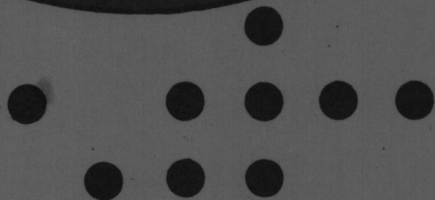
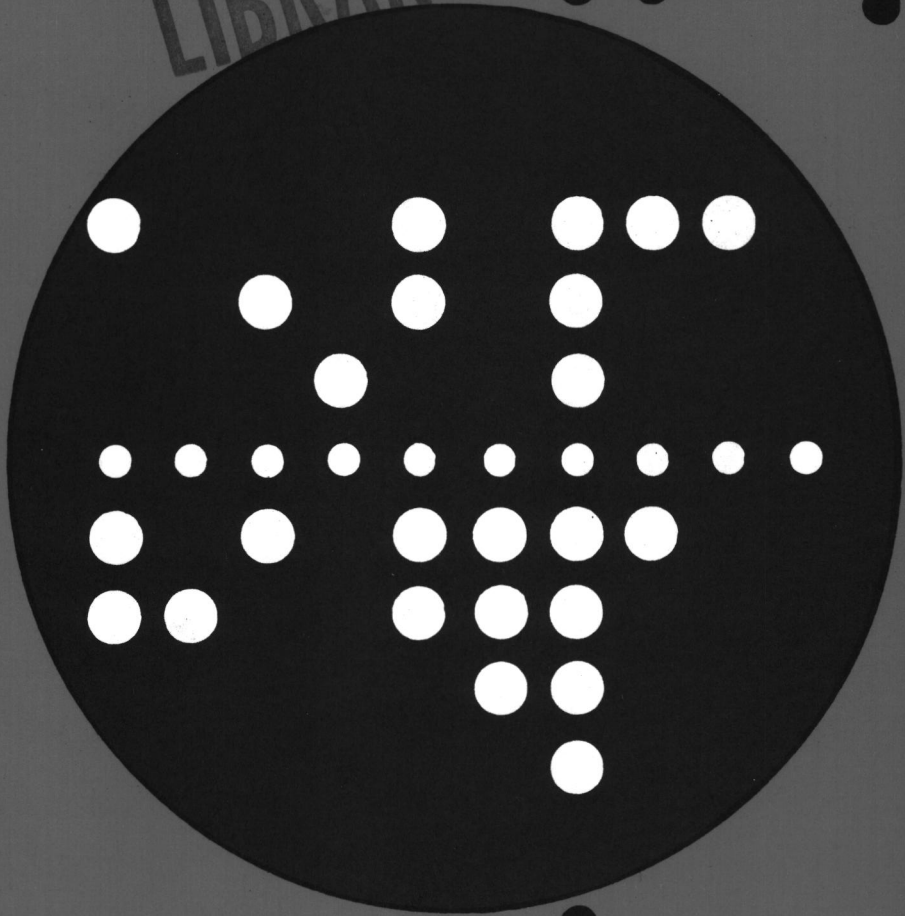


Commission of the European Communities ●

Joint Research Centre - Ispra ● ● ●

LIBRARY

**Computing Centre Newsletter**



April 1977 ● No 10

111. 2/10

Due to technical reasons the publication of this Newsletter has been delayed.

We apologize and we hope to anticipate the appearance of the next edition.

The Editors

# **Contents**

<b>Editorial note</b>	<b>2</b>
<b>The system of software at the Computing Centre</b>	
<b>present status and possible development</b>	<b>3</b>
<b>IBM Fortran IV (H extended)</b>	<b>6</b>
<b>Catalogued procedure LISP</b>	<b>10</b>
<b>Statistics on computer utilization, March</b>	<b>11</b>
<b>Utilization by objectives and accounts, March</b>	<b>12</b>
<b>Table of equivalent time, summary per month</b>	
<b>and cumulative</b>	<b>13</b>
<b>Back-up of PSQ-file</b>	<b>14</b>
<b>Note to the users</b>	<b>21</b>
<b>Visitors</b>	<b>22</b>

---

## Note of the Editor

The present Newsletter will be published monthly except for August and December.

The Newsletter will include:

- Developments, changes, uses of installations
- Announcements, news and abstracts on initiatives and accomplishments.

The Editor thanks in advance those who will want to contribute to the Newsletter by sending articles in English or French to one of the following persons of the Editorial Board.

## Note de la Rédaction

Le présent Bulletin sera publié mensuellement excepté durant les mois d'août et décembre.

Le Bulletin traitera des:

- Développements, changements et emploi des installations
- Avis, nouvelles et résumés concernant les initiatives et les réalisations.

La Rédaction remercie d'avance ceux qui voudront bien contribuer au Bulletin en envoyant des articles en anglais ou français à l'un des membres du Comité de Rédaction.

