNTX
MSGPTR := MSGVAL ; % "ERROR IN FIND/REPLACE STATEMENT
IF TYPE=1 THEN % AN "INSTRUCTION"
BEGIN IF WHICH LSS 2 THEN % "FIND" OR "REPLACE"
BEGIN IF FND+REP NEQ 0 THEN GO ERROR; % REDUNDANT
IF WHICH=0 THEN IF FREP=1 THEN GO ERROR ELSE FND:=1
ELSE % WHICH = 1
BEGIN REP:=1; FREP:=1; END;
FRST:=0; LIT:=0; OK:=0;
END % IF WHICH LSS 2
ELSE IF WHICH LSS 6 THEN % "FIRST","LITERAL","WITH" OR "ARROW"
BEGIN IF REP+FND NEQ 1 THEN GO ERROR;
IF WHICH LSS 4 THEN % "FIRST" OR "LITERAL"
BEGIN IF STRGCNT NEQ 0 THEN GO TO ERROR;
IF WHICH=2 THEN IF FRST=1 THEN GO TO ERROR ELSE FRST:=1
ELSE IF LIT=1 THEN GO TO ERROR ELSE LIT:=1;
END % IF WHICH LSS 4
ELSE IF WHICH=4 THEN GO TO LOOP % "WITH" (OPTIONAL WORD )
ELSE % WHICH = 5 ( "ARROW" )
BEGIN OADR:=AROADR; SIZ:=1; TYPE:=2; GO TO STRING;
END % IF WHICH = 5
END % IF WHICH LSS 6
ELSE IF WHICH=6 THEN % "PRINT"
BEGIN IF PRNT=1 THEN GO TO ERROR; PRNT:=1; TEXT:=0; SEQ:=0;
END % IF WHICH = 6
ELSE IF WHICH=7 THEN % "END"
BEGIN OADR:=ENDADR; SIZ:=8; TYPE:=3; GO TO STRING;
END % IF WHICH=7
ELSE IF SEQ+TEXT NEQ 0 AND WHICH LSS 10 THEN GO TO ERROR
ELSE IF WHICH=8 THEN SEQ := 1
ELSE IF WHICH=9 THEN TEXT:= 1
ELSE IF WHICH=10 THEN % FILE INSTRUCTION
BEGIN IF FREP=1 THEN GO ERROR; FTOG:=1;
IF INFIL=0 AND PRNT=0 THEN INFIL:=1
ELSE IF OUTFIL=0 THEN OUTFIL:=1
ELSE GO TO ERROR; % TOO MANY FILES SPECIFIED
END % IF WHICH = 10
ELSE IF WHICH=11 THEN SITE:=1
ELSE IF WHICH LEQ 17 THEN
BEGIN OADR:=AROADR+(WHICH-11);
SIZ:=1; TYPE:=2;
GO TO STRING;
END;

```

```

76019200 T 0077
76019300 T 0077
76019400 T 0077
START OF SEGMENT ***** 132
76019500 T 0000
76019600 T 0001
76019700 T 0007
76019800 T 0008
76019900 T 0012
76020000 T 0018
76020100 T 0018
76020200 T 0020
76020300 T 0022
76020400 T 0024
76020500 T 0024
76020600 T 0028
76020700 T 0030
76020800 T 0030
76020900 T 0032
76021000 T 0036
76021100 T 0042
76021200 T 0043
76021300 T 0051
76021400 T 0060
76021500 T 0060
76021600 T 0061
76021700 T 0066
76021800 T 0066
76021900 T 0069
76022000 T 0075
76022100 T 0082
76022200 T 0082
76022300 T 0083
76022400 T 0083
76022500 T 0087
76022600 T 0087
76022700 T 0087
76022800 T 0088
76022900 T 0100
76023000 T 0100
76023100 T 0101
76023200 T 0104
76023300 T 0104
76023400 T 0110
76023405 T 0113
76023410 T 0118
76023415 T 0121
76023420 T 0127
76023425 T 0133
76023430 T 0139
76023435 T 0140
76023440 T 0140
76023450 T 0144
76023460 T 0146
76023470 T 0149
76023480 T 0150
76023490 T 0151

```

```

GO TO LOOP;
END * IF TYPE=1
ELSE IF TYPE LSS 5 THEN * A STRING OR SEQUENCE NUMBER
STRING;
  BEGIN IF TYPE=2 THEN * A STRING
  BEGIN IF FND+REP NEQ 1 THEN GO TO ERROR;
  IF REP=1 THEN * A REPLACE STRING
  BEGIN STRGCNT:=STRGCNT + 1;
  IF SIZ=0 AND STRGCNT=1 THEN GO TO ERROR;
  TYPE := STRGCNT; * TELL SAVEIT WHETHER OLD OR NEW STRING
  IF STRGCNT=2 THEN * HAVE BOTH OLD AND NEW STRINGS
  BEGIN REP:=0; STRGCNT:=0; OK:=1; END;
  END * IF A REPLACE STRING
  ELSE OK:=1; * IF A "FIND" STRING
  END * IF TYPE = 2
  ELSE * A SEQUENCE NUMBER, TYPE = 3 OR 4
  BEGIN X:=CNVERT(OADR,SIZ);
  IF X LEQ PREV THEN GO TO ERROR ELSE PREV:=X; * CHECK NUMBER
  END IF A SEQ NO;
  X:=TSIZ + SIZ + 4; * TOTAL NU. OF CHARACTERS TO BE ADDED
  IF X GTR 230 THEN * NOT ENOUGH ROOM
  BEGIN CLEARSTACK(STACK[BASE+2],TSIZ); CONTINUEBIT := FALSE;
  GOERR("TOOLONG",MSGVAL);
  END * IF X GTR 230
  ELSE SAVEIT(STACK[BASE],TSIZ,OADR,SIZ,TYPE,X); * STORE STRING
  GO TO LOOP;
  END * IF TYPE LSS 5
  ELSE IF TYPE = 5 THEN * A CONTINUATION CHARACTER AND LEFT ARROW
  BEGIN TEMP := 32; WAITFOR := VINPUT;
  IF SCHEDULELINE THEN REQUESTIP(LL); GO TO EXIT;
  END * IF TYPE = 5
  ELSE IF TYPE = 6 THEN * CALL OUT THE PROGRAM
  BEGIN
  CONTINUEBIT := FALSE;
  IF SOURCEFILE=0 AND INFIL=0 THEN
  BEGIN CLEARSTACK(STACK[BASE+2],TSIZ); GOERR("NO FILE",MSGVAL);
  END;
  TEMP := 3;
  IF NOT BOOLEAN(OK) THEN
  BEGIN CLEARSTACK(STACK[BASE+2],TSIZ);
  GOERR("INCOMPL",MSGVAL);
  END;
  CLEAR(WORK[0],29);
  MOVE(30, WORDS, STACK[BASE+1], WORK[0]);
  DISKWRITE(ESP2);
  CLEARSTACK(STACK[BASE+2],TSIZ);
  IF TPCOUNT NEQ 0 THEN
  BEGIN
  LITOUT("WAIT. ",NDCRLF);
  DOING := 1;
  WANT := VUPDATE;
  GO TO EXIT;
  END
  ELSE GO TO CASE1;
  END * IF TYPE = 6
  ELSE IF TYPE=7 THEN * FILE NAME
  BEGIN

```

