

22

LOGAN /DISK
=====

```

#####
% B5500 T S S Y S L O G A N A L Y Z E R X V . 2 . 0 0      10/03/73
#####
COMMENT: THIS MATERIAL IS PROPRIETARY TO BURROUGHS
CORPORATION AND IS NOT TO BE REPRODUCED, USED
OR DISCLOSED EXCEPT IN ACCORDANCE WITH PROGRAM
LICENSE OR UPON WRITTEN AUTHORIZATION OF THE
PATENT DIVISION OF BURROUGHS CORPORATION,
DETROIT, MICHIGAN 48232.

COPYRIGHT (C) 1968, 1971, 1972
BURROUGHS CORPORATION DETROIT, MICHIGAN
AA32500;

BEGIN

% SECTION 01 - DEFINES
% SECTION 02 - GLOBALS
% SECTION 03 - UTILITY PROCEDURES, SCANNER, ETC.
% SECTION 04 - SNAP REPORT PROCEDURES
% SECTION 05 - ONSITE REPORTER
% SECTION 06 - DISK LOGGING REPORTER
% SECTION 07 - PARAMETER PROCESSING, INPUT SETUP
% SECTION 08 - INITIALIZATION, OUTER BLOCK
#####
COMMENT ::::: SECTION 01 - DEFINES :::::
#####
COMMENT DEFINES FOR THE VARIOUS LOG RECD TYPES;
  DEFINE BOJ      = 2#,
          EOJ      = 3#,
          PBEOJ    = 4#,
          OPEN     = 5#,
          CLOSEF   = 6#,
          HALT     = 7#,
          EOJSTAT  = 8#,
          CLSSTAT  = 9#,
          ON       = 10#,
          OFF      = 11#,
          CHARGE   = 12#,
          DLOG     = 13#,
          DATEMESS = 14#,
          TIMEMESS = 15#,
          ERRORMSG = 17#

;%
COMMENT DFFINES FOR FIRST WORD OF LOG RECDS;
  DEFINE MIXF    = [2:5]#,
          RLF     = [7:11]#,
          LINEF   = [8:8]#,

```

00001100	T	0000	
00001200	T	0000	
00001300	T	0000	
00001310	T	0000	
00001311	T	0000	
00001312	T	0000	
00001313	T	0000	
00001314	T	0000	
00001315	T	0000	
00001316	T	0000	
00001317	T	0000	
00001318	T	0000	
00001319	T	0000	
00001400	T	0000	
START OF SEGMENT	*****		2
00001500	T	0000	
00001600	T	0000	
00001700	T	0000	
00001800	T	0000	
00001900	T	0000	
00002000	T	0000	
00002100	T	0000	
00002200	T	0000	
00002300	T	0000	
00002400	T	0000	
00002500	T	0000	
00002600	T	0000	
00002700	T	0000	
00002800	T	0000	
00002900	T	0000	
00003000	T	0000	
00003100	T	0000	
00003200	T	0000	
00003300	T	0000	
00003400	T	0000	
00003500	T	0000	
00003600	T	0000	
00003700	T	0000	
00003800	T	0000	
00003900	T	0000	
00004000	P	0000	
00004010	C	0000	
00004100	T	0000	
00004200	T	0000	
00004300	T	0000	
00004400	T	0000	
00004500	T	0000	

McGraw-Hill Business Forms, Inc. 31171

```

TYPEF      = [16:7]#,
SPOF      = [23:1]#,
TIMEF     = [25:23]#;
COMMENT DEFINE REPORT NOS.;%
  DEFINE ONSITE = 0#,
  USERPT      = 2#,
  DISKRPT     = 1#;%
COMMENT LEVEL NAMES;%
  DEFINE SFSSION = 1#,
  USER         = 2#,
  REPORT       = 3#,
  PROGRAM      = 0#;%
COMMENT ASSORTED DEFINES;
  DEFINE UPSI   = SI:=SI+1#,
  UPDI         = DI:=DI+1#,
  UPTALLY      = TALLY:=TALLY+1#,
  GPMK         = "*"#,
  SCAN         = RESULT:=SCAN(PTR,PTR,ACCUM)#,
  NUMBERSCAN=NUMBERSCN(PTR,PTR)#,
  NAMEV        = 1#,
  NUMBERV      = 0#,
  PRNTS        = PRNT#,
  PAGESIZE     = 56#, % LINES PER PAGE
  DATEF        = [9:16] #, % DATE FIELD
  YEAR         = 77000 #, % THIS YEAR x 1000
  SCAN2        = SCAN; SCAN#
;%
  DEFINE LAPSE = (LASTDATE-FIRSTDATE)*5184000 #;
  DEFINE EQUIV(EQUIV1,EQUIV2)=
    REAL(BOOLEAN(EQUIV1) EQV BOOLEAN(EQUIV2))=REAL(NOT FALSE)#;%
COMMENT SORT RECORD SIZES;
  DEFINE OSRECSI = 10#,
  DLRECSI       = 8#,
  MAXSI         = 12#,
  MIXRECSI      = 12#;%
COMMENT SCRATCH FILE BLOCK SIZES;
  DEFINE OSBLKSI = 30xOSRECSI#,
  DLBLKSI       = 30xDLRECSI#
;%
COMMENT SCRATCH FILE ROW SIZES;
  DEFINE OSROWSI = 1000xOSRECSI#,
  DLROWSI       = 1000xDLRECSI#
;%
COMMENT ARRAY SIZES;
  DEFINE HLQSI   = 30#,
  FIDQSI        = 10#,
  NUMLINES      = 12#,
  MAXPARAMS     = 9#,
  MIXMAX        = 12#,
  NUMLINE       = NUMLINES#,
  MAXKEY        = 4#,
  MAXPARAM      = MAXPARAMS#;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
COMMENT ::::: SECTION = 02 = GLOBAL DECS :::::;%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
BOOLEAN COMBOO;% CARD/TTY INPUT
COMMENT GLOBAL ARRAYS;%

```

```

00004600 T 0000
00004700 T 0000
00004800 T 0000
00004900 T 0000
00005000 T 0000
00005100 T 0000
00005200 T 0000
00005300 T 0000
00005400 T 0000
00005500 T 0000
00005600 T 0000
00005700 T 0000
00005800 T 0000
00005900 T 0000
00006000 T 0000
00006100 T 0000
00006200 T 0000
00006300 T 0000
00006400 T 0000
00006500 T 0000
00006600 T 0000
00006700 T 0000
00006710 C 0000
00006720 C 0000
00006730 C 0000
00006800 T 0000
00006900 T 0000
00006910 C 0000
00007000 T 0000
00007100 T 0000
00007200 T 0000
00007300 T 0000
00007400 T 0000
00007500 T 0000
00007600 T 0000
00007700 T 0000
00007800 T 0000
00007900 T 0000
00008000 T 0000
00008100 T 0000
00008200 T 0000
00008300 T 0000
00008400 T 0000
00008500 T 0000
00008600 T 0000
00008700 T 0000
00008800 P 0000
00008900 P 0000
00009000 P 0000
00009100 T 0000
00009200 T 0000
00009300 P 0000
00009400 T 0000
00009500 T 0000
00009600 T 0000
00009610 T 0000
00009700 T 0000

```

```

ARRAY LEVELARRY[0:3],      % LEVEL NAMES                00009800 T 0000
ONOFFFARRY[0:1],          % ON/OFF LINE                00009900 T 0002
UNIT[1:31],               % UNIT MNEMONICS            00010000 T 0003
RTE[1:31],                % RUN TIME ERRORS THIS UNIT 00010100 T 0005
LINE[0:16],               % PRINT LINE                 00010200 T 0007
ULINE[0:16],              % RPTU PRINT LINE           00010300 T 0009
LLTBL[1:NUMLINE,0:3],    % LINE INFO-USER,LOG ON TIME 00010400 T 0010
FIDLINE[0:FIDQSI DIV 9 +1,0:16],% FILE ID PRNT LINES      00010500 T 0012
%
LINEARRY[0:NUMLINE],    % ACCUM LINE USAGE          00010600 P 0016
MIXARRY[0:MIXMAX],      % JOBS IN MIX                00010700 T 0016
HOLDKEY[0:MAXKEY],      % SORT KEY SAVE AREA        00010710 C 0018
PARAWDS[0:MAXPARAM],    % OPTIONS                     00010800 T 0020
DAYSIN[0:11],           % DAYS IN MONTH              00010900 T 0022
FIDQ[0:FIDQSI],         % QUEUE OF FILES TO PROCESS  00011000 T 0023
PSTOTALS[0:1],         % RPT PST TOTS,LOCAL AND REMOTE 00011100 T 0025
IOTOTALS[0:1],         % RPT IOT TOTS,LOCAL AND REMOTE 00011200 T 0027
ETOTALS[0:1],          % RPT ET TOTS,LOCAL AND REMOTE 00011300 T 0029
MERGEINPUT[0:7],       % MERGE INPUT RECORD         00011400 T 0030
CPLRNM[0:9],            % COMPILER NAMES             00011500 T 0032
COMPILE[1:9],           % NUMBER OF COMPILATIONS     00011510 C 0034
DUMMY[0:9],             % USED FOR CLEANER           00011520 C 0036
ASTERISK[0:3],          % ASTERISKS                   00011530 C 0037
INTEGER ARRAY HLQ[0:HLQSI], % QUEUE OF H/L TIMES        00011600 T 0039
HLDATE[1:HLQSI],       % DATES OF H/L S             00011610 C 0041
SAVE ARRAY LOGREC[0:MAXSI], % WORK RECORD                 00011620 C 0043
ARRAY MIXTBL[1:MIXMAX,0:9],%                               00011700 T 0044
COMMENT REAL GLOBALS(SIC),%
REAL GT1,GT2,GT3,GT4;    % GLOBAL TEMPORARIES        00011800 T 0046
REAL GT5,GT6,GT7,GT8,   %                               00011900 T 0048
GT9,GT10,GT11,GT12,    %                               00012000 T 0048
GT13,GT14,GT15;        % MORE OF THE SAME          00012100 T 0048
REAL GT16,GT17,GT18,GT19;%                               00012200 T 0048
REAL DAY ,              % "MON","TUES",ETC.         00012300 T 0048
DATE1,DATE2,DATE3,     % MM/DD/YY                   00012400 T 0048
UCT,                     % USER CANDE TIME           00012500 T 0048
UTT,                     % USER TERMINAL TIME        00012600 T 0048
RCT,                     % REPORT CANDE TIME         00012610 T 0048
RTT,                     % REPORT TERMINAL TIME      00012620 T 0048
TSESSIONS,              % REPORT CANDE TIME         00012630 T 0048
PACKETS,                % REPORT TERMINAL TIME      00012640 T 0048
MFID,                   % TERMINAL SESSIONS         00012650 C 0048
MERGEID,                % BATCH PACKETS             00012660 C 0048
PTR,                    % LOG FILEID                 00013100 T 0048
PTRBASE,                % MERGE FILE ID             00013200 T 0048
RESULT,                 % SCAN POINTER               00013300 T 0048
ACCUM                   % ADDR OF WORK RECORD       00013400 T 0048
;%                       % VALUE RETUREN BY SCANNER  00013500 T 0048
COMMENT GLOBAL INTEGERS FOR ASSORTED PURPOSES;
INTEGER HLP,             % SCANNER STUFF WORD        00013600 T 0048
FIDP,                   %                               00013700 T 0048
PAGENO,                 % INDEX INTO HLQ            00013800 T 0048
OSJOBS,                 % INDEX INT FIDQ            00013900 T 0048
TUJOBS,                 % REPORT PAGE NUMBER        00014000 T 0048
UPAGENO,                % NO. OF ONSITE JOBS       00014100 T 0048
LINECTR,                % NO. OF TERMINAL JOBS     00014200 T 0048
ULINECTR,               % RPTU PAGE NUMBER         00014300 T 0048
NFIDLINFS,              % RPTU PAGE LINE COUNTER   00014400 T 0048
% NO OF FILE ID PRNT LINES 00014500 T 0048
%                               00014600 T 0048
%                               00014700 T 0048

```

Moore Business Forms, Inc. sv 1-127

```

      REMUSERS,      % NO OR REMOTE USERS          00014800 T 0048
      REMUSE,       % 1 IFF THIS USER RAN A REM JOB    00014900 T 0048
      OSUSERS,     % NO OF ONSITE USERS              00015000 T 0048
      OSUSE,       % 1 IFF THIS USER RAN AN OS JOB    00015100 T 0048
      NL,          % NO OF LINES IN USE              00015200 T 0048
      MN,          %NO. OF JOBS IN MIX              00015210 C 0048
      COMLINES,    % NO OF LINES W/ NO LON OR OFF     00015300 T 0048
      FIRSDATE,    % FIRST DATE SEEN IN LOG          00015310 C 0048
      LASTDATE,    % LAST DATE SEEN IN LOG          00015311 C 0048
      BEGINTIME,   % EARLIEST TIME FOUND IN LOG     00015400 T 0048
      LASTIME,     % AS ABOVE,LATEST                00015500 T 0048
COMMENT GLOBAL BOOLEANS, MOSTLY REPORT OPTION TOGS;
      BOOLEAN TERMINALTOG, % TU SUMMARY RPT OUTPUT      00015600 T 0048
      ONSITETOG,    % ONSITE SUMMARY OUTPUT        00015700 T 0048
      DISKTOG,     % DISK LOG SUMMARY OUTPUT        00015800 T 0048
      HLTOG,       % H/L ANALYSIS                  00015900 T 0048
      LINETOG,     % LINE ANALYSIS                 00016000 T 0048
      MERGETOG,    % MERGED INPUT                  00016100 T 0048
      MERGEND,     % NO MORE MERGE INPUT           00016200 T 0048
      SUMTOG,      % SUMMARY OUTPUT                00016300 T 0048
      CLEANER,    % SORT RELEASE TOG              00016400 T 0048
      DETOG,      % PRINT DETAILS AD NAUSEUM       00016500 T 0048
      HLING,      % CLEANING AFTER A H/L           00016510 C 0048
      ERRTOG,     % DEVICE ERRORS ANALYSIS         00016520 C 0048
      ;%
COMMENT FILE DECLARATIONS;
      FILE IN DISK SERIAL(2,10,300),
      MERGEFILE DISK SERIAL(2,OSRECSI,OSBLKSI),
      PRNT 16(2,17),
      UPRT 16(2,17),
      TWX 19(1,10)
      ;%
      SAVF FILE OUT SUMFILE DISK SERIAL[20:OSROWSI]
      "USER" "TSS-LOG" (2,OSRECSI,OSBLKSI,SAVE 10);%
COMMENT SUMFILE GETS CONTENTS OF MERGE FILE, PLUS SUMMARY;
%*****
COMMENT ::::: SECTION - 03 - UTILITY PROCEDURES;
%*****
REAL STREAM PROCEDURE GETABSADDR(WD);
%*****
BEGIN
  DI:=WD; GETABSADDR:=DI;%
END GETABSADDR;%
      00016600 T 0048
      00016700 T 0048
      00016800 T 0048
      00016900 T 0048
      00017000 T 0051
      00017100 T 0055
      00017200 T 0059
      00017300 T 0062
      00017400 T 0065
      00017500 T 0066
      00017600 T 0067
      00017610 C 0073
      00017700 T 0073
      00017800 T 0073
      00017900 T 0073
      00018000 T 0073
      00018100 T 0073
      00018200 T 0075
      00018300 T 0075
      00018400 T 0075
      00018500 T 0075

%*****
REAL STREAM PROCEDURE DECV(N,P);%
  VALUE N,P;%
%*****
BEGIN SI:=LOC P; DI:=LOC DECV;%
  DI:=DI+d; DI:=DI-N; DS:=N DEC;%
END DECV;%
      00018600 T 0076
      00018700 T 0076
      00018800 T 0076
      00018900 T 0076
      00019000 T 0076
      00019100 T 0077
      00019200 T 0078