---

## Editorial Board / Comité de Rédaction

S.R. Gabbai, D.G. Ispra  
H. de Wolde, C.C. Ispra  
C. Pigni, C.C. Ispra  
J. Pire, C.C. Ispra

---

## Computing Centre References

		Room	Tel.
<i>Manager</i>	J. Pire	1816	732
Adjoined	G. Gaggero	1874	787
<i>Computer Room</i>	P. Tomba	1857	797
Adjoined	A. Binda	1857	797
<i>Peripherals</i>	G. Nocera	1825	767
<i>System Group</i>	D. Koenig	1839	742
Adjoined	P.A. Moinil	1841	704
<i>Informatics Support</i>	G. Gaggero	1874	787
○ General Information	G. Hudry	1873	787
○ Program Information Service	G. Gaggero	1874	787
Adjoined	S. Leo Menardi	1884	721
○ Graphics and Support to Users	H.I. de Wolde	1890	753
Adjoined	A. Pollicini	1882	743
Application Packages	A. Inzaghi	1887	755
Programming Languages	C. van den Muyzenberg	1848	781

---

<b>Editor</b> : Sylvia R. Gabbai
<b>Layout</b> : Paul De Hoe
<b>Graphical and Printing Workshop</b> , JRC Ispra

# **The System Software at the Computing Centre Present Status and Possible Development**

D. Koenig

## **Preface**

This article is the first one of a series of three articles. It will describe the present situation of the system software, outline the possibilities for its future development and serve as an introduction for the following articles.

The second article will present the highlights of the present operating system – OS/MVT (1) Release 21.8 – on a more technical level.

The third article will give an introduction to one component of the present operating system – namely the Time Sharing Option (TSO) of OS/MVT (1) – and will present an overview of the facilities which are available to the user of TSO. (Furthermore detailed information on how to use TSO will be provided before the system will be introduced).

In the following the term “system software” is used in a restricted sense to only denote the operating system proper (i.e. OS/MVT and HASP and components thereof). Excluded from the following discussion are the language processors like compilers and assemblers, the data base systems even if they provide their own data communication environment and the large application packages like for instance GENESYS and ICES. This does however not mean that these components are excluded in the planning for the system software development.

## **The System Software and its Development**

For some time there exist (among others) two partly interrelated trends of development of computing. The first one is to provide for the users better access to the computing facilities. This is usually accomplished by the transition from a “batch operating system” to a time-sharing operating system offering access to the computer via on-line terminals. In parallel to the development of time-sharing operating systems new computing facilities were developed which are especially suited to be used from a terminal like for instance text editors <sup>1)</sup>, incremental compilers <sup>2)</sup> or high level language debugging systems <sup>3)</sup>. The second trend is the decentralization of computing services. This is usually done by removing some of the existing service functions from a central machine and implementing them on different, interconnected machines. Some of the problems and

possible advantages of this approach are discussed in Ref. 4, 5 and 6. If executed properly this transition should have only minor influences on the users of a computing service.

It is expected that also the development of the JRC computing facilities will follow these two trends; that is, in a first step the access to the central machine will be improved and in a second step some functions presently incorporated into the software of the central machine will be implemented on other machines.

### **Better Access**

After a general analysis of the computing requirements for the JRC in the future <sup>7)</sup> and after a more detailed analysis of the requirements for conversational computing (contained in <sup>8)</sup>) it became evident that the existing teleprocessing system could not fulfill these requirements because there was no manpower available to further develop it. Therefore it became necessary to replace the operating system OS/MFT (2) by the operating system OS/MVT (1) in order to be able to introduce a teleprocessing monitor which is adapted to the computing needs of the Centre.

After the introduction of OS/MVT at the end of last year an analysis of four TP-monitors:

CONTEXT	(TEL-TEC System S.A.)
IMSJDS	(IBM Corp.)
ROSCOE	(Applied Data Research Inc.)
TSQ	(IBM Corp.)

was made. Based on technical, financial, maintenance and support aspects of the different monitors as well as the computing needs of the JRC, the decision was taken to choose TSO. The system is now in a testing phase and will probably be available for the connection of the first terminals in September this year. The service of the internal TP-system (TELEUR) will be offered for as long as there are major applications using it. It is however recommended that new applications use the TSO-system. Existing, internally developed applications like SIMAS or COREA will be moved to TSO. Since the editing facilities of TSO are more powerful and more convenient to use than those offered by PSQ/FILEDI it is expected that the transition from PSQ/FILEDI usage to TSO usage is made quickly. The user may obtain assistance in the transfer of his data from PSQ to TSO.

---

(1) MVT = Multiprogramming with a variable number of tasks

(2) MFT = Multiprogramming with a fixed number of tasks

## **Decentralization**

The second step in the development of the JRC computing facilities - that is the decentralization of service functions - will not be taken in the near future. In order to introduce a sensible decentralization in the Centre it is necessary to be able to interconnect different types of computers in a uniform way. At present a first study has been performed — with active user participation — on how to solve the interconnection problem within the JRC. The results of this study will be presented in a later issue. Only after the interconnection problem is solved further decisions on how to decentralize the actual computing services can be taken in cooperation with the involved users.

