X-613-70-87

NASA TM X-63861

A METHOD OF HANDLING MULTIPLE FILE TAPES WITH OS/360

JACQUES PACQUET

MARCH 1970

GSFC

GODDARD SPACE FLIGHT CENTER
GREENBELT, MARYLAND

X-613-70-87

A METHOD OF HANDLING MULTIPLE FILE TAPES WITH OS/360

Jacques Pacquet*
Laboratory for Optical Astronomy

March 1970

GODDARD SPACE FLIGHT CENTER
Greenbelt, Maryland

*Service d'Aeronomie, Centre National de la Recherche Scientifique, Fort de Verrieres (Seine et Oise) FRANCE

ABSTRACT

A set of general routines to handle tapes on computers using OS/360 is described. More specifically, a method to read a multi-reel multifile "data set" on unlabeled tape and to generate on unlabeled or labeled tapes one or more multi-reel multifile "data sets" is described using the queued sequential access method of the operating system. The software used consists of a set of assembly language routines, callable from FORTRAN IV. A discussion of their construction and their potential use is given.

CONTENTS

CONTENTS (Continued)

A METHOD OF HANDLING MULTIPLE FILE TAPES WITH OS/360

INTRODUCTION

Before entering into the subject of these pages, we will explain the reasons which led to the writing of these routines.

The problem posed was one of reduction of data from experiments flown aboard the OGO satellites. The information given by the experiment was telemetered from the satellite to the ground station. After separation of the data on a computer, the experimenter receives his information on (decommutated) data tapes. These tapes are 7 track, odd parity, 556 bpi density tapes generated on a Univac 1108 computer. As many as 1500 tapes, each with 10 or more files, could be received.

For different reasons, not pertaining to this discussion, a phase of "reduction" of the data is necessary. The reduction consists in reading these decommutated data tapes (a multi-reel data set on multifile unlabeled tapes) and in writing out more meaningful data tapes.

For an experiment, the output tapes are of two classes: One class will contain all the data consisting of measurements of the physical phenomena with the proper time and experiment status; the second class consists of housekeeping data (attitude error signals, inflight calibration data, etc.). To avoid breaking the structure of the input data set, (each file corresponds to data pertaining to the same pass, same edit tape, etc.,) at this point of the reduction one input file of data will generate one output file for each class. So two multi-reel multifile data sets are to be generated. Due to the complexity and the many logical functions accomplished by the reduction program, it seemed easier to write it in FORTRAN IV language.

These functions cannot be accomplished using FORTRAN IV alone. To FORTRAN the input looks like a specific number of data sets, for each of which a DD statement must be assigned. While for the input it may be possible to compute the number of files and punch accordingly the specified number of DD cards, it is generally impossible to compute the number of output files, as this is dependent on the data processed by the program. For example, an input file may only relate to a time when the experiment was off, therefore no output files will be generated. Not knowing how to easily solve this problem in FORTRAN, it was decided to write appropriate routines in assembly language callable from FORTRAN.

Functions to be Accomplished

To accomplish this goal, the software had to accomplish the following basic functions:

1. Read input and write output tapes (7 and 9 track).

2. Write output tapes with the proper format.

3. Read and write on the same tape unit (the INOUT option does not exist for QSAM).

4. Dismount any volume during the run of the programs.

5. Keep track of mounted volumes and of their position.

6. Generate proper messages to operators for manual interventions.

7. Position input tapes to selected file.

8. Add data on the end of previously written tapes.

9. Allow initial input tapes to be dismounted and output tapes mounted on the same drives.

10. Allow alternate use of two drives for input.

OPERATING SYSTEM 360 FUNCTIONS

Though it is not our purpose to give a course in how the OS/360 processes sequential data sets on tape using the Queued Sequential Access Method (QSAM), we will try to explain from the user's point of view the different phases of the processing.

A. Definitions

Data Set:

For the problem previously defined, one might consider a data set to be one file on an input or output tape.

DCB:

To process a data set, a block of information must be constructed called a Data Control Block using the macro instruction DCB. Using the parameters specified in the DCB macro instruction, the system constructs the Data Control Block. In this DCB you specify characteristics of your data set and of its organization, number of buffers used, etc.

DD Statement:

Besides the DCB, a DD statement must also be supplied which specifies, at execution time, information about the data set omitted in the DCB. In particular, you specify in this DD statement the characteristics of the I/O device. This DD statement is referred to by its DDNAME and is stored on a direct access device in a Job File Control Block (JFCB).

UCB:

With each I/O device is associated a block of information called the Unit Control Block. In this block is found, when the tape is on the drive and positioned, its volume serial number (specified in the DD statement), and at what file it is positioned.

B. Processing of the Data Set

Open Phase:

Before the processing starts, the system must send a mount message to the operator, position it, and construct the necessary block to communicate with the channel. All of these functions are accomplished by opening the data set, using the macro instruction OPEN.

OPEN	(DCB ADDRESS, [(options)], ...)

When using this macro instruction, one specifies which data set is to be processed, and the kind of operations to be accomplished (input or output). Basically, there are three different steps accomplished by the system.

1. Through the DCB address the system gets the data control block from which, through the DDNAME, it gets the DD Statement. (The DDNAME is a parameter of both the DCB and DD Statement.) Using the information from the DD Statement, it completes the Data Control Block.

2. Having obtained the volume serial number on which the data set resides from the DD Statement, the system searches the UCB's allocated to the data set to see if the volume is mounted. If not, and a mount message has not yet been issued, it will signify to the operator to mount the volume.

3. Once the volume is mounted, it positions the tape by comparing the data set sequence number specified in the DD Statement and the one specified in the UCB.

Processing Phase:

Using the macro instructions GET and PUT, information is transmitted between the I/O Buffers and work areas in memory:

GET	DCB address,	[(area address)]
PUT	DCB address,	[(area address)]

Under the QSAM, using this macro instruction, all the functions to read or to write information (construction of the channel program, transmission of information, errors recovery procedures, etc.), are accomplished by the system.

Ending Phase:

Once all the I/O operations have ended, the Data Control Block must be closed. During this phase the system will free the storage used by the blocks used at open time, empty the buffer, and inform the operator of the final disposition of the volume on which the data set resides. For this the CLOSE macro instruction is used.

CLOSE	(DCB address, [(options)] . . .)

C. Restrictions

Basically, one might separate the processing of a data set into three logical steps - the opening phase, the processing phase and the closing phase. Besides this, there are two important restrictions of the OS/360 concerning the processing of data sets on unlabeled tapes:

1. For each data set a DD Statement must be supplied. (This means a DD Statement for each physical file on the input tapes.)

