Title:

RC3600 DATA ENTRY RELEASE 2
STRUCTURE OF ACCOUNT- AND LOGFILE.

A/S REGNECENTRALEN

RC SYSTEM LIBRARY: FALKONERALLE 1 DK-2000 COPENHAGEN F

RCSL No:	43-GL5985
Edition:	January 1978
Author:	Flemming Havmøller

Keywords:

MUS, DOMUS, DATA ENTRY, LOGFILE, LOGF,
ACCOUNTFILE, ACCO.

Abstract:

This manual descripes how the files ACCO (accountfile) and LOGF (logfile) are structured and some examples of how to make a Musilprogram to read these files.

Printed by A/S Regnecentralen, Copenhagen

Users of this manual are cautioned that the specifications contained herein are subject to change by RC at any time without prior notice. RC is not responsible for typographical or arithmetic errors which may appear in this manual and shall not be responsible for any damages caused by reliance on any of the materials presented.

CONTENTS PAGE

1. INTRODUCTION. 1.

This manual descripes how the files ACCO (Accountfile) and LOGF (Logfile) are structured.

Later on in this manual (Appendix A and B) there are some coding examples which shows how to read these files and produce statistics in different ways.

These 2 files, both produced by NANNY during process, has information about: login-, key- and setcommands, operators name, keystationnumber, time, date and the like.

The difference between the accountfile (ACCO) and the logfile (LOGF) is, that the accountfile only contains one recordtype OS (Operator Statistics) and the logfile contains the following recordtypes: OS (Operator Statistic), SC (Supervisor Command), SA (Supervisor Answer), LC (Login Command) and LA (Login Answer).

The files can be used either directly from disc or from magnetic tape by using the supervisor program DUMPSTAT, (see Data Entry Release 2 - Users Guide Part 2).

Please note that it is not allowed to write in accountfile or logfile, because NANNY are using some of the first words in the files, as index to write information.

2. ACCOUNTFILE ACCO.

2.1. Accountfile, Structure as Discfile.

The recordlength of these records is 52 bytes, and each sector in the discfile contains one or more of these records.

Maximum number of records in each sector is 9.

Each sector is starting with the recordtype OS (written with characters in asciicode) in byte number 1 and 2. (See figure 2-2).

The first sector in the discfile (sector 0) has in the first two words 2 counters (see figure 2-1).
The first counter placed in the first word, tells how many sectors used (later on called sector counter), and the second counter placed in the second word tells how many bytes used in the last sector (later on called bytecounter).

SEC: 1, 2, ...
BYTEC: 0, 1, 2...

Therefore, to calculate number of records to read, use the following formel:
Number of records: =
((Sectorcounter - 1)*NORECS)+((Bytecounter + 1)/RECLENGTH,
where NORECS is maximum number of records in a sector,
and RECLENGTH is the length of the record.

In Musilcode the following statements can be used:

```
OPEN (ACCO,1);
SETPOSITION (ACCO,0,0);
DISCLENGTH: = RECLENGTH;
GETREC (ACCO, DISCLENGTH);
RECORDS: = (WORD ACCO↑.SECTOR-1)*NORECS;
RECORDS: = RECORDS+((WORD ACCO↑.BYTES+1)/DISCLENGTH);
```

After that the integer RECORDS will tell how many records in the discfile are relevant.

Please note that the first sector containing records is the second one in the discfile.

For further information about programming in the Musil-language, please see appendix A and B in this manual and the RC3600 Musil Programming Guide.

Figure 2-1:

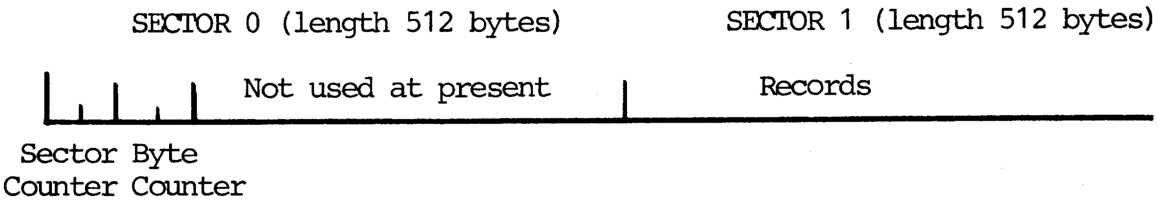


Figure 2-2. Recordformat of Records of Type OS.

FROM BYTE	TO BYTE	NO.OF BYTES	TYPE A/B/X	CONTAINS INFORMATION
1	2	2	A	Recordtype OS
3	10	8	A	Date YY.MM.DD
11	15	5	A	Time HH.MM
16	16	1	X	EMPTY 1
17	18	2	A	Terminal number
19	21	3	A	Operators identification from LOGIN
22	22	1	X	EMPTY 2
23	27	5	A	Jobname
28	28	1	X	EMPTY 3
29	33	5	A	Formatname
34	34	1	X	EMPTY 4
35	39	5	A	Batchname
40	40	1	X	EMPTY 5
41	42	2	B	Batchstatus
43	44	2	B	Keytime
45	46	2	B	Number of Keystrokes
47	48	2	B	Number of Records
49	50	2	B	Number of Corrections
51	52	2	B	Number of invalid Records

Explanation of type:

A = Characters in ASCII-code

B = Binary digits

X = Not used, Normally filled up with Null characters.

2.2 Accountfile, Structure as Magnetic Tape-file.

The discfile ACCO can be transferred to magnetic tape by use of the supervisor program DUMPSTAT.
(See Data Entry Release 2 - Users Guide, Part 2).