```

```

*****
INTEGER STREAM PROCEDURE OCV(WD,POS);%
  VALUE POS;%
  COMMENT CONVERTS 2 DEC AT WD,POS(POS) TO OCT;
*****
BEGIN SI:=WD; SI:=SI+POS;
  DI:=LOC OCV; DS:= 2 OCT;
END OCV;%

```

```

00019300 T 0079
00019400 T 0079
00019500 T 0079
00019600 T 0079
00019700 T 0079
00019800 T 0079
00019900 T 0080
00020000 T 0081

```

```

*****
INTEGER STREAM PROCEDURE SCN(PTRV, PTR, ACCUM);%
  VALUE PTRV;%
  COMMENT RETURNS 0 IFF STRING OF ALPHA,
  RETURNS THE SPEC CHAR IFF ONE FOUND;%
*****
BEGIN
  LABEL L;%
  SI:=PTRV;%
L: IF SC = " " THEN BEGIN UPSI; GO L END;
  IF SC = ALPHA THEN%
  BEGIN COMMENT NAMES;
    DI:=ACCUM; DS:=LIT"0"; DS:=7 LIT " "; DI:=DI-7;%
    7(DS:=CHR; IF SC =ALPHA THEN ELSE JUMP OUT);%
  END ELSE
  BEGIN COMMENT SPEC CHAR;
    DI:=LOC SCN ; DI:=DI+7;
    DS:=CHR;%
  END SPEC CHAR HANDLING;
  DI:=PTR; PTRV:=SI; SI:=LOC PTRV; DS:=WDS;
END SCAN;%

```

```

00020100 T 0082
00020200 T 0082
00020300 T 0082
00020400 T 0082
00020500 T 0082
00020600 T 0082
00020700 T 0082
00020800 T 0083
00020900 T 0083
00021000 T 0083
00021100 T 0085
00021200 T 0086
00021300 T 0086
00021400 T 0088
00021500 T 0091
00021600 T 0091
00021700 T 0091
00021800 T 0091
00021900 T 0092
00022000 T 0092
00022100 T 0093

```

```

*****
REAL STREAM PROCEDURE NUMBERSCN(PTRV, PTR);%
  VALUE PTRV;%
  COMMENT SCANS OFF NEXT NUMBER, CONVERTS TO REAL;
*****
BEGIN LABEL L;
  LOCAL A;%
  SI:=PTRV; DI:= LOC NUMBERSCN;%
L: IF SC LSS "0" THEN BEGIN UPSI; GO L END;%
  A:=SI;
  7(UPSI; UPTALLY;
  IF SC LSS "0" THEN JUMP OUT);
  SI:=A; A:=TALLY; DS:=A OCT;
  A:=SI; DI:=PTR; SI:=LOC A; DS:=WDS;%
END NUMBERSCN;%

```

```

00022200 T 0094
00022300 T 0094
00022400 T 0094
00022500 T 0094
00022600 T 0094
00022700 T 0094
00022800 T 0094
00022900 T 0094
00023000 T 0094
00023100 T 0096
00023200 T 0096
00023300 T 0097
00023400 T 0099
00023500 T 0100
00023600 T 0101

```

```

*****

```

```

00023700 T 0102

```

Moore Business Forms, Inc. sv 10/1/71

```

BOOLEAN STREAM PROCEDURE COMPARE(N,A,B);%
  VALUE N;
  COMMENT RETURNS TRUE IFF ARRAY A SORTS LOWER THAN B;
%*****
BEGIN LABEL XIT;
  LOCAL SA,DA;%
  SI:=A; DI:=B;
  N(UPSI;UPDI;
  SA:=SI; DA:=DI;%
  IF 7 SC GTR DC THEN
  BEGIN TALLY:=0; JUMP OUT TO XIT END;%
  DI:=DA; SI:=SA;
  IF 7 SC LSS DC THEN
  BEGIN TALLY:=1; JUMP OUT TO XIT END;%
  );% EQUAL, REPEAT
  TALLY:=1;%
XIT:COMPARE:=TALLY;%
END COMPARE;%

```

```

00023800 T 0102
00023900 T 0102
00024000 T 0102
00024100 T 0102
00024200 T 0102
00024300 T 0102
00024400 T 0102
00024500 T 0102
00024600 T 0104
00024700 T 0104
00024800 T 0105
00024900 T 0106
00025000 T 0106
00025100 T 0107
00025200 T 0108
00025300 T 0108
00025400 T 0109
00025500 T 0109

```

```

%*****
BOOLEAN STREAM PROCEDURE EQUAL(N,A,B);%
  VALUE N;%
  COMMENT RETURNS TRUE IFF ARRAY A=B FOR N WORDS;
%*****
BEGIN LABEL XIT;
  SI:=A; DI:=B;%
  N(UPSI;UPDI;
  IF 7 SC NEQ DC THEN
  BEGIN TALLY:=0; JUMP OUT TO XIT END);%
  TALLY:=1;%
XIT:EQUAL:=TALLY;%
END EQUAL;%

```

```

00025600 T 0110
00025700 T 0110
00025800 T 0110
00025900 T 0110
00026000 T 0110
00026100 T 0110
00026200 T 0111
00026300 T 0111
00026400 T 0113
00026500 T 0113
00026600 T 0115
00026700 T 0115
00026800 T 0116

```

```

%*****
STREAM PROCEDURE MOVE(N,A,B); VALUE N;%
%*****
BEGIN S1:=A; DI:=B; DS:=N WDS END MOVE;%

```

```

00026900 T 0117
00027000 T 0117
00027100 T 0117
00027200 T 0117

```

```

%*****
INTEGER PROCEDURE TIMER(T);%
  VALUE T; REAL T;%
COMMENT : TAKES BINARY TIME IN T IN 1/60 SECONDS AND
  CONVERTS AS FOLLOWS :
  JOB TIME(T) LSS 1 MINUTE;
    TIMER[12:12] ::= "0M"
    TIMER[24:12] ::= SECONDS
    TIMER[36:12] ::= 1/60 SECONDS
  JOB TIME(T) GTR 1 MINUTE;

```

```

00027300 T 0119
00027400 T 0119
00027500 T 0119
00027600 T 0119
00027700 T 0119
00027800 T 0119
00027900 T 0119
00028000 T 0119
00028100 T 0119
00028200 T 0119

```

Moore Business Forms, Inc. sv 1121

```

TIMER[12:12] ::= HOURS
TIMER[24:12] ::= MINUTES
TIMER[36:12] ::= SECONDS;%
%*****
BEGIN INTFGR H,M,S,SS,A;%

  A:=T;%
  H:=A DIV 216000; A:=A MOD 216000;
  M:=A DIV 3600; A:=A MOD 3600;%
  S:=A DIV 60; SS:=T MOD 60;%
  H:=DECV(3,H);%
  M:=DECV(2,M);%
  S:=DECV(2,S);%
  SS:=DECV(2,SS);%
  IF(H+M)=0 THEN
    TIMER:=SS&S[24:36:12]&"0M"[12:36:12] ELSE
    TIMER:=SRM[24:36:12]&H[12:36:12];%
  END TIMER;%

```

```

00028300 T 0119
00028400 T 0119
00028500 T 0119
00028600 T 0119
00028700 T 0119
START OF SEGMENT ***** 3
00028800 T 0000
00028900 T 0000
00029000 T 0003
00029100 T 0005
00029200 T 0008
00029300 T 0010
00029400 T 0012
00029500 T 0014
00029600 T 0016
00029700 T 0017
00029800 T 0020
00029900 T 0026
3 IS 31 LONG, NEXT SEG 2

```

```

%*****
PROCEDURE STIMER(T,WD1,WD2);%
  VALUE T;%
  INTEGER T,WD1,WD2;
  COMMENT CONVERTS TIME(T) AS FOLLOWS;
  %   WD1[30:12] ::= DD
  %   WD1[42:6]  ::= ":"
  %   WD2[24:12] ::= DD
  %   WD2[30:6]  ::= ":"
  %   WD2[36:12] ::= DD
%*****
BEGIN
  INTEGER I;

  WD1:=":"; WD2:"00:00";%
  WD1:=WD1&(I:=TIMER(T))[30:12:12];%
  WD2:=WD2&I[18:24:12]&I[36:36:12];%
END STIMER;

```

```

00030000 T 0119
00030100 T 0119
00030200 T 0119
00030300 T 0119
00030400 T 0119
00030500 T 0119
00030600 T 0119
00030700 T 0119
00030800 T 0119
00030900 T 0119
00031000 T 0119
00031100 T 0119
00031200 T 0119
START OF SEGMENT ***** 4
00031300 T 0000
00031400 T 0002
00031500 T 0005
00031600 T 0008
4 IS 12 LONG, NEXT SEG 2

```

```

%*****
INTEGER PROCEDURE MATCH(V,A);%
  VALUE V;
  REAL V; ARRAY A[0];%
  COMMENT LOOKS TO SEE IF VALUE V OCCURS IN ARRAY A,
  IF IT DOES RETURNS INDEX, IF NOT RETURNS 0;
%*****
BEGIN INTEGER NAE,I;%

  LABEL XIT;%
  NAE:=A[0]; MATCH:=0;%

```

```

00031700 T 0119
00031800 T 0119
00031900 T 0119
00032000 T 0119
00032100 T 0119
00032200 T 0119
00032300 T 0119
00032400 T 0119
START OF SEGMENT ***** 5
00032500 T 0000
00032600 T 0000

```

```

FOR I:=1 STEP 1 UNTIL NAE DO
  BEGIN
    IF V=A[I] THEN
      BEGIN MATCH:=I; GO XIT END;%
    END LOOP;%
  XIT:
  END MATCH;%

```

```

00032700 T 0001
00032800 T 0003
00032900 T 0003
00033000 T 0004
00033100 T 0005
00033200 T 0008
00033300 T 0008
5 IS 11 LONG, NEXT SEG 2

```

```

%*****
PROCEDURE PRINTLINE(LINE,WHICH);%
  VALUE WHICH; INTEGER WHICH;%
  ARRAY LINE[0];%
%*****
BEGIN
  SWITCH FILE SWF:=PRNT,UPRNT;%

  IF WHICH=USERPT THEN ULINECTR:=ULINECTR+1 ELSE
    LINECTR:=LINECTR+1;
    WRITE(SWF[WHICH DIV 2],15,LINE[*]);%
  END PRINTLINE;%

```

```

00033400 T 0119
00033500 T 0119
00033600 T 0119
00033700 T 0119
00033800 T 0119
00033900 T 0119
00034000 T 0119
START OF SEGMENT ***** 6
00034100 P 0004
00034200 P 0006
00034300 P 0008
00034400 T 0013
6 IS 18 LONG, NEXT SEG 2

```

```

%*****
STREAM PROCEDURE BLANKOUT(LINE);%
%*****
BEGIN
  DI:=LINE; DS:= 8 LIT" ";
  SI:=LINE; DS:=16 WDS;
  END BLANKOUT;%

```

```

00034500 T 0119
00034600 T 0119
00034700 T 0119
00034800 T 0119
00034900 T 0120
00035000 T 0121
00035100 T 0122

```

```

%*****
STREAM PROCEDURE CLEAR(N,A);
  VALUE N;%
  COMMENT SET N+1 WORDS TO ZERO;
%*****
BEGIN
  DI:=A; DS:=8 LIT"0";
  SI:=A; DS:=N WDS;
  END CLEAR;%

```

```

00035200 T 0122
00035300 T 0122
00035400 T 0122
00035500 T 0122
00035600 T 0122
00035700 T 0122
00035800 T 0123
00035900 T 0124
00036000 T 0125

```

```

%*****
PROCEDURE NEXTPAGE(WHICH);%
  VALUE WHICH; INTEGER WHICH;%
%*****

```

```

00036100 T 0125
00036200 T 0125
00036300 T 0125
00036400 T 0125

```


Moore Business Forms, Inc. sv 1412T

```

BEGIN
  REAL I;%
  SWITCH FILE SWF:=PRNT,UPRNT;%
  FORMAT HF(X26,"** B=5500 TIME SHARING SYSTEM LOG ",
  "REPORT **",X7,A6,"DAY  ",,2(I2,"/"),I2,X11
  ,"PAGE ",I5);
  LIST HL(DAY,DATE1,DATE2,DATE3,
  IF WHICH=USERPT THEN UPAGENO:=UPAGENO+1 ELSE
  PAGENO:=PAGENO+1);%
  WRITE(SWF[WHICH DIV 2][PAGE]);%
  WRITE(SWF[WHICH DIV 2][DBL],HF,HL);%
  IF WHICH=USERPT THEN ULINFCTR:=6+NFIDLINES ELSE
  LINECTR:=6+NFIDLINES;
  FOR I:=0 STEP 1 UNTIL NFIDLINES
  DO PRINTLINE(FIDLINE[I,*],WHICH);%
END NEXTPAGE;%

```

```

%*****
PROCEDURE PUTHEADER(WHICH);%
  VALUE WHICH;%
  INTEGER WHICH;
%*****
BEGIN
  SWITCH FILE SWFILE:=PRNT,UPRNT;%
  SWITCH FORMAT SWF:=
  (X51,"U S E R  J O B  S U M M A R Y"),
  (X50,"U S E R  D I S K  S U M M A R Y"),%
  (X55,"U S E R  S U M M A R Y");%
  SWITCH FORMAT SWFF:=
  ("USER",X4,"PN1",X5,"PN2",X5,
  "LL",X5,"BOJ",X7,"EOJ",X7,
  "PST",X7,"IOT",X4,"ET/TT",X9,
  "CORE",X1,"TRMNTN",X2,"MFID",X4,
  "FID",X6,"UNIT",X4,"TIME"),
  ("USER",X5,"MFID",X5,"FID",X6,
  "CREATN DATE & TIME",X1,"LOGOFF TIME",X2,
  "# DISK SEGS",X5,"TIME USED");%
  ("USER",X27,"PST",X7,"IOT",X7,"ET",X8,"C&E",X4,
  "TERMINAL",X1,"CORE/SECS");%
  NEXTPAGE(WHICH);%
  WRITE(SWFILE[WHICH DIV 2][DBL],SWF[WHICH]);
  WRITE(SWFILE[WHICH DIV 2][DBL],SWFF[WHICH]);%
END PUTHEADER;%