2. When an end of file is reached on an input tape, the Data Control Block for the specific data set must be closed.

CONSTRUCTION OF THE SOFTWARE

A. Choice of the Routines

We had in mind to use the capabilities of the operating system as much as possible. This led to the choice of processing the data sets under the Queued Sequential Access Method (QSAM). Besides the automatic handling of buffers by the system, an access method relieves the programmer of remembering the position of the tape, constructing the channel program, etc.

In order to handle many data sets, the routines had to be general. In consideration of the functions needed to process a data set, reviewed in the previous section, the following routines, callable from FORTRAN, were constructed:

OPEN

CLOSE

FREAD

FWRITE

UTILITY

All of these routines are the QSAM macro instructions. Specifically, the OPEN and CLOSE routines use the macro of the same name. Their purpose is to communicate from FORTRAN IV to the operating system through assembler language written code. Using this routine, one is able to read or write any tape selected, provided he has described the physical configuration of his program in the UTILITY routine.

B. Functions of the Routines

OPEN Routine:

Open the necessary DCB's to process the data sets—one DCB is opened at each call. Allow the program to supply only one DD Statement for each multiple data set. Force the system to issue a mount message when necessary.

Make possible reading and writing on the same tape drive (under QSAM) using either locate mode or move mode.

Make the system position tapes to the proper files; in particular allow file (n-1) to be read after file (n).

Give the capability of alternating between two tape drives when reading tapes.

Return a code specifying if the opening of the DCB was successful or not.

CLOSE Routine:

Close the necessary DCB's opened to process the data sets. One DCB is closed at each call.

Force the system to issue messages to the operator for the final disposition of the volumes.

Dismount any volumes specified if necessary. The option of leaving the volume can also be specified.

Return a code specifying if the closing of the DCB was successful or not.

FREAD Routine:

Transmit information from the I/O buffer to work area in core. The data set must have been opened.

Use either locate or move mode as specified at opening time.

Return a code specifying whether a successful read occurred, a tape mark was read, or a read error occurred.

FWRITE Routine:

Transmit information to the I/O buffer from the work area in core. The data set must have been opened.

Use either locate mode or move mode as specified at opening time.

Return a code if the writing was successful.

UTILITY Routine:

Provide the linkage with the FORTRAN Program.

Set up the save areas each time one of the preceding routines is called.

Make sure the data set referred to exists.

Contain the DCB of the data sets associated with the program.

Contain the job file control blocks for the DD Statements associated with the data sets.

Basically, this software accomplished the functions required. Now the use of the operating system to accomplish these functions will be described.

C. Description of the Software

OPEN Routine:

a. FORTRAN calling sequence:

CALL OPEN (DD ADDRESS, VOLUME SER, FILE, MODE, RETURN CODE)

Where:

DD ADDRESS is an array of 8 bytes containing the EBCDIC name of the DD Statement allocating the drive on which the volume containing the data set is or will be mounted.

VOLUME SER is an array of 8 bytes containing the EBCDIC volume serial number of the volume on which resides the data sets. If zero is specified, the volume serial used in the DD Statement will be used.

FILE is the file to which the volume must be positioned so the data set can be processed. If zero is specified, the file sequence number specified in the job file control block will be used.

MODE is a parameter which can take these values (0, 1, 2, 3,).

0 means open for input, locate mode in use

1 means open for input, move mode in use

2 means open for output, locate mode in use

3 means open for output, move mode in use

RETURN CODE is a parameter which can take the values (0, 1)

0 means the data set exists and its DCB has been opened

1 means the data set does not exist and nothing has been done

b. Functions of the parameters:

DD ADDRESS

To process a data set, a DD Statement must be supplied. This DD Statement identifies the volume on which the data set resides and the tape unit on which the volume is mounted. (More than one volume and more than one tape drive might be allocated.) This DD Statement has a DDNAME. Each time reference is made to this DDNAME, one is, in fact, referring to the volume mounted on the I/O device allocated by the DD Statement. This is why this DDNAME is used as a means of communication between the FORTRAN program and the assembly language routines, for one cannot access directly the DCB to be opened.

For each data set processed by this software, one must supply, to each of the routines, the names of the DD Statement allocating the I/O device so the UTILITY routine can identify the data set to which this call is applied. This name (DDNAME) must already exist in the UTILITY routine. In general, each DDNAME points to a DCB. Once the DCB ADDRESS is found, the software can open the DCB.

In the case of a data set residing on two tape drives, one being active at a time, two DDNAMES (of the two DD Statements allocating the tape drives), point to the same DCB.

VOLUME SER AND FILE

One of the restrictions of the O/S is that one has to supply a DD Statement for each data set (file). As the number of output files is assigned dynamically during the run of the program and unknown at the beginning of the run, one has either to generate dynamically the necessary DD Statements or modify a single one. For obvious reasons, the second solution was adopted. It has been seen that the merging of the DD Statement and the DCB happens at open time. Each time a tape mark is read, the DCB has to be closed. To process the next file, it has to be reopened. Suppose we could "update" the DD Statement - the problem would then be solved because the merging would occur with a new DD Statement.

There is a special type of OPEN macro instruction to open the DCB called the type J OPEN macro instruction. By using this type of OPEN, you specify to the OS/360 that you will supply the Job File Control Block which includes the DD Statement.

Thus, when the Data Control Block is closed, the JFCB can be modified (in other words, the DD Statement).

Two particular fields in the JFCB of interest for our purpose are:

name	length	location
JFCBFLSQ	2 bytes	Byte 68
JFCBVOLS	30 bytes	Byte 122

The JFCBFLSQ identifies the data set sequence number of the volume (the file on our tape), and the JFCBVOLS the serial number of the volume on which the data set resides. So, every time one wants to process another file on a tape, one modifies the file parameter and the system will position the tape as before. This allows processing the files on the tapes in the order desired. When one wishes to change volumes, he provides another volume serial number as the volume serial parameter. Provided the tape is unloaded, the system will send a message to the operator to mount the next volume and update the necessary blocks to keep track of this new volume.

Only one volume serial number must be specified in the DD Statement because if a tape mark is hit while reading the tape, the system would not give back control to the problem program, but rather try to process the next volume. This is why just one volume serial number should be specified in the job file control block. To switch tapes, one must be careful with the bypass label processing operation (BLP). The tape must first be unloaded, otherwise, the system will use the tape already on the drive. It will, however, issue the mount message for the right volume.