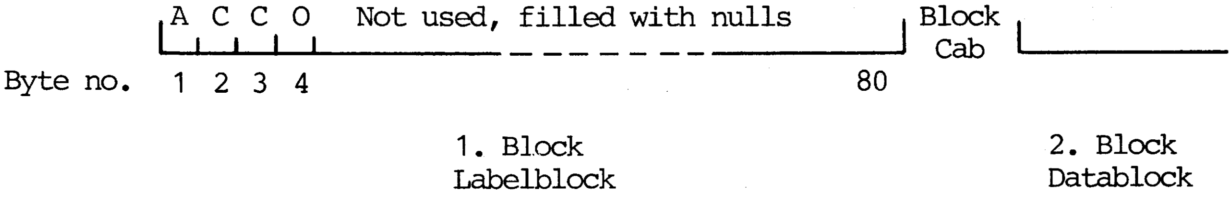
The first block on the tape is a labelblock with the blocklength of 80 bytes, this block has from byte 1 to 4 the ASCII-characters ACCO, and the remaining bytes in the block (from byte 5 to 80) is not used, and are filled up with null characters (see figure 2-3).

This labelblock is followed by, one or more datablocks, these datablocks have the length of 512 bytes, and each of them contains maximum 9 records of length 52 bytes. The information stored in these records are explained in section 2.1.

Please note that the last block on the tape can be a short block, minimum blocklength is 52 bytes.

After the last datablock, the magtape is terminated with two tapemarks.

Figure 2-3.



3. LOGFILE, LOGF. 3.

3.1. Logfile, Structure as Discfile. 3.1.

The recordlength of these records is 102 bytes, and each sector in this discfile contains one or more of these records.

Maximum number of records in each sector is 5.

Each of these records is starting with a recordtype in byte number 1 and 2, this recordtype is written in ASCII-code and there are 5 different types.

OS: Operator Statistic. (Recordformat see figure 3-2). Information about: Jobname, formatname, batchname, batchstatus, keytime, keystrokes, records, corrections and invalid records.

SC: Supervisor Command. (Recordformat see figure 3-3). Information about commands sent from operator to supervisor.

SA: Supervisor Answer. (Recordformat see figure 3-3). Information about the result of running of a supervisorprogram, and errormessages.

LC: Login Command. (Recordformat see figure 3-3). Information about: Login-, set-, key-, rekey- and editcommands.

LA: Login Answer. (Record format see figure 3-3). Information about the result of the login-, set-, key-, rekey- and editcommands.

Each of these recordtypes has, from byte 3 to 21, information about date, time, terminal and operatorname.

The first sector in the discfile (sector 0) has in the first two words 2 counters (see figure 3-1).
The first counter placed in the first word, tells how many sectors used (later on called sectorcounter), and the second counter placed in the second word tells how many bytes used in the last sector (later on called bytecounter).

Therefore, to calculate number of records to read, use the following formel:
Number of records: =
((Sectorcounter-1)*NORECS)+((Bytecounter+1)/RECLENGTH),
where NORECS is maximum number of record in a sector,
and RECLENGTH is the length of the record.

3.1. In Musilcode the following statements can be used:

```
OPEN (LOGF,1);
SETPOSITION (LOGF,0,0);
DISCLENGTH: = RECLENGTH;
GETREC (LOGF,DISCLENGTH);
RECORDS: = (WORD LOGF↑.SECTOR-1)*NORECS;
RECORDS: = RECORDS+((WORD LOGF↑.BYTES+1)/DISCLENGTH);
```

After that the integer RECORDS will tell how many records in the discfile are relevant.

Please note that the first sector containing records is the second one in the discfile.

For further information about programming in the Musil-language please see appendix A and B in this manual and the RC3600 Musil Programming Guide.

Figure 3-1:

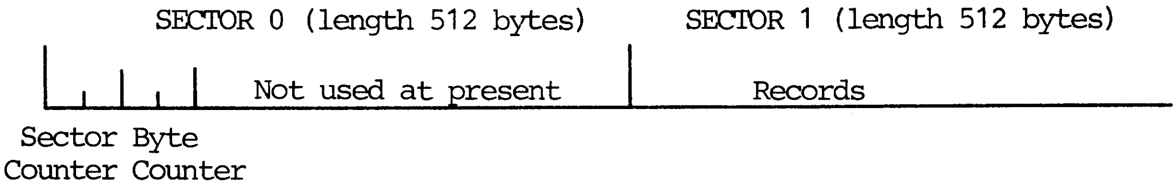


Figure 3-2. Recordformat of Records of Type OS. 3.1.

FROM BYTE	TO BYTE	NO.OF BYTES	TYPE A/B/X	CONTAINS INFORMATION
1	2	2	A	Recordtype = OS
3	10	8	A	Date YY.MM.DD
11	15	5	A	Time HH.MM
16	16	1	X	
17	18	2	A	Terminal number
19	21	3	A	Operator identifikation from LOGIN
22	22	1	X	
23	27	5	A	Jobname
28	28	1	X	
29	33	5	A	Formatname
34	34	1	X	
35	39	5	A	Batchname
40	40	1	X	
41	42	2	B	Batchstatus
43	44	2	B	Keytime
45	46	2	B	Number of Keystrokes
47	48	2	B	Number of records
49	50	2	B	Number of corrections
51	52	2	B	Number of invalid records
53	101	49	X	

Explanation of Type:

A: Characters in ASCII-code.

B: Binary Digits

X: Not used, normally filled up with Null characters.

This record is, except that it is longer, equal to the record described in section 2, figure 2-2.

3.1. Figure 3-3. Recordformat of Records of Type SC, SA, LC, LA.

FROM BYTE	TO BYTE	NO.OF BYTES	TYPE A/B/X	CONTAINS INFORMATION
1	2	2	A	Recordtype = SC,SA, LC or LA
3	10	8	A	Date YY.MM.DD
11	15	5	A	Time HH.MM
16	16	1	X	
17	18	2	A	Terminal number
19	21	3	A	Operator identifikation from LOGIN
22	22	1	X	
23	101	79	A	Text area, descriped below.

Explanation of type:

A: Characters in ASCII-code.
X: Not used, normally filled up with Null characters.