## **References**

1. RICE/VAN DAM: On-line Text Editing, A Survey, *Computing Surveys* 3(1971), pp. 93-114
2. RISHEL, W.J.: Incremental Compilers, *Datamation* Jan. 1970, pp. 129-136
3. EVANS/DARLEY: On-line Debugging Techniques, A Survey, *AFIPS Proceedings* 29 (FJCC 1966), pp. 37-49
4. PATRICK, R.L.: Decentralizing Hardware and Dispersing Responsibility, *Data-mation* vol.22, No. 5, May 1976, pp. 79-84
5. WAGNER, F.V.: Is Decentralization Inevitable? *Datamation* vol.22, No. 10, Nov. 1976, pp. 86-97
6. FERRERA/NILLES: Five Years Planning for Data Communiation, *Datamation* vol.22, No. 10, Oct. 1976, pp. 51-57
7. LANGSFORD: Report on Computing Requirements at Ispra 1976-1980, HL.76/2306 (Aug. 1976), Computer Service and Systems Division, AERE Harwell
8. GRABE/LANGSFORD: Report on the Strategy for Computing Facilities Deve-lopment at JRC Ispra, Christian Rovsing Report (Dec. 1976)

## IBM Fortran IV (H Extended)

C. Pigni

### Introduction

Recently the FORTRAN IV (H Extended) compiler with the Library Model II have been installed. These are two product programs produced by the IBM corporation.

The library replies to the requests of the compiler for mathematical and service subprograms.

The FORTRAN IV (H Extended) compiler accepts any source programs that are valid defined in IBM System/360 and System/370 FORTRAN IV language.

The compiler has several extended features and capabilities.

### Extended-precision

**Data types:** REAL\*16 and COMPLEX\*32 are implemented with function and input/output conversion support.

The magnitude is from  $16^{-65}$  (approximately  $10^{-78}$ ) through  $16^{-63}$  (approximately  $10^{-75}$ ).

The precision is of 28 hexadecimal digits (approximately 35 decimal digits).

A REAL\*16 CONSTANT consists of a string of decimal digits with or without a decimal point, followed by an extended precision exponent. The exponent consists of the letter Q followed by a signed or unsigned one or two digit integer constant.

If the string is not followed by the exponent it is treated as a valid basic real constant of four digits.

#### Examples:

```
.4912718438297165362815Q+33  
67.27951Q09
```

The letter Q is used in the format statement to represent extended REAL data fields.

#### Example:

```
101 FORMAT (Q18.6,2Q24.16,2Q21.18)
```

The first field contains 18 characters, of which 6 are the fractional portion.

The second and the third are identical fields of 24 characters, 16 of which are the fractional portion.



The fourth and fifth are identical fields and contain 21 characters, of which 18 are the fractional portion.

### External Statement Extension

This feature provides a means of declaring certain names that would otherwise be considered FORTRAN library subprograms to be the names of user-supplied functions or subroutines.

The added facility is obtained by writing an AMPERSAND (&) preceding the library function or subroutine name in an EXTERNAL statement. A user-supplied function is regarded as of the type and length predefined by its initial character, unless its type is declared in an IMPLICIT or EXPLICIT specification statement. Compiler checking of the type, number and length of the arguments of both functions and subroutines is suspended.

#### *Example:*

```
EXTERNAL &QSIN
```

```
A=QSIN(E,F,D)
```

*Explanation:* Because of its appearance with an ampersand in an EXTERNAL statement, QSIN is treated as a user-supplied external procedure of type REAL\*4 without checking on the arguments.

### Automatic Function Selection

The automatic function selection facility allows the programmer to use a single generic name when requesting a FORTRAN-supplied function which has several names, depending on argument type. The proper function is selected by the FORTRAN compiler based on the type and the length of the argument of the function. The function is requested by including the GENERIC statement.

#### *Example:*

```
GENERIC  
EXTERNAL SIN  
REAL*8 A,B,C,D  
C=SIN(A)  
D=DSIN(B)  
CALL SUB(SIN)
```

*Explanation:* Because automatic function selection has been invoked, function DSIN is called to calculate the value of C, as well as D. The specific name SIN is passed to the subroutine SUB.

## List-directed Input/Output

The list directed input/output feature permits reading and writing of formatted data without a FORMAT statement. The general form of READ and WRITE statement becomes the following:

```
READ (a,*,ERR=b,END=c) list
WRITE (a,*) list
```

The format statement number is substituted by an asterisk. The BACKSPACE statement may not be used with list-directed data.

The list-directed input data consists of an alternation of constants and separators. An input constant may be of any valid FORTRAN data type, except that a literal constant must be enclosed in quotation marks. The separators are blank and comma, to separate one item from another; slash (/) is a special-purpose separator, indicating that no more data is to be read during the current execution of a READ statement.

### Example:

```
READ(5,*,END=101,ERR=520) (A(I),I=1,50),WORD1,WORD2,B,C,D,E
```

data input on two cards as follows:

```
first card      50*0. 'EXAM' 'PLE '
second card     1.5,0.165D-6,, TRUE.
```

**Explanation:** When the READ is executed, the value 0. is read into each of the first 50 elements of array A; text is read into location WORD1 and WORD2; two values are read into location B and C; location D is skipped because of the null specification and logical value true is placed in location E.

## Asynchronous input

The asynchronous input/output feature providing high-speed data transmission for unformatted sequential data-sets. It is performed by the READ, WRITE and WAIT statements.

The asynchronous READ and WRITE statements initiate a transmission. The WAIT statement, which must be executed for each asynchronous READ and WRITE, terminates the transmission cycle.