```

76023500 T 0151
76023600 T 0151
76023700 T 0151
76023800 T 0152
76023900 T 0154
76024000 T 0154
76024100 T 0159
76024200 T 0161
76024210 T 0166
76024300 T 0170
76024400 T 0172
76024500 T 0174
76024600 T 0184
76024700 T 0184
76024800 T 0187
76024900 T 0187
76025000 T 0187
76025100 T 0190
76025200 T 0192
76025300 T 0192
76025400 T 0195
76025500 T 0196
76025600 T 0202
76025700 T 0209
76025800 T 0209
76025900 T 0215
76026000 T 0215
76026100 T 0215
76026200 T 0217
76026210 T 0219
76026300 T 0225
76026400 T 0225
76026500 T 0227
76026600 T 0227
76026610 T 0229
76026620 T 0233
76026630 T 0243
76026700 T 0243
76026800 T 0244
76026900 T 0245
76027000 T 0249
76027100 T 0256
76027200 T 0256
76027300 T 0257
76027400 T 0260
76027500 T 0263
76027600 T 0267
76027700 T 0268
76027800 T 0269
76027900 T 0270
76028000 T 0270
76028100 T 0271
76028200 T 0273
76028300 T 0273
76028400 T 0281
76028410 T 0281
76028415 T 0282

```

```

X := TSIZ + SIZ + 4; % NO OF CHARACTERS TO BE ADDED
IF X GTR 230 THEN % NOT ENOUGH ROOM
  BEGIN CLEARSTACK(STACK[BASE+2],TSIZ); CONTINUEBIT:=FALSE;
  GOERR("TOOLONG",MSGVAL);
  END % IF X GTR 230
ELSE SAVEIT(STACK[BASE],TSIZ,OADR,SIZ,TYPE,X);
  FTOG:=0;
  GO TO LOOP;
  END % IF TYPE = 7
ELSE % MUST BE AN ERROR
  BEGIN
  MSGPTR := MSGVAL + TYPE - 9;
ERROR:
  TEMP := 3;
  CLEARSTACK(STACK[BASE+2],TSIZ);
  CONTINUEBIT := FALSE;
  GOERR(ERRVALUE(OADR),MSGPTR);
  END IF AN ERROR;
END CASE 0;

BEGIN % CASE 1
CASE1:
  CLEAR(WORK[0],29);
  WORK[0]:=ESP2;
  WORK[3]:=MAKEFN("1S",LL);
  WORK[4]:=WORK[6]:=USERCODE;
  WORK[5]:=SOURCEFILE;
  WORK[7] := FILETYPE; % CHECK FOR TYPE DATA FILES
  RUN1:=IF BOOLEAN(FND) THEN "FIND " ELSE "REPLACE";
  RUN2:=IF BOOLEAN(FND) THEN "DISK " ELSE "CANDE ";
  ERRSW:=USERCODE;
  DISKWRITE(ESP1);
  IF BOOLEAN(FND) THEN
    BEGIN
      IAM:=VDISPATCH; COMMONVALUE:=ESP1; LITOUT("WAIT " ,NOCRLF);
    END
  ELSE
    BEGIN
      WANT:=VDISPATCH;DDING:=2;IF NOT CONCISE THEN LITOUT("WAIT, ",1);
    END;
  EQJMSG := NOT CONCISE; WRITEASTERISK := CONCISE;
% VOID
  GO TO EXIT;
END CASE 1;
BEGIN % CASE 2
LABEL DUMMY;

IF COMMONCELL,[2:1] = 0 THEN % NORMAL EOJ
  BEGIN
  IF BOOLEAN(FREP) THEN
  IF COMMONCELL,[21:27] NEQ 0 THEN % SUBSTITUTIONS MADE
    BEGIN
    SOURCEFILE := MAKEFN("1S",LL);
    GOODOBJ := FALSE;
    CANCELCKPT(A[*]);
    END;
  WRITEASTERISK:=TRUE;

```

```

76028420 T 0283
76028425 T 0286
76028430 T 0286
76028435 T 0293
76028440 T 0299
76028445 T 0299
76028450 T 0305
76028455 T 0308
76028460 T 0308
76028500 T 0308
76028600 T 0308
76028700 T 0309
76028800 T 0311
76028900 T 0312
76029000 T 0312
76029100 T 0316
76029200 T 0318
76029300 T 0326
76029400 T 0326
132 IS 327 LONG, NEXT SEG 131
76029500 T 0079
76029600 T 0079
76029700 T 0080
76029800 T 0081
76029900 T 0082
76030000 T 0084
76030100 T 0087
76030150 T 0088
76030200 T 0090
76030300 T 0094
76030400 T 0098
76030500 T 0099
76030600 T 0102
76030700 T 0104
76030710 T 0104
76030720 T 0107
76030730 T 0107
76030740 T 0107
76030750 T 0114
76030760 T 0118
76030800 T 0118
76030900 T 0124
76031000 T 0124
76031100 T 0126
76031200 T 0126
76031210 T 0126
START OF SEGMENT ***** 133
76031300 T 0000
76031400 T 0001
76031500 T 0002
76031600 T 0003
76031700 T 0005
76031800 T 0006
76031900 T 0008
76032000 T 0010
76032100 T 0011
76032200 T 0011

```

```

END % IF EOJ OK
ELSE
BEGIN % ERROR EOJ
IF SCHEDULELINE AND BOOLEAN(FREP) AND NOT NUSTOP THEN SCHEDERROR;
LINECLEAR(LCW,A[*]);
MSGPTR := MSGVAL;
WRITEASTERISK := FALSE;
END % IF ERROR EOJ
IAM := WANT := WAITFOR := 0;
END CASE 2;

END ALL CASES;

EXIT;
END PROCEDURE REPLACIT;

```

```

76032300 T 0013
76032400 T 0013
76032500 T 0013
76032600 T 0014
76032700 T 0029
76032800 T 0030
76032900 T 0032
76033000 T 0034
76033100 T 0034
76033200 T 0036
133 IS 37 LONG, NEXT SEG 131
76033300 T 0127
START OF SEGMENT ***** 134
134 IS 4 LONG, NEXT SEG 131
76033400 T 0128
76033500 T 0128
131 IS 134 LONG, NEXT SEG 8

```

```

*****%
PROCEDURE REMEMBERTHIS (LCW);
*****%
BOOLEAN LCW;
BEGIN
DEFINE CTR = [316] #, USEDOROPEN = [12136] #;

REAL STREAM PROCEDURE BITNUMBER (T);
VALUE T;
BEGIN
LABEL L;
SI := LOC T; SI := SI+2;
*(IF SC= "" THEN BEGIN SI:=SI+1; TALLY:=TALLY+6; END
ELSE
*(IF SB THEN BEGIN SKIP 1 SB; TALLY:=TALLY+1; END
ELSE
JUMP OUT 2 TO L));
L: BITNUMBER := TALLY;
END STREAM;

```

```

90016000 T 0061
90017000 T 0061
90018000 T 0061
90019000 T 0061
90020000 T 0061
90021000 T 0061
START OF SEGMENT ***** 135
90022000 T 0000
90023000 T 0000
90024000 T 0000
90025000 T 0000
90026000 T 0000
90027000 T 0000
90028000 T 0002
90029000 T 0002
90030000 T 0004
90031000 T 0004
90032000 T 0005
90033000 T 0006

```