The text area from byte 23 to 101 contains all command typed in from the operator, errormessages, messages from supervisor programs.

The text is written in ASCII characters, and not used bytes are filled up with spaces (decimal value = 32), (see figure 3-4).

Figure 3-4:

	S	U	R	V	E	Y	▵	B	A	T	C	H	.	B	0	4	5	▵	▵	▵	------	▵	▵	▵	▵	▵
Byte no.	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42		97	98	99	100	101

This figure shows how the call of supervisor program SURVEY with parameters are stored in the record form byte 23 and on. The character identified with ▵ is the character space.

3.2 Logfile, Structure as Magnetic Tapefile. 3.2

The discfile LOGF can be transferred to magnetic tape by use of the supervisor program DUMPSTAT.
(See Data Entry Release 2 - Users Guide, Part 2).

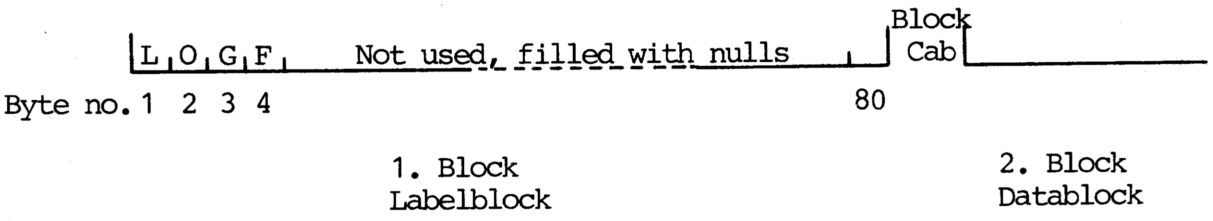
The first block on the tape is a labelblock with a blocklength of 80 bytes, this block has from byte 1 to 4 the ASCII-characters LOGF, and the remaining bytes in the block (from byte number 5 to 80) is not used and are filled up with null characters (see figure 3-5).

This labelblock is followed by one or more datablocks, these datablocks have the length of 512 bytes and each of them contains maximum 5 records of length 102 bytes.
The information stored in these records are explaned in section 3.1.

Please note that the last block on the tape can be a shortblock, minimum blocklength 102 bytes, equal to one record.

After the last datablock, the magtape is terminated with two tapemarks.

Figure 3-5:

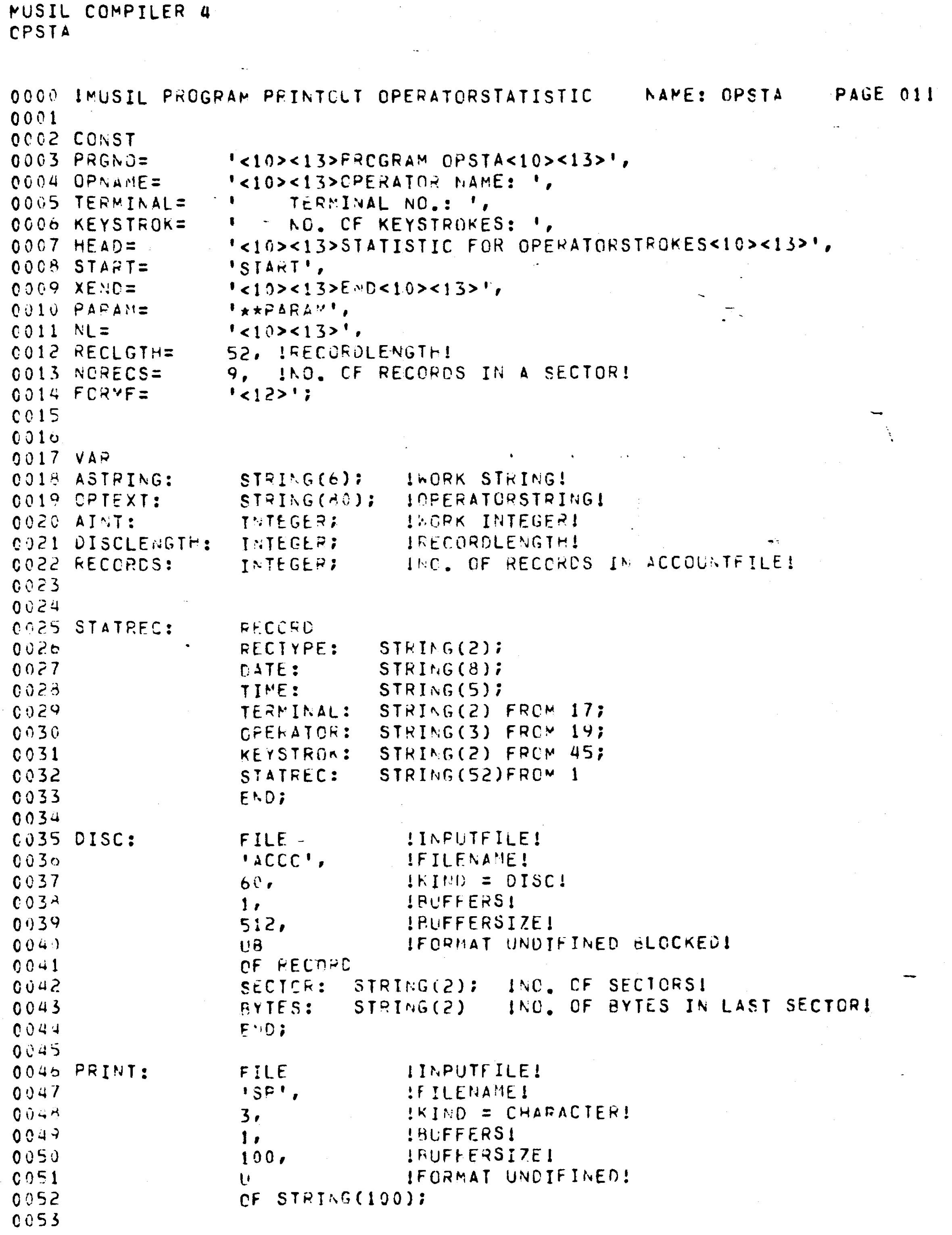


APPENDIX A - CODING EXAMPLE FOR READING ACCOUNTFILE.