```

00036500	T	0125		
00036600	T	0125		
START OF SEGMENT	*****		7	
00036700	T	0000		
00036800	T	0004		
START OF SEGMENT	*****		6	
00036900	T	0004		
00037000	T	0004		
8 IS	23 LONG,	NEXT SEG	7	
00037100	T	0004		
00037200	T	0010		
00037300	T	0014		
00037400	T	0019		
00037500	T	0024		
00037600	P	0028		
00037700	P	0030		
00037800	T	0032		
00037900	T	0032		
00038000	T	0038		
7 IS	42 LONG,	NEXT SEG	2	
00038100	T	0125		
00038200	T	0125		
00038300	T	0125		
00038400	T	0125		
00038500	T	0125		
00038600	T	0125		
00038700	T	0125		
START OF SEGMENT	*****		9	
00038800	T	0004		
START OF SEGMENT	*****		10	
00038900	T	0004		
00039000	T	0004		
00039100	T	0004		
10 IS	30 LONG,	NEXT SEG	9	
00039200	T	0004		
START OF SEGMENT	*****		11	
00039300	T	0004		
00039400	T	0004		
00039500	T	0004		
00039600	T	0004		
00039700	T	0004		
00039800	T	0004		
00039900	T	0004		
00040000	T	0004		
00040100	T	0004		
00040200	T	0004		
11 IS	74 LONG,	NEXT SEG	9	
00040300	T	0004		
00040400	T	0005		
00040500	T	0009		
00040600	T	0014		
9 IS	19 LONG,	NEXT SEG	2	

Moore Business Forms, Inc. sv

```

*****
BOOLEAN PROCEDURE DISKREAD;%
*****
BEGIN
  LABEL GET,EOF,THRU;

GET:READ(DISK,10,LOGREC[*])[EOF];%
  IF LOGREC[0]=REAL(NOT FALSE) THEN GO TO EOF;
  IF LOGREC[0].TYPEF LSS BOJ THEN GO GET;
  PTR:=PTRBASE;
  IF LOGREC[0].TYPEF NEQ DATEMESS THEN GO THRU;
  DO UNTIL SCAN = "("; LASTDATE:=NUMBERSCAN-YEAR;
  IF FIRSTDATE=0 THEN BEGIN FIRSTDATE:=LASTDATE;
  FIRSTDATE:=FIRSTDATE + BEGINTIME DIV 5184000;
  BEGINTIME:=BEGINTIME MOD 5184000 END;
  PTR := PTRBASE; % ALLOWING RECORD TO BE SCANNED AGAIN
  GO THRU;
EOF:IF FIDQ[FIDP:=FIDP+1] NEQ 0 THEN
  BEGIN CLOSE(DISK);
  MFID:=FIDQ[FIDP];%
  FILL DISK WITH MFID;%
  GO TO GET;
  END;%
DISKREAD:=TRUE;
THRU:
END DISKREAD;

```

00040700	T	0125	
00040800	T	0125	
00040900	T	0125	
00041000	T	0125	
00041100	T	0125	
START OF SEGMENT	*****		12
00041200	T	0000	
00041300	T	0005	
00041400	T	0007	
00041500	T	0009	
00041600	P	0009	
00041610	C	0011	
00041620	C	0018	
00041625	C	0020	
00041630	C	0021	
00041635	C	0023	
00041640	C	0023	
00041700	T	0027	
00041800	T	0029	
00041900	T	0031	
00042000	T	0032	
00042100	T	0036	
00042200	T	0036	
00042300	T	0036	
00042400	T	0037	
00042500	T	0038	
12 IS	43 LONG,	NEXT SEG	2

```

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
COMMENT ::::: SECTION - 04 - SNAP REPORTS :::::
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
*****
PROCEDURE HLANA;%
*****
BEGIN
  FORMAT F1("THERE WERE ",I3, " HALT LOADS");%

  FORMAT F2("HALT LOAD TIMES WERE : ");
  F3(X2,A3,A5,X2,I6);
  F4("MEAN TIME BETWEEN HALT LOADS...",A3,A5);

  FORMAT F5("TIME PERIOD COVERED WAS FROM ",A3,A5,
  X2,I6," TO ",A3,A5,X2,I6);

  REAL A,B,C,D,I;%
  LABEL XIT;%
  %
  STIMER(BEGINTIME,A,B); STIMER(LASTIME,C,D);%

```

00042600	T	0125	
00042700	T	0125	
00042800	T	0125	
00042900	T	0125	
00043000	T	0125	
00043100	T	0125	
00043200	T	0125	
00043300	T	0125	
START OF SEGMENT	*****		13
START OF SEGMENT	*****		14
14 IS	8 LONG,	NEXT SEG	13
00043400	T	0000	
START OF SEGMENT	*****		15
00043500	P	0000	
00043600	T	0000	
15 IS	26 LONG,	NEXT SEG	13
00043700	T	0000	
START OF SEGMENT	*****		16
00043800	P	0000	
16 IS	17 LONG,	NEXT SEG	13
00043900	T	0000	
00044000	T	0000	
00044100	P	0000	
00044200	T	0000	

```

WRITE(PRNT,F5,A,B,FIRSTDATE+YEAR,C,D,LASTDATE+YEAR);
WRITE(PRNT[DBL],F1,HLP);%
IF HLP = 0 THEN GO XIT;%
WRITE(PRNT[DBL],F2);%
FOR I:=1 STEP 1 UNTIL HLP DO
BEGIN COMMENT H/L TIMES;
    STIMER(HLQ[I],A,B);
    WRITE(PRNT[DBL],F3,A,B,HLDATE[I]+YEAR);
END PRINTING HL TIMES;
STIMER((LASTIME-BEGINTIME+LAPSE)/HLP,A,B);
WRITE(PRNT[DBL],F4,A,B);%
XIT;
END HLANA;%

```

```

00044300 P 0002
00044400 T 0022
00044500 T 0030
00044600 T 0031
00044700 T 0034
00044800 T 0036
00044900 T 0036
00045000 P 0037
00045100 T 0052
00045200 P 0054
00045300 T 0058
00045400 T 0069
00045500 T 0070
13 IS 74 LONG, NEXT SEG 2

```

```

%*****
PROCEDURE LINEANA;%
    COMMENT PRINTS OUT THE AMOUNTS AND %S OF TIME
    THAT I LINES WERE IN USE;%
%*****
BEGIN BOOLEAN FIRST;%

```

```

    FORMAT F1("NO. OF LINES IN USE    TIME    %TOTAL TIME");

```

```

    FORMAT F2(X11,I4,X4,A3,A5,X3,F8,2);%

```

```

REAL I,L1,L2;%
REAL MAX,T,TT;%
LABEL XIT;%
WRITE(PRNT[DBL],F1);%
FOR I:=NUMLINES STEP -1 UNTIL 0 DO
BEGIN T:=LINEARRY[I];
    TT:=TT+T; IF T NEQ 0 THEN
        IF NOT FIRST THEN BEGIN FIRST:=TRUE;MAX:=I END;
END;
IF TT=0 THEN GO TO XIT;%
FOR I:=0 STEP 1 UNTIL MAX DO
BEGIN
    STIMER(T:=LINEARRY[I],L1,L2);%
    WRITE(PRNT[DBL],F2,I+COMLINES,L1,L2,100*(T/TT));%
END;

```

```

XIT;
END LINEANA;%

```

```

00045600 T 0125
00045700 T 0125
00045800 T 0125
00045900 T 0125
00046000 T 0125
00046100 T 0125
START OF SEGMENT ***** 17
00046200 T 0000
START OF SEGMENT ***** 18
18 IS 10 LONG, NEXT SEG 17
00046300 T 0000
START OF SEGMENT ***** 19
19 IS 10 LONG, NEXT SEG 17
00046400 T 0000
00046500 T 0000
00046600 T 0000
00046700 P 0000
00046800 T 0003
00046890 C 0004
00046900 P 0005
00047000 T 0007
00047100 T 0010
00047200 T 0012
00047300 T 0013
00047400 T 0015
00047500 T 0015
00047600 T 0017
00047700 T 0031
00047800 T 0033
00047900 T 0034
17 IS 38 LONG, NEXT SEG 2

```

```

PROCEDURE MIXANA;
    COMMENT PRINTS OUT THE NUMBER OF JOBS IN MIX A LA LINEANA;
BEGIN BOOLEAN FIRST;

```

```

00047901 C 0125
00047902 C 0125
00047903 C 0125
START OF SEGMENT ***** 20

```

Moore Business Forms, Inc. sv 1412T

FORMAT F1("NO. OF JOBS IN MIX TIME %TOTAL TIME");

FORMAT F2(X11,I4,X4,A3,A5,X5,F8.2);

```

REAL I,L1,L2,MAX,T,TT;
LABFL XIT;
WRITE(PRNT[DBL],F1);
FOR I:=MIXMAX STEP -1 UNTIL 0 DO
  BEGIN T:=MIXARRY[I];
        TT:=TT+T; IF T NEQ 0 THEN
          IF NOT FIRST THEN BEGIN FIRST:=TRUE; MAX:=I END;
        END;
        IF TT=0 THEN GO TO XIT;
        FOR I:=0 STEP 1 UNTIL MAX DO
          BEGIN
            STIMER(T:=MIXARRY[I],L1,L2);
            WRITE(PRNT[DBL],F2,I,L1,L2,100*(T/TT));
          END;
        XIT;
      END MIXANA;

```

```

00047904 C 0000
START OF SEGMENT ***** 21
21 IS 10 LONG, NEXT SEG 20
00047905 C 0000
START OF SEGMENT ***** 22
22 IS 10 LONG, NEXT SEG 20
00047908 C 0000
00047909 C 0000
00047910 C 0000
00047912 C 0003
00047913 C 0004
00047915 C 0005
00047916 C 0007
00047918 C 0010
00047920 C 0012
00047922 C 0013
00047924 C 0015
00047926 C 0015
00047928 C 0017
00047930 C 0030
00047932 C 0032
00047934 C 0033
20 IS 37 LONG, NEXT SEG 2

```

```

*****
PROCEDURE ERRANA;
  COMMENT LISTS THE NO. OF ERRORS PER PERIPHERAL;
*****:
BEGIN FORMAT FT("PERIPHERAL DEVICE ERRORS WERE DISTRIBUTED",

```

```

" AS FOLLOWS :",/, "LUN",X4,"UNIT",X10,"NOE");%
FORMAT F(I3,X2,A6,X5,I8);%

```

```

REAL I;
NEXTPAGE(ONSITE);%
WRITE(PRNT[DBL],FT);%
FOR I:=1 STEP 1 UNTIL 31 %
DO IF RTE[I] NEQ 0 THEN WRITE(PRNT[DBL],F,I-1,UNIT[I],RTE[I]);%
END ERRANA;

```

```

00048000 T 0125
00048100 T 0125
00048200 T 0125
00048300 T 0125
00048400 T 0125
START OF SEGMENT ***** 23
START OF SEGMENT ***** 24
00048500 T 0000
24 IS 19 LONG, NEXT SEG 23
00048600 T 0000
START OF SEGMENT ***** 25
25 IS 8 LONG, NEXT SEG 23
00048700 T 0000
00048800 T 0000
00048900 T 0000
00049000 T 0003
00049100 T 0003
00049200 T 0022
23 IS 25 LONG, NEXT SEG 2

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
COMMENT :::: SECTION - 05 - ONSITE REPORTER ::::;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
COMMENT LAYOUT OF THE ONSITE SORT RECORDS;;
%      S[0] ::= USER ID
%      S[1] ::= LINE NO.
%      S[2] ::= PROG NAME 1

```

```

00049300 T 0125
00049400 T 0125
00049500 T 0125
00049600 T 0125
00049700 T 0125
00049800 T 0125
00049900 T 0125

```

Moore Business Forms, Inc. sv

```

%      S[3] ::= PROG NAME 2
%      S[4] ::= BOJ TIME
%      S[5] ::= EOJ TIME
%      S[6] ::= CPU TIME USED
%      S[7] ::= I/O TIME USED
%      S[8] ::= CORE USED
%      S[9] ::= "DS-ED","EOJ",ETC,
COMMENT LAYOUT OF THE FILE STAT SORT RECORD;
%      F[0]-F[3] ::= S[0]-S[3]
%      F[4] ::= CLOSE STAT TIME
%      F[5]-F[9] ::= SAME AS LOG RECORD
COMMENT LAYOUT OF THE TERMINAL LOGGING RECORD;
%      T[0] ::= USER ID
%      T[1] ::= LINE NO
%      T[2] ::= "LINE &"
%      T[3] ::= "C&E USE"
%      T[4] ::= TIME LOGGED ON
%*****
STREAM PROCEDURE MAKEFILEDETAILLINE(LINE,MFID,FID,UNIT,TIM);%
  VALUE MFID,FID,UNIT,TIM;%
%*****
BEGIN
  DI:=LINE;DI:=DI+60;DI:=DI+35;%
  SI:=LOC MFID; 2(UPSI;DS:=7 CHR; UPDI); % MFID,FID
  S1:=SI+5; DI:=DI+2;DS:=3 CHR; % UNIT
  SI:=SI+2; DI:=DI+2;%
  2(DS:=2 CHR; DS:=LIT";"); DS:=2 CHR;%
END MAKEFILEDETAILLINE;%

```

```

00050000 T 0125
00050100 T 0125
00050200 T 0125
00050300 T 0125
00050400 T 0125
00050500 T 0125
00050600 T 0125
00050700 T 0125
00050800 T 0125
00050900 T 0125
00051000 T 0125
00051100 T 0125
00051200 T 0125
00051300 T 0125
00051400 T 0125
00051500 T 0125
00051600 T 0125
00051700 T 0125
00051800 T 0125
00051900 T 0125
00052000 T 0125
00052100 T 0125
00052200 T 0126
00052300 T 0126
00052400 T 0128
00052500 T 0129
00052600 T 0129
00052700 T 0131

```

```

%*****
PROCEDURE FILEDETAIL(SORTREC);
  ARRAY SORTREC[0];%
  COMMENT FORMATS THEN FILE RECORD PRINTLINE;%
%*****
BEGIN
  DEFINE S=SORTREC#;%

  REAL LUN;%
  LABEL XIT;
  IF(LUN:=S[8].[36:6])=0 THEN GO TO XIT;%
  RTE[LUN]:=RTE[LUN]+S[8].[24:12];% RUN TIME ERRORS
  BLANKOUT(LINE);
  MAKEFILEDETAILLINE(LINE,S[5],S[6],UNIT[LUN],
  TIMR(CABS(S[9])));%
  IF DETOG THEN
    PRINTLINE(LINE,ONSITE);%
XIT:
END FILEDETAIL;%

```

```

00052800 T 0131
00052900 T 0131
00053000 T 0131
00053100 T 0131
00053200 T 0131
00053300 T 0131
00053400 T 0131
START OF SEGMENT ***** 26
00053500 T 0000
00053600 T 0000
00053700 T 0000
00053800 T 0002
00053900 T 0006
00054000 T 0007
00054100 T 0010
00054190 C 0011
00054200 T 0011
00054300 T 0013
00054400 T 0014
26 IS 17 LONG, NEXT SEG 2

```

```

%***** 00054500 T 0131

```

Moore Business Forms, Inc. sv

```

STREAM PROCEDURE MAKELEVELLINE(LINE,ASTERISK,LEVEL,
  PST,IOT,ET,COR);
  VALUE PST,IOT,ET,COR,ASTERISK,LEVEL;%
%*****
BEGIN SI:=LOC ASTERISK; DI:=LINE;%
  DI:=DI+18; 2(UPDI;UPSI;DS:=7 CHR);%
  DS:=12 LIT" TOTALS....";%
  3(SI:=SI+2;DI:=DI+2;%
  2(DS:=2 CHR; DS:=LIT"");%
  DS:=2 CHR);%
  DI:=DI+2; DS:= 8 DEC;%
  DI:=DI-8;DS:=8 FILL;%
END MAKELEVELLINE;%