```

REAL
%
LINK, W, T, S;
S := TOBECONTROL;
TOBECONTROL,USEDOROPEN := R(B(W:= TOBECONTROL,USEDOROPEN)
OR NOT B(T:=R(NOT B(W+1) OR B(W)))));
IF LINK := 35 - BITNUMBER (T) LSS 0 THEN
BEGIN
LITOUT("NOTDONE",CRLF); TOBECONTROL := S;
END IF NOTREMEMBERED ELSE
BEGIN
THINGSTODO := B(W := R(THINGSTODO) + 1);
IF W GTR 1 THEN TOBEDONE[R(THINGSLAST),0],CTR := LINK;
TOBEDONE[LINK,0] := T&LINK[114018];

```

```

90034000 T 0007
90035000 T 0007
90036000 T 0007
90037000 T 0008
90038000 T 0009
90039000 T 0014
90040000 T 0016
90041000 T 0017
90042000 T 0019
90043000 T 0019
90044000 T 0021
90045000 T 0024
90046000 T 0029

```



```

        THINGSLAST:= B(LINK);
        IF W = 1 THEN THINGSLINK:= B(LINK);
        MOVE (15,WORDS,WORK[0],TOBEDONE[LINK,1]);
        MOVE (5,WORDS,EVENTS[0],TOBEDONE[LINK,16]);
        INC(TOBECONTROL,CTR);
        IF TOBECONTROL.CTR GTR 20 THEN WECANUSEDATA := FALSE;
    END IF REMEMBERED;
    END OF REMEMBERTHIS;

```

```

90047000 T 0032
90048000 T 0035
90049000 T 0038
90050000 T 0041
90051000 T 0044
90052000 T 0047
90053000 T 0050
90054000 T 0050

```

135 IS 53 LONG, NEXT SEG 8

```

*****%
PROCEDURE GETSOMETHINGTODO (LCW,A); BOOLEAN LCW; ARRAY A[0];
*****%
    BEGIN
    DEFINE CTR = [316] #, USEDOROPEN = [12136] #;

    REAL LINK, W, TL, I, COUNT, LCOUNT, LMAX;
    LABEL AGAIN;
        AGAIN: LMAX:=REAL(THINGSTODO);
        LCOUNT:=LCOUNT+1;
        TL := REAL(THINGSLAST);
        LINK:=R(THINGSLINK);DECB(THINGSTODO);
        THINGSLINK:=B(I:=(W:=TOBEDONE[LINK,0]),CTR);
        TOBECONTROL,USEDOROPEN:= % TURN OFF THE BIT FOR ROW[LINK]
            R(B(TOBECONTROL,USEDOROPEN) AND B(W));
        MOVE (15,WORDS,TOBEDONE[LINK,0],WORK[0]);
        MOVE (05,WORDS,TOBEDONE[LINK,16],EVENTS[0]);
        IF EVENTREASON=VINPUT OR EVENTREASON=VTEACHER OR
        EVENTREASON=VEQUATE THEN
            BEGIN
                WORKADRS := ADDRESS(WORK[1]);
                COUNT := (WORK[0]-1),[3919];
                WORKENDADRS:=(WORKADRS+COUNT,[4015])&COUNT[30:4513];
            END;
            DECR(TOBECONTROL,CTR);
            IF TOBECONTROL.CTR LEQ 20 THEN WECANUSEDATA := TRUE;
            REASON:= EVENTREASON;
            IF SAVEDWHILEINITBUSY THEN
                BEGIN % GET THAT ONE FIRST.
                    IF REASON LSS VINPUT OR REASON = WAITINGSCW,SCHWAITBITS
                    THEN SAVEDWHILEINITBUSY := FALSE ELSE
                BEGIN % THIS ISNT IT.
                    IF LCOUNT GTR LMAX THEN
                        BEGIN
                            REASON:=VERROR; ERRSW:="SYS,ERR"; BUSYLINE:=FALSE;
                            LINE[LL,MSGPTRLOC]:=28; % TASK WAS DISCONTINUED
                            GO TO EXIT;
                        END;
                    REMEMBERTHIS(LCW); GO AGAIN;
                END;
            WHILE I NEQ TL DO
                IF W:=TOBEDONE[I,16],[1817] LSS VINPUT
                OR W = WAITINGSCW,SCHWAITBITS THEN
                    BEGIN

```

```

90055000 T 0061
90056000 T 0061
90057000 T 0061
90058000 T 0061
90059000 T 0061
90060000 T 0000
90061000 T 0000
90061100 T 0000
90062000 T 0001
90063000 T 0003
90064000 T 0004
90065000 T 0009
90066000 T 0013
90067000 T 0014
90068000 T 0016
90069000 T 0019
90069010 T 0022
90069015 T 0025
90069020 T 0027
90069030 T 0028
90069040 T 0030
90069050 T 0032
90069060 T 0034
90070000 T 0034
90071000 T 0037
90072000 T 0041
90073000 T 0042
90074000 T 0043
90075000 T 0043
90076000 T 0045
90077000 T 0049
90077100 T 0049
90077200 T 0050
90077300 T 0050
90077400 T 0054
90077500 T 0056
90077600 T 0058
90078000 T 0058
90079000 T 0059
90080000 T 0059
90081000 T 0061
90082000 T 0063
90083000 T 0065

```

START OF SEGMENT ***** 136

```

                SAVEDWHILEINITBUSY:=TRUE; GO EXIT;
            END ELSE I:=TOBEDONE[I,0],CTR;
    END IF SAVEDWHILEINITBUSY;
EXIT: END OF GETSOMETHINGTODU;

```

```

90084000 T 0066
90085000 T 0069
90086000 T 0072
90087000 T 0072
136 IS 77 LONG, NEXT SEG 8

```

```

*****%
PROCEDURE FINISHED (LCW, A); BOOLEAN LCW; ARRAY A[0];
*****%
    BEGIN
        LABEL RESTRT, EXIT;

```

```

90088000 T 0061
90089000 T 0061
90090000 T 0061
90091000 T 0061
90091100 T 0061
START OF SEGMENT ***** 137
90091200 T 0000
90091300 T 0000
90091400 T 0000
90091500 T 0000
90091600 T 0006
90091700 T 0008

```

```

%.....%
STREAM PROCEDURE PLSLOGOFF(WORK);
BEGIN DI:=WORK;
    DS:=45LIT"YOUR SCHEDULED TIME IS UP, PLEASE LOG OFF AS ";
    DS:=17LIT"SOON AS POSSIBLE, ";
END PLSLOGOFF;

```

```

%.....%
STREAM PROCEDURE TELLHIMBYE(WORK);
BEGIN DI:=WORK;
    DS:=47LIT"SORRY, BUT YOU ARE PAST YOUR SCHEDULED TIME AND ";
    DS:=23LIT"WE MUST DISCONNECT YOU, ";
END TELLHIMBYE;

```

```

90091800 T 0009
90091900 T 0009
90092000 T 0009
90092100 T 0009
90092200 T 0015
90092300 T 0018

```

```

%.....%
BOOLEAN STREAM PROCEDURE TIMEOK(N, A); VALUE N, A;
BEGIN SI:=LOC A; SI:=SI+4; N(SKIP SB);
    IF SB THEN TALLY:=1; TIMEOK:=TALLY;
END TIMEOK;

```

