

LABEL 000000000PRINTER0017509800 EXECUTE OBJECT/READ;FILE SOURCEFILE=SYMBOL/LOGAN;END*

OBJECT /READ

SYMBOL/LOGAN

Data Documents Inc.

% 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

00001100
00001200
00001300

1 COMMENT: THIS MATERIAL IS PROPRIETARY TO BURROUGHS
2 CORPORATION AND IS NOT TO BE REPRODUCED, USED
3 OR DISCLOSED EXCEPT IN ACCORDANCE WITH PROGRAM
4 LICENSE OR UPON WRITTEN AUTHORIZATION OF THE
5 PATENT DIVISION OF BURROUGHS CORPORATION,
6 DETROIT, MICHIGAN 48232.

00001310
00001311
00001312
00001313
00001314
00001315

7
8 COPYRIGHT (C) 1968, 1971, 1972
9 BURROUGHS CORPORATION DETROIT, MICHIGAN
10 AA32500;

00001316
00001317
00001318
00001319
00001400

11 BEGIN

12 % SECTION 01 - DEFINES
13 % SECTION 02 - GLOBALS
14 % SECTION 03 - UTILITY PROCEDURES, SCANNER, ETC.
15 % SECTION 04 - SNAP REPORT PROCEDURES
16 % SECTION 05 - ONSITE REPORTER
17 % SECTION 06 - DISK LOGGING REPORTER
18 % SECTION 07 - PARAMETER PROCESSING, INPUT SETUP
19 % SECTION 08 - INITIALIZATION, OUTER BLOCK

00001500
00001600
00001700
00001800
00001900
00002000
00002100
00002200

20 *****
21 COMMENT ::::: SECTION 01 - DEFINES :::::
22 *****
23 COMMENT DEFINES FOR THE VARIOUS LOG RECD TYPES;

00002300
00002400
00002500
00002600
00002700

24 DEFINE BOJ = 2#,
25 EOJ = 3#,
26 PBEOJ = 4#,
27 OPEN = 5#,
28 CLOSEF = 6#,
29 HALT = 7#,
30 EOJSTAT = 8#,
31 CLSSTAT = 9#,
32 ON = 10#,
33 OFF = 11#,
34 CHARGE = 12#,
35 DLOG = 13#,
36 DATEMESS = 14#,
37 TIMEMESS = 15#

00002800
00002900
00003000
00003100
00003200
00003300
00003400
00003500
00003600
00003700
00003800
00003900
00004000

38 ;%
39 COMMENT DEFINES FOR FIRST WORD OF LOG RECDS;

00004100
00004200

40 DEFINE MIXF = [2:5]#,
41 RLF = [7:1]#,
42 LINEF = [8:8]#,
43 TYPEF = [16:7]#,
44 SPOF = [23:1]#,
45 TIMEF = [25:23]#;

00004300
00004400
00004500
00004600
00004700
00004800

46 COMMENT DEFINE REPORT NOS. ;%
47 DEFINE ONSITE = 0#,
48 USERPT = 2#,

00004900
00005000
00005100

49 DISKRPT = 1# ;%
50 COMMENT LEVEL NAMES ;%
51 DEFINE SESSION = 1#,

00005200
00005300
00005400

52 USER = 2#,
53 REPORT = 3#,
54 PROGRAM = 0# ;%

00005500
00005600
00005700

55 COMMENT ASSORTED DEFINES;

00005800

56 DEFINE UPSI = SI:=SI+1#,
57 UPDI = DI:=DI+1#,

00005900
00006000

Data Documents/Inc

```

      UPTALLY = TALLY:=TALLY+1#,          00006100
      GPMK    = "*"#,                    00006200
      SCAN    = RESULT:=SCN(PTR,PTR,ACCUM)#, 00006300
      NUMBERSCAN=NUMBERSCN(PTR,PTR)#,     00006400
      NAMEV   = 1#,                      00006500
      NUMBERV = 0#,                      00006600
      PRNTS   = PRNT#,                   00006700
      SCAN2   = SCAN; SCAN#              00006800
;X
      DEFINE EQUIV(EQUIV1,EQUIV2)=
      REAL(BOOLEAN(EQUIV1) EQV BOOLEAN(EQUIV2))=REAL(NOT FALSE)#;X 00007000
      COMMENT SORT RECORD SIZES;        00007100
      DEFINE OSRECSI = 10#,              00007200
      DLRECSI = 8#,                     00007300
      MAXSI = 12#,                      00007400
      MIXRECSI = 12#;X                  00007500
      COMMENT SCRATCH FILE BLOCK SIZES; 00007600
      DEFINE OSBLKSI = 30xOSRECSI#,     00007700
      DLBLKSI = 30xDLRECSI#            00007800
;X
      COMMENT SCRATCH FILE ROW SIZES;   00007900
      DEFINE OSROWSI = 1000xOSRECSI#,   00008000
      DLROWSI = 1000xDLRECSI#          00008100
;X
      COMMENT ARRAY SIZES;               00008200
      DEFINE HLQSI = 30#,                00008300
      FIDQSI = 10#,                     00008400
      NUMLINES = 48#,                   00008500
      MAXPARAMS = 8#,                   00008600
      MIXMAX = 29#,                     00008700
      NUMLINE = NUMLINES#,               00008800
      MAXKEY = 4#,                       00008900
      MAXPARAM = 8#;                     00009000
      *****
      COMMENT :::: SECTION = 02 = GLOBAL DECS ::::;X 00009100
      *****
      BOOLEAN COMBOO;% CARD/TTY INPUT    00009200
      COMMENT GLOBAL ARRAYS;%           00009300
      ARRAY LEVELARRY[0:3],             % LEVEL NAMES 00009400
      ONOFFARRY[0:1],                   % ON/OFF LINE 00009500
      UNIT[1:31],                        % UNIT MNEMONICS 00009600
      RTE[1:31],                          % RUN TIME ERRORS THIS UNIT
      LINE[0:16],                         % PRINT LINE 00009700
      ULINE[0:16],                        % RPTU PRINT LINE 00009800
      LLTBL[1:NUMLINE,0:3],% LINE INFO-USER,LOG ON TIME 00009900
      FIDLINE[0:FIDQSI DIV 9 +1,0:16],% FILE ID PRNT LINES 00010000
      HLQ[0:HLQSI],                       % QUEUE OF H/L TIMES 00010100
      LINEARRY[0:NUMLINE],% ACCUM LINE USAGE 00010200
      HOLDKEY[0:MAXKEY], % SORT KEY SAVE AREA 00010300
      PARAWDS[0:MAXPARAM],% OPTIONS 00010400
      DAYSIN[0:11], % DAYS IN MONTH 00010500
      FIDQ[0:FIDQSI], % QUEUE OF FILES TO PROCESS 00010600
      PSTOTALS[0:1], % RPT PST TOTS,LOCAL AND REMOTE 00010700
      IOTOTALS[0:1], % RPT IOT TOTS,LOCAL AND REMOTE 00010800
      ETOTALS[0:1], % RPT ET TOTS,LOCAL AND REMOTE 00010900
      MERGEINPUT[0:7], % MERGE INPUT RECORD 00011000
      ASTERISK[0:3]; % ASTERISKS 00011100
      SAVE ARRAY LOGREC[0:MAXSI]; % WORK RECORD 00011200
      ARRAY MIXTBL[1:MIXMAX,0:9];X 00011300
      COMMENT REAL GLOBALS(SIC);X 00011400