### Example:

```
.
.
.
READ (10,ID=1) (A(I),I=1,200)
.
.
WAIT (10,ID=1) (A(I),I=1,200)
```

After the execution of the WAIT, the specified array elements may be referred to.

### **General Observation**

The FORTRAN IV (H extended) compiler with the Library mod. II, have some other improvements; please consult the publications specified in the bibliography. We remember here the ANSF option of the compiler to recognize only those library and build-in functions specified by the American National Standards Institute (ANS); and we signal the data conversion routines, which provide greater accuracy in conversions.

### **BIBLIOGRAPHY**

1. FORTRAN IV LANGUAGE - GC28-6515 - IBM Corporation
2. IBM OS FORTRAN IV (H Extended) Compiler - Programmer's Guide  
- SC28-6852 - IBM Corporation

## Catalogued Procedure LISP

For the users of the LISP (LISt Processing) language, the procedure to execute the LISP interpreter (using 300K bytes storage, card \$ class 3) is inserted in the library.

### Examples

1. LISP program/data on cards  
// EXEC LISP  
//GO.SYSIN DD \*

PROGRAM/DATA

/\*

2. LISP program/data on a disk dataset MYDATA on disk USER02  
// EXEC LISP,DSND=MYDATA,VOLD=USER02
3. LISP program/data on a member (P1) of a partitioned dataset (DATASET) followed by punched cards  
// EXEC LISP,DSND=DATASET(P1),VOLD=USER02  
//GO.SYSIN DD \*

CARDS

/\*

4. LISP program/data on tape  
// EXEC LISP  
//GO.SYSIN DD DSN=LISPROG,VOL=SER=EU9999,DISP=OLD,  
// UNIT=TP9

The parameters that may be used are:

DSNA	the dataset with ATOMS, default=LISP(ATOMS)
DSNF	the dataset with FUNCTIONS, default=LISPD(FUNC)
DSND	the dataset with input data, default=LISPD(EMPTY), an empty dataset
VOLD	volume of the input data (on disk), default=COPIC1

For more information about LISP see:

*Analisi di alberi logici AND/OR mediante tecniche non numeriche  
il codice SALP.*

M. Astolfi, S. Contini, G. Volta  
EUR Report No. 5641

## Statistics of computing installation utilization

### Report of computing installation exploitation for the month of

	YEAR 1977	YEAR 1976
Number of working days _____	23 d	22 d
Work hours from 8.00 to 24.00 for _____	16.00 h	15.00 h
Duration of scheduled maintenance _____	19.49 h	33.96 h
Duration of unexpected maintenance _____	14.92 h	4.34 h
Total maintenance time _____	34.51 h	37.30 h
Total exploitation time _____	304.82 h	286.70 h
CPU time in problem mode _____	142.24 h	118.58 h

#### Teleprocessing:

CPU time _____	2.20 h	0.80 h
I/O number _____	298,862	433,655
Equivalent time _____	2.31 h	3.90 h
Elapsed time _____	136.00 h	87.00 h

#### Batch processing:

Number of jobs _____	11,107	10,067
Number of cards read _____	3,392,000	2,904,000
Number of cards punched _____	220,000	191,000
Number of lines printed _____	30,337,000	28,563,000
Number of pages printed _____	679,000	648,000

#### BATCH PROCESSING DISTRIBUTION BY REQUESTED CORE MEMORY SIZE

	100	200	300	400	600	800	1000	1400	total
Number of jobs	2805	3993	2597	1056	187	94	24	-	10036
Elapsed time (hrs)	53	183	193	137	38	26	16	-	646
CPU time (hrs)	3.4	31	40	32	14	10	7	-	137
Equivalent time (hrs)	18	72	87	71	20	11	12	-	291
Turn around time (hrs)	0.6	1.5	2.1	2.9	4.3	4.4	4.4	-	1.7

#### PERCENTAGE OF JOBS FINISHED IN LESS THAN

TIME	15'	30'	1h	2h	4h	8h	1D	2D	3D	6D
% year 1976	36.2	55.5	73.1	86.2	95.6	98.8	99.5	99.6	99.9	100
% year 1977	26.8	42.2	59.6	75.1	88.8	97.1	99.6	99.9	100	-

**Utilization of computer center by the objectives and appropriation accounts for the month of March**

**IBM 370/165**  
**equivalent time in hours**

1.20.2	General Services - Administration-Ispra	75.25
1.20.3	General Services - Technical-Ispra	3.10
1.50.2	Department B	0.04
1.90.0	ESSOR	13.91
1.92.0	Support to the Commission	2.43
2.10.1	Reactor Safety	82.47
2.10.2	Plutonium Fuel and Actinide Research	1.46
2.10.3	Nuclear Materials	2.17
2.20.1	Solar Energy	0.61
2.20.2	Hydrogen	0.27
2.20.4	Design Studies on Thermonuclear Fusion	0.65
2.30.0	Environment and Resources	39.34
2.40.0	METRE	4.12
2.50.1	Data Processing	85.95
2.50.3	Safeguards	0.33
	<b>TOTAL</b>	<b>312.10</b>
1.94.0	Services to external Users	17.86
	<b>TOTAL</b>	<b>329.96</b>



## **Back-up of PSQ-File**

Ole Nórager, ECDIN

### **General Description**

During the pilot phase of the ECDIN project it has been necessary to place a great number of files in the PSQ-system containing either data which needs checking and updating or computer programs under development. With many files in the PSQ system and with the repeated experience of destruction of data, when the system was overloaded, the lack of an automatic back-up system for PSQ files worried us. During 1976 we made several attempts to construct such a system and in June we succeeded in the construction, and the ECDIN group has since used an in-stream procedure BCKP to back-up PSQ files as members of unloaded partitioned data sets (PDS) on magnetic tapes.