The following example in Musil language shows how it is possible to read in the Accountfile and produce statistics.

In the shown program there is no procedures to take care of errors from disc or from the printer.

The program is able to read the accountfile (ACCO) and write out a statistic which shows the operatorname, terminalnumber and number of Keystrokes done by the operator.



```
MUSIL COMPILER 4
OPSTA

!MUSIL PROGRAM PRINTOUT OPERATORSTATISTIC    NAME: OPSTA    PAGE 01!

CONST
PRGNO=      '<10><13>PROGRAM OPSTA<10><13>',
OPNAME=     '<10><13>OPERATOR NAME: ',
TERMINAL=   '    TERMINAL NO.: ',
KEYSTROK=   '    NO. OF KEYSTROKES: ',
HEAD=       '<10><13>STATISTIC FOR OPERATORSTROKES<10><13>',
START=      'START',
XEND=       '<10><13>END<10><13>',
PARAM=      '**PARAM',
NL=         '<10><13>',
RECLGTH=    52, !RECORDLENGTH!
NORECS=     9,  !NO. OF RECORDS IN A SECTOR!
FORMF=      '<12>';


VAR
ASTRING:     STRING(6);   !WORK STRING!
OPTEXT:      STRING(80);  !OPERATORSTRING!
AINT:        INTEGER;     !WORK INTEGER!
DISCLENGTH:  INTEGER;     !RECORDLENGTH!
RECORDS:     INTEGER;     !NO. OF RECORDS IN ACCOUNTFILE!


STATREC:     RECORD
             RECTYPE:   STRING(2);
             DATE:      STRING(8);
             TIME:      STRING(5);
             TERMINAL:  STRING(2) FROM 17;
             OPERATOR:  STRING(3) FROM 19;
             KEYSTROK:  STRING(2) FROM 45;
             STATREC:   STRING(52)FROM 1
             END;

DISC:        FILE         !INPUTFILE!
             'ACCO',      !FILENAME!
             60,          !KIND = DISC!
             1,           !BUFFERS!
             512,         !BUFFERSIZE!
             UB           !FORMAT UNDIFINED BLOCKED!
             OF RECORD
             SECTOR:  STRING(2);  !NO. OF SECTORS!
             BYTES:   STRING(2)   !NO. OF BYTES IN LAST SECTOR!
             END;

PRINT:       FILE         !INPUTFILE!
             'SP',        !FILENAME!
             3,           !KIND = CHARACTER!
             1,           !BUFFERS!
             100,         !BUFFERSIZE!
             U            !FORMAT UNDIFINED!
             OF STRING(100);

```

```
!MUSIL PROGRAM PRINTOUT OPERATORSTATISTIC     NAME: OPSTA     PAGE 02!

          PROCEDURE DELCHAR;      !DELETE ILLEGAL CHARACTERS!
          BEGIN
          AINT:= 0;
          REPEAT MOVE(ASTRING,AINT,OPTEXT,0,1);  !MOVE ONE CHARACTER!
          IF BYTE OPTEXT < 32 THEN INSERT(32,ASTRING,AINT); !IF CHAR. ILLG!
          AINT:= AINT + 1 UNTIL AINT = 3;
          END;

          PROCEDURE PRINTSTAT;        !PRINT ONE LINE!
          BEGIN
          OUTTEXT(PRINT,OPNAME);      !OPERATORS NAME!
          ASTRING:= STATREC.OPERATOR;
          DELCHAR;    !CALL OF PROCEDURE DELCHAR!
          INSERT(0,ASTRING,3);
          OUTTEXT(PRINT,ASTRING);
          OUTTEXT(PRINT,TERMINAL);    !TERMINALNUMBER!
          ASTRING:= STATREC.TERMINAL;
          INSERT(0,ASTRING,2);
          OUTTEXT(PRINT,ASTRING);
          OUTTEXT(PRINT,KEYSTROK);    !KEYSTROCKES!
          BINDEC(WORD STATREC.KEYSTROK,ASTRING);
          OUTTEXT(PRINT,ASTRING);
          END;