As long as tapes are processed on the same drive, the JFCB need only to be read once. However, in the case of alternation between two allocated drives, the program will read the JFCB every time a switch of drive occurs. This is because one JFCB area and one DCB is used for the two drives.

MODE

One last function to accomplish is to read or write on the same drive. Under QSAM there is no INOUT option as provided by the Basic Sequential Access Method (BSAM). However, there is a parameter of the DCB called the MACRF parameter which specifies the macro instruction and the facilities to be used to process the data set. Of all the options available only the following were kept:

GL means GET	LOCATE	MODE
GM means GET	MOVE	MODE
PL means PUT	LOCATE	MODE
PM means PUT	MOVE	MODE

The locate mode allows one to process within the buffer of the machine. The move mode allows one, in writing, to transfer a FORTRAN built buffer to the I/O buffer of the computer, or to transfer information in a FORTRAN accessible buffer. The system knows what is used by the bit setting on the foundation of the data control block. These bits are set by the MACRF parameter in the DCB. Therefore, by just specifying at the opening time which macro type to use, one can read or write using locate or move mode on any tape drive.

RETURN CODE

This return code merely states whether the proper DDNAME exists in the control program.

CLOSE Routine:

a. FORTRAN Calling sequence:

```
CALL      CLOSE      (DD ADDRESS, OPTION, RETURN CODE)
```

Where:

DD ADDRESS

Same as for OPEN

OPTION

Is a parameter which can take two values (0, 1)

0 means close with leave option specified

1 means close without option specified

RETURN CODE

Same as for OPEN

b. Functions of the Parameters:

DD ADDRESS and RETURN CODE

Same as for OPEN

OPTION

Every time a tape mark is read or written, the Data Control Block associated with the data set must be closed. Two options are available. Should the next file have to be processed, the volume must remain mounted and positioned. The macro instruction CLOSE has the capability of specifying options.

The leave option positions the current volume to the logical end of the data set. In other words, the tape remains positioned, and to process the following file one need only to open the DCB using the OPEN routine, with the same volume serial number and the file number incremented by one.

In case the processing for that volume has ended, and the drive needs to be freed to process the next volume, the tape needs to be unloaded. To accomplish this, it is not necessary to specify any options in the CLOSE macro instruction (you may, however, specify the option DISP), and specify in the disposition parameter in the DD Statement, the KEEP option, e.g., DISP = (OLD, KEEP). In this case, not only does the system unload the tape, but also it informs the operator of the action to be taken concerning the volume.

FREAD Routine:

a. FORTRAN Calling Sequence:

```
CALL  FREAD  (DD ADDRESS, BUF ADDRESS, NUMBER, RETURN CODE)
```

Where:

DD ADDRESS

Same as above

BUF ADDRESS is either a word or an array. It will contain the address of the I/O buffer containing the next record read in the case of locate mode. In move mode, it will contain the record just read, including the control word in case of a variable record (RECFM = V).

NUMBER

A half-word containing the number of full words just read.

RETURN CODE is a parameter which can take the values 0, 1, 2, 3.

0 means normal return, a record has been read.

1 means an end of file has been read while trying to read the record. The DCB must be closed by the user.

2 means an error occurred while reading.

3 the data set does not exist and nothing has been done.

b. Functions of the Parameters:

DD ADDRESS

Same as for OPEN.

BUF ADDRESS

In the case of locate mode, BUF address contains the address of the record just accessed by the GET macro instruction. This buffer cannot be directly accessed by a FORTRAN program and must be processed through an assembly language program.

The move mode is best used when the data are already unpacked and can be used by FORTRAN. In this case, BUF ADDRESS will be the name of an array in the FORTRAN calling program, large enough to accommodate the entire record.

If the record format specified for the data set is undefined or fixed, the buffer contains only the data. In the case of variable (blocked or unblocked) record format, the first word will contain the number of bytes contained in the logical record just accessed. This is a control word generated at the time the record was written on the volume.

NUMBER

This parameter will contain the number of full words just read. This parameter is taken from the DCB record length parameter. In case of fixed or undefined record formats, it will be the exact length of the record. In case of variable records, it will be the length of the record, including the control word at the beginning.

FWRITE Routine:

a. FORTRAN calling sequence:

CALL FWRITE (DD ADDRESS, BUF ADDRESS, NUMBER, RETURN CODE)

Where:

DD ADDRESS

Same as OPEN.

BUF ADDRESS

Will contain the address of the I/O Buffer where the record to be written can be packed for locate mode. In the move mode it will contain the record to write, including the control word for variable record format.

NUMBER

A half word containing the number of words to be written.

RETURN CODE

0 - normal return. The record has been moved to the I/O Buffer.

2 - error while writing (not used because the system will ABEND the task).

3 - the data set does not exist and nothing has been done.

b. Function of the Parameters:

DD ADDRESS

Same as for OPEN.

BUF ADDRESS

Same as for FREAD routine. It should be pointed out here that for the locate mode the first call to the routine provides the address of the buffer where the record can be constructed. The last record is written when the DCB is closed.

Undefined, fixed or variable record formats can be written as specified in the DD Statement for the Data set. In the case of variable record, the first word of the logical record to be written must not be used. The routine uses it to generate the control word.

NUMBER

This parameter contains the number of words to be written. In the case of variable records, the control word must be included in the count.

RETURN CODE

Only the normal return and the not found data set return can be used. In the case of a write error, the system ABENDS the job. In the case of an end of volume, the system will automatically ask for the next volume to be mounted.

UTILITY Routine:

This routine is a control section which is used by the preceding subroutines. It cannot be accessed by FORTRAN directly. Its main use, besides setting of save areas and restoring registers, consists of the linkage with the FORTRAN program and the setting of the Return Codes.

Basically, its contents can be described simply as:

a. One Table of DDNAME(S)—This table of double words (one for each DD NAME) must contain all the DD names and the DD Statements allocating the I/O devices for the data sets to be processed by this software.

Example:

Suppose a program to read one input tape and write out two output tapes using this software:

Input	Name of DD Statement	DTPIN (//GO·DTPIN DD)
Output	Name of DD Statement	DTPOUT1 (//GO·DTPOUT1)
		DTPOUT2 (//GO·DTPOUT2)

The UTILITY routine must contain the following table:

```
DDNMTBLE   DS   OCL
           DC   CL8'  DTPIN'
                CL8'  DTPOUT1'
                CL8'  DTPOUT2'
```

b. One Table of DCB Address--The DD Name table is just a means to communicate with the assembly package. The purpose is to reach the Data Control Block Address. Therefore, a table of DCB address corresponding to each data set must be created.