In the present version the procedure makes use of eight different programs. The main program BACKUP is written in the COBOL programming language and this program invokes through three assembler routines, ASSPSQ, ASSIEC, and ASSIEH, the 'PSQ program', PSQB, and the two IBM utility programs IEBCOPY and IEHMOVE. The program IEFBR14 is used to allocate disk space.

The procedure is written to perform three operations :

- to back-up selected PSQ files as members of a new unloaded PDS on magnetic tape
- to generate PSQ files which are copies of members of an unloaded PDS, i.e. to restore PSQ files from the back-up tape
- to update a back-up tape by replacing members in or adding members to the PDS.

The necessary information about the PSQ files which are to be backed-up or restored are input to the main program as a card deck and this program generates the control statements for the PSQB program and the utility programs. In a back-up run PSQB copies a PSQ file to a member, MMM, of a PDS with the Data Set Name INTM. IEBCOPY copies MMM to a member of another PDS with the DSN WKSET and renames the member with a name indicated on the input cards (see the examples). When all the wanted PSQ files are copied to members of WKSET, IEHMOVE copies this PDS to a magnetic tape. To avoid lack of space in the PDS INTM, this data set is compressed each time a new PSQ file has been copied to INTM(MMM). In a restore run the data-flow goes in the opposite direction and for an



'update of a back-up tape'-run the old PDS is first loaded to a disk as the PDS WKSET and the operations then go on as for generation of a new back-up PDS. Figure 1 gives an overview of the data flow and figure 2 is the outline of the logic in the program BACKUP.

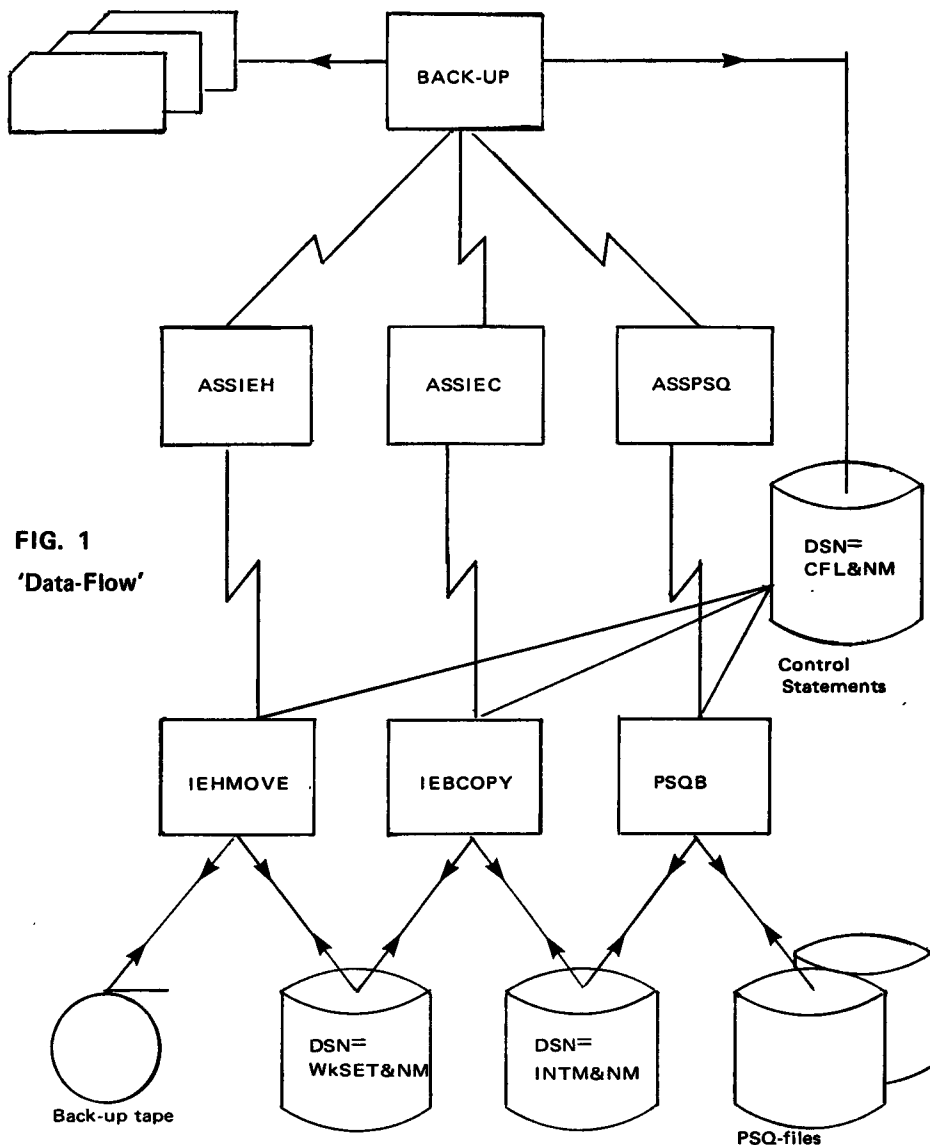


FIG. 1  
'Data-Flow'