```

```

00054600 T 0131
00054700 T 0131
00054800 T 0131
00054900 T 0131
00055000 T 0131
00055100 T 0132
00055200 T 0134
00055300 T 0135
00055400 T 0136
00055500 T 0137
00055600 T 0138
00055700 T 0138
00055800 T 0139

```

```

%*****
PROCEDURE LEVEL(WHICH,PST,IOT,ET,COR);
  VALUE WHICH,PST,IOT,ET,COR;%
  INTEGER WHICH;
  REAL PST,IOT,ET,COR;%
%*****
BEGIN
  BLANKOUT(LINE);
  MAKELEVELLINE(LINE,ASTERISK[WHICH],LEVELARRY[WHICH],
  TIMER(PST),TIMER(IOT),TIMER(ET),COR);%
  IF DETOG THEN
    PRINTLINE(LINE,ONSITE);%
END LEVEL;%

```

```

00055900 T 0139
00056000 T 0139
00056100 T 0139
00056200 T 0139
00056300 T 0139
00056400 T 0139
00056500 T 0139
00056600 T 0139
00056700 T 0141
00056800 T 0142
00056890 C 0145
00056900 T 0145
00057000 T 0147

```

```

%*****
PROCEDURE SPLIT(I); VALUE I;%
  REAL I;%
%*****
BEGIN DEFINE P=GT10#,I0=GT11#,E=GT12#;%

  PSTOTALS[I]:=PSTOTALS[I]+P;%
  IOTOTALS[I] := IOTOTALS[I] + I0;%
  ETOTALS[I]:=ETOTALS[I]+E;%
  IF I=0 THEN OSUSE:=1 ELSE REMUSE:=1;%
END SPLIT;%

```

```

00057100 T 0148
00057200 T 0148
00057300 T 0148
00057400 T 0148
00057500 T 0148
START OF SEGMENT ***** 27
00057600 T 0000
00057700 T 0002
00057800 T 0004
00057810 T 0006
00057900 T 0009
27 IS 10 LONG, NEXT SEG 2

```

```

%*****
STREAM PROCEDURE MADETAILLINE(LINE,USER,PN1,PN2,LL,BOJT,
  EOJT,PST,IOT,ET,COR,HEND);%
  VALUE USER,PN1,PN2,LL,BOJT,EOJT,PST,IOT,ET,COR,HEND;%
%*****
BEGIN
  SI:=LOC USER; DI:=LINE;

```

```

00058000 T 0148
00058100 T 0148
00058200 T 0148
00058300 T 0148
00058400 T 0148
00058500 T 0148
00058600 T 0148

```

```

3(UPSI; DS:=7 CHR; UPDI);          % USER,PN1,PN2
SI:=SI+6;DS:=2 CHR; DI:=DI+2;% LINE NO
5(SI:=SI+2; 2(DS:=2 CHR; DS:=LIT";"); DS:=2 CHR;
DI:=DI+2);%
DS:=8 DEC; DI:=DI+2;
UPSI;DS:=5 CHR;%
DI:=DI-15; DS:= 5 FILL;%
END MAKEDETAILLINE;%

```

```

00058700 T 0148
00058800 T 0149
00058900 T 0150
00059000 T 0152
00059100 T 0153
00059200 T 0153
00059300 T 0154
00059400 T 0154

```

```

%*****
PROCEDURE DETAIL(USER,PN1,PN2,LL,BOJT,EOJT,PST,IOT,ET,
COR,HEND);%
VALUE USER,PN1,PN2,LL,BOJT,EOJT,PST,IOT,ET,COR,HEND;%
REAL USER,PN1,PN2,BOJT,EOJT,PST,IOT,ET,COR,HEND,LL;%
%*****

```

```

00059500 T 0154
00059600 T 0154
00059700 T 0154
00059800 T 0154
00059900 T 0154
00060000 T 0154
00060100 T 0154
00060110 C 0154

```

```

BEGIN
INTEGER I;

BLANKOUT(LINE);%
MAKEDETAILLINE(LINE,USER,PN1,PN2,LL,TIMER(BOJT MOD 5184000),
TIMER(EOJT MOD 5184000),TIMER(PST),TIMER(IOT),TIMER(ET),
COR,HEND);%
IF I:=MATCH(PN1,CPLRNM)>0 THEN COMPILS[I]:=COMPILS[I]+1;
IF DETOG THEN
PRINTLINE(LINE,ONSITE);%
END DETAIL;%

```

```

START OF SEGMENT ***** 28
00060200 T 0000
00060300 P 0001
00060400 P 0004
00060500 T 0007
00060510 C 0008
00060590 C 0014
00060600 T 0014
00060700 T 0016
28 IS 20 LONG, NEXT SEG 2

```

```

COMMENT LAYOUT OF THE USER MASTER RECORD;%
% U[0] ::= USER
% U[1] ::= CPU TIME
% U[2] ::= I-O TIME
% U[3] ::= ELAPSED TIME
% U[4] ::= CANDE CPU TIME
% U[5] ::= TERMINAL TIME
% U[6] ::= CORE
% U[7] ::= DISK SPACE
%*****

```

```

00060900 T 0154
00061000 T 0154
00061100 T 0154
00061200 T 0154
00061300 T 0154
00061400 T 0154
00061500 T 0154
00061600 T 0154
00061700 T 0154
00061710 T 0154
00061800 T 0154
00061900 T 0154
00062000 T 0154
00062100 T 0154
00062200 T 0154
00062210 T 0154
00062300 T 0154

```

```

PROCEDURE REPORTU(USER,PST,IOT,ET,CET,TT,COR);%
VALUE PST,IOT,ET,CET,TT,COR,USER;%
REAL PST,IOT,ET,CET,TT,COR,USER;%
COMMENT OUTPUTS THE REPORT WITH USER TOALS AS
THE DETAIL;
%*****

```

```

START OF SEGMENT ***** 29
00062400 T 0000
00062500 T 0000
00062600 T 0000
00062700 P 0000
00062800 T 0002

```

```

BEGIN INTEGER I,J;%

LABEL EOF,XIT;%
DEFINE INPUTREADY=EQUIV(M[0],0) AND NOT MERGEND#;%
DEFINE M=MERGEINPUT#;%
IF ULINCTR GTR PAGESIZE THEN PUTHEADER(USERPT);
MAKEDETAILLINE(ULINE,USER,"CURRENT","TOTALS ","*");

```

11/27
Moore Business Forms, Inc. sv

```

TIMER(PST),TIMER(IOT),TIMER(ET),TIMER(CET),
TIMER(TT),DECV(8,I:=(COR/1024)*(PST+IOT)), " ");%
PRINTLINE(ULINE,USERPT);%
BLANKOUT(ULINE);%
IF MERGETOG THEN IF INPUTREADY THEN
DO BEGIN READ(MERGEFILE,8,M[*])[EOF]; WRITE(SUMFILE,8,M[*]) END
UNTIL COMPARE(1,USER,M[0]);%
IF MERGETOG THEN
IF EQUIV(USER,M[0]) THEN
BEGIN COMMENT HIT;
  MAKEDETAILLINE(ULINE,USER,"MERGE  ","TOTALS  ","**",
  TIMER(M[1]),TIMER(M[2]),TIMER(M[3]),TIMER(M[4]),
  TIMER(M[5]),DECV(8,J:=(M[6]/1024)*(M[1]+M[2])), " ");%
  MAKEDETAILLINE(ULINE,USER,"GRAND  ","TOTALS  ","***",
  TIMER(PST:=PST+M[1]),TIMER(IOT:=IOT+M[2]),
  TIMER(ET:=ET+M[3]),TIMER(CET:=CET+M[4]),
  TIMER(TT:=TT+M[5]),DECV(8,I:=I+J), " ");%
  M[0]:=0;%
END;
WRITE(UPRNT[DBL]);%
IF MERGETOG THEN IF COMPARE(1,M[0],USER) AND M[0] NEQ 0
THEN M[0]:=0;%
IF SUMTOG THEN WRITE(SUMFILE,*,USER,PST,IOT,ET,
CET,TT,I);%
GO TO XIT;
EOF:MERGEND:=TRUE;%
XIT:
END USERPT;%

```

PRINTLINE(ULINE,USER)

```

00062900 T 0003
00063000 T 0006
00063100 T 0011
00063200 T 0013
00063300 T 0014
00063400 T 0017
00063500 T 0027
00063600 T 0030
00063700 T 0030
00063800 T 0033
00063900 T 0033
00064000 T 0035
00064100 T 0039
00064200 T 0045
00064300 T 0046
00064400 T 0050
00064500 T 0054
00064600 T 0059
00064700 T 0061
00064800 T 0061
00064900 T 0065
00065000 T 0068
00065100 T 0071
00065200 T 0088
00065300 T 0095
00065400 T 0095
00065500 T 0096
00065600 T 0097
29 IS 102 LONG, NEXT SEG 2

```

```

*****
BOOLEAN PROCEDURE OSIN(SORTREC);%
  ARRAY SORTREC[0];%
*****
BEGIN
%.....
PROCEDURE JOBON(U,P1,P2,LL,M,T);%
  VALUE U,P1,P2,LL,M,T;%
  REAL U,P1,P2,LL,M,T;%
  COMMENT USES PASSED PARAMETERS TO BUILD MIX TABLE ENTRIES;
%.....
BEGIN DEFINE MT=MIXTRBL#;%
  MT[M,0]:=U; IF LL=0 THEN MT[M,1]:=0 ELSE
  MT[M,1]:=DECV(2,LL) & DECV(5,LLTBL[LL,2]) [6:18:30];%
  MT[M,2]:=P1; MT[M,3]:=P2;%
  MT[M,4]:=T;%
  IF LL NEQ 0 THEN
  LLTBL[LL,3]:=U;% PUT IN USER CODE
  MIXARRY[MN]:=MIXARRY[MN]+T;
  MIXARRY[MN:=MN+1]:=MIXARRY[MN]-T;
END JOBON;%

```

```

00065700 T 0154
00065800 T 0154
00065900 T 0154
00066000 T 0154
00066100 T 0154
00066200 T 0154
00066300 T 0154
START OF SEGMENT ***** 30
00066400 T 0000
00066500 T 0000
00066600 T 0000
00066700 T 0000
00066800 T 0000
START OF SEGMENT ***** 31
00066900 T 0000
00067000 T 0006
00067100 T 0014
00067200 T 0019
00067300 T 0021
00067400 T 0022
00067410 C 0025
00067420 C 0027
00067500 T 0030
31 IS 31 LONG, NEXT SEG 30

```


Moore Business Forms, Inc. sv 12121

```

%.....
PROCEDURE JOBOFF(S,LL,MIX,TIM,HEND);%
  VALUE LL,MIX,TIM,HEND;%
  REAL LL,MIX,TIM,HEND;%
  ARRAY S[0];%
  COMMENT CREATES SORT RECORD FROM PASSED INFO AND
  INFO IN MIXTABLE;%
%.....
BEGIN
  DEFINE M=MIXTBL#;%

  MOVF(9,M[MIX,*],S[*]);%
  S[5]:=TIM;% EOJ TIME
  S[9]:=HEND;% HOW JOB ENDED ,MAY BE PSEUDO
  CLEAR(9,M[MIX,*]);% CLEAR MIX TABLE ENTRY
  MIXARRY[MN]:=MIXARRY[MN]+TIM;
  MIXARRY[MN:=MN-1]:=MIXARRY[MN]-TIM;
END JOBOFF;%

```

```

00067600 T 0000
00067700 T 0000
00067800 T 0000
00067900 T 0000
00068000 T 0000
00068100 T 0000
00068200 T 0000
00068300 T 0000
00068400 T 0000
00068500 T 0000
START OF SEGMENT ***** 32
00068600 T 0000
00068700 T 0002
00068800 T 0003
00068900 T 0005
00068910 C 0007
00068920 C 0009
00069000 T 0012
32 IS 13 LONG, NEXT SEG 30

```

```

%.....
PROCEDURE LOGOFF(S,LL,OFFT,CET,BOO,HEND);%
  VALUE LL,OFFT,CET,BOO,HEND;%
  REAL LL,OFFT,CET,HEND;%
  BOOLEAN BOO;%
  ARRAY S[0];%
%.....
BEGIN
  REAL I;%

  LABEL XIT;%
  S[1]:=DECV(2,LL) & DECV(5,LLTBL[LL,2]) [6:18:30];%
  S[0]:=LLTBL[LL,0]; S[2]:=" LINE &";%
  S[3]:="C&E USE"; S[4]:=LLTBL[LL,1]; S[5]:=OFFT; S[6]:=CET;
  S[7]:=0; S[8]:=" "; S[9]:=HEND;%
  FOR I:=0,1,3 DO LLTBL[LL,I]:=0;
  LLTBL[LL,2]:=LLTBL[LL,2]+1;% BUMP SESSION NO.
  IF BOO THEN GO XIT;%
  LINEARRY[NL]:=LINEARRY[NL]+OFFT;%
  LINEARRY[NL:=NL-1]:=LINEARRY[NL]-OFFT;%
XIT:
END LOGOFF;

```

```

00069100 T 0000
00069200 T 0000
00069300 T 0000
00069400 T 0000
00069500 T 0000
00069600 T 0000
00069700 T 0000
00069800 T 0000
00069900 T 0000
START OF SEGMENT ***** 33
00070000 T 0000
00070100 T 0000
00070200 T 0006
00070300 T 0010
00070400 T 0016
00070500 P 0020
00070600 T 0033
00070700 T 0037
00070800 T 0038
00070900 T 0040
00071000 T 0043
00071100 T 0044
33 IS 47 LONG, NEXT SEG 30

```

```

%.....
PROCEDURE LOGON(LL,ONT,WHO);%
  VALUE LL,ONT,WHO;

```

```

00071200 T 0000
00071300 T 0000
00071400 T 0000

```

Macroe Business Forms, Inc. sv

```

REAL LL,ONT,WHO;%
%.....
BEGIN LLTBL[LL,0]:=WHO; LLTBL[LL,1]:=ONT;%
  LLTBL[LL,3]:=0; % SO WE KNOW HE LOGGED IN
  LINEARRY[NL]:=LINEARRY[NL]+ONT;%
  LINEARRY[NL:=NL+1]:=LINEARRY[NL]-ONT;%
END LOGON;%

```

```

00071500 T 0000
00071600 T 0000
00071700 T 0000
00071800 T 0005
00071900 T 0007
00072000 T 0009
00072100 T 0012

```

```

DEFINE K=GT1#,S=SORTREC#;%
DEFINE W=LOGREC#;%
DEFINE I=GT2#,M=MIXTBL#;%
DEFINE HLTIME = (IF HLP=0 THEN BEGINTIME ELSE (HLQ[HLP]
+5184000*(HLDATE[HLP]-FIRSTDATE)))#;
LABEL RD,ONN,OFN,HL,DONE;%
LABEL XXT,XIT,BOJR,EOJR,FILER,EOJS;
LABEL MRCLEAN,MRAGN,MRSCLEAN,MRSAGN;%
SWITCH SWL:=RD,BOJR,EOJR,
RD,RD,RD,HL,EOJS,FILER,ONN,OFN,RD,RD,RD,RD;%
SWITCH CLEANSW:=MRCLEAN,MRAGN,MRSCLEAN,MRSAGN;%
REAL L,LL,PN1,PN2,USERID;%
INTEGER TIM,MIX;
BOOLEAN B;%