Preceding example:

Input	Names of DCB	TPIN
Output	Names of DCB	TPOUT1
		TPOUT2

The UTILITY routine must contain the following table:

```
DCBTABLE   DS   OF
           DC   A(TPIN)
           DC   A(TPOUT1)
           DC   A(TPOUT2)
```

The order must be the same as the one in the DD Name table.

c. One Table of EXIT LIST Address—These EXIT LIST addresses are the ones specified in the EXIT LIST for each DCB so the job file control block can be read in using the RDJFCB macro instruction.

Preceding example:

Suppose the EXLST parameter specifies

For the DCB	TPIN	EXLST = LIST1
For the DCB	TPOUT1	EXLST = LIST2
For the DCB	TPOUT3	EXLST = LIST3

The UTILITY routine must contain the following table:

```
ADDLIST    DS   OF
LIST1      DC   X'87',AL3(JFIN)
LIST2      DC   X'87',AL3(JFOUT1)
LIST3      DC   X'87',AL3(JFOUT2)
```

Where JFIN, JFOUT1 and JFOUT2 are areas of 176 characters long aligned on a double word boundary.

d. The DATA CONTROL BLOCK—For each data set active at one time, a DCB must be supplied. The parameter specified for this DCB depends on the programmer. However, the EXIT LIST address must be specified and match the one specified in the EXIT LIST table. The EODAD parameter must specify EODAD = EOF and the SYNAD parameter SYNAD = ERR if they are to be used.

EOF and ERR are addresses in the UTILITY routines where the Control program branches after gaining control from the system in case of an error or a tape mark encountered while reading.

Preceding example:

Here is an example of a DCB macro instruction:

```
TPIN    DCB    DSORG=PS,MACRF=(GL),
                DDNAME=DTPIN,DEVD=TA,
                BUFNO=2,BFALN=D,EROPT=ACC,
                EODAD=EOF,SYNAD=ERR,
                EXLST=LIST1,BFTEK=S.
```

The MACRF parameter and the DD Name parameter are set up by the OPEN ROUTINE.

e. The JFCB AREAS—For each data set, an area of 176 characters long must be supplied. The RDJFCB macro instruction will read the job file control block for the data set whose DCB is specified as the parameter of the RDJFCB macro instruction. It will read in core in the area specified by the EXIT LIST of the DCB.

Preceding example:

The JFCB DTPIN will be read in JFIN
DTPOUT1 will be read in JFOUT1
DTPOUT2 will be read in JFOUT2

The UTILITY must contain:

```
            DS    OD
JFIN        DS    176C
JFOUT1      DS    176C
JFOUT2      DS    176C
```

f. The EXIT LIST—Two EXIT LISTs addresses are specified.

EOF when a tape mark is read on a tape.

ERR when an error occurs while reading the tape.

If the user wants to use the return code specified in the FREAD routine, in case of a tape mark or an error while reading he must first specify in the DCB for the concerned data set

EROPT = ACC
EODAD = EOF
SYNAD = ERR

In this case the RETURN CODE will be properly set up and the problem program will regain control.

This parameter, though specified in the DCB, will not be used if an end of volume is encountered while writing a tape or an error occurs. If an error occurs, the system will abend the task. If an end of volume occurs, the system will request mounting of the next volume automatically, the FWRITE routine will regain control when the Buffer is to be written out on the new volume. A normal return will be specified in this case.

POTENTIAL USE

Let us recall that this software has been written for the purpose of handling multifile multi-reel data sets. It might not be suitable to use it for other applications. To be able to use this software, one must first set up the UTILITY program. It would be wasteful in time if that part of the software had to be reprogrammed for each program. Though a little wasteful in core, it is better to set it up such that it will suit every requirement. For this one needs to know the maximum number of data sets to be handled using these routines - once decided, the following actions will have to be accomplished:

1. Set the parameter of the loop for the DDNAME checks. If it is the number of tape units you want to handle (it might be greater than the number of data sets), this parameter is equal to (n-1) x 8.

2. Set the table of DDNAMEs, each eight characters long (may be padded by blanks).

3. Set up the Data Control Blocks. The amount of information specified there depends on the programmer. However, let us recall that only the PUT and GET macro instructions are supported for both Locate and Move mode. Also, if the Return Code of the FREAD routine is to be used, EROPT=ACC, EODAD=EOF and SYNAD=ERR must be specified. At least one DCB must be supplied per data set.

4. Set the table of DCB address. This table must have n entries, some entries may point to the same DCB.

5. Set the table of EXIT LIST entries, some pointing to the same job file control area. Each EXIT LIST must have its name specified in the Data Control Block using this EXLST parameter.

6. Set the job file control block areas. These areas must be of 176 characters long, aligned on a double word boundary.

Once all the information is specified in the UTILITY routine, the software is ready to be used.

An example will now be given to show the possibilities and the use of the software.

EXAMPLE

Suppose one has a data reduction task which must handle many tapes in various programs. We will assume that the maximum number of data sets to be handled within a job is four, for the main data reduction programs. This program consists in reading a batch of tapes and writing out information on three different output data tapes. Two drives are to be assigned for the input to be able to alternate between the two.

Here one has four data sets: one input, three output data sets and five drives: two for input, three for output.

All the remaining program will fit within this configuration, so one is now ready to set the UTILITY routine.

A. Setting of the UTILITY Routine

1. Parameter of the address loop n = 32

2. DDNAME table.

For example, we will assume the following DDNAMEs:

```
DDNTBLE   DS    OCL8
          DC    CL8'DDECOM1'
                CL8'DDECOM2'
                CL8'DTPOUT'
                CL8'DORBIT'
                CL8'DDIRECT'
```

For each of these DDNAMEs corresponds a DD Statement.

INPUT TAPES:

```
//GO·DDECOM1      DD . . . .
//GO·DDECOM2      DD . . . .
```

OUTPUT TAPES:

```
//GO·DTPOUT       DD . . . .
//GO·DORBIT       DD . . . .
//GO·DDIRECT      DD . . . .
```

REMARKS:

Assigning certain drives for input or output is made at opening time. You could, for example, assign DTPOUT for input and alternate between two output tapes DDECOM1 and DDECOM2.