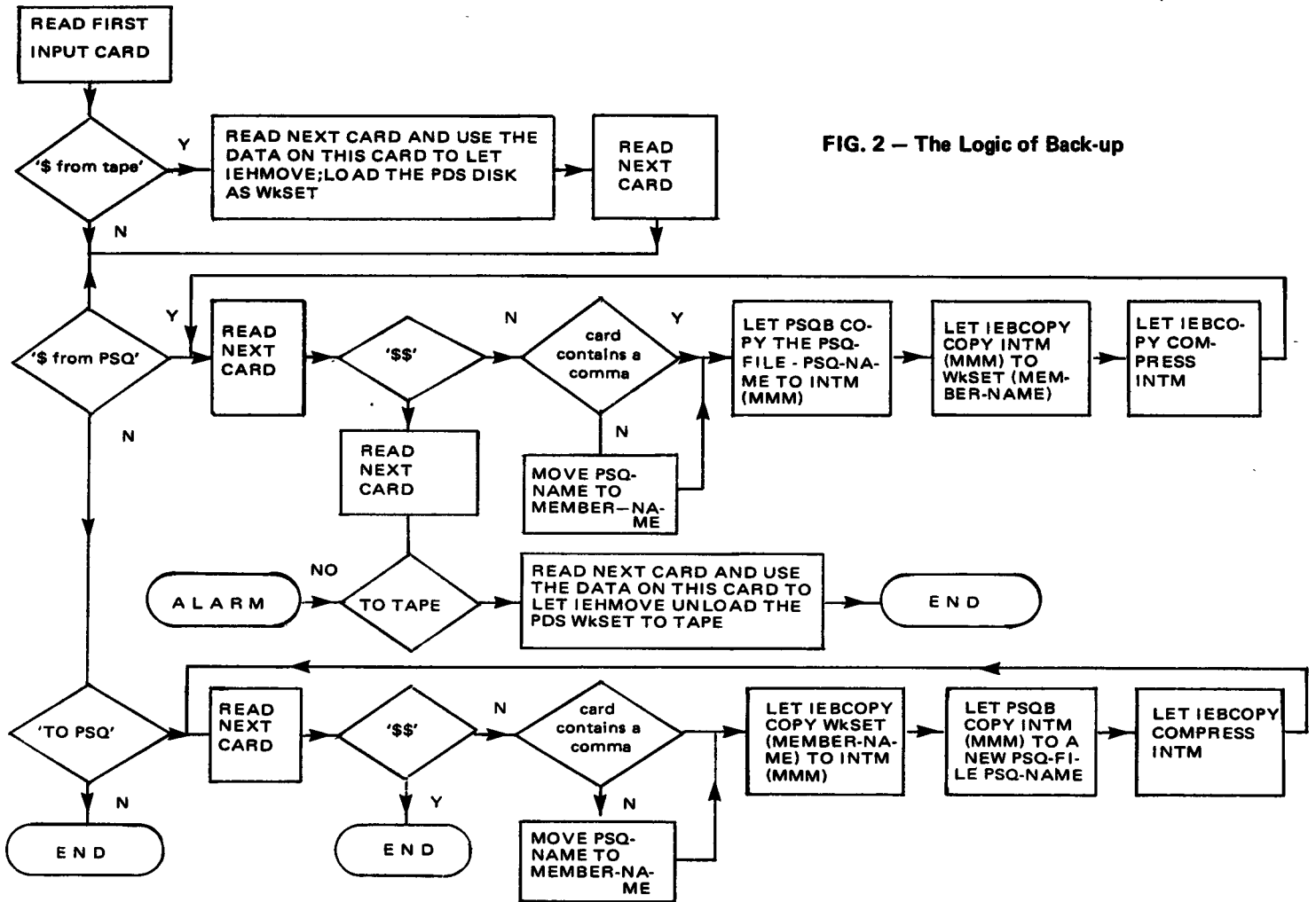


FIG. 2 - The Logic of Back-up

## Examples of Control Statement for the BCKP Procedure

### a) Generation of a new back-up file

This example shows the control statements necessary for the creation of a back-up file with copies of four PSQ files with the PSQ numbers 250, 251, 255 and 1003 and the PSQ names PSQ1, PSQ2, PSQ3 and PSQ4. The two first files use the PSQ names as member names in the generated PDS, where as PSQ3 and PSQ4 are given the member MEMBC and MEMBD. The unloaded PDS is placed as the first data set on a tape with the serial number EU9998 and given the data set name PSQBCK1.

```
$FROM PSQ
$0250PSQ1
$0251PSQ2
$0255PSQ3,MEMBC
$1003PSQ4.MEMBD
$$
$TO TAPE
$TO=TP9=EU9998,RENAME=PSQBCK1
/*
```

All cards have the first character in column 1. The PSQ numbers must be written with four digits, including leading zeros. The PSQ names, the member-names and the DSN follow the normal conventions for these names, i.e. up to 8 characters of which the first should be alphabetic. Whenever both a PSQ name and a member name is specified the member name shall follow immediately after the PSQ name only separated by a comma. The PSQB will as default be controlled by the PSQ statement \$PN, but this option can be changed by entering a card with \$\$PY between the \$FROM PSQ and the \$\$ card. All PSQ files described after the \$\$PY and before the \$\$ or a \$\$PN card will be listed.