```

Data Documents/Inc.

```

REAL GT1,GT2,GT3,GT4;      % GLOBAL TEMPORARIES      00012000
REAL GT5,GT6,GT7,GT8,
GT9,GT10,GT11,GT12,
GT13,GT14,GT15;          % MORE OF THE SAME        00012300
REAL GT16,GT17,GT18,GT19;%                          00012400
REAL DAY,                  % "MON","TUES",ETC.        00012500
DATE1,DATE2,DATE3,        % MM/DD/YY          00012600
UCT,                        % USER CANDE TIME  00012610
UTI,                        % USER TERMINAL TIME 00012620
RCT,                        % REPORT CANDE TIME  00012630
RTT,                        % REPORT TERMINAL TIME 00012640
MFID,                       % LOG FILEID        00013100
MERGEID,                    % MERGE FILE ID     00013200
PTR,                        % SCAN POINTER      00013300
PTRBASE,                    % ADDR OF WORK RECURD 00013400
RESULT,                     % VALUE RETURN BY SCANNER 00013500
ACCUM                       % SCANNER STUFF WORD 00013600
;%
COMMENT GLOBAL INTEGERS FOR ASSORTED PURPOSES;
INTEGER HLP,                % INDEX INTO HLG    00013900
FIDP,                      % INDEX INT FIDG    00014000
PAGENO,                    % REPORT PAGE NUMBER 00014100
OSJOBS,                    % NO. OF ONSITE JOBS 00014200
TUJOBS,                    % NO. OF TERMINAL JOBS 00014300
UPAGENO,                   % RPTU PAGE NUMBER  00014400
LINECTR,                   % REPORT PAGE LINE COUNTER 00014500
ULINECTR,                  % RPTU LINE COUNTER 00014600
NFIDLINES,                 % NO OF FILE ID PRNT LINES 00014700
REMUSERS,                  % NO OR REMOTE USERS 00014800
REMUSE,                    % 1 IFF THIS USER RAN A REM JOB 00014900
OSUSERS,                   % NO OF ONSITE USERS 00015000
OSUSE,                     % 1 IFF THIS USER RAN AN OS JOB 00015100
NL,                        % NO OF LINES IN USE 00015200
COMLINES,                  % NO OF LINES W/ NO LON OR OFF 00015300
BEGINTIME,                 % EARLIEST TIME FOUND IN LOG 00015400
LASTIME;                   % AS ABOVE,LATEST   00015500
COMMENT GLOBAL BOOLEANS, MOSTLY REPORT OPTION TOGS;
BOOLEAN TERMINALTOG,        % TU SUMMARY RPT OUTPUT 00015600
ONSITETO,                  % ONSITE SUMMARY OUTPUT 00015800
DISKTOG,                   % DISK LOG SUMMARY OUTPUT 00015900
HLTOG,                     % H/L ANALYSIS        00016000
LINETO,                     % LINE ANALYSIS       00016100
MERGETOG,                  % MERGED INPUT        00016200
MERGEND,                   % NO MORE MERGE INPUT 00016300
SUMTOG,                    % SUMMARY OUTPUT      00016400
CLEANER,                   % SORT RELEASE TOG    00016500
ERRTOG,                    % DEVICE ERRORS ANALYSIS 00016600
;%
COMMENT FILE DECLARATIONS;
FILE IN DISK DISK SERIAL(2,10,300),
MERGEFILE DISK SERIAL(2,OSRECS1,OSBLKSI),
PRNT 16(2,17),
UPRNT 16(2,17),
TWX 19(1,10)
;%
SAVE FILE OUT SUMFILE DISK SERIAL[20:OSROWSI]
"USER" "ISS-LOG" (2,OSRECS1,OSBLKSI,SAVE 10);%
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
COMMENT ::::: SECTION - 03 - UTILITY PROCEDURES;
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

```

%*****
REAL STREAM PROCEDURE GETABSADDR(WD);
%*****
1 BEGIN
2   DI:=WD; GETABSADDR:=DI;%
3 END GETABSADDR;%
4 %*****
5 REAL STREAM PROCEDURE DECV(N,P);%
6   VALUE N,P;%
7 %*****
8 BEGIN SI:=LOC P; DI:=LOC DECV;%
9   DI:=DI+8; DI:=DI-N; DS:=N DECV;%
10 END DECV;%
11 %*****
12 INTEGER STREAM PROCEDURE OCV(WD,POS);%
13   VALUE POS;%
14   COMMENT CONVERTS 2 DEC AT WD,POS(POS) TO OCT;
15 %*****
16 BEGIN SI:=WD; SI:=SI+POS;
17   DI:=LOC OCV; DS:= 2 OCT;
18 END OCV;%
19 %*****
20 INTEGER STREAM PROCEDURE SCN(PTRV,PTR,ACCUM);%
21   VALUE PTRV;%
22   COMMENT RETURNS 0 IFF STRING OF ALPHA,
23   RETURNS THE SPEC CHAR IFF ONE FOUND;%
24 %*****
25 BEGIN
26   LABEL L;%
27   SI:=PTRV;%
28 L: IF SC = " " THEN BEGIN UPSI; GO L END;
29   IF SC = ALPHA THEN%
30   BEGIN COMMENT NAMES;
31     DI:=ACCUM;DS:=LIT"0"; DS:=7 LIT " ";DI:=DI-7;%
32     7(DS:=CHR; IF SC =ALPHA THEN ELSE JUMP OUT);%
33   END ELSE
34   BEGIN COMMENT SPEC CHAR;
35     DI:=LOC SCN ; DI:=DI+7;
36     DS:=CHR;%
37   END SPEC CHAR HANDLING;
38   DI:=PTR;PTRV:=SI; SI:=LOC PTRV; DS:=WDS;
39 END SCAN;%
40 %*****
41 REAL STREAM PROCEDURE NUMBERSCN(PTRV,PTR);%
42   VALUE PTRV;%
43   COMMENT SCANS OFF NEXT NUMBER, CONVERTS TO REAL;
44 %*****
45 BEGIN LABEL L;
46   LOCAL A;%
47   SI:=PTRV; DI:= LOC NUMBERSCN;%
48 L: IF SC LSS "0" THEN BEGIN UPSI; GO L END;%
49   A:=SI;
50   7(UPSI;UPTALLY;
51   IF SC LSS "0" THEN JUMP OUT);
52   SI:=A; A:=TALLY; DS:=A OCT;
53   A:=SI; DI:=PTR; SI:=LOC A; DS:=WDS;%
54 END NUMBERSCN;%
55 %*****
56 BOOLEAN STREAM PROCEDURE COMPARE(N,A,B);%
57   VALUE N;

```

```

00018000
00018100
00018200
00018300
00018400
00018500
00018600
00018700
00018800
00018900
00019000
00019100
00019200
00019300
00019400
00019500
00019600
00019700
00019800
00019900
00020000
00020100
00020200
00020300
00020400
00020500
00020600
00020700
00020800
00020900
00021000
00021100
00021200
00021300
00021400
00021500
00021600
00021700
00021800
00021900
00022000
00022100
00022200
00022300
00022400
00022500
00022600
00022700
00022800
00022900
00023000
00023100
00023200
00023300
00023400
00023500
00023600
00023700
00023800
00023900

```



```

%*****
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;
%*****
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;%
  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;%
%*****
PROCEDURE PRINTLINE(LINE,WHICH);%
  VALUE WHICH; INTEGER WHICH;%
  ARRAY LINE[0];%
%*****
BEGIN
  SWITCH FILE SWF:=PRNT,UPRNT;%
  IF WHICH=USERPT THEN ULINECTR:=ULINECTR+2 ELSE
  LINECTR:=LINECTR+2;
  WRITE(SWF[WHICH DIV 2][DBL],17,LINE[*]);%
END PRINTLINE;%
%*****
STREAM PROCEDURE BLANKOUT(LINE);%
%*****
BEGIN
  DI:=LINE;DS:= 8 LIT" ";
  SI:=LINE; DSI:=16 WDS;
END BLANKOUT;%
%*****
STREAM PROCEDURE CLEAR(N,A);
  VALUE N;%
  COMMENT SET N+1 WORDS TO ZERO;
%*****
BEGIN
  DI:=A; DS:=8 LIT"0";
  SI:=A; DSI:=N WDS;

```

```

00030000
00030100
00030200
00030300
00030400
00030500
00030600
00030700
00030800
00030900
00031000
00031100
00031200
00031300
00031400
00031500
00031600
00031700
00031800
00031900
00032000
00032100
00032200
00032300
00032400
00032500
00032600
00032700
00032800
00032900
00033000
00033100
00033200
00033300
00033400
00033500
00033600
00033700
00033800
00033900
00034000
00034100
00034200
00034300
00034400
00034500
00034600
00034700
00034800
00034900
00035000
00035100
00035200
00035300
00035400
00035500
00035600
00035700
00035800
00035900

```

Data Documents/Inc.