3. DATA CONTROL BLOCKs:

One detailed example of a typical DCB will be given and only a skeleton (with the most necessary information) will be given for the others.

```
ORBIT     DCB    DSORG=PS,MACRF=(PM),DDNAME=DORBIT,
                 DEVD=TA,BUFNO=2,BFALN=D,EROPT=ACC,
                 EODAD=EOF,SYNAD=ERR,EXLST=LIST3,
                 BFTEK=S.

TPOUT     DCB    . . . . . ., DDNAME=DTPOUT. . .EXLST=LIST2,
                 EROPT=ACC,EODAD=EOF,SYNAD=ERR
```

```
DIRECT   DCB   . . . . . ., DDNAME=DDIRECT. . .EXLST=LIST4,
                EROPT=ACC,EODAD=EOF,SYNAD=ERR

DECOM    DCB   . . . . . ., DDNAME=DDECOM1. . .EXLST=LIST1,
                EROPT=ACC,EODAD=EOF,SYNAD=ERR
```

REMARK:

In fact, the DDNAME parameter and the MACRF parameter are set at opening time.

4. DATA CONTROL BLOCK address table:

```
DCBTABLE   DS   OF
           DC   A(DECOM)
           DC   A(DECOM)
           DC   A(TPOUT)
           DC   A(ORBIT)
           DC   A(DIRECT)
```

One sees that the first two entries point to the same DCB, two tape drives being assigned at one data set.

5. EXIT LISTS table:

```
ADDLIST   DS   OF
LIST1     DC   X'87',AL3(JFDECOM)
          DC   X'87',AL3(JFDECOM)
LIST2     DC   X'87',AL3(JETPOUT)
LIST3     DC   X'87',AL3(JFORBIT)
LIST4     DC   X'87',AL3(JFDIRECT)
```

6. JOB FILE CONTROL BLOCK area:

```
            DS   OD
JFDECOM     DS   CL176
JETPOUT     DS   CL176
JFORBIT     DS   CL176
JFDIRECT    DS   CL176
```

With this information specified in the UTILITY routine, the software is now ready to be used.

B. Application

1. Definition:

We will take, as example, the main data reduction job. The input tapes will be processed using the locate mode and the output tapes using the move mode. We will assume the following routines written: SUBROUTINE UNPAK (ADD, L, OUT, N), whose function is to unpack the data in the buffer of length L and address specified in the ADD parameter and return unpacked data in the array OUT of length N. SUBROUTINES BUF1, BUF2, BUF3, (OUT, N, BUF, M.)

Function:

To generate output Buffers respectively for output tapes 1, 2 and 3. Information from the last call to the UNPAK routine is packed in an array called BUF of length M.

2. FORTRAN Program:

```
      INTEGER*2 NOB, M

      DIMENSION DDEOM(4), DTPOUT(2), DORBIT(2), DIRECT(2),

     1OUT(1000), BUF(1000), NFILE(20), OLSER(2),
     2OUT 1(2), OUT2(2), OUT3(2)

     DATA DDECOM/'DDECOM1 DDECOM2'/,DTPOUT/'DTPOUT'/,
     1DDORBIT/'DORBIT '/, DIRECT/'DDIRECT'/

C
C  IND IS THE POINTER USED FOR ALTERNATING BETWEEN
C  TAPE DRIVES.
C
           IND = 3
           K   = 0

C
C  READ VOLUME-SER OF OUTPUT TAPES.
C
           READ (5,2) OUT1, OUT2, OUT3
```

```
C
C READ VOL-SER OF INPUT TAPES, NUMBER OF FILES
C AND LIST
C NUMBERS

   26 READ (5,1, END = 100) OLSER, N, (NFILE(J), J=1, N)
      IF (IND.EQ.3) GO TO 10
      GO TO 11
   10 IND = 1
   11 CONTINUE

C
C LOOP TO PROCESS N FILES ON 1 TAPE
C

      DO  12 J = 1, N

C
C OPEN INPUT TAPE FOR LOCATE MODE
C

      CALL OPEN (DDECOM(IND), OLSER, N FILE(J), O, NC)

C
C OPEN OUTPUT TAPE 1 FOR MOVE MODE
C

      CALL OPEN (DTPOUT, OUT1, (K+J),3,NC)

C
C OPEN OUTPUT TAPE 2 FOR MOVE MODE
C

      CALL OPEN (DIRECT, OUT2,(K+J),3,NC)

C
C OPEN OUTPUT TAPE 3
C

      CALL OPEN (DORBIT, OUT3,(K+J),3,NC)
```

```
C
C  READ INPUT TAPE
C
  23 CALL FREAD(DDECOM(IND),ADD1,NOB,NC)
  GO TO (21,22) NC

C
C  UNPACK DATA
C
      CALL UNPACK(ADD1,NOB,OUT,NUMB)

C
C  PREPARE BUFFER FOR OUTPUT TAPE 1.
C
      CALL BUF1(OUT,NUMB,BUF,M)

C
C  WRITE RECORD ON OUTPUT TAPES
C
      CALL FWRITE (DTPOUT,BUF,M,NC)
      CALL BUF2(OUT,NUMB,BUF,M)
      CALL FWRITE(DIRECT,BUF,M,NC)
      CALL BUF3(OUT,NUMB,BUF,M)
      CALL FWRITE (DORBIT,BUF,M,NC)
      GO TO 23

C
C  A READ ERROR - SKIP THE RECORD
C
  22 WRITE (6,3) (list)
  GO TO 23

C
C  END OF ON INPUT TAPE
C
  21 IF(J.EQ.N)
  GO TO 24
```

```
C
C IF N FILES HAVE NOT BEEN READ-CLOSE AND LEAVE
C
        CALL CLOSE (DDECOM(IND),0,NC)
C
C LAST FILE DONE-CLOSE AND KEEP
C
     24 CALL CLOSE (DDECOM (IND),1,NC)
C
C CLOSE OUTPUT DCB's(LEAVE)
C
     25 CALL CLOSE (DTPOUT,0,NC)
        CALL CLOSE (DIRECT,0,NC)
        CALL CLOSE (DORBIT,0,NC)
     12 CONTINUE
C
C UPDATE PARAMETERS POINTING TO OUTPUT FILES
C
        K = K+N
        GO TO 26

   1 FORMAT (2A4, I3,20I2)
   2 FORMAT (6A4)
   3 FORMAT (USERS CHOICE)

 100 . . .
     RETURN
     END
```

CONCLUSION

This method of handling tapes has been proven successful for two experiments flown aboard OGO IV and OGO V. It completely frees the programmer of the burden of writing an assembly language program, which speeds up his work. It is hoped that this work can be as helpful for others as it has been for us.