*The Newsletter is available at:*

**Mrs. A. Cambon**  
Support to Computing  
Bldg. 36 - Tel. 753

*Des exemplaires du Bulletin  
sont disponibles chez:*

**Mme A. Cambon**  
Support to Computing  
Bât. 36 - Tel. 753

## b) Restoring of the PSQ files

This example shows how the four files in the previous example are restored in the PSQ system with the PSQ names MEMBA, MEMBB, MEMBC, and MEMBD. The file MEMBC is listed.

```
$FROM TAPE
$FROM=TP9=EU9999,PDS=PSQBCK1
$TO PSQ
$MEMBA, PSQ1
$MEMBA, PSQ2
$$PY
$MEMBC
$$PN
$MEMBD
$$
/*
```

The remarks relating to example 2.a) are also valid here for the names. No PSQ numbers are specified as these numbers are generated by the PSQ system. Notice the PSQ names are always written before the member name, when these two names are different.

## c) Updating a back-up data set

This example shows how to update the back-up file used in the previous example by replacing MEMBA and MEMBC with copies of two new PSQ files 501 PSQ5 and 502 PSQ6 and by adding a new member 503 PSQ7 with member name MEMBE. The updated PDS is unloaded to a tape with serial number EU998 and the unloaded PDS is given the name PSQBCK2.

```
$FROM TAPE
$FROM=TP9=EU9999,PDS=PSQBCK1
$FROM PSQ
$0501PSQ5,MEMBA
$0502PSQ6,MEMBC
$0503PSQ7,MEMBE
$$
$TO TAPE
$TO=TP9=EU9999,RENAME=PSQBCK2
/*
```

It has been assumed in all examples, that the data sets on tape are the first data sets on the tape. Following the format conventions for the utility programs, it is possible to use data sets which are not the first on a tape. If

the PDS PSQBCK1 was the third data set on the tape EU9999 the statements in example b) and c) should be:

```
$FROM=TP9=(EU9999,3),PDS=PSQBCK1
```

and in example a)

```
$TO=TP9=(EU9999,3),RENAME=PSQBCK1
```

## The JCL Cards

The JCL cards in the instream procedure are listed below (the name and the location of the ECDIN program library is changed).

```
//BCKP PROC NM=XXX,PLIB='SYS1.LIB',PVOL=USER,
//          PVI='SER=EURSY1',PV2='SER=EURSY2',PV3='SER=SPJOL1'
//STP1 EXEC PGM=IEFBR14
//DD1 DD DSN=&CFLNM,DISP=(NEW,PASS,DELETE),UNIT=3330,
//          VOL=&PV1,SPACE=(80,(100,5,1)),
//          DCB=(RECFM=F,LRECL=80,BLKSIZE=80)
//DD3 DD DSN=&INTNM,DISP=(NEW,PASS,DELETE),UNIT=3330,
//          VOL=&PV1,SPACE=(12000,(40,5,2)),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=12000)
//GO EXEC PGM=BCKP,PARM='&NM'
//STEPLIB DD DSN=&PLIB,DISP=SHR,VOL=SER=&PVOL,UNIT=3330
//          DD DSN=SYS1.ANSLIB,DISP=SHR
//          DD DSN=SYS1.LIBUTIL,DISP=SHR,UNIT=3330,VOL=SER=USER04
//JLDTAP DD DDNAME=TAPIN
//NEWTAP DD DDNAME=TAPOUT
//SYSUT1 DD DISP=JLD,UNIT=3330,VOL=&PV3
//SYSUT2 DD DSN=WKSET&NM,DISP=(NEW,DELETE,DELETE),UNIT=3330,
//          VOL=&PV2,SPACE=(12000,(600,20,10)),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=12000)
//PRINT DD SYSJUT=A
//SYSIN DD DSN=&CFLNM(MMM),DISP=(OLD,DELETE),VOL=&PV1,UNIT=3330
//CAKDS DD DSN=&CFLNM(MMM),DISP=(OLD,DELETE),VOL=&PV1,UNIT=3330
//PSQIN DD DSN=&INTNM,DISP=(OLD,DELETE),VOL=&PV1,UNIT=3330
//FTJIFOJ1 DD DSN=&INTNM(MMM),DISP=(OLD,DELETE),VOL=&PV1,
//          UNIT=3330
//SYSUT4 DD UNIT=3330,VOL=&PV1,SPACE=(256,(10,10))
//PSJDISK1 DD UNIT=3330,VOL=SER=USER02,DSN=SYS1.PSQ1,DISP=SHR
//PSJDISK2 DD UNIT=3330,VOL=SER=USER03,DSN=SYS1.PSQ2,DISP=SHR
//PSJDISK3 DD UNIT=3330,VOL=SER=USER04,DSN=SYS1.PSQ3,DISP=SHR
//PSJDISK4 DD UNIT=3330,VOL=SER=USER05,DSN=SYS1.PSQ4,DISP=SHR
//SYSPRINT DD SYSJUT=A
//PKINTSK DD SYSJUT=A,DCB=(RECFM=FA,LRECL=133,BLKSIZE=133)
//PRINTCMD DD SYSOUT=A,DCB=(RECFM=FA,LRECL=133,BLKSIZE=133)
//SYSDBJUT DD SYSJUT=A
//KURT DD DDNAME=SYSIN
//SYSDDUMP DD SYSJUT=A
//BCKPEND PEND
```