```

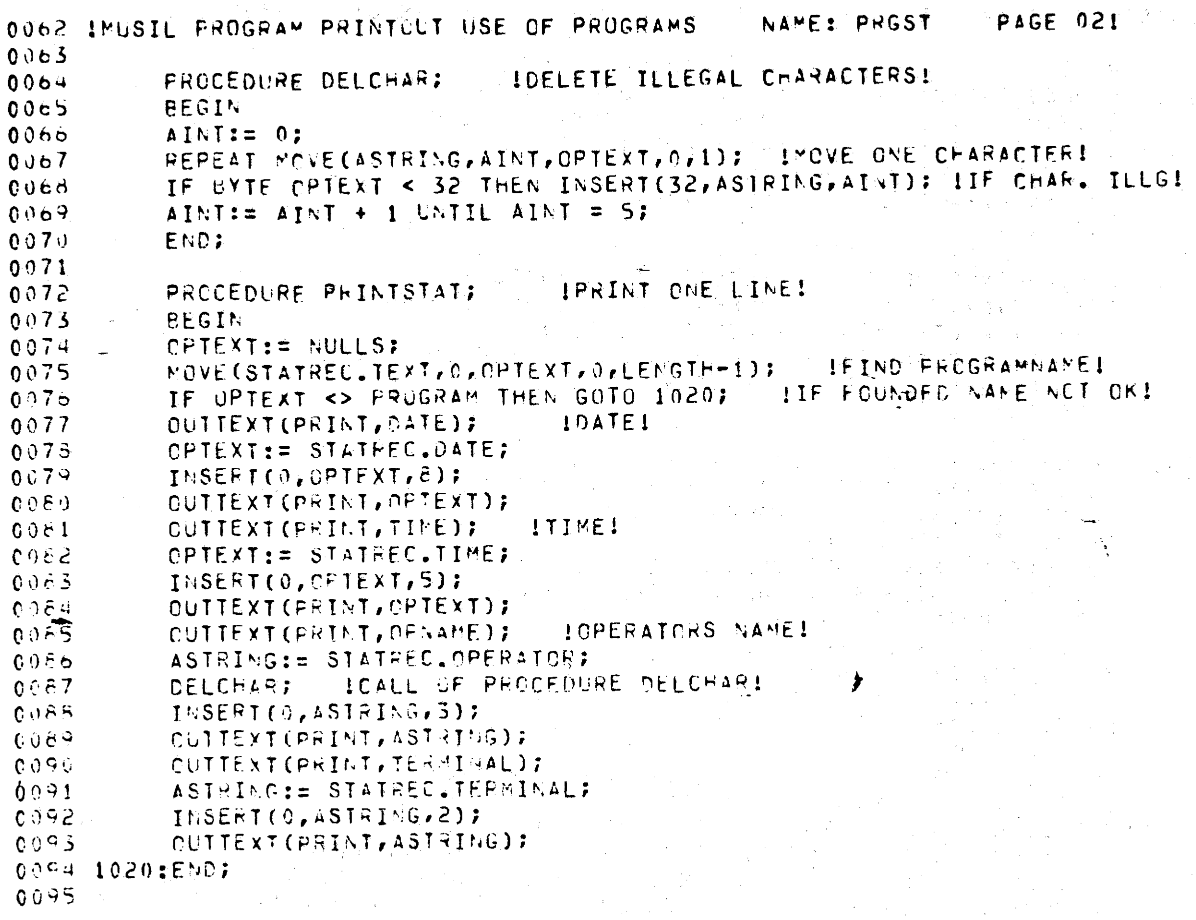
```
!MUSIL PROGRAM PRINTOUT OPERATORSTATISTIC     NAME: OPSTA     PAGE 03!

     BEGIN                     !MAIN PROGRAM!
10:  OPMESS(PRGNO);            !MESSAGE TO OPERATOR!
     OPIN(OPTEXT);             !PREPARE OPTEXT FOR INPUT!
     OPWAIT(ATNT);             !WAIT FOR INPUT!
     IF OPTEXT = START THEN GOTO 15;
     OPMESS(PARAM);            !IF NOT START!
     GOTO 10;

15:  OPEN(PRINT,3);            !OPEN PRINTER!
     SETPOSITION(PRINT,0,0);
     OUTTEXT(PRINT,FORMF);!NEW PAGE!
     OUTTEXT(PRINT,HEAD); !WRITE HEAD ON PAGE!
     OUTTEXT(PRINT,NL);

     OPEN(DISC,1);             !OPEN DISC FOR READING!
     SETPOSITION(DISC,0,0);!POSITION TO FIRST SECTOR!
     DISCLENGTH:= RECLGTH;        !AJUST RECORDLENGTH!
     GETREC(DISC,DISCLENGTH);!READ THE RECORD WITH LENGTH!
     RECORDS:= (WORD DISC↑.SECTOR-1)*NORECS; !NO OF RECORDS IN SECTOR!
     RECORDS:= RECORDS+((WORD DISC↑.BYTES+1)/DISCLENGTH); !NO OF RECORDS!
     SETPOSITION(DISC,0,1);

     WHILE RECORDS > 0 DO
       BEGIN
       GETREC(DISC,DISCLENGTH);
       MOVE(DISC↑,0,STATREC,0,DISCLENGTH);
       PRINTSTAT;
       RECORDS:= RECORDS - 1;
       END;
     CLOSE(DISC,1);    !CLOSE INPUT!
     OUTTEXT(PRINT,NL);
     OUTTEXT(PRINT,XEND); !PRINT END ON PRINTER!
     OUTTEXT(PRINT,NL);
     OUTTEXT(PRINT,FORMF); !NEW PAGE!
     CLOSE(PRINT,1);           !CLOSE PRINTER!
     OPMESS(XEND);             !MESSAGE TO OPERATOR!
     GOTO 10;
     END;

SIZE: 01568
```

```
STATISTIC FOR OPERATORSTROKES

OPERATOR NAME: JHA    TERMINAL NO.: 01    NO. OF KEYSTROKES: 00032
OPERATOR NAME: JHA    TERMINAL NO.: 01    NO. OF KEYSTROKES: 00278
OPERATOR NAME: JHA    TERMINAL NO.: 01    NO. OF KEYSTROKES: 00063
OPERATOR NAME: JHA    TERMINAL NO.: 01    NO. OF KEYSTROKES: 00087
OPERATOR NAME: JHA    TERMINAL NO.: 01    NO. OF KEYSTROKES: 00058
OPERATOR NAME: JHA    TERMINAL NO.: 01    NO. OF KEYSTROKES: 00006
OPERATOR NAME: JHA    TERMINAL NO.: 01    NO. OF KEYSTROKES: 00177
OPERATOR NAME: JHA    TERMINAL NO.: 01    NO. OF KEYSTROKES: 00099
OPERATOR NAME: JHA    TERMINAL NO.: 01    NO. OF KEYSTROKES: 00214
OPERATOR NAME: JHA    TERMINAL NO.: 01    NO. OF KEYSTROKES: 00631
OPERATOR NAME: JHA    TERMINAL NO.: 01    NO. OF KEYSTROKES: 00545
OPERATOR NAME: JHA    TERMINAL NO.: 01    NO. OF KEYSTROKES: 00549
OPERATOR NAME: JHA    TERMINAL NO.: 01    NO. OF KEYSTROKES: 00062
OPERATOR NAME: JHA    TERMINAL NO.: 01    NO. OF KEYSTROKES: 00267
OPERATOR NAME: JHA    TERMINAL NO.: 01    NO. OF KEYSTROKES: 00346
OPERATOR NAME: JHA    TERMINAL NO.: 01    NO. OF KEYSTROKES: 01007
OPERATOR NAME: JHA    TERMINAL NO.: 01    NO. OF KEYSTROKES: 00249
OPERATOR NAME: JHA    TERMINAL NO.: 01    NO. OF KEYSTROKES: 00554
OPERATOR NAME: JHA    TERMINAL NO.: 01    NO. OF KEYSTROKES: 01261
OPERATOR NAME: JHA    TERMINAL NO.: 01    NO. OF KEYSTROKES: 00066
OPERATOR NAME: JHA    TERMINAL NO.: 01    NO. OF KEYSTROKES: 00006
OPERATOR NAME: JHA    TERMINAL NO.: 01    NO. OF KEYSTROKES: 00193
OPERATOR NAME: JHA    TERMINAL NO.: 01    NO. OF KEYSTROKES: 00035
OPERATOR NAME: JHA    TERMINAL NO.: 01    NO. OF KEYSTROKES: 00076
OPERATOR NAME: JHA    TERMINAL NO.: 01    NO. OF KEYSTROKES: 00242
OPERATOR NAME: JHA    TERMINAL NO.: 01    NO. OF KEYSTROKES: 00004
OPERATOR NAME: JHA    TERMINAL NO.: 01    NO. OF KEYSTROKES: 00044
OPERATOR NAME: JHA    TERMINAL NO.: 01    NO. OF KEYSTROKES: 00127
OPERATOR NAME: JHA    TERMINAL NO.: 01    NO. OF KEYSTROKES: 00039
OPERATOR NAME: JHA    TERMINAL NO.: 01    NO. OF KEYSTROKES: 00046
OPERATOR NAME: JHA    TERMINAL NO.: 01    NO. OF KEYSTROKES: 00033
OPERATOR NAME: JHA    TERMINAL NO.: 01    NO. OF KEYSTROKES: 00082
OPERATOR NAME: JHA    TERMINAL NO.: 01    NO. OF KEYSTROKES: 00105
OPERATOR NAME: JHA    TERMINAL NO.: 01    NO. OF KEYSTROKES: 00052
OPERATOR NAME: JHA    TERMINAL NO.: 01    NO. OF KEYSTROKES: 00363
OPERATOR NAME: JHA    TERMINAL NO.: 01    NO. OF KEYSTROKES: 00154
OPERATOR NAME: JHA    TERMINAL NO.: 01    NO. OF KEYSTROKES: 00284
OPERATOR NAME: JHA    TERMINAL NO.: 01    NO. OF KEYSTROKES: 00071
OPERATOR NAME: JHA    TERMINAL NO.: 01    NO. OF KEYSTROKES: 00069
OPERATOR NAME: JHA    TERMINAL NO.: 01    NO. OF KEYSTROKES: 00203
OPERATOR NAME: JHA    TERMINAL NO.: 01    NO. OF KEYSTROKES: 00164
OPERATOR NAME: JHA    TERMINAL NO.: 01    NO. OF KEYSTROKES: 00065
OPERATOR NAME: JHA    TERMINAL NO.: 01    NO. OF KEYSTROKES: 00354
OPERATOR NAME: JHA    TERMINAL NO.: 01    NO. OF KEYSTROKES: 00636
OPERATOR NAME: JHA    TERMINAL NO.: 01    NO. OF KEYSTROKES: 00034
OPERATOR NAME: JHA    TERMINAL NO.: 01    NO. OF KEYSTROKES: 00049
OPERATOR NAME: JHA    TERMINAL NO.: 01    NO. OF KEYSTROKES: 00096
OPERATOR NAME: JHA    TERMINAL NO.: 01    NO. OF KEYSTROKES: 00047
OPERATOR NAME: JHA    TERMINAL NO.: 01    NO. OF KEYSTROKES: 00110
OPERATOR NAME: FLH    TERMINAL NO.: 02    NO. OF KEYSTROKES: 00142
OPERATOR NAME: FLH    TERMINAL NO.: 02    NO. OF KEYSTROKES: 00127
OPERATOR NAME: FLH    TERMINAL NO.: 02    NO. OF KEYSTROKES: 00365
OPERATOR NAME: FLH    TERMINAL NO.: 01    NO. OF KEYSTROKES: 00307
OPERATOR NAME: FLH    TERMINAL NO.: 01    NO. OF KEYSTROKES: 00157
OPERATOR NAME: FLH    TERMINAL NO.: 01    NO. OF KEYSTROKES: 00186
OPERATOR NAME: FLH    TERMINAL NO.: 01    NO. OF KEYSTROKES: 00399
OPERATOR NAME: FLH    TERMINAL NO.: 01    NO. OF KEYSTROKES: 00379
OPERATOR NAME: FLH    TERMINAL NO.: 01    NO. OF KEYSTROKES: 00465
OPERATOR NAME: FLH    TERMINAL NO.: 02    NO. OF KEYSTROKES: 00011

END
```