```

```

00072200 T 0012
00072300 T 0012
00072400 T 0012
00072410 C 0012
00072410 C 0012
00072500 T 0012
00072600 T 0012
00072700 T 0012
00072800 T 0012
00072900 T 0015
00073000 T 0023
00073100 T 0029
00073200 P 0029
00073300 T 0029
00073400 T 0029
00073500 P 0029
00073600 T 0033
00073700 T 0034
00073800 T 0036
00073900 T 0037
00074000 T 0038
00074100 T 0038
00074105 C 0040
00074110 C 0041
00074200 T 0042
00074300 T 0047
00074310 C 0049
00074400 T 0051
00074500 T 0051
00074600 T 0052
00074610 C 0054
00074611 C 0057
00074612 C 0060
00074700 T 0065
00074800 T 0069
00074900 T 0072
00075000 T 0074
00075100 T 0078
00075200 P 0084
00075300 T 0087
00075400 T 0088
00075500 T 0090
00075600 T 0090
00075700 T 0091
00075800 T 0093
00075900 T 0096

```

```

COMMENT * * * * * OSIN STARTS HERE * * * * * ;
IF CLEANER OR HLING THEN GO CLEANSW[K];
RD: IF DISKREAD THEN
  BEGIN CLOSE(DISK);
    CLEANER:=TRUE;%
    GO TO MRCLEAN;%
  END;
TIM:=(L:=LOGREC[0]),TIMEF;%
TIM:=TIM MOD 5184000;
LASTIME := TIM;
MIX:=L.MIXF; LL:=IF L.RLF=0 THEN 0 ELSE L.LINEF;%
GO TO SWL[LL,TYPEF];%
GO TO RD; % TYPE NOT HANDLED HERE BY SWITCH SWL
BOJR:COMMENT BOJ RECORD;
IF LL=0 THEN IF NOT ONSITETOG THEN GO RD ELSE
ELSE IF NOT TERMINALTOG THEN GO RD;%
IF MIMIX,0] NEQ 0 THEN BEGIN
  MIXARRY[MN]:=MIXARRY[MN]+TIM+LAPSE;
  MIXARRY[MN:=MN-1]:=MIXARRY[MN]-(TIM+LAPSE) END;
DO UNTIL SCAN="/"; PN1:=ACCUM;% PROG NAME 1
SCAN; PN2:=ACCUM;% PROG NAME 2
IF SCAN NEQ "/" THEN
  USERID:=IF LL=0 THEN "NO-USER" ELSE ACCUM ELSE
  BEGIN SCAN; USERID:=ACCUM END;% USER ID
JOBON(USERID,PN1,PN2,LL,MIX,TIM+LAPSE);%
LASTIME:=TIM;%
GO TO RD;%
EOJR:COMMENT EOJ RECORD;
IF LL=0 THEN IF NOT ONSITETOG THEN GO RD ELSE
ELSE IF NOT TERMINALTOG THEN GO RD;%
IF EQUIV(MIMIX,0),0) THEN
  BEGIN COMMENT NO BOJ,TRY TO SALVAGE DATA ELSEWHERE;%

```

```

        IF LL NEQ 0 THEN
        USERID:=IF EQUIV(LLTBL[LL,0],0) THEN "UNKNOWN"
        FLSE LLTBL[LL,0] ELSE USERID:="UNKNOWN";%
        DO UNTIL SCAN="/";PN1:=ACCUM; SCAN; PN2:=ACCUM;%
        JOBON(USERID,PN1,PN2,LL,MIX,TIM+LAPSE);
    END SALVEAGE;%
    DO UNTIL SCAN=",";%
    DO SCAN UNTIL L:=ACCUM,[6:12]="EO" OR
    L="DS" OR L="SY";%
    JOBOFF(S,LL,MIX,TIM+LAPSE,IF L="DS" THEN "DS=ED " ELSE ACCUM);;%
    GO TO XIT;%
ONN: COMMENT LOG ON RECORD;
    IF NOT(LINETOG OR TERMINALTOG) THEN GO RD;
    DO UNTIL SCAN=NUMBERV;%
    IF LL=0 THEN GO TO RD;% THRW AWAY SCHED STUFF
    IF LLTBL[LL,0] NEQ 0 THEN
    BEGIN COMMENT NOT LOGEED OFF;
        LOGOFF(S,LL,TIM+LAPSE,0,FALSE,"ES=ON ");
        LOGON(LL,TIM+LAPSE,ACCUM);
        GO TO IF TERMINALTOG THEN XIT ELSE RD;%
    END PSEUDO LOGOFF;
    LOGON(LL,TIM+LAPSE,ACCUM);
    LASTIME:=TIM;%
    GO TO RD;%
OFN: COMMENT LOG OFF RECORD;
    IF NOT(LINETOG OR TERMINALTOG) THEN GO RD;%
    DO UNTIL SCAN=NUMBERV;%
    IF LLTBL[LL,0]=0 THEN LOGON(LL,TIM+LAPSE,ACCUM);
    DO UNTIL SCAN = "(";
    LOGOFF(S,LL,TIM+LAPSE,6xNUMBERSCAN,FALSE," ");
    GO TO IF TERMINALTOG THEN XIT ELSE RD;%
HL: HLQ[HLP:=HLP+1]:=TIM;%
    HLDATE[HLP]:= LASTDATE;
    IF (MN=0) AND (NL=0) THEN GO TO RD; % DOESN'T NEED CLEANING
    HLING:=TRUE; GO TO MRSCLEAN;
    GO TO RD;
EOJS: COMMENT EOJ STAT RECORD;
    IF LL=0 THEN IF NOT ONSITETOG THEN GO RD ELSE
    ELSE IF NOT TERMINALTOG THEN GO RD;%
    M[MIX,6]:=W[1];% CPU TIME
    M[MIX,7]:=W[2];% I O TIME
    M[MIX,8]:=W[3];% CORE USED
    GO TO RD;%
FILER: COMMENT FILE STAT RECORD;%
    IF LL=0 THEN IF NOT ONSITFTOG THEN GO RD ELSE
    ELSE IF NOT TERMINALTOG THEN GO RD;%
    IF EQUIV(M[MIX,0],0) THEN GO TO RD;% CANT RECOVER,NO INFO
    MOVF(4,M[MIX,0],S[0]);%
    S[4]:=M[MIX,4]+1;%
    MOVE(5,W[1],S[5]);%
    S[9]:=-ABS(S[9]);%
    GO TO XIT;%
MRSCLEAN:
    COMMENT GENERATE PSEUDO LOG-OFFS FOR THOSE USERS STILL ON;
    IF NOT(TERMINALTOG OR LINETOG) THEN GO MRSCLEAN;%
    K:=2; I:=0;
MRAGN:

```

```

00076000 T 0096
00076100 T 0097
00076200 T 0101
00076300 T 0106
00076400 P 0114
00076500 T 0117
00076600 T 0117
00076700 T 0121
00076800 T 0126
00076900 P 0128
00077000 T 0134
00077100 T 0141
00077200 T 0141
00077300 T 0142
00077310 T 0146
00077400 T 0147
00077500 T 0149
00077600 P 0150
00077700 P 0154
00077800 T 0157
00077900 T 0161
00078000 P 0161
00078100 T 0164
00078200 T 0165
00078300 T 0168
00078400 T 0168
00078500 T 0169
00078600 P 0173
00078610 C 0178
00078700 P 0182
00078800 T 0188
00078900 T 0192
00078910 C 0195
00078920 C 0197
00078930 C 0199
00079000 T 0203
00079100 T 0203
00079200 T 0204
00079300 T 0205
00079400 T 0207
00079500 T 0210
00079600 T 0212
00079700 T 0215
00079800 T 0216
00079900 T 0216
00080000 T 0217
00080100 T 0219
00080200 T 0222
00080300 T 0225
00080400 T 0229
00080500 T 0230
00080600 T 0232
00080700 T 0233
00080800 T 0234
00080900 T 0234
00081000 P 0235
00081100 T 0236

```

```

IF I:=I+1 GTR NUMLINES THEN GO MRSCLEAN;%
IF LLTBL[I,0] NEQ 0 THEN
BEGIN LOGOFF(S,I,LASTIME+LAPSE,0,FALSE,"ES=OF "); GO XXT END ELSE
BEGIN COMMENT HE MAY NOT BE LOGGED IN;%
LLTBL[I,1]:=BEGINTIME;% ASSUME ON
IF LLTBL[I,0]:=LLTBL[I,3] NEQ 0 THEN
BEGIN LOGOFF(S,I,LASTIME+LAPSE,0,FALSE,"ESTIME ");
COMLINES:=COMLINES+1;%
GO TO XXT;%
END ELSE GO MRAGN;%
END LOGON CHECK;%
MRSCLEAN: IF NOT HLING THEN BEGIN
LINEARRY[0]:=LINEARRY[0]+LASTIME+LAPSE;
WHILE LINEARRY[0]<0 DO LINEARRY[0]:=LINEARRY[0]+5184000; END;
COMMENT GENERATE PSEUDO EOJS FOR JOBS STILL RUNNING;
IF NOT(ONSITETOG OR TERMINALTOG) THEN GO DONE;%
K:=4; I:=0;%
MRSAGN:
IF (I:=I+1) GTR MIXMAX THEN GO DONE;%
IF MCI,0] NEQ 0 THEN
BEGIN B:=(MCI,4] LSS HLTIME); JOBOFF(S,LL,I,LASTIME+LAPSE,
IF B THEN "HL=ED " ELSE "INMIX ");
GO TO XXT;%
END PSEUDO EOJ ELSE GO MRSAGN;%
DONE: IF NOT HLING THEN BEGIN
MIXARRY[0]:=MIXARRY[0]+LASTIME+LAPSE;
WHILE MIXARRY[0]<0 DO MIXARRY[0]:=MIXARRY[0]+5184000; END;
K:=I:=0;%
IF NOT HLING THEN OSIN:=TRUE; HLING:=FALSE;
GO TO XXT;%
XIT; LASTIME:=TIM;%
XXT:
END OSIN;%

```

```

00081200 T 0237
00081300 T 0239
00081400 P 0241
00081500 T 0249
00081600 T 0249
00081700 T 0252
00081800 P 0256
00081900 T 0261
00081910 T 0262
00082000 T 0265
00082100 T 0265
00082200 P 0265
00082205 C 0266
00082210 C 0269
00082300 T 0275
00082400 T 0275
00082500 T 0276
00082600 T 0277
00082700 T 0278
00082800 T 0280
00082900 P 0282
00083000 P 0293
00083100 T 0295
00083200 T 0299
00083300 P 0299
C 0300
00083310 C 0303
00083400 T 0309
00083500 P 0310
00083600 T 0312
00083700 T 0313
00083800 T 0314
00083900 T 0315

```

30 IS 322 LONG, NEXT SEG 2

```

%*****
PROCEDURE OSOUT(NODATA, SORTREC);
VALUE NODATA;%
BOOLEAN NODATA; ARRAY SORTREC[0];%
%*****
BEGIN
DEFINE TPST=GT2#, % ACCUM REPORT CPU TIME

TIOT=GT3#, % ACCUM REPORT I=0 TIME
TET =GT4#, % ACCUM REPORT ELAPSED TIME
TCOR=GT5#, % ACCUM REPORT CORE(WDS)
UPST=GT6#, % ACCUM USER CPU TIME
UIOT=GT7#, % ACCUM USER I=0 TIME
UET =GT8#, % ACCUM USER ELAPSED TIME
UCOR=GT9#, % ACCUM USER CORE(WDS)
SPST=GT10#, % ACCUM SESSION CPU TIME
SIOT=GT11#, % ACCUM SESSION I=0 TIME
SET =GT12#, % ACCUM SESSION ELAPSED TIME
SCOR=GT13#, % ACCUM SESSION CORE(WDS)

```

```

00084000 T 0154
00084100 T 0154
00084200 T 0154
00084300 T 0154
00084400 T 0154
00084500 T 0154
00084600 T 0154
START OF SEGMENT *****
00084700 T 0000
00084800 T 0000
00084900 T 0000
00085000 T 0000
00085100 T 0000
00085200 T 0000
00085300 T 0000
00085400 T 0000
00085500 T 0000
00085600 T 0000
00085700 T 0000

```

START OF SEGMENT ***** 34

```

PPST=GT14#,% ACCUM PROGRAM CPU TIME
PIOT=GT15#,% ACCUM PROGRAM I=O TIME
PET =GT16#,% ACCUM PROGRAM ELAPSED TIME
PCOR=GT17#,% ACCUM PROGRAM CORE(WDS)
NU  =GT18#,% NO. OF USERS
NR  =GT1#,% REC NO.
NJ  =GT19#,% NO. OF JOBS PROCESSED

;%
DEFINE S=SORTREC#;%
DEFINE H=HOLDKEY#;%
INTEGER PST,IOT,ET;%
INTEGER COR;%
LABEL DET,FILEDET,THRU;%
IF NOT NODATA THEN%
BEGIN COMMENT DATA CASE;%
  IF (NR:=NR+1)=1 THEN GO TO DET;%
  IF NOT EQUAL(4,S[0],HOLDKEY) THEN
  BEGIN COMMENT SUMMARIZE LAST PROG DATA;
    LEVEL(PROGRAM,PPST,PIOT,PET,PCOR);%
    SPST:=SPST+PPST; SIOT:=SIOT+PIOT;
    SET:=SET+PET; SCOR:=SCOR+PCOR;
    CLEAR(3,PPST);%
  IF DETOG THEN BEGIN
    WRITE(PRNT[DBL]); LINECTR:=LINECTR+2;%
  END
  END LAST PROG ELSE GO DET;%
  IF NOT EQUAL(2,S[0],HOLDKEY) THEN
  BEGIN COMMENT SUMMARIZE LINE DATA;%
    IF S[1],[36:12] NEQ 0 THEN TSESSIONS:=TSESSIONS+1;
    LEVEL(SESSION,SPST,SIOT,SET,SCOR);%
    UPST:=UPST+SPST; UIOT:=UIOT+SIOT;%
    UCOR:=UCOR+SCOR; UET:=UET+SET;%
    SPLIT(IF H[1],[36:12]=0 THEN 0 ELSE 1);%
    CLEAR(3,SPST);%
  IF DETOG THEN BEGIN
    WRITE(PRNT[DBL]); LINECTR:=LINECTR+2;%
  END
  END LINE SUMMARY ELSE GO DET;%
  IF NOT EQUIV(H[0],S[0]) THEN
  BEGIN COMMENT SUMMARIZE USER DATA;
    LEVEL(USER,UPST,UIOT,UET,UCOR);%
    REPORTU(HOLDKEY[0],UPST,UIOT,
    UET,UET,UTT,UCOR); %
    TPST:=TPST+UPST; TIOT:=TIOT+UIOT;
    TET:=TET+UET; TCOR:=TCOR+UCOR;
    RTT := RTT+UTT; RCT := RCT+UCT;%
    UTT := UCT := 0;%
    CLEAR(3,UPST);
    OSUSERS:=OSUSERS+OSUSE; OSUSE:=0;%
    REMUSERS:=REMUSERS+REMUSE; REMUSE:=0;%
    NU:=NU+1;%
  IF DETOG THEN
    PUTHEADER(ONSITE);%
  END USER SUMMARY;%
DET: COMMENT PRECESS DETAIL RECORD;%
  IF LINECTR GTR PAGESIZE THEN PUTHEADER(ONSITE);
  IF SIGN(S[9])=-1 THEN GO FILEDET;%