```

END CLEAR;%                                00036000
%*****                                00036100
PROCEDURE NEXTPAGE(WHICH);%                00036200
    VALUE WHICH; INTEGER WHICH;%          00036300
%*****                                00036400
BEGIN                                      00036500
    REAL I;%                               00036600
    SWITCH FILE SWF:=PRNT,UPRNT;%         00036700
    FORMAT HF(X26,"** B-5500 TIME SHARING SYSTEM LOG ", 00036800
    "REPORT **",X7,A6,"DAY  ",,2(I2,"/"),I2,X11 00036900
    ,"PAGE ",I5);                          00037000
    LIST HL(DAY,DATE1,DATE2,DATE3,        00037100
    IF WHICH=USERPT THEN UPAGENO:=UPAGENO+1 ELSE 00037200
    PAGENO:=PAGENO+1);%                   00037300
    WRITE(SWF[WHICH DIV 2][PAGE1]);%      00037400
    WRITE(SWF[WHICH DIV 2][DBL],HF,HL);%  00037500
    IF WHICH=USERPT THEN ULINECTR:=6+2*NFIDLINES ELSE 00037600
    LINECTR:=6+2*NFIDLINES;%             00037700
    FOR I:=0 STEP 1 UNTIL NFIDLINES      00037800
    DO PRINTLINE(FIDLINE[I,*],WHICH);%   00037900
END NEXTPAGE;%                             00038000
%*****                                00038100
PROCEDURE PUTHEADER(WHICH);%              00038200
    VALUE WHICH;%                         00038300
    INTEGER WHICH;                        00038400
%*****                                00038500
BEGIN                                      00038600
    SWITCH FILE SWFILE:=PRNT,UPRNT;%     00038700
    SWITCH FORMAT SWF:=                   00038800
    (X51,"U S E R  J O B  S U M M A R Y"), 00038900
    (X50,"U S E R  D I S K  S U M M A R Y"),% 00039000
    (X55,"U S E R  S U M M A R Y");%       00039100
    SWITCH FORMAT SWFF:=                  00039200
    ("USER",X4,"PN1",X5,"PN2",X5,        00039300
    "LL",X5,"BOJ",X7,"EOJ",X7,         00039400
    "PST",X7,"IOT",X4,"ET/TT",X9,      00039500
    "CORE",X1,"TRMNTN",X2,"MFID",X4,    00039600
    "FID",X6,"UNIT",X4,"TIME"),        00039700
    ("USER",X5,"MFID",X5,"FID",X6,      00039800
    "CREATN DATE & TIME",X1,"LOGOFF TIME",X2, 00039900
    "# DISK SEGS",X5,"TIME USED");%     00040000
    ("USER",X27,"PST",X7,"IOT",X7,"ET",X8,"C&E",X4, 00040100
    "TERMINAL",X1,"CORE/SECS");%       00040200
    NEXTPAGE(WHICH);%                   00040300
    WRITE(SWFILE[WHICH DIV 2][DBL],SWF[WHICH]); 00040400
    WRITE(SWFILE[WHICH DIV 2][DBL],SWFF[WHICH]);% 00040500
END PUTHEADER;%                          00040600
%*****                                00040700
BOOLEAN PROCEDURE DISKREAD;%              00040800
%*****                                00040900
BEGIN                                      00041000
    LABEL GET,EOF,THRU;                  00041100
    GET:READ(DISK,10,LOGREC[*])[EOF];%   00041200
    IF LOGREC[0]*REAL(NOT FALSE) THEN GO TO EOF; 00041300
    IF LOGREC[0].TYPEF LSS BOJ THEN GO GET; 00041400
    PTR:=PTRBASE;                        00041500
    GO THRU;                              00041600
    EOF:IF FIDQ[FIDP]=FIDP+1 NEQ 0 THEN 00041700
    BEGIN CLOSE(DISK);                  00041800
    MFID:=FIDQ[FIDP];%                 00041900

```

Data Documents, Inc.

Data Documents/Inc.

FILL DISK WITH MFID;%

GO TO GET;

00042000

END;%

00042100

DISKREAD:=TRUE;

00042200

THRU;

00042300

END DISKREAD;

00042400

00042500

COMMENT ::::: SECTION - 04 - SNAP REPORTS :::::;

00042600

00042700

00042800

PROCEDURE HLANA;%

00042900

00043000

BEGIN

00043100

FORMAT F1("THERE WERE ",I3, " HALT LOADS");%

00043200

FORMAT F2("HALT LOAD TIMES WERE : ");%

00043300

F3(X2,A3,A5);%

00043400

F4("MEAN TIME BETWEEN HALT LOADS... ",A3,A5);%

00043500

FORMAT F5("TIME PERIOD COVERED WAS FROM ",A3,A5,

00043600

" TO ",A3,A5);%

00043700

REAL A,B,C,D,I;%

00043800

LABEL XIT;%

00043900

NEXTPAGE(ONSITE);%

00044000

STIMER(BEGINTIME,A,B); STIMER(LASTIME,C,D);%

00044100

WRITE(PRNT[DBL],F5,A,B,C,D);%

00044200

WRITE(PRNT[DBL],F1,HLP);%

00044300

IF HLP = 0 THEN GO XIT;%

00044400

WRITE(PRNT[DBL],F2);%

00044500

FOR I:=1 STEP 1 UNTIL HLP DO

00044600

BEGIN COMMENT H/L TIMES;

00044700

STIMER(HLQ[I],A,B);

00044800

WRITE(PRNT[DBL],F3,A,B);

00044900

END PRINTING HL TIMES;

00045000

STIMER((LASTIME-BEGINTIME)/HLP,A,B);%

00045100

WRITE(PRNT[DBL],F4,A,B);%

00045200

XIT;

00045300

END HLANA;%

00045400

00045500

PROCEDURE LINEANA;%

00045600

COMMENT PRINTS OUT THE AMOUNTS AND %S OF TIME

00045700

THAT I LINES WERE IN USE;%

00045800

00045900

BEGIN BOOLEAN FIRST;%

00046000

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

00046100

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

00046200

REAL I,L1,L2;%

00046300

REAL MAX,T,TT;%

00046400

LABEL XIT;%

00046500

NEXTPAGE(ONSITE); WRITE(PRNT[DBL],F1);%

00046600

FOR I:=NUMLINES STEP -1 UNTIL 0 DO

00046700

BEGIN TT:=TT+LINEARRY[I]; IF LINEARRY[I] NEQ 0 THEN

00046800

IF NOT FIRST THEN BEGIN FIRST:=TRUE;MAX:=I END;

00046900

END;

00047000

IF TT=0 THEN GO TO XIT;%

00047100

FOR I:=0 STEP 1 UNTIL MAX DO

00047200

BEGIN

00047300

STIMER(T:=LINEARRY[I],L1,L2);%

00047400

WRITE(PRNT[DBL],F2,I+COMLINES,L1,L2,100*(T/TT));%

00047500

END;

00047600

XIT;

00047700

END LINEANA;%

00047800

00047900

```

%*****
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;
%*****
COMMENT ::::: SECTION - 05 - ONSITE REPORTER :::::
%*****
COMMENT LAYOUT OF THE ONSITE SORT RECORDS;
%
S[0] ::= USER ID
%
S[1] ::= LINE NO.
%
S[2] ::= PRQG NAME 1
%
S[3] ::= PRQG NAME 2
%
S[4] ::= BOJ TIME
%
S[5] ::= EOJ TIME
%
S[6] ::= QPU 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 LUGGING 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
SI:=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;%
%*****
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);

```

```

00048000
00048100
00048200
00048300
00048400
00048500
00048600
00048700
00048800
00048900
00049000
00049100
00049200
00049300
00049400
00049500
00049600
00049700
00049800
00049900
00050000
00050100
00050200
00050300
00050400
00050500
00050600
00050700
00050800
00050900
00051000
00051100
00051200
00051300
00051400
00051500
00051600
00051700
00051800
00051900
00052000
00052100
00052200
00052300
00052400
00052500
00052600
00052700
00052800
00052900
00053000
00053100
00053200
00053300
00053400
00053500
00053600
00053700
00053800
00053900

```

Data Documents/Inc.

Data Documents/Inc.

```

MAKEFILEDETAILLINE(LINE,S[5],S[6],UNIT[LUN],
TIMER(ABS(S[9])));%
PRINTLINE(LINE,ONSITE);%

```

00054000
00054100
00054200

```

XIT:
END FILEDETAIL;%

```

```

STREAM PROCEDURE MAKELEVELLINE(LINE,ASTERISK,LEVEL,
PST,IOT,ET,COR);
VALUE PST,IOT,ET,COR,ASTERISK,LEVEL;%

```

00054300
00054400
00054500
00054600
00054700
00054800

```

*****
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;%

```

00054900
00055000
00055100
00055200
00055300
00055400
00055500
00055600
00055700

```

END MAKELEVELLINE;%
*****
PROCEDURE LEVEL(WHICH,PST,IOT,ET,COR);
VALUE WHICH,PST,IOT,ET,COR;%
INTEGER WHICH;
REAL PST,IOT,ET,COR;%

```

00055800
00055900
00056000
00056100
00056200
00056300

```

*****
BEGIN
BLANKOUT(LINE);
MAKELEVELLINE(LINE,ASTERISK[WHICH],LEVELARRY[WHICH],
TIMER(PST),TIMER(IOT),TIMER(ET),COR);%
PRINTLINE(LINE,ONSITE);%

```

00056400
00056500
00056600
00056700
00056800
00056900

```

END LEVEL;%
*****
PROCEDURE SPLIT(I); VALUE I;%

```

00057000
00057100
00057200

```

REAL I;%
*****
BEGIN DEFINE P=GT10#, IO=GT11#, E=GT12#;%
PSTOTALS[I]:=PSTOTALS[I]+P;%
IOTOTALS[I] := IOTOTALS[I] + IO;%
ETOTALS[I]:=ETOTALS[I]+E;%
IF I=0 THEN OSUSE:=1 ELSE REMUSE:=1;%

```

00057300
00057400
00057500
00057600
00057700
00057800
00057810
00057900