APPENDIX I

SOURCE CODE LISTING

```
// EXEC FORTRAN
//SOURCE.SYSIN DD *
      DIMENSION NUF(200),BUF(200),NN(2)
      DATA NN/'DORBIT'/
      INTEGER * 2 NOB
      INTEGER * 2 NUB
      DO 4 J=1,100
 4    NUF(J)=J
      NUB=100
      CALL OPEN(NN,0,1,3,NC)
      CALL FWRITE(NN,NUF,NUB,NC)
      CALL FWRITE(NN,NUF,NUB,NC)
      CALL CLOSE(NN,0,NC)
      CALL OPEN(NN,0,1,1,NC)
      DO 3 K=1,2
      CALL FREAD(NN,BUF,NOB,NC)
      WRITE(6,1) NOB,(BUF(I),I=1,NOB)
 3    CONTINUE
 1    FORMAT(I4,(' ',10Z8) )
      STOP
      END
/*
// EXEC ASSEMBLR,PARM='LOAD,DECK'
//SOURCE.SYSPUNCH DD DSNAME=&DECK,SYSOUT=B
//SOURCE.SYSIN DD *
MAST    TITLE 'MASTER CONTROL PROGRAM FOR ''OPEP SYSTEM'' J.PACQUET'
OPEN     START 0
 TITLE 'CONTROL PROGRAM ENTRY NAME - ''OPEN'''
         ENTRY DIRECT
         ENTRY UT1
         ENTRY UT2
         ENTRY DECOM
         ENTRY TPOUT
         ENTRY ORBIT
         ENTRY CLOSE
         ENTRY FWRITE
         ENTRY FREAD
*   *   *                    OPEN ROUTINE                    *   *   *
*                                                                     *
* FORTRAN CALL#    CALL OPEN (ADDRDDNAME,VOLSERIAL,FILSEQ,CODE,RC)
*                                                                     *
*         WHERE, PARM 1 IS THE LOCATION OF THE  D D N A M E .         *
*                   =  2 IS THE LOCATION OF THE SERIAL NUMBERS.       *
*                   =  3 IS THE LOCATION OF THE FILE SEQ NUMBER.      *
*                   =  4 IS    TYPE OF OPEN(0=GL,1=GM,2=PL,3=PM)      *
*                   =  5 IS THE LOCATION FOR THE RETURN CODE.         *
*                                                                     *
*         RETURNS ARE AS FOLLOWS# RC = 0, NORMAL RETURN               *
*                                  = = 1, DDNAME NOT RECOGNIZABLE     *
*                                                                     *
*   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *
```

```
 SPACE 2
MASTRPGM SAVE   (14,12),,*     O P E N  ENTRY
         BALR   11,0
         USING  *,11
         USING  UTILITY,10
         L      10,DATABAS1
         ST     1,SAV1
         BAL    1,SAVTRACE
         BAL    14,DDNMCHCK
*                               GET DCB OFFSET IN GPR-4
         B      *+4(15)         R15 = RC
         B      FOUND
         LA     15,1(0)         DDNAME NOT RECOGNIZABLE
STO1     ST     15,RETURNCD
         B      RETURN1
 SPACE 1
FOUND    LA     2,DCBTABLE(4)           DCB ADDRESS POINTER
         CLI    0(2),X'00'          IS DD IN JFCB AREA
         BNE    DONE                YES
         SR     5,5
         LA     6,4
         LA     7,8
         LA     8,DCBTABLE(4)
LOOP7    LA     2,DCBTABLE(5)
         CLC    1(3,2),1(8)         IS IT GOOD JFCB AREA
         BNE    SAUT                NO
         MVI    0(2),X'00'          DEACTIVATE OLD JFCB
SAUT     BXLE   5,6,LOOP7
         LA     2,DCBTABLE(4)
         MVI    0(2),X'01'          ACTIVATE NEW JFCB AREA
         L      2,DCBTABLE(4)
         L      1,SAV1              PICK UP ADDRESS OF DDNAME
         L      3,0(1)
         MVC    40(8,2),0(3)        MOOVE   DDNAME
         RDJFCB ((2))
DONE     L      5,ADDLIST(4)        PICK UP ADDRESS OF JFCB AREA
         L      2,DCBTABLE(4)
         L      1,SAV1
         L      3,4(1)
         SR     6,6
         A      6,0(3)
         BZ     ZERO1
         MVC    118(6,5),0(3)       MOVE   SERIAL    #
ZERO1    L      3,8(1)
         SR     6,6
         A      6,0(3)
         BZ     ZERO2
         MVC    68(2,5),2(3)        MOVE   FILE   SEQUENCE   #
ZERO2    L      3,12(1)
         L      3,0(3)
         S      3,=F'2'
         BM     INPUT
         NI     50(2),X'01'
         NI     51(2),X'01'
         LTR    3,3
```

```
          BZ     OUTLOC        OPEN FOR OUTPUT LOC MODE
          OI     51(2),X'50'
          B      OUTPUT
OUTLOC    OI     51(2),X'48'
OUTPUT    OPEN   ((2),(OUTPUT)),TYPE=J
          B      RETURN1
INPUT     NI     51(2),X'01'
          NI     50(2),X'01'
          A      3,=F'2'
          BZ     INLOC
          OI     50(2),X'50'
          B      TOOPEN
INLOC     OI     50(2),X'48'
TOOPEN    OPEN   ((2),(INPUT)),TYPE=J
RETURN1   L      1,SAV1
          L      3,16(1)
          MVC    0(4,3),RETURNCD     STORE RETURN CODE
          B      RETURN
*
DATABAS1  DC     A(UTILITY)     ADDRESS OF GENERAL ROUTINES & DATA
*
 TITLE 'CONTROL PROGRAM ENTRY NAME - ''CLOSE'''
*   *   *                      CLOSE ROUTINE                        *   *   *
*                                                                             *
* FORTRAN CALL#     CALL CLOSE (ADDRDDNAME,TYPE,RC)                           *
*                                                                             *
*          WHERE, PARM 1 IS THE LOCATION OF THE  D D N A M E .              *
*                    =  2 IS TYPE OF CLOSE(0=LEAVE,1=KEEP )                 *
*                    =  3 IS THE LOCATION FOR THE RETURN CODE.              *
*                                                                             *
*          RETURNS ARE AS FOLLOWS# RC = 0, NORMAL RETURN                    *
*                                   = = 1, DDNAME NOT RECOGNIZABLE            *
*                                                                             *
*   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *
 SPACE 2
          USING *,11
CLOSE     SAVE   (14,12),,*