```

90092400 T 0019
90092500 T 0019
90092600 T 0019
90092700 T 0021
90092800 T 0022

```

```

%.....%
                ALLDONE := TRUE ;
IF BOOLEAN(TIMERRESTRICTBIT) THEN % USER HAS RESTRICTED TIME PERIOD
IF NOT TIMEOK(ENTIER(TIMENOW:=TIME(1)/216000), TIMELIMITS) THEN
IF NOT NOTIFIED THEN % USER IS NOT AWARE HIS TIME IS UP
    BEGIN PLSLOGOFF(WORK); TWXOUT(LL, WORK[0], 62, 2);
        NOTIFYBIT := REAL(TRUE); NOTIFYTIME := TIMENOW; END
ELSE
    BEGIN IF TIMENOW = NOTIFYTIME GTR GRACEPERIOD THEN
        BEGIN TELLHIMBYE(WORK); TWXOUT(LL, WORK[0], 70, 2); NOSAVE:=TRUE;
            LINECLEAR(LCW, A[*]); A[BASE]:=0; BYE(LCW, A[*]); GO EXIT; END; END;
        IF INITIATEDWHILEBUSY THEN
            BEGIN

```

```

90092900 T 0023
90093000 T 0023
90093100 T 0025
90093200 T 0026
90093300 T 0031
90093400 T 0033
90093500 T 0036
90093520 T 0040
90093600 T 0040
90093700 T 0044
90093800 T 0050
90094000 T 0057
90095000 T 0057

```

```

IF LLINFO NEQ 0 THEN WAITINGSCW := LLINFO
ELSE WAITINGSCW, SCWAITBITS := 0;
INITIATEDWHILEBUSY := BOOLEAN(LLINFO:=0);
IF SAVEDWHILEINITBUSY THEN GO RESTRT ELSE GO EXIT;
END;
BUSYLINE := FALSE;
% VOID
% VOID
BEGIN
IF REAL (THINGSTODO) = 0 THEN
ALLMCPMSG := FALSE;
LIBMSGCTR := 0;
IF SCHEDULELINE THEN REQUESTIP(LL)
ELSE
IF WRITEASTERISK THEN
TWXOUT(LL, ASTOR, 1, 1);
WRITEASTERISK := FALSE; GO EXIT;
END;
COMMENT: SOMETHING HAS BEEN REMEMBERED;
RESTRT: RESTARTNEEDED := TRUE;
GETSOMETHINGTODO(LCW, A[*]);
EXIT:END;

```

```

90096000 T 0058
90097000 T 0060
90098000 T 0064
90099000 T 0067
90100000 T 0069
90101000 T 0069
90102000 T 0071
90103000 T 0071
90104000 T 0071
90105000 T 0072
90106000 T 0073
90107000 T 0075
90107100 T 0076
90107200 T 0077
90108000 T 0081
90109000 T 0082
90110000 T 0084
90111000 T 0087
90112000 T 0087
90113000 T 0087
90114000 T 0090
90115000 T 0092

```

137 IS 93 LONG, NEXT SEG 8

```

*****
DEFINE BUILDSCW =SCW:=0&WAITFOR[10:41:7]&PREVRCW[17:40:8]&BASE
[25:40:8]&DOING[33:41:7]&IAM[40:40:8]&1[1:46:2]
#;
DEFINE BUILDRCW =RCW:=SCW&SREG[03:40:8]&PARAM[11:41:7]&TEMP
[18:41:7]&DOING[33:41:7]&IAM[40:40:8]&2[1:46:2]
#;
DEFINE STORERCW =A[PREVRCW:=SCWPREV-1] := RCW #;
%
DEFINE RECALLRCW=BEGIN RCW :=A[PREVRCW:=SCWPREV];
PARAM :=RCWPARAM;
TEMP :=RCWTEMP ;
BASE :=RCWBASE ;
DOING :=RCWDOING;
IAM :=RCWIAM ;
SREG :=RCWSREG;
PREVRCW:=PREVRCW +1;
WANT :=WAITFOR:=0;
END OF RECALLRCW #;

```

```

90116000 T 0061
90117000 T 0061
90118000 T 0061
90119000 T 0061
90120000 T 0061
90121000 T 0061
90122000 T 0061
90123000 T 0061
90124000 T 0061
90125000 T 0061
90126000 T 0061
90127000 T 0061
90128000 T 0061
90129000 T 0061
90130000 T 0061
90131000 T 0061
90132000 T 0061
90133000 T 0061
90134000 T 0061
90135000 T 0061
90136000 T 0061
90137000 T 0061
90138000 T 0061
90139000 T 0061
90140000 T 0061
90141000 T 0061
90142000 T 0061
90143000 T 0061
90144000 T 0061

```

%PAGE

```

*****
BOOLEAN PROCEDURE ANOTHEROPERATION (LCW, A);
*****

```

```

BOOLEAN LCW;
ARRAY A[0];
BEGIN
LABEL L1;

```

START OF SEGMENT ***** 138

```

        IF IAM NEQ 0 THEN
L1: BEGIN COMMENT THIS PROCEDURE WILL BE RECALLED;
        BUILDRCW; STORERCW;
        END % PROCEDURE IS NOW IDLE AND CAN BE RESTARTED;
        ELSE
        DOING := 0;
        SREG:=BASE := BASE + TEMP + PARAM;
        IAM := WANT;
        BUILDSCW;
        IF WANT NEQ 0 THEN
        BEGIN COMMENT A NEW PROCEDURE IS TO BE GIVEN CONTROL;
        A [BASE] := PARAM;
        DOING := WANT := 0;
        END THE NEW PROCEDURE IS NOW READY TO BE CALLED;
        TEMP := PARAM := 0;
        IF WAITFOR NEQ 0 THEN
        BEGIN COMMENT WE MUST WAIT BEFORE THIS LINE CAN CONTINUE;
        IF IAM NEQ 0 THEN GO TO L1; COMMENT MAKE LINE IDLE;
        ANOTHEROPERATION := FALSE;
        WAITINGSCW := SCW;
        END LINE WILL NOW WAIT ON EVENTQUE
        ELSE
        BEGIN COMMENT NO WAIT IS REQUIRED;
        IF (ANOTHEROPERATION := NOT ALLDONE) AND IAM EQL 0 THEN
        BEGIN COMMENT RESET RETURN INFO;
        RECALLRCW; WAITINGSCW:=BUILDSCW;
        END WE CAN NOW RETURN TO THE LAST UNFINISHED PROCEDURE;
        END IF LINE DID NOT HAVE TO WAIT;
        END OF ANOTHEROPERATION;

```

```

90145000 T 0000
90146000 T 0000
90147000 T 0002
90148000 T 0011
90149000 T 0011
90150000 T 0011
90151000 T 0012
90152000 T 0015
90153000 T 0015
90154000 T 0022
90155000 T 0023
90156000 T 0023
90157000 T 0025
90158000 T 0026
90159000 T 0026
90160000 T 0027
90161000 T 0028
90162000 T 0028
90163000 T 0030
90164000 T 0030
90165000 T 0032
90166000 T 0032
90167000 T 0032
90168000 T 0032
90169000 T 0035
90170000 T 0035
90171000 T 0055
90172000 T 0055
90173000 T 0055

```

138 IS 58 LONG, NEXT SEG 8