```

END SPLIT;%
*****
STREAM PROCEDURE MAKEDETAILLINE(LINE,USER,PN1,PN2,LL,BOJT,
EOJT,PST,IOT,ET,COR,HEND);%
VALUE USER,PN1,PN2,LL,BOJT,EOJT,PST,IOT,ET,COR,HEND;%

```

00058000
00058100
00058200
00058300

```

*****
BEGIN
SI:=LOC USER; DI:=LINE;
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;%

```

00058400
00058500
00058600
00058700
00058800
00058900
00059000
00059100
00059200

```

END MAKEDETAILLINE;%
*****
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;%

```

00059300
00059400
00059500
00059600
00059700
00059800

Data Documents/Inc.

	REAL USER,PN1,PN2,BOJT,EOJT,PST,IOT,ET,COR,HEND,LL;%	00059900
	%*****	00060000
	BEGIN	00060100
1	BLANKOUT(LINE);%	00060200
2	MAKETAILLINE(LINE,USER,PN1,PN2,LL,TIMER(BOJT),	00060300
3	TIMER(EOJT),TIMER(PST),TIMER(IOT),TIMER(ET),	00060400
4	COR,HEND);%	00060500
5	PRINTLINE(LINE,ONSITE);%	00060600
6	END DETAIL;%	00060700
7	COMMENT LAYOUT OF THE USER MASTER RECORD;%	00060900
8	% U[0] := USER	00061000
9	% U[1] := CPU TIME	00061100
10	% U[2] := I-O TIME	00061200
11	% U[3] := ELAPSED TIME	00061300
12	% U[4] := CANDE CPU TIME	00061400
13	% U[5] := TERMINAL TIME	00061500
14	% U[6] := CORE	00061600
15	% U[7] := DISK SPACE	00061700
16	%*****	00061710
17	PROCEDURE REPORTU(USER,PST,IOT,ET,CET,TT,COR);%	00061800
18	VALUE PST,IOT,ET,CET,TT,COR,USER;%	00061900
19	REAL PST,IOT,ET,CET,TT,COR,USER;%	00062000
20	COMMENT OUTPUTS THE REPORT WITH USER TOALS AS	00062100
21	THE DETAIL;	00062200
22	%*****	00062210
23	BEGIN INTEGER I,J;%	00062300
24	LABEL EOF,XIT;%	00062400
25	DEFINE INPUTREADY=EQUIV(M[0],0) AND NOT MERGEND#;%	00062500
26	DEFINE M=MERGEINPUT#;%	00062600
27	IF ULINECTR GTR 68 THEN PUTHEADER(USERPT);%	00062700
28	MAKETAILLINE(ULINE,USER,"CURRENT","TOTALS ","*")	00062800
29	TIMER(PST),TIMER(IOT),TIMER(ET),TIMER(CET),	00062900
30	TIMER(TT),DECV(8,I:=(CUR/1024)*(PST+IOT)), " ");%	00063000
31	PRINTLINE(ULINE,USERPT);%	00063100
32	BLANKOUT(ULINE);%	00063200
33	IF MERGETOG THEN IF INPUTREADY THEN	00063300
34	DO BEGIN READ(MERGEFILE,8,M[*]) (EOF); WRITE(SUMFILE,8,M[*]) END	00063400
35	UNTIL COMPARE(1,USER,M[0]);%	00063500
36	IF MERGETOG THEN	00063600
37	IF EQUIV(USER,M[0]) THEN	00063700
38	BEGIN COMMENT HIT;	00063800
39	MAKETAILLINE(ULINE,USER,"MERGE ","TOTALS ","*")	00063900
40	TIMER(M[1]),TIMER(M[2]),TIMER(M[3]),TIMER(M[4]),	00064000
41	TIMER(M[5]),DECV(8,J:=(M[6]/1024)*(M[1]+M[2])), " ");%	00064100
42	MAKETAILLINE(ULINE,USER,"GRAND ","TOTALS ","*")	00064200
43	TIMER(PST:=PST+M[1]),TIMER(IOT:=IOT+M[2]),	00064300
44	TIMER(ET:=ET+M[3]),TIMER(CET:=CET+M[4]),	00064400
45	TIMER(TT:=TT+M[5]),DECV(8,I:=I+J), " ");%	00064500
46	M[0]:=0;%	00064600
47	END;	00064700
48	WRITE(UPRNT[DBL]);%	00064800
49	IF MERGETOG THEN IF COMPARE(1,M[0],USER) AND M[0] NEQ 0	00064900
50	THEN M[0]:=0;%	00065000
51	IF SUMTOG THEN WRITE(SUMFILE,**,USER,PST,IOT,ET,	00065100
52	CET,TT,I);%	00065200
53	GO TO XIT;	00065300
54	EOF:MERGEND:=TRUE;%	00065400
55	XIT;	00065500
56	END USERPT;%	00065600
57	%*****	00065700

```

1  BOOLEAN PROCEDURE MSIN(SORTREC);%                               00065800
2  ARRAY SORTREC(0);%                                             00065900
3  %*****                                                         00066000
4  BEGIN                                                           00066100
5  %.....                                                         00066200
6  PROCEDURE JOBN(U,P1,P2,LL,M,T);%                               00066300
7  VALUE U,P1,P2,LL,M,T;%                                         00066400
8  REAL U,P1,P2,LL,M,T;%                                          00066500
9  COMMENT USES PASSED PARAMETERS TO BUILD MIX TABLE ENTRIES;  00066600
10 %.....                                                         00066700
11 BEGIN DEFINE MT=MIXTBL#;%                                       00066800
12 MT[M,0]:=U; IF LL=0 THEN MT[M,1]:=0 ELSE                       00066900
13 MT[M,1]:=DECV(2,LL) & DECV(5,LLTBL[LL,2]) [6:18:30];%       00067000
14 MT[M,2]:=P1; MT[M,3]:=P2;%                                       00067100
15 MT[M,4]:=T;%                                                     00067200
16 IF LL NEQ 0 THEN                                               00067300
17 LLTBL[LL,3]:=U;% PUT IN USER CODE                               00067400
18 END JOBN;%                                                       00067500
19 %.....                                                         00067600
20 PROCEDURE JOBOFF(S,LL,MIX,TIM,HEND);%                           00067700
21 VALUE LL,MIX,TIM,HEND;%                                         00067800
22 REAL LL,MIX,TIM,HEND;%                                          00067900
23 ARRAY S(0);%                                                    00068000
24 COMMENT CREATES SORT RECORD FROM PASSED INFO AND             00068100
25 INFO IN MIXTABLE;%                                             00068200
26 %.....                                                         00068300
27 BEGIN                                                           00068400
28 DEFINE M=MIXTBL#;%                                             00068500
29 MOVE(9,M[MIX,*],S[*]);%                                         00068600
30 S[5]:=TIM;% EQU TIME                                           00068700
31 S[9]:=HEND;% HOW JOB ENDED ,MAY BE PSEUDO                     00068800
32 CLEAR(9,M[MIX,*]);% CLEAR MIX TABLE ENTRY                    00068900
33 END JOBOFF;%                                                    00069000
34 %.....                                                         00069100
35 PROCEDURE LOGOFF(S,LL,OFFT,CET,BOO,HEND);%                     00069200
36 VALUE LL,OFFT,CET,BOO,HEND;%                                    00069300
37 REAL LL,OFFT,CET,HEND;%                                         00069400
38 BOOLEAN BOO;%                                                  00069500
39 ARRAY S(0);%                                                    00069600
40 %.....                                                         00069700
41 BEGIN                                                           00069800
42 REAL I;%                                                         00069900
43 LABEL XIT;%                                                     00070000
44 S[1]:=DECV(2,LL) & DECV(5,LLTBL[LL,2]) [6:18:30];%           00070100
45 S[0]:=LLTBL[LL,0]; S[2]:=" LINE &";%                           00070200
46 S[3]:="C&E USE"; S[4]:=LLTBL[LL,1]; S[5]:=OFFT; S[6]:=CET;%   00070300
47 S[7]:=0; S[8]:=" "; S[9]:=HEND;%                                00070400
48 FOR I:=0,1 DO LLTBL[LL,I]:=0;%                                   00070500
49 LLTBL[LL,2]:=LLTBL[LL,2]+1;% BUMP SESSION NO.                 00070600
50 IF BOO THEN GO XIT;%                                           00070700
51 LINEARRY[NL]:=LINEARRY[NL]+OFFT;%                               00070800
52 LINEARRY[NL:=NL+1]:=LINEARRY[NL]-OFFT;%                         00070900
53 XIT;                                                             00071000
54 END LOGOFF;%                                                    00071100
55 %.....                                                         00071200
56 PROCEDURE LOGON(LL,ONT,WHO);%                                   00071300
57 VALUE LL,ONT,WHO;%                                             00071400
58 REAL LL,ONT,WHO;%                                              00071500
59 %.....                                                         00071600
60 BEGIN LLTBL[LL,0]:=WHO; LLTBL[LL,1]:=ONT;%                     00071700

```

Data Documents/Inc.