          LR     11,15
          L      10,DATABAS2
          ST     1,SAV1
          BAL    1,SAVTRACE
          BAL    14,DDNMCHCK
          B      *+4(15)
          B      IHADFILE       NORMAL RETURN
          LA     15,1           DDNAME NOT RECOGNIZABLE
STOR      ST     15,RETURNCD
          B      RETURN2
  SPACE 1
IHADFILE  L      2,DCBTABLE(4)        PICK ADDRESS OF DCB
          L      6,SAV1
          L      6,4(6)
          L      6,0(6)
          LTR    6,6
          BZ     LEAVE
          CLOSE ((2))                 UNLOAD TAPE
```

```
          B      RETURN2
LEAVE     CLOSE ((2),LEAVE)          LEAVE AND NOT UNLOAD TAPE
RETURN2   L      1,SAV1
          L      1,8(1)
          MVC    0(4,1),RETURNCD     STORE RETURN CODE
          B      RETURN
*
DATABAS2  DC     A(UTILITY)
*
          TITLE 'WRITE ROUTINE'
*                                                                      *
* FORTRAN CALL#     CALL FWRITE(ADDRDDNAME,BUF,NCOUNT,RC)              *
*                                                                      *
*          WHERE, PARM 1 IS THE LOCATION OF THE  D D N A M E .        *
*                   '  2 IS THE ARRAY OF DATA TO WRITE. FOR PM        *
*                   '  2 CONTAIN ADDRES OF NXT BUFFER FOR PL          *
*                   =  3 IS THE NUMBER OF WORDS IN THE ARRAY. PM      *
*                                NCOUNT HALF WORD                      *
*                                                                      *
*          RETURNS ARE AS FOLLOWS# RC = 0, NORMAL RETURN              *
*                                   = = 2 ERROR RETURN                 *
*                                   = = 3, DDNAME NOT RECOGNIZABLE     *
*                                                                      *
*   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *  *
 SPACE 2
          USING *,11
FWRITE    SAVE  (14,12),,*
          LR     11,15
          L      10,DATABAS3
          ST     1,SAV1
          BAL    1,SAVTRACE
          BAL    14,DDNMCHCK
          B      *+4(15)
          B      IHAVFILE       NORMAL RETURN
          LA     15,3
          ST     15,RETURNCD
          B      RETURN3
  SPACE 1
IHAVFILE  L      1,DCBTABLE(4) LOAD ADD OF DCB
          LR     5,1
          L      2,SAV1
          L      2,4(2)
          LR     0,2           LOAD ADD OF BUFFER
          L      2,SAV1
          L      2,8(2)
          LH     2,0(2)
          SLA    2,2
          SR     3,3
          IC     3,36(1)
          SRA    3,6
          S      3,=F'1'
          BZ     VFMAT
          S      3,=F'1'
          BZ     TOPUT
UFMAT     STH    2,82(1)        MOVE NUMBER OF BYTES
```

```
         B      TOPUT
VFMAT    LR     3,0
         STH    2,0(3)
TOPUT    PUT    (1),(0)
         SR     4,4
         IC     4,43(5)
         N      4,=F'16'
         BNZ    APRES
         L      2,SAV1
         L      2,4(2)
         ST     1,0(2)
APRES    B      RETURN3
*
DATABAS3 DC     A(UTILITY)
*
         TITLE 'READ  ROUTINE'
*   *   *                      FREAD  ROUTINE                    *   *   *
*                                                                          *
* FORTRAN CALL#    CALL FREAD (ADDRDDNAME,BUF,NCOUNT,RC)                   *
*         WHERE, PARM 1 IS THE LOCATION OF THE  D D N A M E .              *
*                                                                          *
*                  '  CONTAINS THE ADDRESS OF THE RCORD READ   GL          *
*                                                                          *
*                  '  CONTAINS                 THE RCORD READ   GM       *
*                                                                          *
*                  =  3 IS THE NUMBER OF WORDS IN THE ARRAY.               *
*                                 NCOUNT HALF WORD                         *
*                                                                          *
*         RETURNS ARE AS FOLLOWS# RC = 0, NORMAL RETURN                    *
*                                  = = 1, EOF RETURN                       *
*                                                                          *
*                                  = = 2  ERROR RETURN                     *
*                                                                          *
*                                  = = 3, DDNAME NOT RECOGNIZABLE          *
*                                                                          *
*   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *   *
 SPACE 2
         USING *,11
FREAD    SAVE   (14,12),,*
         LR     11,15
         L      10,DATABAS4
         ST     1,SAV1
         BAL    1,SAVTRACE
         BAL    14,DDNMCHCK
         B      *+4(15)
         B       HAVFILE       NORMAL RETURN
         LA     15,3           DDNAME NOT RECOGNOZIBLE
         ST     15,RETURNCD
         B      RETURN3
  SPACE 1
HAVFILE  L      1,DCBTABLE(4) LOAD ADD OF DCB
         LR     3,1
         L      2,SAV1
         L      2,4(2)
         LR     0,2
```

```
         GET   (1),(0)
         SR    4,4
         IC    4,42(3)
         N     4,=F'16'
         BNZ   AFTER
         L     2,SAV1
         L     2,4(2)
         ST    1,0(2)
AFTER    L     2,SAV1
         L     2,8(2)
         LH    4,82(3)
         SRA   4,2
         STH   4,0(2)
         B     RETURN3
*
DATABAS4 DC    A(UTILITY)
*