```

*****%
PROCEDURE HANDLETHISLINE(LCW, A);
*****%
BOOLEAN LCW;
ARRAY A[0];
BEGIN
LABEL RESTART, AGAIN, EXIT;
RESTART:
IAM := WANT := WAITFOR := TEMP := PARAM := 0;
ALLDONE := FALSE;
RESTARTNEEDED := FALSE;
IF WAITINGSCW, SCWWAITBITS = REASON
OR WAITINGSCW, SCWWAITBITS = R(NOT FALSE).[41:7]
THEN
BEGIN COMMENT: SOMEONE IS WAITING FOR THIS EVENT;
SCW := WAITINGSCW;
WAITINGSCW := 0;
PREVRCW := SCWPREV;
END
ELSE
BEGIN COMMENT: WE WERE NOT WAITING FOR IT SO;
WANT := REASON;

```

```

90174000 T 0061
90175000 T 0061
90176000 T 0061
90177000 T 0061
90178000 T 0061
90179000 T 0061
90180000 T 0061
START OF SEGMENT ***** 139
90181000 T 0000
90182000 T 0000
90183000 T 0002
90184000 T 0005
90185000 T 0007
90186000 T 0008
90187000 T 0010
90188000 T 0011
90189000 T 0011
90190000 T 0012
90191000 T 0014
90192000 T 0015
90193000 T 0015
90194000 T 0015
90195000 T 0015

```

```

IF IDLELINE THEN
BEGIN COMMENT: MUST BE A NEW EVENT FOR THIS LINE;
IF NOT LOGGEDON THEN
BEGIN
CHARGE:= -1;
END LOGGING ON THE SYSTEM;
BUSYLINE := TRUE;
BASE := SBASE;
PREVRCW := LLWORDS;
END
ELSE
BEGIN
IF REASON=VTEACHER THEN % DONT PERMIT MULTIPLE INQUIRIES
IF INQUIRY THEN GO EXIT ELSE INQUIRY := TRUE;
IF REASON LSS VINPUT AND NOT INITIATEDWHILEBUSY THEN
BEGIN
INITIATEDWHILEBUSY := TRUE;
LLINFO := SCW := WAITINGSCW;
BASE := SCWBASE + 1;
PREVRCW := SCWPREV - 1;
END
ELSE
BEGIN
IF R(THINGSTODO) GTR 2 THEN
BEGIN
LITOUT("NOTDONE",CRLF);
END
ELSE
BEGIN
REMEMBERTHIS (LCW);
IF REASON = LLINFO.SCWWAITBITS
OR REASON LSS VINPUT THEN SAVEDWHILEINITBUSY := TRUE;
END
IF REMEMBERED;
GO EXIT;
END
REASON VS 15;
END
IDLE OR NOT;
BUILDSCW;
A[PREVRCW] := BUILDRCW;
END
WAITING OR NOT;
-----
AGAIN:
IF NOT ANOTHEROPERATION (LCW, A[*]) THEN GO EXIT;
COMMENT: IF WE GOT TO HERE IT MEANS THERE IS SOMETHING
THAT CAN BE INITIATED FOR THIS LINE NOW,
"IAM" IS SET TO THE CASE VALUE WHICH WILL CALL
OUT THE APPROPRIATE PROCEDURE, SO= LETS TRYIT;
CASE IAM OF
BEGIN
FINISHED(LCW, A[*]); % 00 = LAST THING DONE ON LINE.
HITHERE(LCW, A[*]); % 01 = DIAL UP.
BREAK (LCW, A[*]); % 02 = BREAK OCCURRED.
WRU (LCW, A[*]); % 03 = WHO ARE YOU RECD.
LINECLEAR(LCW, A[*]); % 04 = CLEAR LINE AFTER 02 OR 03.
DISKCHUNK; % 05 = NEW CHUNK OF DISK.
DISCONNECT(LCW, A[*]); % 06 = DISCONNECT OCCURRED.
INPUTDATA(LCW, A[*]); % 07 = INPUT DATA IS IN OUR BUFFER.
BEGIN % 08 = BREAK FINISHED.
MSGPTR:=18; % I AM THE GENIE ...

```

```

90196000 T 0016
90197000 T 0017
90198000 T 0018
90199000 T 0019
90201000 T 0019
90202000 T 0021
90203000 T 0021
90204000 T 0023
90205000 T 0024
90206000 T 0024
90207000 T 0024
90208000 T 0024
90208100 T 0025
90208200 T 0026
90209000 T 0030
90210000 T 0032
90211000 T 0032
90212000 T 0035
90213000 T 0037
90214000 T 0038
90215000 T 0040
90216000 T 0040
90217000 T 0040
90218000 T 0041
90219000 T 0042
90220000 T 0042
90221000 T 0043
90222000 T 0043
90223000 T 0046
90224000 T 0047
90225000 T 0047
90226000 T 0052
90227000 T 0052
90228000 T 0052
90229000 T 0052
90230000 T 0052
90231000 T 0059
90232000 T 0067
90233000 T 0067
90234000 T 0067
90235000 T 0068
90236000 T 0070
90237000 T 0070
90238000 T 0070
90239000 T 0070
90240000 T 0070
90241000 T 0070
90242000 T 0071
90243000 T 0073
90244000 T 0075
90245000 T 0077
90246000 T 0080
90247000 T 0082
90248000 T 0083
90249000 T 0085
90250000 T 0087
90250100 T 0087

```

```

IAM:=WANT:=WAITFOR:=0;
END BREAKCLEAR;
RESTARTIT(LCW,A[*]); % 09 = AFTER H/L SAVE WRKFILE
IAM:=WAITFOR:=EVENTS[1]; % 10 = MCP USE.
MAKEDATE; % 11 = DATE CHANGED.
HITHERE(LCW,A[*]); % 12 (RETURN FROM NEWS/CANDE FIND)
CALL(NOTUSED); % 13 =
MCPMSG(LCW,A[*]); % 14 = HANDLES MCP MESSAGES ABOUT JOBS
COOLIT(A[*]); % 15
MCPOK(A[*]); % 16
CALL(NOTUSED); % 17 =
CALL(NOTUSED); % 18 =
CALL(NOTUSED); % 19 =
CALL(NOTUSED); % 20 =
CALL(NOTUSED); % 21 =
CALL(NOTUSED); % 22 =
CALL(NOTUSED); % 23 =
CALL(NOTUSED); % 24 =
TEACHER(LCW,A[*]); % 25 = WANTS ELABORATION.
INPUT(LCW,A[*],WORK[1],EVENTS[1],EVENTS[3]); % USER INPUT
DISPATCH(LCW,A[*]); % 27 = FIRES UP RUNNERS.
DISKREAD(LCW,A[*]); % 28 = READS DISK FOR PROCEDURES
CALL(NOTUSED); % 29 =
DOPS(LCW,A[*]); % 30 = ERR DURING SCAN
DOGCATCHER(LCW,A[*]); % 31 = DOGCATCHER.
COMPILEIT(LCW,A[*]); % 32 = COMPILE IT.
SETUPFILES(LCW,A[*]); % 33 = SET UP ON HELLO
ERROR(SREG,LCW,A[*]); % 34 = ERROR HANDLER
CLOSEWORKTABLE(LCW,A[*]); % 35 = TO MAKE SURE TABLE IS TIDY
CHNGPSWD(LCW,A[*]); % 36 = CHANGE PASSWORD
THYME(LCW,A[*]); % 37 = TIME
CARRIAGE(LCW,A[*]); % 38 = CARRIAGE CONTROL.
CHNGNAME(LCW,A[*]); % 39 = CHANGENAME
MONITORR(LCW,A[*]); % 40 =
SETIT(LCW,A[*]); % 41 = "SET" & "RESET"
SSFILE(LCW,A[*]); % 42 TRANSMIT FILE
EQUATE(LCW,A[*]); % LABEL EQUATION
STATUSCHECK(LCW,A[*]); % 44 = STATUS OF RUNNING JOB
CALL(NOTUSED); % 45 =
CALL(NOTUSED); % 46 =
CALL(NOTUSED); % 47 =
CALL(NOTUSED); % 48 =
CALL(NOTUSED); % 49 =
BEGIN % 50 = SEND SS MSGS
MOVE(4,WORDS,EVENTS[1],WORK[10]);
TWXOUT(LL,WORK[0],112,2);
IAM := 0;
END;
PDIT(LCW,A[*]); % 51 = LIST FILES
COPYIT(LCW,A[*]); % 52 = COPY.
SEQ(LCW,A[*]); % 53 = WANTS SEQ #S.
;15 ;40
BYE(LCW,A[*]); % 54 = BYE.
MAKENEWFILE(A[*]); % 55 = MAKE
LISTIT(LCW,A[*]); % 56 = LIST
EXECUTE(LCW,A[*]); % 57 = RUN, COMPILE OR EXECUTE
LOADIT(LCW,A[*]); % 58 = LOAD

```