```

LLTBL[LL,3]:=0; % SO WE KNOW HE LOGGED IN
LINEARRY[NL]:=LINEARRY[NL]+UNT;%
LINEARRY[NL:=NL+1]:=LINEARRY[NL]-UNT;%
END LOGON;%
DEFINE K=GT1#,S=SORTREC#;%
DEFINE W=LOGREC#;%
DEFINE I=GT2#,M=MIXTBL#;%
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;%
REAL TIM,MIX;%
BOOLEAN B;%
COMMENT * * * * * OSIN STARTS HERE * * * * * ;
IF CLEANER THEN GO CLEANSWK1;%
RD: IF DISKREAD THEN
BEGIN CLOSE(DISK);
CLEANER:=TRUE;%
GO TO MRCLEAN;%
END;
TIM:=L-LOGREC[0],TIMEF;%
MIX:=L.MIXF; LL:=IF L.RLF=0 THEN 0 ELSE L.LINEF;%
GO TO SWL[L,TYPEF];%
BOJR:COMMENT BOJ RECCRD;
IF LL=0 THEN IF NOT ONSITETOG THEN GO RD ELSE
ELSE IF NOT TERMINALTOG THEN GO RD;%
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);%
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(M[MIX,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"
ELSE LLTBL[LL,0] ELSE USERID:="UNKNOWN";%
DO UNTIL SCAN="/";PN1:=ACCUM; SCAN; PN2:=ACCUM;%
JOBON(USERID,PN1,PN2,LL,MIX,BEGINTIME);%
END SALVEAGE;%
DO UNTIL SCAN=",";%
DO SCAN UNTIL L:=ACCUM.[6:12]="EU" OR
L="DS" OR L="SY";%
JOBOFF(S,LL,MIX,TIM,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,HLQ[HL],0,FALSE,"ES-ON ");%

```

```

00071800
00071900
00072000
00072100
00072200
00072300
00072400
00072500
00072600
00072700
00072800
00072900
00073000
00073100
00073200
00073300
00073400
00073500
00073600
00073700
00073800
00073900
00074000
00074100
00074200
00074300
00074400
00074500
00074600
00074700
00074800
00074900
00075000
00075100
00075200
00075300
00075400
00075500
00075600
00075700
00075800
00075900
00076000
00076100
00076200
00076300
00076400
00076500
00076600
00076700
00076800
00076900
00077000
00077100
00077200
00077300
00077310
00077400
00077500
00077600

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

Data Documents/Inc.

```

LOGON(LL,TIM,ACCUM);%
GO TO IF TERMINALTOG THEN XIT ELSE RD;%
END PSEUDO LOGOFF;
LOGON(LL,TIM,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,BEGINTIME,ACCUM);%
LOGOFF(S,LL,TIM,0,TRUE,"");%
GO TO IF TERMINALTOG THEN XIT ELSE RD;%
HL: HLQ[HLP:=HLP+1]:=TIM;%
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 ONSITETOG 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
MOVE(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;%
MRCLEAN:
COMMENT GENERATE PSEUDO LOG-OFFS FOR THOSE USERS STILL ON;
IF NOT(TERMINALTOG OR LINETOG) THEN GO MRSCLEAN;%
K:=2; LINEARRY[NL]:=LINEARRY[NL]+TIM; I:=0;
MRAGN:
IF I:=I+1 GTR NUMLINES THEN GO MRSCLEAN;%
IF LLTBL[I,0] NEQ 0 THEN
BEGIN LOGOFF(S,I,LASTIME,0,TRUE,"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,0,TRUE,"ESTIME ");
COMLINES:=COMLINES+1;%
GO TO XXT;%
END ELSE GO MRAGN;%
END LOGON CHECK;%
MRSCLEAN:
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 M[I,0] NEQ 0 THEN
BEGIN JOBOFF(S,LL,I,IF B=( M[I,4] LSS HLQ[HLP]) THEN HLQ[HLP]
ELSE LASTIME,IF B THEN "HL-ED " ELSE "INMIX ");%
GO TO XXT;%
END PSEUDO EOJ ELSE GO MRSAGN;%
DONE:
K:=I:=0;%
OS/NI=TRUE;%

```

```

00077700
00077800
00077900
00078000
00078100
00078200
00078300
00078400
00078500
00078600
00078700
00078800
00078900
00079000
00079100
00079200
00079300
00079400
00079500
00079600
00079700
00079800
00079900
00080000
00080100
00080200
00080300
00080400
00080500
00080600
00080700
00080800
00080900
00081000
00081100
00081200
00081300
00081400
00081500
00081600
00081700
00081800
00081900
00081910
00082000
00082100
00082200
00082300
00082400
00082500
00082600
00082700
00082800
00082900
00083000
00083100
00083200
00083300
00083400
00083500

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

```

GO TO XXT;%                                00083600
XIT:LASTIME:=TIM;%                          00083700
XXT;%                                        00083800
END OSIN;%                                  00083900
%*****                                  00084000
PROCEDURE OSOUT(NODATA, SORTREC);          00084100
  VALUE NODATA;%                            00084200
  BOOLEAN NODATA; ARRAY SORTREC(0);%      00084300
%*****                                  00084400
BEGIN                                       00084500
  DEFINE TPST=GT2#, % ACCUM REPORT CPU TIME 00084600
    TIOT=GT3#, % ACCUM REPORT I=O TIME      00084700
    TET =GT4#, % ACCUM REPORT ELAPSED TIME 00084800
    TCOR=GT5#, % ACCUM REPORT CORE(WDS)     00084900
    UPST=GT6#, % ACCUM USER CPU TIME        00085000
    UIOT=GT7#, % ACCUM USER I=O TIME        00085100
    UET =GT8#, % ACCUM USER ELAPSED TIME    00085200
    UCOR=GT9#, % ACCUM USER CORE(WDS)       00085300
    SPST=GT10#, % ACCUM SESSION CPU TIME    00085400
    SIOT=GT11#, % ACCUM SESSION I=O TIME    00085500
    SET =GT12#, % ACCUM SESSION ELAPSED TIME 00085600
    SCOR=GT13#, % ACCUM SESSION CORE(WDS)   00085700
    PPST=GT14#, % ACCUM PROGRAM CPU TIME    00085800
    PIOT=GT15#, % ACCUM PROGRAM I=O TIME    00085900
    PET =GT16#, % ACCUM PROGRAM ELAPSED TIME 00086000
    PCOR=GT17#, % ACCUM PROGRAM CORE(WDS)   00086100
    NU  =GT18#, % NO. OF USERS               00086200
    NR  =GT1#, % REC NO.                    00086300
    NJ  =GT19# % NO. OF JOBS PROCESSED      00086400
;%                                           00086500
  DEFINE S=SORTREC#;%                       00086600
  DEFINE H=HOLDKEY#;%                       00086700
  INTEGER PST, IOT, ET;%                   00086800
  INTEGER COR;%                             00086900
  LABEL DET, FILEDET, JHRU;%              00087000
  IF NOT NODATA THEN%                      00087100
  BEGIN COMMENT DATA CASE;%              00087200
    IF (NR:=NR+1)=1 THEN GO TO DET;%       00087300
    IF NOT EQUAL(4, S(0), HOLDKEY) THEN    00087400
    BEGIN COMMENT SUMMARIZE LAST PROG DATA; 00087500
      LEVEL(PROGRAM, PPST, PIOT, PET, PCOR);% 00087600
      SPST:=SPST+PPST; SIOT:=SIOT+PIOT;      00087700
      SET:=SET+PET; SCOR:=SCOR+PCOR;         00087800
      CLEAR(3, PPST);%                      00087900
      WRITE(PRNT(0BL)); LINECTR:=LINECTR+2;% 00088000
    END LAST PROG ELSE GO DET;%            00088100
    IF NOT EQUAL(2, S(0), HOLDKEY) THEN    00088200
    BEGIN COMMENT SUMMARIZE LINE DATA;%    00088300
      LEVEL(SESSION, SPST, SIOT, SET, SCOR);% 00088400
      UPST:=UPST+SPST; UIOT:=UIOT+SIOT;%    00088500
      UCOR:=UCOR+SCOR; UET:=UET+SET;%      00088600
      SPLIT(IF H(1), (36:12)=0 THEN 0 ELSE 1);% 00088700
      CLEAR(3, SPST);%                     00088800
      WRITE(PRNT(0BL)); LINECTR:=LINECTR+2;% 00088900
    END LINE SUMMARY ELSE GO DET;%         00089000
    IF NOT EQUIV(H(0), S(0)) THEN          00089100
    BEGIN COMMENT SUMMARIZE USER DATA;    00089200
      LEVEL(USER, UPST, UIOT, UET, UCOR);%  00089300
      REPORTU(HOLDKEY(0), UPST, UIOT,      00089400
        UET, UCT, UTT, UCOR); %           00089500

```

Data Documents/Inc.