 TITLE 'CONTROL PROGRAM ''LIBRARY'' ROUTINES.'
*   *   *   *   *      DDNAME SEARCH ROUTINE      *   *   *   *   *
*
UTILITY  CSECT
*
DDNMCHCK STM   2,6,SAV2
         L     6,SAV1
         L     6,0(6)          DDNAME ADDRESS
         SR    4,4
         LA    3,48          MAX# OF DD'S=7 TEMPORARILY
         LA    2,8
*
LOOP     LA    5,DDNMTBLE(4)
         CLC   0(8,6),0(5)
         BE    *+16
         BXLE  4,2,LOOP
*
         LA    15,4(0)         ERR RETURN
         B     *+10
*
         SR    15,15           NORMAL RETURN
         SRL   4,1             N * 8 / 2
*                              PNTR. IN FILE TABLES(FULL WORD)
         LM    2,3,SAV2
         LM    5,6,SAV2+12
         BR    14              INTER-RETURN
*
*        THIS ROUTINE USES R4 - TO RETURN POINTER TO DCB.
*                           R15- FOR THE RETURN CODE(RC).
*   *   *
*
*                   SETUP FOR SAVE AREA TRACE
*
SAVTRACE ST    13,SAV13+4      HSA
         LA    2,SAV13
         ST    2,8(13)         LSA
         LR    13,2            MY SAVE AREA
         SR    2,2
```

```
         ST    2,RETURNCD     SET FINAL RC INITIALLY TO ZERO
         BR    1              INTER-RETURN
*                                                              . .
*
*   *   *
 SPACE 1
EOF      MVI   RETURNCD+3,X'01'
         B     RETURN3
ERR      MVI   RETURNCD+3,X'02'
         BR    14
RETURN3  L     1,SAV1
         L     1,12(1)
         MVC   0(4,1),RETURNCD
RETURN   L     13,SAV13+4
 RETURN  (14,12)              TO CALLER
 SPACE 2
 TITLE 'CONTROL PROGRAM DATA SECTION.'
*    *    *    *              D   A   T   A              *    *    *
*
SAV13    DS    18F
SAV1     DS    F              MUST NOT BE MOLESTED AFTER ENTRY
SAV2     DS    F
SAV3     DS    F
SAV4     DS    F
SAV5     DS    F
SAV6     DS    F
SAV7     DS    F
SAV8     DS    F
SAV9     DS    F
SAV10    DS    F
 SPACE 2
DCBTABLE DS    0F
         DC    A(DECOM)
         DC    A(DECOM)
         DC    A(TPOUT)
         DC    A(ORBIT)
         DC    A(OUT1)
         DC    A(OUT2)
         DC    A(DIRECT)
*
*
DDNMTBLE DS    0CL8
         DC    CL8'DDECOM1'
         DC    CL8'DDECOM2'
         DC    CL8'DTPOUT'
         DC    CL8'DORBIT'
         DC    CL8'DOUT1'
         DC    CL8'DOUT2'
         DC    CL8'DDIRECT'
*
*
ADDLIST  DS    0F
LIST1    DC    X'87',AL3(JFDECOM)
         DC    X'87',AL3(JFDECOM)
LIST2    DC    X'87',AL3(JFTPOUT)
```

```
LIST3     DC     X'87',AL3(JFORBIT)
LIST4     DC     X'87',AL3(JFUT1)
LIST5     DC     X'87',AL3(JFUT2)
LIST6     DC     X'87',AL3(JFDIRECT)
   SPACE 5
          DS     0F
RETURNCD  DS     F
          DS     0D
JFDECOM    DS    0CL176
JFCBDSNM  DS     44C
JFCBELNM  DS     8C
JFCBTSDM  DS     C
JFCBSYSC  DS     13C
JFCBLTYP  DS     C
          DS     C
JFCBFLSQ  DS     2C
JFCBVLSQ  DS     2C
JFCBMASK  DS     8C
JFCBCRDT  DS     3C
JFCBXPDT  DS     3C
JFCBIND1  DS     C
JFCBIND2  DS     C
JFCBUFNO  DS     C
JFCBFTEK  DS     C
JFCBBUFL  DS     2C
JFCEROPT  DS     C
JFCBDVCH  DS     C
JFCDEN    DS     C
          DS     3C
JFCDSORG  DS     2C
JFCRECFM  DS     C
JFCOPTCD  DS     C
JFCBLKSI  DS     2C
JFCLRECL  DS     2C
JFCNCP    DS     C
          DS     10C
JFCBNVOL  DS     C
JFCBVOLS  DS     5CL6
JFCBEXTL  DS     C
JFCBEXAD  DS     3C
JFCBPQTY  DS     3C
JFCBCTRI  DS     C
JFCBSQTY  DS     3C
          DS     C
JFCBDQTY  DS     3C
JFCBSPNM  DS     3C
JFCBABST  DS     2C
JFCBSBNM  DS     3C
JFCBDRLH  DS     3C
JFCBVLCT  DS     C
JFCBSPTN  DS     C
JFTPOUT   DS     176C
JFORBIT   DS     176C
JFUT1     DS     176C
JFUT2     DC     176C
```

```
JFDIRECT DS     176C
*                    DCB FO RATTITUDE ORBIT TAPE
ORBIT    DCB    DSORG=PS,MACRF=(GM),DDNAME=DORBIT,DEVD=TA,                P
                BUFNO=2,BFALN=D,EROPT=ACC,EODAD=EOF,                      P
                SYNAD=ERR,EXLST=LIST3,BFTEK=S
*                    DCB FOR WRITING OUTPUT ANALYSIS TAPE
TROUT    DCB   DSORG=PS,MACRF=(PM),DDNAME=DTPOUT,DEVD=TA,                P
                BUFNO=2,BFALN=D,EODAD=EOF,                                P
                SYNAD=ERR,EXLST=LIST2,BFTEK=S
*                    DCB FOR DIRECTORY TAPE
DIRECT   DCB    DSORG=PS,MACRF=(GM),DDNAME=DDIRECT,DEVD=TA,               P
                BUFNO=2,BFALN=D,EROPT=ACC,EODAD=EOF,                      P
                SYNAD=ERR,EXLST=LIST6,BFTEK=S
*                    DCB FOR READING DECUM TAPE
DECOM    DCB   DSORG=PS,MACRF=(GL),DDNAME=DDECOM,DEVD=TA,                 P
                BUFNO=2,BFALN=D,EROPT=ACC,EODAD=EOF,                      P
                SYNAD=ERR,EXLST=LIST1,BFTEK=S
*                    DCB FOR DATA SET UTILITAIRE 1
UT1      DCB    DSORG=PS,MACRF=(GM),DDNAME=DUT1,DEVD=DA,                  P
                BUFNO=2,BFALN=D,EROPT=ACC,EODAD=EOF,                      P
                SYNAD=ERR,EXLST=LIST4,BFTEK=S
*                    DCB FOR DATA SET UTILITAIRE 2
UT2      DCB    DSORG=PS,MACRF=(GM),DDNAME=DUT2,DEVD=DA,                  P
                BUFNO=2,BFALN=D,EROPT=ACC,EODAD=EOF,                      P
                SYNAD=ERR,EXLST=LIST5,BFTEK=S
 END
/*
// EXEC LINKGO
//OBJECT DD *
/*
//GO.DORBIT DD UNIT=2400-9,VOLUME=SER=Z943,LABEL=(,BLP),                  P
//             DCB=(DEN=2,BLKSIZE=600),                                   P
//             DISP=(OLD,KEEP),DSNAME=OPEP
//GO.SYSUDUMP DD SYSOUT=A
/*
/*
                                                          0472 CARDS
```