```

90250200 T 0089
90250300 T 0090
90251000 T 0091
90252000 T 0093
90253000 T 0095
90254000 T 0100
90255000 T 0103
90256000 T 0104
90257000 T 0106
90258000 T 0108
90259000 T 0110
90260000 T 0111
90261000 T 0112
90262000 T 0113
90263000 T 0115
90264000 T 0116
90265000 T 0117
90266000 T 0118
90267000 T 0120
90268000 T 0122
90269000 T 0126
90270000 T 0128
90271000 T 0130
90272000 T 0131
90273000 T 0134
90274000 T 0136
90275000 T 0138
90276000 T 0140
90277000 T 0143
90278000 T 0145
90279000 T 0147
90280000 T 0150
90281000 T 0152
90282000 T 0154
90283000 T 0156
90284000 T 0159
90285000 T 0161
90286000 T 0163
90287000 T 0165
90288000 T 0167
90289000 T 0168
90290000 T 0169
90291000 T 0170
90292000 T 0172
90293000 T 0172
90294000 T 0174
90295000 T 0176
90296000 T 0176
90297000 T 0177
90298000 T 0179
90299000 T 0181
90300000 T 0184
90301000 T 0184
90302000 T 0186
90303000 T 0188
90305000 T 0190
90307000 T 0192

```

```

SAVER (LCW, A[*]); % 59 = SAVE
CALL(NOTUSED); % 60
REMOVE (LCW, A[*]); % 61 = REMOVE
CHANGE(LCW, A); % 62 = CHANGE
RENAMEIT (LCW, A[*]); % 63 = RENAME
WHATSIT (LCW, A[*]); % 64 = PRINT FILE TYPE
CALL (NOTUSED); % 65 = WAS "TYPE" %0404=
CALL(NOTUSED); % 66
GUARDIT(LCW, A[*]); % 67 = GUARD(MAKE LOCK)
LOCKIT(LCW, A[*]); % 68 = LOCK OR UNLOCK
TAPEIT(LCW, A[*]); % 69 = PAPER TAPE
LISTIT(LCW, A[*]); % 70 = PRINT(NO HDING)
CHARGEIT(LCW, A[*]); % 71 = CHARGE CODE
SCHEDULE(LCW, A[*]); % 72 = SCHEDULE A TASK
SCHEDSTATUS(LCW, A[*]); % 73 = SCHED STATUS/STOP
LISTFILE(LCW, A[*]); % 74 = LIST FILES
UPDATE(LCW, A[*]); % 75 = UPDATE WORK FILE
REPLACEIT(LCW, A[*]); % 76 = REPLACEMENT ROUTINE
END OF CASE SOMETHING MORE MAY BE NEEDED;

```

```

90308000 T 0194
90310000 T 0197
90312000 T 0200
90313000 T 0202
90314000 T 0204
90315000 T 0207
90316000 P 0209
90317000 T 0210
90318000 T 0211
90319000 T 0214
90320000 T 0216
90321000 T 0218
90322000 T 0220
90322100 T 0223
90322200 T 0225
90322300 T 0227
90322400 T 0229
90322500 T 0232
90323000 T 0234
START OF SEGMENT ***** 140
140 IS 78 LONG, NEXT SEG 139
90324000 T 0234
90325000 T 0236
90326000 T 0237
139 IS 238 LONG, NEXT SEG 8

```

```

IF RESTARTNEEDED THEN GO RESTART ELSE GO AGAIN;
EXIT:
END OF HANDLETHISLINE;

```

```

*****
PROCEDURE RUNCANDE;
*****
BEGIN
LABEL HAVEEVENT;
DO % THE LOOP STARTS HERE
BEGIN
IF NODATA AND WECANUSEDATA THEN
IF NODATAREQUESTED THEN
BEGIN
REQUESTINPUT (TTYINPUT [*]);
NODATAREQUESTED := FALSE;
END;
IF TIMETOOKLINES THEN
IF MAKEOKEVENT THEN GO TO HAVEEVENT;
WAITBIT:=REAL(NODATA);
NEXTEVENT (WAITBIT, CLOCK, EVENT [*]);
IF BOOLEAN (WAITBIT) THEN
BEGIN
MOVE (5, WORDS, EVENT[0], EVENTS[0]);
LL :=EVENTLL;
IF REASON:=EVENTREASON = VMCPMSG THEN
% VOID 90347001.
MOVE(15, WORDS, EVENT[4], WORK[0]);
IF REASON=VHELLO THEN
IF LINE[LL, 13], SCHWAITBITS NEQ REASON THEN
LINE[LL, 23] := EVENTS[2]; % TERMINAL TYPE

```

```

90327000 T 0061
90328000 T 0061
90329000 T 0061
90330000 T 0061
90331000 T 0061
START OF SEGMENT ***** 141
90332000 T 0000
90333000 T 0000
90334000 T 0000
90335000 T 0001
90336000 T 0003
90337000 T 0003
90338000 T 0004
90339000 T 0006
90339100 T 0006
90339200 T 0007
90340000 T 0009
90341000 T 0010
90342000 T 0012
90343000 T 0012
90344000 T 0012
90345000 T 0015
90346000 T 0016
90347000 T 0018
90348000 T 0018
90348100 T 0021
90348200 T 0022
90348300 T 0024

```