```

TPST:=TPST+UPST;TIOT:=TIOT+UIOT; 00089600
TET:=TET+UET;TCOR:=TCOR+UCOR; 00089700
RTI := RTI+UTI; RCT := RCT+UCT;% 00089710
1 UTT := UCT := 0;% 00089720
2 CLEAR(3,UPST); 00089800
3 OSUSERS:=OSUSERS+OSUSE; OSUSE:=0;% 00089810
4 REMUSERS:=REMUSERS+REMUSE; REMUSE:=0;% 00089820
5 NU:=NU+1;% 00089900
6 PUTHEADER(ONSITE);% 00090000
7 END USER SUMMARY;% 00090100
8 DET: COMMENT PREPROCESS DETAIL RECORD;% 00090200
9 IF LINECTR GTR 68 THEN PUTHEADER(ONSITE);% 00090300
10 IF SIGN(S[9])=-1 THEN GO FILEDET;% 00090400
11 IF S[1],[36:12]=0 THEN OSJOBS:=OSJOBS+1 00090500
12 ELSE TUJOBS:=TUJOBS+1;% 00090600
13 NJ:=NJ+1;% 00090700
14 DETAIL(S[0],S[2],S[3],S[1],S[4],S[5],PST:=S[6], 00090800
15 IOT:=S[7],ET:=S[5]-S[4],COR:=S[8],S[9]); 00090900
16 COMMENT ACCUM PROG TOTALS;% 00091000
17 PPST := PPST+PST; PIOT := PIOT+IOT;% 00091100
18 IF S[3] = "C&E USE" THEN PET := PET+ET ELSE UTT := UTT+ET;% 00091110
19 PCOR:=PCOR+COR;% 00091200
20 MOVE(5,S[0],HOLDKEY);% 00091300
21 GO THRU;% 00091400
22 FILEDET: 00091500
23 FILEDETAIL(S);% 00091600
24 END DATA CASE ELSE 00091700
25 BEGIN COMMENT NO DATA CASE;% 00091800
26 IF SUMTOG THEN LOCK(SUMFILE);% 00091900
27 LEVEL(PROGRAM,PPST,PIOT,PET,PCOR);% 00092000
28 WRITE(PRINT[DBL]);% 00092100
29 SPST:=SPST+PPST; SIOT:=SIOT+PICT; 00092200
30 SET:=SET+PET; SCOR:=SCOR+PCUR;% 00092300
31 LEVEL(SESSION,SPST,SIOT,SET,SCOR);% 00092400
32 WRITE(PRINT[DBL]);% 00092500
33 IF S[3] NEQ "C&E USE" THEN 00092510
34 SPLIT(IF H[1],[36:12]=0 THEN 0 ELSE 1);% 00092600
35 UPST:=UPST+SPST; UIOT:=UIOT+SIOT; 00092700
36 UET:=UET+SET; UCOR:=UCOR+SCOR;% 00092800
37 LEVEL(USER,UPST,UIOT,UET,UCOR);% 00092900
38 REPORTU(HOLDKEY[0],UPST,UIOT,UET, 00093000
39 UCT,UTT,UCOR);% 00093100
40 WRITE(PRINT[DBL]);% 00093200
41 TPST:=TPST+UPST; TIOT:=TIOT+UIOT; 00093300
42 TET:=TET+UET; TCOR:=TCOR+UCOR; 00093400
43 NU:=NU+1;% 00093500
44 LEVEL(REPORT,TPST,TIOT,TET,TCOR); 00093600
45 CLEAR(7,UPST);% 00093700
46 END NODATA CASE; 00093800
47 THRU: 00093900
48 END OSOUT;% 00094000
49 ***** 00094100
50 BOOLEAN PROCEDURE OSCOMP(A,B);% 00094200
51 ARRAY A,B[0]; 00094300
52 ***** 00094400
53 OSCOMP:=IF EQUAL(4,A[0],B[0]) THEN A[4] LEQ B[4] 00094500
54 ELSE COMPARE(4,A[0],B[0]);% 00094600
55 ***** 00094700
56 PROCEDURE OSVAL(A); 00094800
57 ARRAY A[0];% 00094900

```

```

%*****
FILL A[*] WITH OCT1414141414141414,OCT1414141414141414,
OCT1414141414141414,OCT1414141414141414,
OCT1414141414141414;%
%*****
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;%
  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 * * * * *");%
  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);%
  STIMER(PST/NJ,P1,P2); STIMER(IOT/NJ,I1,I2);%
  STIMER(ET/NU,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:=PSTTOTALS[0];%
    IOT:=IOTTOTALS[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:=PSTTOTALS[1];%
    IOT:=IOTTOTALS[1];%

```