```

```

00085800 T 0000
00085900 T 0000
00086000 T 0000
00086100 T 0000
00086200 T 0000
00086300 T 0000
00086400 T 0000
00086500 T 0000
00086600 T 0000
00086700 T 0000
00086800 T 0000
00086900 T 0000
00087000 T 0000
00087100 T 0000
00087200 T 0000
00087300 T 0001
00087400 T 0003
00087500 T 0005
00087600 T 0006
00087700 T 0008
00087800 T 0010
00087900 T 0013
00087990 C 0014
00088000 T 0014
00088010 C 0020
00088100 T 0020
00088200 T 0020
00088300 T 0022
00088340 C 0023
00088400 T 0026
00088500 T 0028
00088600 T 0030
00088700 T 0033
00088800 T 0036
00088890 C 0037
00088900 T 0038
00088910 C 0043
00089000 T 0043
00089100 T 0043
00089200 T 0045
00089300 T 0046
00089400 T 0047
00089500 T 0049
00089600 T 0050
00089700 T 0052
00089710 T 0055
00089720 T 0057
00089800 T 0059
00089810 T 0060
00089820 T 0062
00089900 T 0064
00089990 C 0065
00090000 T 0065
00090100 T 0066
00090200 T 0066
00090300 P 0067
00090400 T 0069

```

1412T
McGraw-Hill Business Forms, Inc. sv

```

IF S[1],[36:12]=0 THEN BEGIN
  OSJOBS:=OSJOBS+1;
  IF (S[2]="LDCNTRL") AND (S[3]="DISK ") THEN
    PACKETS:=PACKETS+1
  END
  ELSE TUJOBS:=TUJOBS+1;%
  NJ:=NJ+1;%
  DETAIL(S[0],S[2],S[3],S[1],S[4],S[5],PST:=S[6],
  IOT:=S[7],ET:=S[5]-S[4],COR:=S[8],S[9]);
  COMMENT ACCUM PROG TOTALS;%
  IF S[3] = "C&E USE" THEN BEGIN
    UTT:=UTT+ET; UCT:=UCT+PST END
  ELSE BEGIN
    PET:=PET+ET; PPST:=PPST+PST; PIOT:=PIOT+IOT;
    PCOR:=PCOR+COR END;
    MOVE(5,S[0],HOLDKEY);%
    GO THRU;%
  FILEDET;
  FILEDETAIL(S);%
  END DATA CASE ELSE
  BEGIN COMMENT NO DATA CASE;%
    IF SUMTOG THEN LOCK(SUMFILE);%
    LEVEL(PROGRAM,PPST,PIOT,PET,PCOR);%
  IF DETOG THEN
    WRITE(PRNT[DBL]);%
    SPST:=SPST+PPST; SIOT:=SIOT+PIOT;
    SET:=SET+PET; SCOR:=SCOR+PCOR;%
    IF H[1],[36:12] NEQ 0 THEN TSESSIONS:=TSESSIONS+1;
    LEVEL(SESSION,SPST,SIOT,SET,SCOR);%
  IF DETOG THEN
    WRITE(PRNT[DBL]);%
    % ELIMINATE A STUPID LINE OF CODE
    SPLIT(IF H[1],[36:12]=0 THEN 0 ELSE 1);%
    UPST:=UPST+SPST; UIOT:=UIOT+SIOT;
    UET:=UET+SET; UCOR:=UCOR+SCOR;%
    LEVEL(USER,UPST,UIOT,UET,UCOR);%
    REPORTU(HOLDKEY[0],UPST,UIOT,UET,
    UCT,UTT,UCOR);%
  IF DETOG THEN
    WRITE(PRNT[DBL]);%
    TPST:=TPST+UPST; TIOT:=TIOT+UIOT;
    TET:=TET+UET; TCOR:=TCOR+UCOR;
    RTT:=RTT+UTT; RCT:=RCT+UCT;
    OSUSERS:=OSUSERS+OSUSE; REMUSERS:=REMUSERS+REMUSE;
    NU:=NU+1;%
    LEVEL(REPORT,TPST,TIOT,TET,TCOR);
    CLEAR(7,UPST);%
  END NODATA CASE;
  THRU:
  END OSOUT;%

```

```

00090500 P 0072
00090510 C 0074
00090520 C 0075
00090540 C 0078
00090550 C 0078
00090600 T 0079
00090700 T 0084
00090800 T 0085
00090900 T 0089
00091000 T 0094
00091100 P 0094
00091110 P 0095
00091120 C 0098
00091130 C 0100
00091200 P 0103
00091300 T 0105
00091400 T 0106
00091500 T 0107
00091600 T 0108
00091700 T 0109
00091800 T 0109
00091900 T 0109
00092000 T 0112
00092090 C 0114
00092100 T 0114
00092200 T 0118
00092300 T 0121
00092310 C 0123
00092400 T 0127
00092490 C 0128
00092500 T 0129
00092510 P 0133
00092600 T 0133
00092700 T 0137
00092800 T 0139
00092900 T 0142
00093000 T 0143
00093100 T 0145
00093190 C 0146
00093200 T 0146
00093300 T 0151
00093400 T 0153
00093410 C 0156
00093420 C 0158
00093500 T 0161
00093600 T 0162
00093700 T 0164
00093800 T 0165
00093900 T 0165
00094000 T 0165

```

34 IS 168 LONG, NEXT SEG 2

```

%*****
BOOLEAN PROCEDURE OSCOMP(A,B);%

```

```

00094100 T 0154
00094200 T 0154

```

Moore Business Forms, Inc. sv 1-127

```

      ARRAY A,B[0];
%*****
OSCOMP:=IF EQUAL(4,A[0],B[0]) THEN A[4] LEQ B[4]
ELSE COMPARE(4,A[0],B[0]);%

```

```

00094300 T 0154
00094400 T 0154
00094500 T 0154
00094600 T 0158

```

```

%*****
PROCEDURE OSVAL(A);
  ARRAY A[0];%
%*****
FILL A[*] WITH OCT1414141414141414,OCT1414141414141414,
      OCT1414141414141414,OCT1414141414141414,
      OCT1414141414141414;%

```

```

00094700 T 0165
00094800 T 0165
00094900 T 0165
00095000 T 0165
00095100 T 0165
START OF SEGMENT ***** 35
00095200 T 0167
00095300 T 0167
35 IS 5 LONG, NEXT SEG 2

```

```

%*****
PROCEDURE OSFINAL;%
%*****
BEGIN
  DEFINE PST = GT2#, % CPU TIME

      IOT = GT3#, % I=O TIME
      ET = GT4#, % ELAPSED TIME
      NJ = GT19#, % NO. OF JOBS PROCESSED
      NU = GT18#, % NO. OF USERS
  LABEL AGAIN;%
  REAL WALLT; % WALL CLOCK TIME
  INTEGER P1,P2,I1,I2,E1,E2;%
  BOOLEAN TUTOG;%
  LIST TIMELIST(P1,P2,I1,I2,E1,E2);%
  FORMAT F1("NO. OF JOBS PROCESSED WAS : ",I6,X2,
      "NO. OF USERS WAS : ",I6);%
  FORMAT F2("TOTAL TIMES",10(","),3(A3,A5,X3)),
      F3("PERCENTAGES(TIME/ET)",",",3(F8,2,X3)),
      F4("AVERAGES PER JOB.....",3(A3,A5,X3)),
      F5("AVERAGES PER USER.....",3(A3,A5,X3));%
  FORMAT F6(X22,"CPU",X8,"I=O",X9,"ET");%

  FORMAT F7(" * * * * * ALL JOBS * * * * *");%

  FORMAT F8(" * * * * * REMOTE JOBS * * * * *");%

  FORMAT F9(" * * * * * ONSITE JOBS * * * * *");%

```

```

00095400 T 0168
00095500 T 0168
00095600 T 0168
00095700 T 0168
00095800 T 0168
START OF SEGMENT ***** 36
00095900 T 0000
00096000 T 0000
00096100 T 0000
00096200 T 0000
00096300 T 0000
00096310 C 0000
00096400 T 0000
00096500 T 0000
00096600 T 0000
00096700 T 0012
START OF SEGMENT ***** 37
00096800 T 0012
37 IS 15 LONG, NEXT SEG 36
00096900 T 0012
START OF SEGMENT ***** 38
00097000 T 0012
00097100 T 0012
00097200 T 0012
38 IS 49 LONG, NEXT SEG 36
00097300 T 0012
START OF SEGMENT ***** 39
39 IS 9 LONG, NEXT SEG 36
00097400 T 0012
START OF SEGMENT ***** 40
40 IS 8 LONG, NEXT SEG 36
00097500 T 0012
START OF SEGMENT ***** 41
41 IS 9 LONG, NEXT SEG 36
00097600 T 0012
START OF SEGMENT ***** 42

```

```

FORMAT F10("TOTAL CANDE TIME ",A3,A5," TOTAL TERMINAL TIME ",
          A3,A5," T-S SESSIONS",I5," CONNECT/SESSION",A3,A5);
FORMAT F11("COMPILATIONS",I5); FORMAT F12(A2,A5,":",I5,F8,2,"%");

FORMAT F13 ("WALI. CLOCK ELAPSED TIME ",A3,A5);

FORMAT F14 ("PERCENT OF WALLTIME,.",3(F8,2,X3));

```

```

WALLT:=LASTIME-BEGINTIME+LAPSE; STIMER(WALLT,E1,E2);
WRITE(PRNT[DBL],F13,E1,E2);
WRITE(PRNT[DBL],F7);%
AGAIN:
WRITE(PRNT[DBL],F1,NJ,NU);%
WRITE(PRNT[DBL],F6);%
STIMER(PST,P1,P2); STIMER(IOT,I1,I2);
STIMER(ET,E1,E2);
WRITE(PRNT[DBL],F2,TIMELIST);%
WRITE(PRNT[DBL],F3,100*(PST/ET),100*(IOT/ET),100);%
WRITE(PRNT[DBL],F14,100*PST/WALLT,100*IOT/WALLT,100*ET/WALLT);
STIMER(PST/NJ,P1,P2); STIMER(IOT/NJ,I1,I2);%
STIMER(ET/NJ,E1,E2);%
WRITE(PRNT[DBL],F4,TIMELIST);%
STIMER(PST/NU,P1,P2); STIMER(IOT/NU,I1,I2);%
STIMER(ET/NU,E1,E2);%
WRITE(PRNT[DBL],F5,TIMELIST);%
WRITE(PRNT[DBL]);%
IF ONSITETOG THEN
BEGIN COMMENT ONSITE TOTALS;
  WRITE(PRNT[DBL],F9);%
  ONSITETOG:=FALSE;%
  PST:=PSTOTALS[0];%
  IOT:=IOTOTALS[0];%
  ET := ETOTALS[0];%
  NJ:=OSJOBS;%
  NU:=OSUSERS;%
  GO AGAIN;%
END ONSITE TOTALS;%
IF TERMINALTOG THEN
IF NOT TUTOG THEN
BEGIN COMMENT REMOTE TOTALS;%
  TUTOG:=TRUE;%
  WRITE(PRNT[DBL],F8);%
  PST:=PSTOTALS[1];%
  IOT:=IOTOTALS[1];%
  ET := ETOTALS[1];%
  NJ:=TUJOBS;%
  NU:=REMUSERS;%
  GO AGAIN;%
END

```

END

```

42 IS 9 LONG, NEXT SEG 36
00097610 C 0012
START OF SEGMENT ***** 43
00097620 C 0012
43 IS 23 LONG, NEXT SEG 36
00097630 C 0012
START OF SEGMENT ***** 44
44 IS 6 LONG, NEXT SEG 36
START OF SEGMENT ***** 45
45 IS 9 LONG, NEXT SEG 36
00097640 C 0012
START OF SEGMENT ***** 46
46 IS 9 LONG, NEXT SEG 36
00097645 C 0012
START OF SEGMENT ***** 47
47 IS 11 LONG, NEXT SEG 36
00097690 C 0012
00097695 C 0016
00097700 T 0027
00097800 T 0030
00097900 T 0031
00098000 T 0040
00098100 T 0043
00098200 T 0045
00098300 T 0047
00098400 T 0050
00098410 C 0065
00098500 T 0081
00098600 T 0084
00098700 T 0086
00098800 T 0089
00098900 T 0093
00099000 T 0095
00099100 T 0098
00099200 T 0102
00099300 T 0102
00099400 T 0103
00099500 T 0106
00099600 T 0106
00099700 T 0107
00099800 T 0108
00099900 T 0109
00099910 T 0110
00100000 T 0111
00100100 T 0111
00100200 T 0111
00100300 T 0112
00100400 T 0113
00100500 T 0113
00100600 T 0114
00100700 T 0117
00100800 T 0118
00100900 T 0119
00101000 T 0120
00101010 T 0121
00101100 T 0121
00101200 P 0122

```



```

ELSE BEGIN STIMER(RCT,P1,P2); STIMER(RTT,E1,E2);
      STIMER(RTT/TSESSIONS,I1,I2);
      WRITE(PRNT[DBL],F10,P1,P2,E1,E2,TSESSIONS,I1,I2) END;
COMPILS[6]:=COMPILS[6]-COMPILS[8]; % DON-T DOUBLE COUNT PASCAL
MIXANA; I1:=0; FOR I2:=1 STEP 1 UNTIL CPLRNM[0] DO
      I1:=I1+COMPILS[I2];
WRITE(PRNT[DBL],F11,I1); FOR P1:=1 STEP 1 UNTIL CPLRNM[0] DO
      WRITE(PRNT,F12,CPLRNM[P1],[6:12],CPLRNM[P1],[18:30],
      I2:=COMPILS[P1],100*I2/I1);

```

END OSFINAL;%

WRITE(PNCH,<IS,SI8,4,5>)

FIRSTDATE,WALLT, PSTOTALS[0], ETOTALS[0], PSTOTALS[1], ETOTALS[1],
OSJOBS, OSUSERS, TSESSIONS, REMUSERS,

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
COMMENT ::::: SECTION = 06 = DISK LOGGING :::::;%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
COMMENT LAYOUT OF THE DISK LOGGING SORT RECORD;
%      S[0] ::= USER ID
%      S[1] ::= MFID
%      S[2] ::= FID
%      S[3] ::= CREATION DATE
%      S[4] ::= CREATION TIME
%      S[5] ::= LOG-OFF TIME
%      S[6] ::= NO OF DISK SEGS

```