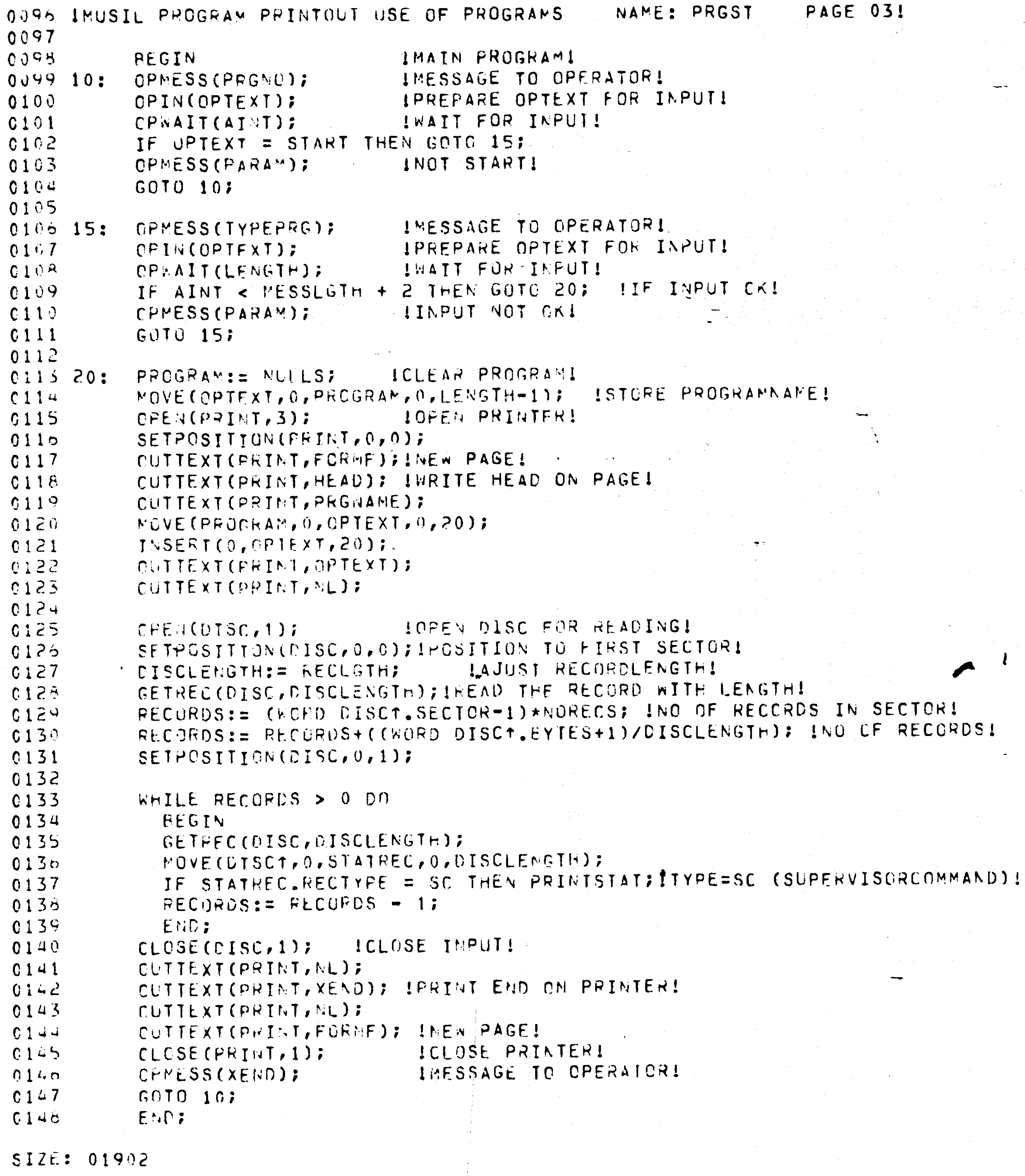
APPENDIX B - CODINGEXAMPLE FOR READING LOGFILE.