```

00095000
00095100
00095200
00095300
00095400
00095500
00095600
00095700
00095800
00095900
00096000
00096100
00096200
00096300
00096400
00096500
00096600
00096700
00096800
00096900
00097000
00097100
00097200
00097300
00097400
00097500
00097600
00097700
00097800
00097900
00098000
00098100
00098200
00098300
00098400
00098500
00098600
00098700
00098800
00098900
00099000
00099100
00099200
00099300
00099400
00099500
00099600
00099700
00099800
00099900
00099910
00100000
00100100
00100200
00100300
00100400
00100500
00100600
00100700
00100800

```

1	ET := ETOTALS[1];%	00100900	
2	NJ:=TUJORS;%	00101000	
3	NU:=REMUSERS;%	00101010	
4	GO AGAIN;%	00101100	
5	END REMOTE TOTALS;%	00101200	
6	END OSFINAL;%	00101300	
7	*****	00101400	
8	COMMENT ::::: SECTION - 06 - DISK LOGGING :::::;%	00101500	
9	*****	00101600	
10	COMMENT LAYOUT OF THE DISK LOGGING SORT RECORD;	00101700	
11	% S[0] := USER ID	00101800	
12	% S[1] := MFID	00101900	
13	% S[2] := FID	00102000	
14	% S[3] := CREATION DATE	00102100	
15	% S[4] := CREATION TIME	00102200	
16	% S[5] := LOG-OFF TIME	00102300	
17	% S[6] := NO OF DISK SEGS	00102400	
18	*****	00102500	
19	BOOLEAN PROCEDURE DLIN(S); ARRAY S[0];	00102600	
20	COMMENT FORMATS THE DISK SORT RECORD;	00102700	
21	*****	00102800	
22	BEGIN	00102900	
23	DEFINE L=LOGREC#;%	00103000	
24	REAL T;%	00103100	
25	LABEL GET,XIT;%	00103200	
26	GET:	00103300	
27	IF DISKREAD THEN BEGIN DLIN:=TRUE; GO XIT END;	00103400	
28	IF LOGREC[0].TYPEF NEQ DLOG THEN GO GET;%	00103500	
29	DO UNTIL SCAN="/";%	00103600	
30	S[1]:=ACCUM; % MFID	00103700	
31	DO UNTIL SCAN="/";%	00103800	
32	S[2]:=ACCUM; % FID	00103900	
33	SCAN; S[0]:=ACCUM; % USER ID	00104000	
34	S[5]:=L[0].TIMEF; % LOG-OFF TIME	00104100	
35	SCAN;S[6]:=NUMBERSCAN;	00104200	
36	DO UNTIL SCAN="/";	00104300	
37	S[3]:=IF ACCUM.[12:6]=" " THEN (O&ACCUM[12:6:6])	00104400	
38	ELSE ACCUM;	00104500	
39	SCAN2;	00104600	
40	S[3]:=IF ACCUM.[12:6]=" " THEN(S[3]&ACCUM[24:6:6])	00104700	
41	ELSE (S[3]&ACCUM[18:6:12]);%	00104800	
42	SCAN; S[3]:=S[3] & ACCUM[30:6:12];% AND YEAR	00104900	
43	T:=216000*NUMBERSCAN;% HOURS	00105000	
44	T:=T+3600*NUMBERSCAN;% MINUTES	00105100	
45	T:=T+ 60*NUMBERSCAN;% SECONDS	00105200	
46	T:=T+ NUMBERSCAN;% 1/60 SECONDS	00105300	
47	XIT:	00105400	
48	END DLIN;	00105500	
49	*****	00105600	
50	PROCEDURE OVAL(A); ARRAY A[0];	00105700	
51	*****	00105800	
52	FILL A[*] WITH OCT14141414141414,OCT14141414141414,	00105900	
53	OCT14141414141414;%	00106000	
54	*****	00106100	
55	BOOLEAN PROCEDURE DCOMP(A,B); ARRAY A,B[0];	00106200	
56	*****	00106300	
57	DCOMP:=COMPARE(3,A[0],B[0]);%	00106400	
58	*****	00106500	
59	STREAM PROCEDURE MAKEDISKLEVEL(LINE,ASTERISK,LEVEL,	00106600	
60	SEGS,MTHS,DAYS,TIM);%	00106700	

Data Documents/Inc.

	VALUE ASTERISK,LEVEL,SEGS,MTHS,DAYS,TIM;%	00106800	
	%*****	00106900	
	BEGIN	00107000	
1	LOCAL DA;%	00107100	1
2	DI:=LINE; DI:=DI+33;%	00107200	2
3	SI:=LOC ASTERISK;%	00107300	3
4	2(UPSI;UPDI; DS:=7 CHR);%	00107400	4
5	DS:=12 LIT " TOTALS,....";%	00107500	5
6	UPDI; DA:=DI; DS:=8 CHR;% NO SEGS	00107600	6
7	DI:=DI+3;%	00107700	7
8	SI:=SI+6; DS:=2 CHR; DS:=LIT" ";% MTHS	00107800	8
9	SI:=SI+6; DS:=2 CHR; DS:=LIT"-";% DAYS	00107900	9
10	SI:=SI+2; 2(DS:=2CHR; DS:=LIT""); DS:=2 CHR;% TIME	00108000	10
11	DI:=DA; DS:=6 FILL;%	00108100	11
12	END MAKEDISKLEVEL;%	00108200	12
13	%*****	00108300	13
14	PROCEDURE DISKLEVEL(WHICH,SEGS,MTHS,DAYS,TIM);%	00108400	14
15	VALUE WHICH,SEGS,MTHS,DAYS,TIM;%	00108500	15
16	REAL WHICH,SEGS,MTHS,DAYS,TIM;%	00108600	16
17	%*****	00108700	17
18	BEGIN	00108800	18
19	INTEGER T;	00108900	19
20	BLANKOUT(LINE);%	00109000	20
21	MAKEDISKLEVEL(LINE,ASTERISK[WHICH=2],LEVELARRY[WHICH],	00109100	21
22	DECV(8,T:=SEGS),DECV(2,T:=MTHS),DECV(2,T:=DAYS),	00109200	22
23	TIMER(TIM));%	00109300	23
24	PRINTLINE(LINE,DISKRPT);%	00109400	24
25	END DISKLEVEL;%	00109500	25
26	%*****	00109600	26
27	STREAM PROCEDURE MAKEDISKDETAIL(LINE,A,CTIM,LTIM,SEGS,	00109700	27
28	M,D,T);%	00109800	28
29	VALUE CTIM,LTIM,SEGS,M,D,T;%	00109900	29
30	%*****	00110000	30
31	BEGIN	00110100	31
32	LOCAL DA;%	00110200	32
33	DI:=LINE; SI:=A;%	00110300	33
34	3(UPSI; DS:=7 CHR; DI:=DI+2);%	00110400	34
35	UPSI; 2(DS:=2 CHR; DS:=LIT "/"); DS:=2 CHR;	00110500	35
36	SI:=LOC CTIM;	00110600	36
37	2(SI:=SI+2; DI:=DI+3; 2(DS:=2 CHR; DS:=LIT" "); DS:=2 CHR);%	00110700	37
38	DI:=DI+5; DA:=DI; DS:=8 CHR; DI:=DI+3;%	00110800	38
39	SI:=SI+6; DS:=2 CHR; DS:=LIT" ";	00110900	39
40	SI:=SI+6; DS:=2CHR;	00111000	40
41	DS:=LIT" ";%	00111100	41
42	SI:=SI+2; 2(DS:=2 CHR; DS:=LIT" "); DS:=2 CHR;%	00111200	42
43	DI:=DA; DS:=6 FILL;%	00111300	43
44	END MAKE DISKDETAIL;%	00111400	44
45	%*****	00111500	45
46	PROCEDURE DISKDETAIL(SORTREC,MTHS,DAYS,TIM);%	00111600	46
47	VALUE MTHS,DAYS,TIM;	00111700	47
48	REAL TIM; INTEGER MTHS,DAYS;%	00111800	48
49	ARRAY SORTREC[0];%	00111900	49
50	%*****	00112000	50
51	BEGIN DEFINE A#SORTREC#;	00112100	51
52	INTEGER I;%	00112200	52
53	BLANKOUT(LINE);%	00112300	53
54	MAKEDISKDETAIL(LINE,A,TIMER(A[4]),TIMER(A[5]),	00112400	54
55	DECV(8,I:=A[6]),DECV(2,MTHS),DECV(2,DAYS),	00112500	55
56	TIMER(TIM));%	00112600	56
57	PRINTLINE(LINE,DISKRPT);%	00112700	57

```

END DISKDETAIL;%                                00112800
%*****                                        00112900
PROCEDURE DLOUT(B,A);%                          00113000
  VALUE B;%                                     00113100
  BOOLEAN B; % TRUE IF LAST REC PROCESSEE      00113200
  ARRAY A[0]; % SORTED DISK LOG REGRD          00113300
%*****                                        00113400
BEGIN LABEL DET,THRU;                           00113500
  LABEL BRK;%                                   00113600
  DEFINE RPTMO = GT6#,% ACCUM REPORT MONTHS     00113700
         RPTDA = GT7#,% ACCUM REPORT DAYS       00113800
         RPTDT = GT8#,% ACCUM REPORT TIME       00113900
         RPTDS = GT9#,% ACCUM REPORT DISK SEGS  00114000
         USERMO =GT10#,% ACCUM USER MONTHS      00114100
         USERDA =GT11#,% ACCUM USER DAYS        00114200
         USERDT =GT12#,% ACCUM USER TIME        00114300
         USERDS =GT13#,% ACCUM USER DISK SEGS   00114400
         NF =GT14#,% NO. OF FILES PROCESSED     00114500
         NU =GT15#;% NO OF USERS                00114600
  REAL CDATE,CTIME,LDATE,LTIME;%               00114700
  INTEGER MTHS,DAYS;                            00114800
  REAL Z,T;%                                     00114900
  IF(NF:=NF+1) = 1 THEN GO TO DET;%             00115000
  IF B THEN GO BRK;%                             00115100
  IF NOT EQUAL(1,A[0],HOLDKEY) THEN            00115200
  BEGIN %NEW USER OR END OF REPORT             00115300
  BRK;                                          00115400
  USERDA+USERDA+USERDT DIV 5184000;           00115500
  USERDT+USERDT MOD 5184000;                   00115600
  USERMO+USERMO+USERDA DIV 31;                 00115700
  USERDA+USERDA MOD 31;                        00115800
  DISKLEVEL(USER,USERDS,USERMO,USERDA,USERDT);% 00115900
  RPTMO:=RPTMO+USERMO; RPTDA:=RPTDA+USERDA;%   00116000
  RPTDT:=RPTDT+USERDT; RPTDS:=RPTDS+USERDS;%   00116100
  CLEAR(3,USERMO);%                             00116200
  IF NOT B THEN                                 00116300
  BEGIN NU:=NU+1;PUTHEADER(DISKRPT); GO DET END;% 00116400
  END USER TOTAL ELSE GO TO DET;%              00116500
  COMMENT END OF REPORT;                       00116600
  RPTDA:=RPTDA+RPTDT DIV 5184000;%             00116700
  RPTDT:=RPTDT MOD 5184000;%                   00116800
  RPTMO:=RPTMO+RPTDA DIV 31;%                  00116900
  RPTDA:=RPTDA MOD 31;%                         00117000
  DISKLEVEL(REPORT,RPTDS,RPTMO,RPTDA,RPTDT);% 00117100
  GO THRU;%                                     00117200
  DET:CDATE:=A[3];                              00117300
  LDATE:=MFID & CDATE [30:30:12];%            00117400
  IF LTIME:=A[5] LSS CTIME:=A[4] THEN Z:=1;%   00117500
  IF LINECTR GTR 68 THEN PUTHEADER(DISKRPT);%  00117600
  DISKDETAIL(A,                                  00117700
  MTHS:=OCV(LDATE,2)-OCV(CDATE,2),             00117800
  DAYS:=OCV(LDATE,4)-OCV(CDATE,4),             00117900
  T:=LTIME+Z*5184000-CTIME);%                 00118000
  USERMO:=USERMO+MTHS; USERDA:=USERDA+DAYS;    00118100
  USERDT:=USERDT+T; USERDS:=USERDS+A[6];%     00118200
  HOLDKEY[0]:=A[0];%                           00118300
  THRU;                                         00118400
  END DLOUT;                                    00118500
%*****                                        00118600
PROCEDURE DLFINAL;                              00118700

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

```

%*****
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;%
%*****
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" - MERGE INPUT
%   OPTN 8 - "SUM"  - OUTPUT SUMMARY RECORD ON DISK
%*****
PROCEDURE PROCESSPARAMS;%
%*****
BEGIN
  FORMAT F1("PARAMETERS"),
  F2("INV PARAMETER(S)...TRY AGAIN");
  INTEGER I,J;
  REAL P;%
  ARRAY A[1:MAXPARAMS];%
  LIST PARALIST(FBR 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;%
  WHILE P:=A[J:=J+1] NEQ 0 DO%
  BEGIN COMMENT PROCESS PARAMETER LIST;
    IF I:=MATCH(P,PARAMDS) = 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;
      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

```

```

00118800
00118900
00119000
00119100
00119200
00119300
00119400
00119500
00119600
00119700
00119800
00119900
00120000
00120100
00120200
00120300
00120400
00120500
00120600
00120700
00120800
00120900
00121000
00121100
00121200
00121300
00121400
00121500
00121600
00121700
00121800
00121900
00122000
00122100
00122200
00122300
00122400
00122500
00122600
00122700
00122800
00122900
00123000
00123100
00123200
00123300
00123400
00123500
00123600
00123700
00123800
00123900
00124000
00124100
00124200
00124300
00124400
00124500
00124600
00124610

```

Data Documents/Inc.