```

END ELSE BUILDADATAEVENT;
  IF LL GTR MAXLINES THEN
  BEGIN LL:=0; REASON:=13; END; %BAD LINE NUMBER
  IF REASON=VHELLO AND LOGGINGON THEN
  BEGIN LL:=0; REASON:=13; END;
HAVEVENT;
  IF LLPREV NEQ LL THEN
  BEGIN COMMENT A DIFFERENT USER IS GOING TO USE TIME;
  IF LLPREV NEQ 0 THEN
  BEGIN COMMENT CHARGE PREV USER FOR HIS TIME;
  LINE[LLPREV,PTLOC]:=CLOCK-CLOCKPREV+LINE[LLPREV,PTLOC];
  END IF USER TO BE CHARGED;
  LLPREV:=LL;
  CLOCKPREV:=CLOCK;
  END IF NEW USER TO BE HANDLED;
  IF REASON NEQ 13 THEN % THROW AWAY #13,BAD LUCK.
  HANDLETHISLINE(LCW,LINE[LL,*]);
  END OF LOOP UNTIL FALSE;
END RUNCANDE;

```

```

90349000 T 0027
90350000 T 0028
90351000 T 0029
90352000 T 0031
90353000 T 0033
90353500 T 0035
90354000 T 0036
90355000 T 0036
90356000 T 0037
90357000 T 0038
90358000 T 0038
90359000 T 0042
90360000 T 0042
90361000 T 0043
90362000 T 0044
90363000 T 0044
90364000 T 0044
90365000 T 0049
90366000 T 0049

```

141 IS 50 LONG, NEXT SEG 8

```

*****
PROCEDURE INITIALIZECANDE;
*****
  BEGIN LABEL DUMMY;

  STREAM PROCEDURE SETUPCOMM2(COMM2);
  BEGIN DI:=COMM2;
    DS:=8 LIT"088I894("; % 3"0010103110110435"
    DS:= LIT"="; DI:=DI+4; % 3"75"
    SI:=LOC COMM2; SI:=SI+5; DS:=3 CHR;
  END;

```

```

90367000 T 0061
90367100 T 0061
90367200 T 0061
90367300 T 0061
START OF SEGMENT ***** 142
90367400 T 0000
90367410 T 0000
90367420 T 0000
90367430 T 0001
90367440 T 0002
90367450 T 0003

```

```

REAL STREAM PROCEDURE ADDRESS(W);
  BEGIN SI:=W; ADDRESS:=SI; END;

```

```

90367500 T 0003
90367510 T 0003

```

```

X.....
*****
PROCEDURE SPOLOGON(LCW,A); BOOLEAN LCW; ARRAY A[0];
*****
  BEGIN
  REAL VRB,N;

  STREAM PROCEDURE SETBIT(N,WORD); VALUE N;
  BEGIN LOCAL N1;
    SI:=LOC N; SI:=SI+6; DI:=LOC N1; DI:=DI+7; DS:=CHR;
    DI:=WORD; N1(DI:=DI+8); SKIP N DB; DS:=SET;
  END;

```

```

90367600 T 0005
90367610 T 0005
90367620 T 0005
90367630 T 0005
90367640 T 0005
90367650 T 0005
START OF SEGMENT ***** 143
90367660 T 0000
90367670 T 0000
90367680 T 0000
90367690 T 0001
90367700 T 0003

```



```

ESP1 := GETESPDISK;
ESP2 := GETESPDISK;
USERCODE := "SPO ";
LOGGEDON := TRUE;
LINECLEAR(LCW,A[*]);
FILESOK := FALSE;
FILL WORK[*] WITH
  "1SEQ ", "5HELLO ", "1BYE ", "2TAPE ",
  "2PUNCH ", "6CHARGE ", "3SCHEDUL", "3STATUS ",
  "3STOP ", "2CC ", "3SET ", "5RESET ",
  "2TIME ", "2MONITOR";

FOR VRB := 0 STEP 1 UNTIL 13 DO
  BEGIN
  N := WORK[VRB];
  IF YOUFINDAVERB(N, VERBTABLE[0], NUMOFVERBS) THEN
    BEGIN
    N := (N.[9:9]) DIV 2;
    SETBIT(N+1, RESTRICTEDVERBS);
    END;
  END;
END PROCEDURE SPOLOGON;

```

```

90367710 T 0004
90367720 T 0004
90367730 T 0005
90367740 T 0007
90367750 T 0008
90367760 T 0010
90367770 T 0012
90367780 P 0014
90367790 P 0015
START OF SEGMENT ***** 144
90367800 P 0016
90367810 P 0016
90367820 P 0016
144 IS 14 LONG, NEXT SEG 143
90367830 P 0016
90367840 P 0018
90367850 P 0018
90367860 P 0019
90367870 P 0020
90367880 P 0021
90367890 P 0022
90367900 P 0024
90367910 P 0024
90367920 P 0027
143 IS 30 LONG, NEXT SEG 142

```

```

MAKEDATE;
COMMENT VERBTABLE[WORD],[6:3] IS USED AS A FLAG FOR
SCANNER ( FOR "SPECIAL" VERB );
FILL VERBTABLE[*] WITH

```

```

"1LIST ", 3"30000000000000070", % #56
"1RUN ", 3"30000200000000071", % #57
"2SAVE ", 3"30000400000000073", % #59
"1SEQ ", 3"30000600000000065", % #53
"1EXECUTE", 3"30001000000000071", % #57
"2DD ", 3"30001000000000071", % #57 = EXECUTE
"1MAKE ", 3"30001400000000067", % #55
"2LOAD ", 3"30001600000000072", % #58
"1COMPILE", 3"30002000000000071", % #57
"3REMOVE ", 3"30002200000000075", % #61
"2CHANGE ", 3"30002400000000076", % #62
"3RENAME ", 3"30002600000000077", % #63
"1FILES ", 3"30003000000000063", % #51
"1WHATS ", 3"30003200000000100", % #64 ALSO
"2CREATE ", 3"30003400000000067", % #55 ALSO
"5HELLO ", 3"30103600000000001", % 01
"2TYPE ", 3"30004000000000076", % #62 ALSO
"1BYE ", 3"30004200000000066", % #54,
"3COPY ", 3"30004400000000064", % #52

```

```

90368000 P 0005
90368010 P 0010
90368020 P 0010
90369000 P 0010
90370000 P 0011
START OF SEGMENT ***** 145
90371000 P 0012
90372000 P 0012
90373000 P 0012
90374000 P 0012
90375000 P 0012
90376000 P 0012
90377000 P 0012
90378000 P 0012
90379000 P 0012
90380000 P 0012
90381000 P 0012
90382000 P 0012
90383000 P 0012
90384000 P 0012
90385000 P 0012
90386000 P 0012
90387000 P 0012
90388000 P 0012

```