This coding example shows a way to read in Logfile (LOGF) and produce a statistic which is able to print out the use of a certain program, when it has been used and of which operator.

```
MUSIL COMPILER 4
PROG


!MUSIL PROGRAM PRINTOUT USE OF PROGRAMS    NAME: PRGST    PAGE 01!

CONST
PRGNO=      '<10><13>PROGRAM PRGST<10><13>',
PRGNAME=    '      PROGRAM NAME: ',
OPNAME=     '  OPERATOR NAME: ',
TERMINAL=   ' TERMINAL NO.: ',
NULLS=      '<0><0><0><0><0><0><0><0><0><0><0><0><0><0><0><0><0><0><0><0>',
SC=         'SC',
TIME=       ' TIME: ',
DATE=       '<10><13>USED ON DATE: ',
TYPEPRG=    '<10>TYPE SEARCHPROGRAMNAME: ',
HEAD=       '<10><13>STATISTIC FOR USE OF PROGRAMS',
START=      'START',
XEND=       '<10><13>END<10><13>',
PARAM=      '**PARAM',
NL=         '<10><13>',
RECLGTH=    102,  !RECORDLENGTH!
NORECS=     5,    !NO. OF RECORDS IN ONE SECTOR!
MESSLGTH=   20,   !MAX. NO. OF BYTES TO SEARCH!
FORMF=      '<12>';


VAR
ASTRING:     STRING(6);       !WORK STRING!
OPTEXT:      STRING(80);      !OPERATORSTRING!
AINT:        INTEGER;         !WORK INTEGER!
DISCLENGTH:  INTEGER;         !RECORDLENGTH!
RECORDS:     INTEGER;         !NO. OF RECORDS IN LOGFFILE!
LENGTH:      INTEGER;         !LENGTH OF PROGRAMNAME!
PROGRAM:     STRING(MESSLGTH);!NAME OF PROGRAM TO SEARCH!


STATREC:     RECORD
             RECTYPE:   STRING(2);
             DATE:      STRING(6);
             TIME:      STRING(5);
             TERMINAL:  STRING(2)  FROM 17;
             OPERATOR:  STRING(3)  FROM 19;
             TEXT:      STRING(79) FROM 23;
             STATREC:   STRING(102)FROM 1
             END;

DISC:        FILE        !INPUTFILE!
             'LOGF',     !FILENAME!
             60,         !KIND = DISC!
             1,          !BUFFERS!
             512,        !BUFFERSIZE!
             UB          !FORMAT UNDIFINED BLOCKED!
             OF RECORD
             SECTOR:    STRING(2);  !NO. OF SECTORS!
             BYTES:     STRING(2)   !NO. OF BYTES IN LAST SECTOR!
             END;

PRINT:       FILE        !INPUTFILE!
             'SP',       !FILENAME!
             3,          !KIND = CHARACTER!
             1,          !BUFFERS!
             100,        !BUFFERSIZE!
             U           !FORMAT UNDIFINED!
             OF STRING(100);