```

*****
BOOLEAN PROCEDURE DLIN(S); ARRAY S[0];
COMMENT FORMATS THE DISK SORT RECORD;
*****
BEGIN
  DEFINE L=LOGREC#;%

```

```

  REAL T;%
  LABEL GET,XIT;%
  GET:
    IF DISKREAD THEN BEGIN DLIN:=TRUE; GO XIT END;
    IF LOGREC[0].TYPEF NEQ DLOG THEN GO GET;%
    DO UNTIL SCAN="/";%
    S[1]:=ACCUM; % MFID
    DO UNTIL SCAN="/";%
    S[2]:=ACCUM; % FID
    SCAN; S[0]:=ACCUM; % USER ID
    S[5]:=L[0].TIMEF; % LOG-OFF TIME
    SCAN; S[6]:=NUMBERSCAN;
    DO UNTIL SCAN="/";
    S[3]:=IF ACCUM.[12:6]=" " THEN (0&ACCUM[12:6:6])
    ELSE ACCUM;
    SCAN2;
    S[3]:=IF ACCUM.[12:6]=" " THEN(S[3]&ACCUM[24:6:6])
    ELSE (S[3]&ACCUM[18:6:12]);%
    SCAN; S[3]:=S[3] & ACCUM[30:6:12];% AND YEAR
    T:=216000*NUMBERSCAN;% HOURS
    T:=T+3600*NUMBERSCAN;% MINUTES
    T:=T+ 60*NUMBERSCAN;% SECONDS
    T:=T+ NUMBERSCAN;% 1/60 SECONDS

```

XIT:

00101210	C	0122	
00101220	C	0125	
00101221	C	0127	
00101222	C	0142	
00101224	C	0146	
00101226	C	0151	
00101230	C	0153	
00101240	C	0165	
00101241	C	0175	
00101300	T	0184	
36 IS	189 LONG,	NEXT SEG	2
00101400	T	0168	
00101500	T	0168	
00101600	T	0168	
00101700	T	0168	
00101800	T	0168	
00101900	T	0168	
00102000	T	0168	
00102100	T	0168	
00102200	T	0168	
00102300	T	0168	
00102400	T	0168	
00102500	T	0168	
00102600	T	0168	
00102700	T	0168	
00102800	T	0168	
00102900	T	0168	
00103000	T	0168	
START OF SEGMENT	*****		48
00103100	T	0000	
00103200	T	0000	
00103300	T	0000	
00103400	T	0000	
00103500	T	0002	
00103600	T	0004	
00103700	T	0008	
00103800	T	0009	
00103900	T	0013	
00104000	T	0014	
00104100	T	0018	
00104200	T	0020	
00104300	T	0025	
00104400	T	0028	
00104500	T	0031	
00104600	T	0033	
00104700	T	0037	
00104800	T	0041	
00104900	T	0043	
00105000	T	0048	
00105100	T	0051	
00105200	T	0054	
00105300	T	0057	
00105400	T	0060	

Moorc. Business Forms, Inc. sv 14121

END DLIN;

00105500 T 0061
48 IS 66 LONG, NEXT SEG 2

PROCEDURE DVAL(A); ARRAY A[0];

FILL A[*] WITH OCT14141414141414,OCT14141414141414,
OCT14141414141414;%

00105600 T 0168
00105700 T 0168
00105800 T 0168
00105900 T 0168
START OF SEGMENT ***** 49
00106000 T 0169
49 IS 3 LONG, NEXT SEG 2

BOOLEAN PROCEDURE DCOMP(A,B); ARRAY A,B[0];

DCOMP:=COMPARE(3,A[0],B[0]);%

00106100 T 0170
00106200 T 0170
00106300 T 0170
00106400 T 0170

STREAM PROCEDURE MAKEDISKLEVEL(LINE,ASTERISK,LEVEL,
SEGS,MTHS,DAYS,TIM);%
VALUE ASTERISK,LFVEL,SEGS,MTHS,DAYS,TIM;%

BEGIN
LOCAL DA;%
DI:=LINE; DI:=DI+33;%
SI:=LOC ASTERISK;%
2(UPSI;UPDI; DS:=7 CHR);%
DS:=12 LIT " TOTALS.....";%
UPDI; DA:=DI; DS:=8 CHR;% NO SEGS
DI:=DI+3;%
SI:=SI+6; DS:=2 CHR; DS:=LIT";";% MTHS
SI:=SI+6; DS:=2 CHR; DS:=LIT"-";% DAYS
SI:=SI+2; 2(DS:=2CHR; DS:=LIT";"); DS:=2 CHR;% TIME
DI:=DA; DS:=6 FILL;%
END MAKEDISKLEVEL;%

00106500 T 0175
00106600 T 0175
00106700 T 0175
00106800 T 0175
00106900 T 0175
00107000 T 0175
00107100 T 0176
00107200 T 0176
00107300 T 0176
00107400 T 0176
00107500 T 0178
00107600 T 0179
00107700 T 0180
00107800 T 0180
00107900 T 0181
00108000 T 0182
00108100 T 0184
00108200 T 0185

PROCEDURE DISKLEVEL(WHICH,SEGS,MTHS,DAYS,TIM);%
VALUE WHICH,SEGS,MTHS,DAYS,TIM;%
REAL WHICH,SEGS,MTHS,DAYS,TIM;%

BEGIN
INTEGER T;
FORMAT F1 (A2,A5);

00108300 T 0185
00108400 T 0185
00108500 T 0185
00108600 T 0185
00108700 T 0185
00108800 T 0185
00108900 T 0185
START OF SEGMENT ***** 50
00108910 C 0000
START OF SEGMENT ***** 51

1-12T
Moore Business Forms, Inc. 3v

```

BLANKOUT(LINE);%
MAKEDISKLEVEL(LINE,ASTERISK[WHICH=2],LEVELARRY[WHICH],
DECV(8,T:=SEGS),DECV(2,T:=MTHS),DECV(2,T:=DAYS),
TIMER(TIM));%
IF WHICH = USER THEN
WRITE(PRTINO),F1,HOLDKFY[0],[6;12],HOLDKEY[0],[18;30]);
PRINTLINE(LINE,DISKRPT);%
END DISKLEVEL;%

```

```

51 IS      5 LONG, NEXT SEG  50
00109000 T  0000
00109100 T  0001
00109200 T  0003
00109300 T  0009
00109310 C  0010
00109320 C  0011
00109400 T  0024
00109500 T  0025
50 IS     28 LONG, NEXT SEG  2

```

```

%*****
STREAM PROCEDURE MAKEDISKDETAIL(LINE,A,CTIM,LTIM,SEGS,
M,D,T);%
VALUE CTIM,LTIM,SEGS,M,D,T;%
%*****
BEGIN
LOCAL DA;%
DI:=LINE; SI:=A;%
3(UPSI; DS:=7 CHR; DI:=DI+2);%
UPSI; 2(DS:=2 CHR; DS:=LIT "/"); DS:=2 CHR;
SI:=LOC CTIM;
2(SI:=SI+2; DI:=DI+3; 2(DS:=2 CHR; DS:=LIT":"); DS:=2 CHR);%
DI:=DI+5; DA:=DI; DS:=8 CHR; DI:=DI+3;%
SI:=SI+6; DS:=2 CHR; DS:=LIT":";
SI:=SI+6; DS:=2CHR;
DS:=LIT"-";%
SI:=SI+2; 2(DS:=2 CHR;DS:=LIT":"); DS:=2 CHR;%
DI:=DA; DS:=6 FILL;%
END MAKE DISKDETAIL;%

```

```

00109600 T  0185
00109700 T  0185
00109800 T  0185
00109900 T  0185
00110000 T  0185
00110100 T  0185
00110200 T  0186
00110300 T  0186
00110400 T  0186
00110500 T  0187
00110600 T  0189
00110700 T  0189
00110800 T  0192
00110900 T  0193
00111000 T  0194
00111100 T  0194
00111200 T  0195
00111300 T  0197
00111400 T  0197

```

```

%*****
PROCEDURE DISKDETAIL(SORTREC,MTHS,DAYS,TIM);%
VALUE MTHS,DAYS,TIM;
REAL TIM; INTEGER MTHS,DAYS;%
ARRAY SORTREC[0];%
%*****
BEGIN DEFINE A=SORTREC#;

INTEGER I;%
BLANKOUT(LINE);%
MAKEDISKDETAIL(LINE,A,TIMER(A[4]),TIMER(A[5]),
DECV(8,I:=A[6]),DECV(2,MTHS),DECV(2,DAYS),
TIMER(TIM));%
IF DETOG THEN
PRINTLINE(LINE,DISKRPT);%
END DISKDETAIL;%

```

```

00111500 T  0197
00111600 T  0197
00111700 T  0197
00111800 T  0197
00111900 T  0197
00112000 T  0197
00112100 T  0197
START OF SEGMENT ***** 52
00112200 T  0000
00112300 T  0000
00112400 T  0001
00112500 T  0004
00112600 T  0009
00112690 C  0010
00112700 T  0011
00112800 T  0013
52 IS     16 LONG, NEXT SEG  2

```

```

%*****
PROCEDURE DLOUT(B,A);%
  VALUE B;%
  BOOLEAN B; % TRUE IF LAST REC PROCESSEE
  ARRAY A[0]; % SORTED DISK LOG RECORD
%*****
BEGIN LABEL DET,THRU;

  LABFL BRK;%
  DEFINE RPTMO = GT6#,% ACCUM REPORT MONTHS
         RPTDA = GT7#,% ACCUM REPORT DAYS
         RPTDT = GT8#,% ACCUM REPORT TIME
         RPTDS = GT9#,% ACCUM REPORT DISK SEGS
         USERMO =GT10#,% ACCUM USER MONTHS
         USERDA =GT11#,% ACCUM USER DAYS
         USERDT =GT12#,% ACCUM USER TIME
         USERDS =GT13#,% ACCUM USER DISK SEGS
         NF      =GT14#,% NO. OF FILES PROCESSED
         NU      =GT15#,% NO OF USERS

  REAL CDATE,CTIME,LDATE,LTIME;%
  INTEGER MTHS,DAYS;
  REAL Z,T;%
  IF(NF:=NF+1) = 1 THEN GO TO DET;%
  IF B THEN GO BRK;%
  IF NOT EQUAL(1,A[0],HOLDKEY) THEN
  BEGIN %NEW USER OR END OF REPORT
  BRK;
    USERDA←USERDA+USERDT DIV 5184000;
    USERDT←USERDT MOD 5184000;
    USERMO←USERMO+USERDA DIV 31;
    USERDA←USERDA MOD 31;
    IF LINECTR GTR PAGESIZE THEN PUTHEADER(DISKRPT);
    DISKLEVEL(USER,USERDS,USERMO,USERDA,USERDT);%
    RPTMO:=RPTMO+USERMO; RPTDA:=RPTDA+USERDA;%
    RPTDT:=RPTDT+USERDT; RPTDS:=RPTDS+USERDS;%
    CLEAR(3,USERMO);%
    IF NOT B THEN
      BEGIN NU:=NU+1; GO DET END;
    END USER TOTAL ELSE GO TO DET;%
    COMMENT END OF REPORT;
    RPTDA:=RPTDA+RPTDT DIV 5184000;%
    RPTDT:=RPTDT MOD 5184000;%
    RPTMO:=RPTMO+RPTDA DIV 31;%
    RPTDA:=RPTDA MOD 31;%
    DISKLEVEL(REPORT,RPTDS,RPTMO,RPTDA,RPTDT);%
    GO THRU;%
  DET:CDATE:=A[3];
    LDATE:=MFID & CDATE [30:30:12];%
    IF LTIME:=A[5] LSS CTIME:=A[4] THEN Z:=1;%
    IF LINECTR GTR PAGESIZE THEN PUTHEADER(DISKRPT);%
    DISKDETAIL(A,
    MTHS:=OCV(LDATE,2)-OCV(CDATE,2),
    DAYS:=OCV(LDATE,4)-OCV(CDATE,4),
    T:=LTIME+Z×5184000-CTIME);%
    USERMO:=USERMO+MTHS; USERDA:=USERDA+DAYS;
    USERDT:=USERDT+T; USERDS:=USERDS+A[6];%
    HOLDKEY[0]:=A[0];%

```

```

00112900 T 0197
00113000 T 0197
00113100 T 0197
00113200 T 0197
00113300 T 0197
00113400 T 0197
00113500 T 0197
START OF SEGMENT ***** 53
00113600 T 0000
00113700 T 0000
00113800 T 0000
00113900 T 0000
00114000 T 0000
00114100 T 0000
00114200 T 0000
00114300 T 0000
00114400 T 0000
00114500 T 0000
00114600 T 0000
00114700 T 0000
00114800 T 0000
00114900 T 0000
00115000 T 0000
00115100 T 0002
00115200 T 0003
00115300 T 0005
00115400 T 0006
00115500 T 0007
00115600 T 0008
00115700 T 0010
00115800 T 0011
00115810 C 0013
00115900 T 0015
00116000 T 0016
00116100 T 0019
00116200 T 0021
00116300 T 0022
00116400 P 0023
00116500 T 0027
00116600 T 0027
00116700 T 0027
00116800 T 0028
00116900 T 0030
00117000 T 0031
00117100 T 0033
00117200 T 0034
00117300 T 0037
00117400 T 0038
00117500 T 0039
00117600 P 0043
00117700 T 0045
00117800 T 0046
00117900 T 0050
00118000 T 0053
00118100 T 0056
00118200 T 0058
00118300 T 0061

```

THRU;
END DLOUT;

00118400 T 0063
00118500 T 0063
53 IS 68 LONG, NEXT SEG 2

PROCEDURE DLFINAL;

BEGIN
 FORMAT F ("THERE WERE ",I8," FILES ",/

 "OCCUPYING ",I9," DISK SEGMENTS");%

 FORMAT FB("NO. OF USERS PROCESSED WAS:",I8,/,

 "AVE. DISK SPACE PER USER ",I8);%

 WRITE(PRNT[DBL],F,GT14-1,GT9);%
 WRITE(PRNT[DBL],FB,GT15,GT9/GT15);%
END DLFINAL;%

00118600 T 0197
00118700 T 0197
00118800 T 0197
00118900 T 0197
00119000 T 0197
START OF SEGMENT ***** 54
START OF SEGMENT ***** 55
00119100 T 0000
55 IS 15 LONG, NEXT SEG 54
00119200 T 0000
START OF SEGMENT ***** 56
00119300 T 0000
56 IS 16 LONG, NEXT SEG 54
00119400 T 0000
00119500 T 0010
00119600 T 0021
54 IS 22 LONG, NEXT SEG 2

COMMENT ::::: SECTION = 07 = PARAMETER PROCESSING :::::

COMMENT PARAMETERS ARE*;
% OPTN 1 = "OS" - ONSITE REPORT
% OPTN 2 = "TU" - TERMINAL USER REPORT
% OPTN 3 = "DL" - DISK LOGGIN REPORT
% OPTN 4 = "HL" - H/L ANALYSIS
% OPTN 5 = "LINE" - LINE ANALYSIS
% OPTN 6 = "ERR" - ERROR ANALYSIS
% OPTN 7 = "MERGE" - MFRGE INPUT
% OPTN 8 = "SUM" - OUTPUT SUMMARY RECORD ON DISK
% OPTN 9 = "DET" - PRINT GRUBBY DETAILS

PROCEDURE PROCESSPARAMS;%

BEGIN
 FORMAT F1("PARAMETERS"),

 F2("INV PARAMETER(S)...TRY AGAIN");

 INTFGER I,J;
 REAL P;%
 ARRAY A[1:MAXPARAMS];%
 LIST PARALIST(FOR J:= 1 STEP 1 UNTIL MAXPARAMS DO A[J]);
 LABEL GET;
GET:IF NOT COMBOO THEN WRITE(TWX[STOP],F1);%
 READ(TWX,/,PARALIST); J:=0;%

00119700 T 0197
00119800 T 0197
00119900 T 0197
00120000 T 0197
00120100 T 0197
00120200 T 0197
00120300 T 0197
00120400 T 0197
00120500 T 0197
00120600 T 0197
00120700 T 0197
00120800 T 0197
00120810 C 0197
00120900 T 0197
00121000 T 0197
00121100 T 0197
00121200 T 0197
00121300 T 0197
START OF SEGMENT ***** 57
START OF SEGMENT ***** 58
00121400 T 0000
58 IS 13 LONG, NEXT SEG 57
00121500 T 0000
00121600 T 0000
00121700 T 0000
00121800 T 0001
00121900 T 0012
00122000 T 0012
00122100 T 0017

Moore Business Forms, Inc. sv 14121