```

"2MERGE ", 3"30004600000000064", X #52, ALSO X0904- 90389000 P 0012
"3RESEQ ", 3"30005000000000064", X #52, ALSO X0904- 90390000 P 0012
"2DELETE ", 3"30005200000000064", X #52, ALSO X0904- 90391000 P 0012
"1APPEND ", 3"30005400000000064", X #52, ALSO X0904- 90392000 P 0012
"1ADD ", 3"30005400000000064", X 52, ALSO X0904- 90393000 P 0012
"2TO ", 3"30006000000000062", X 50 X0904- 90394000 P 0012
"2SS ", 3"30006200000000062", X 50, ALSO X0904- 90395000 P 0012
"1GUARD ", 3"30006400000000103", X 67 X0904- 90396000 P 0012
"3LOCK ", 3"30006600000000104", X 68 X0904- 90397000 P 0012
"1UNLOCK ", 3"30007000000000104", X 68, ALSO X0904- 90398000 P 0012
"2TAPE ", 3"30007200000000105", X 69 X0904- 90399000 P 0012
"1PRINT ", 3"30007400000000106", X 70 X0904- 90400000 P 0012
"2PUNCH ", 3"30007600000000064", X 52, ALSO X0904- 90401000 P 0012
"6CHARGE ", 3"30010000000000107", X 71 X0904- 90402000 P 0012
"3SCHEDUL", 3"30010200000000110", X 72 X0904- 90402100 P 0012
"3STATUS ", 3"30010400000000111", X 73 X0904- 90402200 P 0012
"3STOP ", 3"30010600000000111", X 73, ALSO X0904- 90402300 P 0012
"2TIME ", 3"30011000000000045", X 37 X0904- 90402400 P 0012
"1UPDATE ", 3"30011200000000113", X 75 X0904- 90402860 P 0012
"3REPLACE", 3"30111400000000114", X 76 X0904- 90402870 P 0012
"4FIND ", 3"30111600000000114", X 76 ALSO X0904- 90402880 P 0012
"2CC ", 3"30012000000000046", X 38 X0904- 90402890 P 0012
"2RMERGE ", 3"30012200000000064", X 52 ALSO X0904- 90403000 P 0012
"1DISPLAY", 3"30012400000000070", X 56 X0904- 90403100 P 0012
"3SET ", 3"30112600000000051", X 41 X0904- 90403200 P 0012
"5RESET ", 3"30113000000000051", X 41 ALSO X0904- 90403300 P 0012
"2MONITOR", 3"30013200000000050", X 40 X0904- 90403400 P 0012
"3SSFILE ", 3"30013400000000052", X 42 X0904- 90403410 P 0012
"2EQUATE ", 3"30013600000000053", X 43 X0904- 90403420 P 0012
"3PUBLIC ", 3"30014000000000104", X 68 ALSO X0904- 90403430 P 0012
"7*****", 0; X LAST ENTRY X0904- 90403900 P 0012

```

```

LOCK(VERBTABLE[*]); % MAKE READ ONLY ARRAY
FILL TYPETABLE[*] WITH "7UNKNOWN",

```

```

"EBASIC ",
"EALGOL ",
"ECOBOL ",
"GFORTAN",
"ETSPOL ",
"FXALGOL ",
"3SEQ ",
"4DATA ",
"3LOCK ",
"4INFO ",
"GCQBOL68";

```

```

LOCK(TYPETABLE[*]); % MAKE READ ONLY ARRAY
FILL RESWRDTABLE[*] WITH "7UNKNOWN",

```

```

"5BASIC ", X 1
"5ALGOL ", X 2
"5COBOL ", X 3
"7FORTAN", X 4
"5TSPOL ", X 5
"6XALGOL ", X 6
"3SEQ ", X 7

```

```

145 IS 100 LONG, NEXT SEG 142
X0904- 90403950 P 0012
X0904- 90404000 P 0014
START OF SEGMENT ***** 146
X0904- 90405000 P 0016
X0904- 90406000 P 0016
X0904- 90407000 P 0016
X0904- 90408000 P 0016
X0904- 90409000 P 0016
X0904- 90410000 P 0016
X0904- 90411000 P 0016
X0904- 90412000 P 0016
X0904- 90413000 P 0016
X0904- 90413100 P 0016
X0904- 90413900 P 0016
146 IS 12 LONG, NEXT SEG 142
X0904- 90413950 P 0016
X0904- 90414000 P 0018
START OF SEGMENT ***** 147
X0904- 90415000 P 0020
X0904- 90416000 P 0020
X0904- 90417000 P 0020
X0904- 90418000 P 0020
X0904- 90419000 P 0020
X0904- 90420000 P 0020
X0904- 90421000 P 0020

```

```

"4DATA " % 8
"4LOCK " % 9
"4INFO " % 10
"7COBOL68" % 11
"6PUBLIC " % 12
"7SOLEUSE" % 13
"7UNLOCKE" % 14
"6LOCKED " % 15
"6OBJECT " % 16
"6SOURCE " % 17 % 1 - 17 ARE POSITION DEPENDENT
"4CARDS " % 18
"4DISK " % 19
"3END " % 20
"5ERRORS " % 21
"4FILES " % 22
"4FROM " % 23
"7LIBRARY" % 24
"7PRINTER" % 25
"5PUNCH " % 26
"5RESEQ " % 27
"4TAPE " % 28
"4TYPE " % 29
"4WITH " % 30
"7*****" % 31

```

```

LOCK(RESWRDTABLE[*J]); % MAKE READ ONLY ARRAY
SETUPCOMM2(COMM2);
ASTOR,[1:5] := "#";
SPOLOGON(LLCONTROL[0],LINE[0,*]);
FLAGS := (NOT FALSE) & FALSE[6:6:2];
WORK[0]:=0; DISKWAIT(1,WORK[*],30,0);
DISKBOTTOM:=WORK[4]-2; %INITIALIZE IT.
WORK[0]:=0; WORKADRS:=ADDRESS(WORK[1]);
WORKENDADRS := ADDRESS(WORK[9]);
END INITIALIZECANDE;

```

```

INITIALIZECANDE;
RUNCANDE;
END PROCEDURES BLOCK;
END COMMUNICATES BLOCK;
END ARRAYS BLOCK.

```

```

NUMBER OF ERRORS DETECTED = 0. COMPILATION TIME = 1124 SECONDS,
PRT SIZE = 395; TOTAL SEGMENT SIZE = 10840 WORDS; DISK SIZE = 540 SEGS; NO. PGM. SEGS = 153
ESTIMATED CORE STORAGE REQUIRED = 11490 WORDS.
ESTIMATED AUXILIARY MEMORY REQUIRED = 0 WORDS.
NUMBER OF CARD-IMAGES PROCESSED = 7624.

```

```

%0904= 90422000 P 0020
%0904= 90423000 P 0020
%0904= 90424000 P 0020
%0904= 90425000 P 0020
%0904= 90426000 P 0020
%0904= 90427000 P 0020
%0904= 90428000 P 0020
%0904= 90429000 P 0020
%0904= 90430000 P 0020
%0904= 90431000 P 0020
%0904= 90432000 P 0020
%0904= 90433000 P 0020
%0904= 90434000 P 0020
%0904= 90435000 P 0020
%0904= 90436000 P 0020
%0904= 90437000 P 0020
%0904= 90438000 P 0020
%0904= 90438100 P 0020
%0904= 90438200 P 0020
%0904= 90438300 P 0020
%0904= 90438400 P 0020
%0904= 90438500 P 0020
%0904= 90438600 P 0020
%0904= 90439000 P 0020
147 IS 32 LONG, NEXT SEG 142
90439100 T 0020
90439200 T 0022
90440000 T 0023
90441000 T 0025
90442000 T 0027
90442100 T 0029
90442200 T 0032
90442300 T 0034
90442400 T 0037
90442600 T 0039
142 IS 41 LONG, NEXT SEG 8
90443000 T 0061
90444000 T 0061
90445000 T 0062
8 IS 66 LONG, NEXT SEG 3
90446000 T 0018
3 IS 22 LONG, NEXT SEG 2
90447000 T 0032
2 IS 35 LONG, NEXT SEG 1
1 IS 2 LONG, NEXT SEG 0

```