```



```
!MUSIL PROGRAM PRINTOUT USE OF PROGRAMS    NAME: PRGST    PAGE 02!

      PROCEDURE DELCHAR;    !DELETE ILLEGAL CHARACTERS!
      BEGIN
      AINT:= 0;
      REPEAT MOVE(ASTRING,AINT,OPTEXT,0,1);  !MOVE ONE CHARACTER!
      IF BYTE OPTEXT < 32 THEN INSERT(32,ASTRING,AINT); !IF CHAR. ILLG!
      AINT:= AINT + 1 UNTIL AINT = 5;
      END;

      PROCEDURE PRINTSTAT;      !PRINT ONE LINE!
      BEGIN
      OPTEXT:= NULLS;
      MOVE(STATREC.TEXT,0,OPTEXT,0,LENGTH-1);    !FIND PROGRAMNAME!
      IF OPTEXT <> PROGRAM THEN GOTO 1020;    !IF FOUNDED NAME NOT OK!
      OUTTEXT(PRINT,DATE);      !DATE!
      OPTEXT:= STATREC.DATE;
      INSERT(0,OPTEXT,8);
      OUTTEXT(PRINT,OPTEXT);
      OUTTEXT(PRINT,TIME);    !TIME!
      OPTEXT:= STATREC.TIME;
      INSERT(0,OPTEXT,5);
      OUTTEXT(PRINT,OPTEXT);
      OUTTEXT(PRINT,OPNAME);    !OPERATORS NAME!
      ASTRING:= STATREC.OPERATOR;
      DELCHAR;    !CALL OF PROCEDURE DELCHAR!
      INSERT(0,ASTRING,3);
      OUTTEXT(PRINT,ASTRING);
      OUTTEXT(PRINT,TERMINAL);
      ASTRING:= STATREC.TERMINAL;
      INSERT(0,ASTRING,2);
      OUTTEXT(PRINT,ASTRING);
1020:END;
```



```
!MUSIL PROGRAM PRINTOUT USE OF PROGRAMS    NAME: PRGST    PAGE 03!

        BEGIN                     !MAIN PROGRAM!
10:     OPMESS(PRGNO);            !MESSAGE TO OPERATOR!
        OPIN(OPTEXT);             !PREPARE OPTEXT FOR INPUT!
        OPWAIT(AINT);             !WAIT FOR INPUT!
        IF OPTEXT = START THEN GOTO 15;
        OPMESS(PARAM);            !NOT START!
        GOTO 10;

15:     OPMESS(TYPEPRG);          !MESSAGE TO OPERATOR!
        OPIN(OPTEXT);             !PREPARE OPTEXT FOR INPUT!
        OPWAIT(LENGTH);           !WAIT FOR INPUT!
        IF AINT < MESSLGTH + 2 THEN GOTO 20;  !IF INPUT OK!
        OPMESS(PARAM);            !INPUT NOT OK!
        GOTO 15;

20:     PROGRAM:= NULLS;     !CLEAR PROGRAM!
        MOVE(OPTEXT,0,PROGRAM,0,LENGTH-1);  !STORE PROGRAMNAME!
        OPEN(PRINT,3);       !OPEN PRINTER!
        SETPOSITION(PRINT,0,0);
        OUTTEXT(PRINT,FORMF);!NEW PAGE!
        OUTTEXT(PRINT,HEAD); !WRITE HEAD ON PAGE!
        OUTTEXT(PRINT,PRGNAME);
        MOVE(PROGRAM,0,OPTEXT,0,20);
        INSERT(0,OPTEXT,20);
        OUTTEXT(PRINT,OPTEXT);
        OUTTEXT(PRINT,NL);

        OPEN(DISC,1);            !OPEN DISC FOR READING!
        SETPOSITION(DISC,0,0);!POSITION TO FIRST SECTOR!
        DISCLENGTH:= RECLGTH;        !AJUST RECORDLENGTH!
        GETREC(DISC,DISCLENGTH);!READ THE RECORD WITH LENGTH!
        RECORDS:= (WORD DISC↑.SECTOR-1)*NORECS; !NO OF RECORDS IN SECTOR!
        RECORDS:= RECORDS+((WORD DISC↑.BYTES+1)/DISCLENGTH); !NO OF RECORDS!
        SETPOSITION(DISC,0,1);

        WHILE RECORDS > 0 DO
          BEGIN
          GETREC(DISC,DISCLENGTH);
          MOVE(DISC↑,0,STATREC,0,DISCLENGTH);
          IF STATREC.RECTYPE = SC THEN PRINTSTAT;!TYPE=SC (SUPERVISORCOMMAND)!
          RECORDS:= RECORDS - 1;
          END;
        CLOSE(DISC,1);   !CLOSE INPUT!
        OUTTEXT(PRINT,NL);
        OUTTEXT(PRINT,XEND); !PRINT END ON PRINTER!
        OUTTEXT(PRINT,NL);
        OUTTEXT(PRINT,FORMF); !NEW PAGE!
        CLOSE(PRINT,1);          !CLOSE PRINTER!
        OPMESS(XEND);            !MESSAGE TO OPERATOR!
        GOTO 10;
        END;

SIZE: 01902
```

```
STATISTIC FOR USE OF PROGRAMS     PROGRAM NAME: COPY

USED ON DATE: 78.02.06 TIME: 13.18  OPERATOR NAME: FLH TERMINAL NO.: 02
USED ON DATE: 78.02.06 TIME: 11.12  OPERATOR NAME: FLH TERMINAL NO.: 02
USED ON DATE: 78.02.06 TIME: 11.12  OPERATOR NAME: FLH TERMINAL NO.: 02
USED ON DATE: 78.02.06 TIME: 11.13  OPERATOR NAME: FLH TERMINAL NO.: 02

END
```