Data Documents/Inc.

```

WRITE(TWX[STOP],FT);%
READ(TWX,/,MERGEID);%
FILL MERGEFILE WITH MERGEID,"LOG";%
1 END GETMERGE;%
2 %*****%
3 REAL PROCEDURE FINDLOG;%
4 %*****%
5 BEGIN
6 REAL I,J;%
7 FORMAT FN("LOG MFID(S)...");%
8 LABEL AGAIN,INS;%
9 IF NOT COMBOO THEN
10 WRITE(TWX[STOP],FN);%
11 READ(TWX[NO]);%
12 PTR:=GETABSAADR(TWX(O));%
13 AGAIN:
14 WHILE SCAN=NUMBERV DO
15 BEGIN COMMENT MAINTAIN FILE QUEUE IN ASC ORDER;
16 IF FIDP=0 THEN GO TO INS;%
17 FOR I:=0 STEP 1 UNTIL FIDP-1
18 DO IF ACCUM LSS FIDQ[I] THEN
19 BEGIN COMMENT MOVE EM UP;
20 FOR J:=FIDP STEP -1 UNTIL I
21 DO FIDQ[J+1]:=FIDQ[J];
22 GO INS;%
23 END;
24 INS: FIDP:=FIDP+1; FIDQ[I]:=ACCUM;%
25 END FILE QUEUE MAINTENANCE;%
26 IF RESULT NEQ GPMK THEN GO TO AGAIN;
27 FINDLOG:=FIDQ[0];%
28 END FINDLOG;%
29 %*****%
30 PROCEDURE GETWORK;%
31 %*****%
32 BEGIN
33 LABEL AGAIN;%
34 REAL I,T;%
35 INTEGER N;%
36 %.....%
37 STREAM PROCEDURE MAKEFIDLINE(N,FIDS,TOG,MERG,ARRY);%
38 VALUE MERG,N,TOG;%
39 %.....%
40 BEGIN DI:=ARRY; SI:=FIDS;%
41 DS:=19 LIT"FILES: CURRENT...";
42 N(CUPSI)UPDI; DS:=7 CHR; DS:=2 LIT ", ";
43 TOG(DI:=ARRY; DI:=DI+63;%
44 DI:=DI+47; SI:=LOC MERG;%
45 DS:=8 LIT"MERGE..."; DS:=WDS));%
46 END MAKE FIDLINE;%
47 AGAIN:
48 IF MFID:=FINDLOG NEQ 0 THEN
49 BEGIN FILL DISK WITH MFID,"LOG ";
50 READ(DISK[NO],10,LOGREC[*]);%
51 BEGINTIME:=LOGREC[0],TIMEF;%
52 PROCESSPARAMS;%
53 IF MERGETOG THEN GETMERGE;%
54 NFIDLINES:=FIDP DIV 9 + 1;%
55 FOR I:=0 STEP 1 UNTIL NFIDLINES
56 DO BLANKOUT(FIDLINE[I,*]);%
57 FOR I:=0 STEP 1 UNTIL NFIDLINES-2

```

```

00124700
00124800
00124900
00125000
00125100
00125200
00125300
00125400
00125500
00125600
00125700
00125710
00125800
00125900
00126000
00126100
00126200
00126300
00126400
00126500
00126600
00126700
00126800
00126900
00127000
00127100
00127200
00127300
00127400
00127500
00127600
00127700
00127800
00127900
00128000
00128100
00128200
00128300
00128400
00128500
00128600
00128700
00128800
00128900
00129000
00129100
00129200
00129300
00129400
00129500
00129600
00129700
00129800
00129900
00130000
00130100
00130200
00130300
00130400
00130500

```

```

DO BEGIN
  MAKEFIDLINE(9,FIDQ[1],0,0,FIDLINE[1,*]);%
  T:=T+9;%
END;%
MAKEFIDLINE(N:=FIDP MOD 10,FIDQ[1],MERGETOG,MERGEID,
FIDLINE[NFIDLINES-1,*]);%
FIDP:=0;%
END ELSE GO AGAIN;%
END GETWORK;%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
COMMENT ::::: SECTION - 08 - INITIALIZE, OUTER BLOCK :::::;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
*****
PROCEDURE INITIALIZE;%
*****
BEGIN
  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";
  Y:=TIME(0);%
  D+Y.[30:6]*100+Y.[36:6]*10+Y.[42:6];
  Y:=Y.[18:6]*10+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+1; P:=Y-1 END ELSE
  BEGIN Q:=M-1; P:=Y END;%
  M:=M+1;%
  DAY:=A[((Q*26-2) DIV 10+D+P+P.[36:10]+1) MOD 7];%
  FILL UNIT[*] WITH "MTA","MTB","MTC","MTD",
"MTF","MTG","MTH","MTJ","MTK","MTL","MTM",
"MTP","MTR","MTS","MTT","DRA","DRB",
"DKA","DKB","LPA","LPB","CPA","CRA","CRB",
"SPO","PPA","PRA","PPB","PRB","UCA";%
  FILL ASTERISK[*] WITH "  *",
"  ***","  ****";%
  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";%
  BLANKOUT(ULINE);%
END INITIALIZE;%
*****
COMMENT OUTER BLOCK;
*****
INITIALIZE)
GETWORK;%
IF TERMINALTOG OR ONSITETOG OR SUMTOG THEN

```

```

00130600
00130700
00130800
00130900
00131000
00131100
00131200
00131300
00131400
00131500
00131600
00131700
00131800
00131900
00132000
00132100
00132200
00132300
00132400
00132500
00132600
00133000
00133100
00133200
00133300
00133400
00133500
00133600
00133700
00133800
00133900
00134000
00134100
00134200
00134300
00134400
00134500
00134600
00134700
00134800
00134900
00135000
00135100
00135200
00135300
00135400
00135500
00135600
00135700
00135800
00135900
00136000
00136100
00136200
00136300
00136400
00136500
00136600
00136700
00136800

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57


```

BEGIN COMMENT REMOTE AND/OR LOCAL REPORTS;%
PTRBASE:=GETABSADDR(LOGREC[1]);%
IF SUMTOG THEN PUTHEADER(USERPT);%
IF TERMINALTOG OR ONSITETOG THEN PUTHEADER(ONSITE);%
SORT(OSOUT,OSIN,0,OSVAL,OSCOMP,OSRECSI,4000);%
NEXTPAGE(ONSITE); OSFINAL;%
HLANA; HLTG:=FALSE;%
ERRANA; ERRTG:=FALSE;%
IF TERMINALTOG OR LINETOG THEN
  BEGIN LINEANA; LINETG:=FALSE END;%
END REMOTE LOCAL REPORTS;%
IF DISKTG 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;%
  IF HLTG OR LINETG OR ERRTG THEN %
  BEGIN CLEANER:=TRUE; CLEANER:=OSIN(HLQ); END;%
  IF HLTG THEN HLANA;%
  IF LINETG THEN LINEANA;%
  IF ERRTG THEN ERRANA;%
END. % OF PROGRAM
END;END. LAST CARD ON OCRDING TAPE
DEFINE MOREDPBITS[MOREDPBITS1]=UV[MOREDPBITS1,31];#
OLAYUSED[OLAYUSED1]=UV[OLAYUSED1,32];#

```

```

00136900
00137000
00137100
00137200
00137300
00137400
00137500
00137600
00137700
00137800
00137900
00138000
00138100
00138200
00138300
00138400
00138500
00138600
00138700
00138800
00138900
00139000
00139100
00139200
00139300
00139400
00139500
99999999
00470510
00470520

```

Data Documents Inc.

LABEL 00000000PRINTER00175098CC EXECUTE OBJECT/READ;FILE SOURCEFILE=SYMBOL/LOGAN;END+

OBJECT /READ

Data Documents/Inc.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57