```

WHILE P:=A[J:=J+1] NEQ 0 DO%
BEGIN COMMENT PROCESS PARAMETER LIST;
  IF I:=MATCH(P,PARAWDS) = 0 THEN
  BEGIN COMMENT INVALID PARAMETER;
    WRITE(TWX,F2);
    GO GET;
  END ELSE CASE (I=1) OF
  BEGIN COMMENT SET TOGS ACCORDING TO PARAMETER;
    ONSITETOG:=TRUE;%
    TERMINALTOG:=TRUE;%
    DISKTOG:=TRUE;%
    HLTOG:=TRUE;%
    LINETOG:=TRUE;%
    ERRTOG:=TRUE;%
    MERGETOG:=TRUE;
    SUMTOG:=TRUE;
    DETOG := TRUE;
  END TOG SETTING;

END PROCESSING OF PARAMETER LIST;
END PROCESSPARAMS;%

```

```

%*****:
PROCEDURE GETMERGE;%
  COMMENT ASKS FOR AND SETS UP MERGE FILE ID;
%*****
BEGIN
  FORMAT FT("MERGE FILE MFID...");%

IF NOT COMBOO THEN
  WRITE(TWX[STOP],FT);%
  READ(TWX,/,MERGEID);%
  FILL MERGEFILE WITH MERGEID,"LOG";%
END GETMERGE;%

```

```

%*****
REAL PROCEDURE FINDLOG;%
%*****
BEGIN
  REAL I,J;%

  FORMAT FN("LOG MFID(S)...");%

  LABEL AGAIN,INS;%
IF NOT COMBOO THEN

```

00122200	T	0021	
00122300	T	0025	
00122400	T	0025	
00122500	T	0028	
00122600	T	0028	
00122700	T	0031	
00122800	T	0032	
00122900	T	0033	
00123000	T	0033	
00123100	T	0035	
00123200	T	0036	
00123300	T	0037	
00123400	T	0038	
00123500	T	0040	
00123600	T	0041	
00123700	T	0042	
00123710	C	0043	
00123800	T	0045	
START OF SEGMENT	*****		59
59 IS	10 LONG,	NEXT SEG	57
00123900	T	0045	
00124000	T	0046	
57 IS	51 LONG,	NEXT SEG	2
00124100	T	0197	
00124200	T	0197	
00124300	T	0197	
00124400	T	0197	
00124500	T	0197	
00124600	T	0197	
START OF SEGMENT	*****		60
START OF SEGMENT	*****		61
61 IS	6 LONG,	NEXT SEG	60
00124610	T	0000	
00124700	T	0000	
00124800	T	0004	
00124900	T	0011	
00125000	T	0015	
60 IS	17 LONG,	NEXT SEG	2
00125100	T	0197	
00125200	T	0197	
00125300	T	0197	
00125400	T	0197	
00125500	T	0197	
START OF SEGMENT	*****		62
00125600	T	0000	
START OF SEGMENT	*****		63
63 IS	6 LONG,	NEXT SEG	62
00125700	T	0000	
00125710	T	0000	

```

WRITE(TWX[STOP],FN);%
READ(TWX[NO]);%
PTR:=GETABSADDR(TWX(0));%
AGAIN:
  WHILE SCAN=NUMBERV DO
  BEGIN COMMENT MAINTAIN FILE QUEUE IN ASC ORDER;
    IF FIDP=0 THEN GO TO INS;%
    FOR I:=0 STEP 1 UNTIL FIDP-1
    DO IF ACCUM LSS FIDQ[I] THEN
    BEGIN COMMENT MOVE EM UP;
      FOR J:=FIDP STEP -1 UNTIL I
      DO FIDQ[J+1]:=FIDQ[J];
      GO INS;%
    END;
    INS: FIDP:=FIDP+1; FIDQ[I]:=ACCUM;%
  END FILE QUEUE MAINTENANCE;%
  IF RESULT NEQ GPMK THEN GO TO AGAIN;
  FINDLOG:=FIDQ[0];%
END FINDLOG;%

```

```

00125800 T 0000
00125900 T 0004
00126000 T 0008
00126100 T 0012
00126200 T 0013
00126300 T 0016
00126400 T 0016
00126500 T 0017
00126600 T 0020
00126700 T 0022
00126800 T 0023
00126900 T 0023
00127000 T 0028
00127100 T 0028
00127200 T 0029
00127300 T 0032
00127400 T 0033
00127500 T 0034
00127600 T 0035

```

62 IS 38 LONG, NEXT SEG 2

```

%*****
PROCEDURE GETWORK;%
%*****
BEGIN
  LABEL AGAIN;%

  REAL I,T;%
  INTEGER N;%
%.....
STREAM PROCEDURE MAKEFIDLINE(N,FIDS,TOG,MERG,ARRY);%
  VALUE MERG,N,TOG;%
%.....
BEGIN DI:=ARRY; SI:=FIDS;%
  DS:=19 LIT"FILES: CURRENT...";
  NCUPSI;UPDI; DS:=7 CHR; DS:=2 LIT ", ";
  TOG(DI:=ARRY; DI:=DI+63;%
  DI:=DI+47; SI:=LOC MERG;%
  DS:=8 LIT"MERGE..."; DS:=WDS);%
END MAKE FIDLINE;%

```

```

00127700 T 0197
00127800 T 0197
00127900 T 0197
00128000 T 0197
00128100 T 0197
00128200 T 0000
00128300 T 0000
00128400 T 0000
00128500 T 0000
00128600 T 0000
00128700 T 0000
00128800 T 0000
00128900 T 0000
00129000 T 0003
00129100 T 0005
00129200 T 0007
00129300 T 0007
00129400 T 0009

```

START OF SEGMENT ***** 64

```

AGAIN:
  IF MFID:=FINDLOG NEQ 0 THEN
  BEGIN FILL DISK WITH MFID,"LOG ";
    READ(DISK[NO],10,LOGREC[*]);%
    LINEARRY[0]:=MIXARRY[0]:=-BEGINTIME:=LOGREC[0].TIMEF;
    PROCESSPARAMS;%
    IF MERGETOG THEN GETMERGE;%
    NFIDLINES:=FIDP DIV 9 + 1;%
    FOR I:=0 STEP 1 UNTIL NFIDLINES
    DO BLANKOUT(FIDLINES[I,*]);%

```

```

00129500 T 0009
00129600 T 0010
00129700 T 0011
00129800 T 0015
00129900 P 0020
00130000 T 0023
00130100 T 0024
00130200 T 0025
00130300 T 0027
00130400 T 0027

```

```

FOR I:=0 STEP 1 UNTIL NFIDLINES-2
DO BEGIN
    MAKEFIDLINE(9,FIDQ[T],0,0,FIDLINE[I,*]);%
    T:=T+9;%
END;%
MAKEFIDLINE(N:=FIDP MOD 10,FIDQ[T],MERGETOG,MERGEID,
FIDLINE[NFIDLINES-1,*]);%
FIDP:=0;%
END ELSE GO AGAIN;%
END GETWORK;%

```

```

00130500 T 0032
00130600 T 0035
00130700 T 0036
00130800 T 0039
00130900 T 0040
00131000 T 0041
00131100 T 0043
00131200 T 0045
00131300 T 0045
00131400 T 0045

```

64 IS 48 LONG, NEXT SEG 2

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
COMMENT ::::: SECTION = 08 = INITIALIZE, OUTER BLOCK :::::
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%*****
PROCEDURE INITIALIZE;%
%*****
BEGIN

```

```

00131500 T 0197
00131600 T 0197
00131700 T 0197
00131800 T 0197
00131900 T 0197
00132000 T 0197
00132100 T 0197
00132200 T 0197

```

START OF SEGMENT ***** 65

```

    DEFINE Y=DATE3#,D=DATE2#,M=DATE1#;%

```

```

    INTEGER H,P,Q;%
    REAL I;%
    ARRAY A[0:6];%
    LABEL OWT;%
    FILL A[*] WITH "    SUN",

    "    MON",
    "    TUES",
    "WEDNES",
    "    THURS",
    "    FRI",
    "    SATUR";

```

```

00132300 T 0000
00132400 T 0000
00132500 T 0000
00132600 T 0001
00133000 T 0001

```

START OF SEGMENT ***** 66

```

    Y:=TIME(0);%
D+Y,[30:6]X100+Y,[36:6]X10+Y,[42:6];
Y:=Y,[18:6]X10+Y,[24:6];%SAVE
I:=-1;%
FOR H:=31,IF Y MOD 4 =0 THEN 29 ELSE 28,31,30,31,
30,31,31,30,31,30,31
DO IF D LEQ (DAYSIN[I:=I+1]:=H) THEN GO OWT ELSE
BEGIN D:=D-H; M:=M+1 END;%
OWT:IF M LSS 2 THEN BEGIN Q:=M+11; P:=Y-1 END ELSE
BEGIN Q:=M-1; P:=Y END;%
M:=M+1;%
DAY:=A[(QX26-2) DIV 10+D+P+P,[36:10]+1) MOD 7];%
FILL UNIT[*] WITH "MTA","MTB","MTC","MTD",

```

66 IS 7 LONG, NEXT SEG 65

```

00133100 T 0003
00133200 T 0003
00133300 T 0003
00133400 T 0003
00133500 T 0003
00133600 T 0003
00133700 T 0003
00133800 T 0004
00133900 T 0009
00134000 T 0011
00134100 T 0012
00134200 T 0025
00134300 T 0037
00134400 T 0042
00134500 T 0046
00134600 T 0049
00134700 T 0052
00134800 T 0053
00134900 T 0059

```

START OF SEGMENT ***** 67

```

    "MTE","MTF","MTH","MTJ","MTK","MTL","MTM",
    "MTP","MTR","MTR","MTS","MTT","DRA","DRB",
    "DKA","DKB","LPA","LPB","CPA","CRA","CRB",
    "SPO","PPA","PRA","PPB","PRB","DCA";%

```

67 IS 31 LONG, NEXT SEG 65

```

    FILL ASTERISK[*] WITH "    *",

```

```

00135000 T 0060
00135100 T 0060
00135200 T 0060
00135300 T 0060
00135400 T 0060

```



```

"    **", "    **", "    **";%
FILL LEVELARRY[*] WITH "PROGRAM",
"SESSION", "USER  ", "REPORT ";%
FILL ONOFFARRY[*] WITH "ON LINE ", "OFF LINE";%
FILL PARAWDS[*] WITH MAXPARAM, "OS", "TU", "DL",
"HL", "LINE", "ERR", "MERGE", "SUM", "DET";
FILL CPLRNM[*] WITH 9, "BASIC  ", "ALGOL  ", "COBOL  ", "FORTRAN",
    "TSPOL  ", "XALGOL  ", "COBOL68", "PASCAL  ", "ESPOL  ";
FOR H:=1 STEP 1 UNTIL CPLRNM[0] DO COMPILS[H]:=0;
FOR H:= 0 STEP 1 UNTIL NUMLINE DO LINEARRY[H]:=0;
FOR H:=0 STEP 1 UNTIL MIXMAX DO MIXARRY[H]:=0;
BLANKOUT(ULINE);%
END INITIALIZE;%

```

```

START OF SEGMENT ***** 68
00135500 T 0062
68 IS 4 LONG, NEXT SEG 65
00135600 T 0062
START OF SEGMENT ***** 69
00135700 T 0064
69 IS 4 LONG, NEXT SEG 65
00135800 T 0064
START OF SEGMENT ***** 70
70 IS 2 LONG, NEXT SEG 65
00135900 T 0066
START OF SEGMENT ***** 71
00136000 P 0067
71 IS 10 LONG, NEXT SEG 65
00136010 C 0067
START OF SEGMENT ***** 72
00136020 C 0069
72 IS 10 LONG, NEXT SEG 65
00136030 C 0069
00136040 C 0075
00136050 C 0080
00136100 T 0085
00136200 T 0086
65 IS 92 LONG, NEXT SEG 2

```

```

%*****
COMMENT OUTER BLOCK;
%*****
INITIALIZE;
GETWORK;%
PTRBASE := GETABSADDR(LOGREC[1]);
IF TERMINALTOG OR ONSITETOG OR SUMTOG THEN
BEGIN COMMENT REMOTE AND/OR LOCAL REPORTS;%
% DELETE A CARD
    PUTHEADER(USERPT);
    IF DETOG THEN
        IF TERMINALTOG OR ONSITETOG THEN PUTHEADER(ONSITE);%
        SORT(OSOUT, OSIN, 0, OSVAL, OSCOMP, OSRECSI, 4000);%
        NEXTPAGE(ONSITE); OSFINAL;%
        HLANA; HLTG:=FALSE;%
        ERRANA; ERRTOG:=FALSE;%
        IF TERMINALTOG OR LINETOG THEN
            BEGIN LINEANA; LINETOG:=FALSE END;%
        END REMOTE LOCAL REPORTS;%
    IF DISKTOG THEN
        BEGIN COMMENT DISK LOGGING REPORT;
            MFID:=FIDQ[FIDP:=0];%
            FILL DISK WITH MFID;%
            CLEAR(9,GT6);%
            PAGENO:=LINECTR:=0;%
            PUTHEADER(DISKRPT);%
            SORT(DLOUT, DLIN, 0, DVAL, DCOMP, 7, 4000);%
            NEXTPAGE(DISKRPT); DLFINAL;%
        END DISK LOGGING;%

```

```

00136300 T 0197
00136400 T 0197
00136500 T 0197
00136600 T 0197
00136700 T 0198
00136710 C 0199
00136800 T 0201
00136900 T 0202
00137000 P 0202
00137100 P 0202
00137190 C 0203
00137200 T 0203
00137300 T 0206
00137400 T 0226
00137500 T 0227
00137600 T 0229
00137700 T 0230
00137800 T 0231
00137900 T 0232
00138000 T 0232
00138100 T 0233
00138200 T 0233
00138300 T 0235
00138400 T 0239
00138500 T 0240
00138600 T 0241
00138700 T 0242
00138800 T 0262
00138900 T 0263

```

```
IF HLTOG OR LINETOG OR ERRTOG THEN %
BEGIN CLEANER:=TRUE; WHILE NOT(COSIN(DUMMY)) DO; END;
IF HLTOG THEN HLANA;%
IF LINETOG THEN LINEANA;%
IF ERRTOG THEN ERRANA;%
END. % OF PROGRAM
```

```
00139000 T 0263
00139100 P 0264
00139200 T 0270
00139300 T 0271
00139400 T 0272
00139500 T 0273
2 IS 277 LONG, NEXT SEG 1
1 IS 2 LONG, NEXT SEG 0
83 IS 69 LONG, NEXT SEG 0
```

NUMBER OF ERRORS DETECTED = 0. COMPILATION TIME = 1035 SECONDS.

PRT SIZE = 238; TOTAL SEGMENT SIZE = 2612 WORDS; DISK SIZE = 145 SEGS; NO. PGM. SEGS = 83

ESTIMATED CORE STORAGE REQUIRED = 12232 WORDS.

ESTIMATED AUXILIARY MEMORY REQUIRED = 0 WORDS.

NUMBER OF CARD-IMAGES PROCESSED = 1529.

LABEL 00000000LINE 00177069?COMPILE LOGAN/DISK TSPOL LIBRARY

TSPOL /LOGAN

Moore Business Forms, Inc. sv