The parameter NM is introduced to ensure a unique file name for the file with the DSN=WKSET by adding extra characters to this name, and thereby allows the procedure to be executed by several users at the same time. The value of the parameters should be a combination of up to three characters allowed in DSN's. The JCL cards required for an execution of the procedure are:

```
// EXEC BCKP,NM=XXX  
//GO.TAPIN DD  
//GO.TAPOUT DD  
//GO.SYSIN DD *
```

(XXX being any allowed character combination)

followed by the control statements discussed above. TAPIN is a magnetic tape containing an unloaded PDS and TAPOUT is a magnetic tape to which the generated/updated PDS is unloaded. The TAPIN DD statement is only required when a \$FROM TAPE card is present among the control statements, and the TAPOUT DD statement when a \$TO TAPE card is present. Both DD cards must, when present, contain the DSN of the unloaded PDS, and the DCB parameters:

```
DCB=(RECFM=FB, LRECL=80, BLKSIZE=800)
```

Whenever a data set is unloaded to magnetic tape the IEHMOVE returns the condition code 04; this does not mean that the job has failed, only that the PDS is unloaded.

The procedure is written for placing the back-up on magnetic tape, but it is also possible to use the procedure to generate and read PDS on disc. The DD card for TAPIN and TAPOUT shall then contain the specification of the disc used and the control cards used for the generation of control statements for IEHMOVE should instead of T0=TP9= and FROM=TP9= contain T0=3330= and FROM=3330=. The T0 TAPE and FROM TAPE cards remain unchanged.

If the program in a back-up run tries to open a non existing PSQ file, the PSQB program will print a warning, but the main program will not detect the error, and the utility program will copy the last opened PSQ file (if any) once more to the unloaded PDS with the name of the non-existing file.

The program prints a list of 'accepted command cards'. These cards have passed a format check, but if the program goes to ABEND for other reasons than format error, the last card listed as accepted will normally be the one which has introduced the error.

The procedure is now catalogued in the procedure library and can be executed by all users of the PSQ system. The catalogued procedure differs from the shown instream procedure only in the location of the load module BCKP and in the omission of the SYSUDUMP DD card.

Due to a system error step GO always terminates with an error code C03. Also since the change of system in November 76 an OC5 (addressing exception) error has occurred from time to time in the GO step, caused by an addressing error during the execution of the IEHMOVE module IEHMOVXSE. The C03 error does not affect the execution, whereas the OC5 error brings the execution to abnormal termination. The response to an OC5 error is to re-enter the same job again.

The ECDIN group does not guarantee any maintenance or development of the procedure and the included programs. However I would like to hear about any problems encountered during the execution of the procedure (building 2, tel. 330), and these extensions of the procedure, which might be developed to satisfy our own requirements will, when they have been tested by the group, be made part of the catalogued procedure.

The procedure can be executed in a class 2 job.

---

### **Note to the Users**

As a consequence of the change from OS/MFT to OS/MVT, the IEC209 message which provides error statistics by volume for tape volumes does no more appear in the HASP SYSTEM LOG. This fact could not be avoided for technical reasons. In the meantime a solution was worked out to provide users who need it with the information contained in this message. Interested users are asked to contact D. Koenig, tel. 742, to discuss their requirements concerning this problem.

---

## **Visitors**

The already established good contacts of the Informatics Department to outside research centers were deepened by the visits of two guests:

- Professor Peter Kirstein (Professor of computer systems at the University College, London) presented in a stimulating lecture the "UK Experiences with the ARPA Computer Network".
- Doctor Leland H. Williams (president of the Triangle University Computing Centre, North Carolina and at present visiting professor at the Edinburgh Regional Computing Centre) gave an interesting lecture on the "Management of Computing Services for Network Delivery".



Les personnes intéressées et désireuses de recevoir régulièrement "Computing Centre Newsletter" sont priées de remplir le bulletin suivant et de l'envoyer à

**Mme A. Cambon**  
**Support to Computing**  
**Bât. 36, Tel. 753**

**Nom** .....

**Adresse** .....

.....

**Tel.** .....



The persons interested in receiving regularly the "Computing Centre Newsletter" are requested to fill out the following form and to send it to:

**Mrs. A. Cambon**  
**Support to Computing**  
**Building 36, Tel. 753**

**Nom** .....

**Address** .....

.....

**Tel.** .....

