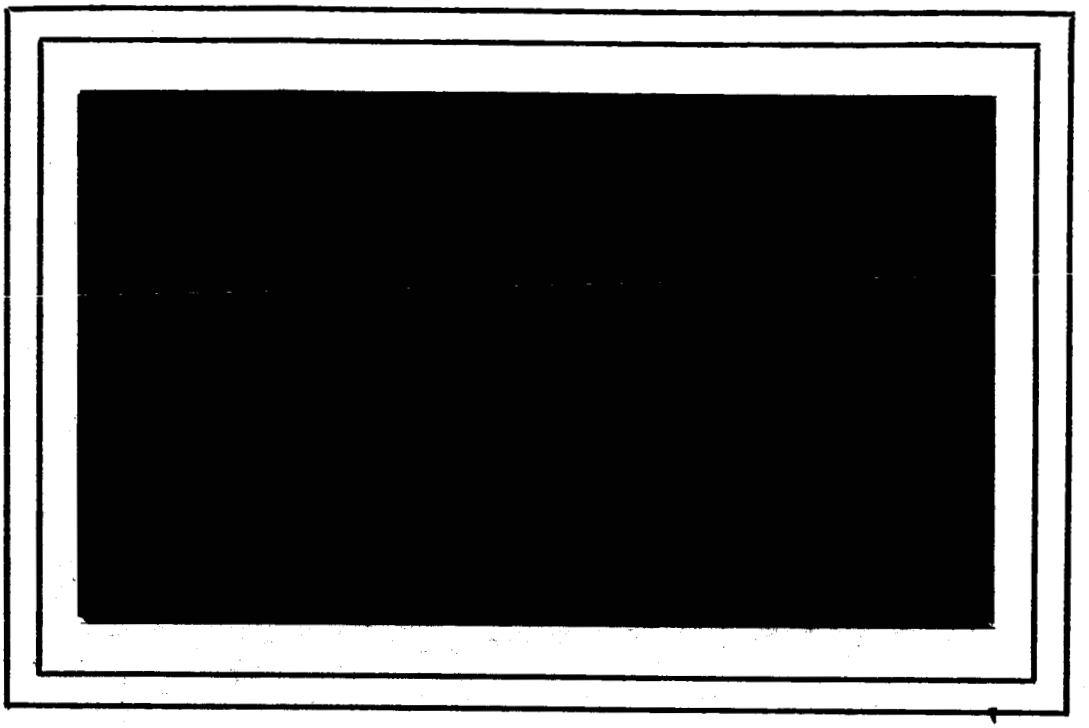
https://ntrs.nasa.gov/search.jsp?R=19660009907 2020-03-16T23:22:25+00:00Z



GPO PRICE $ _____

CFSTI PRICE(S) $ _____

Hard copy (HC) 3.00

Microfiche (MF) .50

ff 653 July 65

UNIVERSITY OF MARYLAND
COMPUTER SCIENCE CENTER

COLLEGE PARK, MARYLAND

FACILITY FORM 602

N66 18196 (ACCESSION NUMBER) — (THRU)

31 (PAGES) — 1 (CODE)

CR-70596 (NASA CR OR TMX OR AD NUMBER) — 08 (CATEGORY)

Technical Report TR-66-26
NsG-398 January 1966

INPUT-OUTPUT SUBROUTINE PACKAGE

UOM IOS
for the
IBM 7090/7094

by

Alfred E. Beam
Senior Computer Systems Analyst
Computer Science Center

The work on this program was performed at the Computer Science Center of the University of Maryland and was supported by Grant NsG-398 of the National Aeronautics and Space Administration.

Abstract

1996

This report describes an input-output subroutine package for use under the IBSYS or DC-IBSYS Monitor System on the IBM 7090/7094. The package is most useful in handling of non-standard tape records under the FORTRAN II and IBJOB sub-monitors of IBSYS. Handling of extremely long records, input records with incorrect parity, short records, variable length records, and mixed mode records are examples of good uses for the package. Also the package may be easily used to write efficient buffering routines.

Introduction

Much of the information which must be processed on the IBM 7090/7094 consists of tape recordings in a format not easily handled by the standard library programs of systems such as IBJOB and FORTRAN II. Such information may come from satellites, radio telescopes, and medical applications. This report describes a basic tape input-output package which allows one to write a small program that handles the non-standard input-output, and the bulk of the program to process the information may be written in other languages such as FORTRAN. The package can also be used to pre-process information into standard FORTRAN records.

A new system to be put under IBSYS could utilize the package to accomplish all of its input-output.

The package requires the IOEX routines of IBSYS and will operate on any 7090/7094 IBSYS system which has tapes. Tapes are referenced by logical numbers which correspond to the standard IBSYS units (e.g. SYSIN1). Assembly parameters are provided for the package to be assembled as an IBMAP or FAP subroutine. An important feature of the routines is that they may be called at interrupt times -- allowing the easy writing of buffering routines.

This package has already had wide use. Versions of the package were used exclusively by two sub-systems to accomplish all of their input-output operations. IOS is in both the IBLIB and FORTRAN II libraries on the UOM IBSYS operating systems tapes.

Identification

a. 7090/7094 I/O Subroutine Package For IBSYS UOM IOS
b. A. Beam, January, 1966
c. Computer Science Center, University of Maryland, College Park, Md.

Purpose

To provide a set of basic tape input-output routines for use in the FORTRAN II and IBJOB Monitor Systems which operate under IBSYS or DC-IBSYS.

Restrictions

The symbolic deck for IOS is distributed to handle 10 logical tape units which are assigned to specific SYSuni functions. The user may alter the assignments or change the number of logical units by simple changes in the symbolic deck. The package requires IOEX for execution.

Method

Tape units are referenced by means of logical numbers which are associated in a table with SYSuni functions (SYSOU1, SYSUT1, etc.), and this table may be changed to suit the purpose of the user. All data selects (read or write in binary or BCD modes) go through a general IOEX select routine with four entries. The select routine is coded so that it may be called at interrupt time if certain conditions are met. Non-data selects are handled by a routine which in turn uses the (NDATA routine of IOEX.

Usage

Usage of the various subroutines is described later. IOS is probably most useful for special purpose tape I/O, i.e. input-output which FORTRAN II and IBJOB library routines cannot handle. Special buffering routines are comparatively easy to write using IOS.

Storage

As distributed, IOS requires about 600 cells. Storage is increased or decreased depending on two assembly parameters which are described later.

Timing

The timing depends upon the way the package is used, and on the model of the tape drives.

Checkout

Various versions of IOS have been used many hours on the 7090 and 7094. It has also been used extensively under DC-IBSYS.

Entries to IOS

A - Non-data selects

The following entries are used for non-data reference to logical tape number N, N=1,2,.... In the calling sequences, N is the logical tape number; and except when specified differently, return is always to the location following the TIX instruction.

1) Rewind logical tape N

```
CALL    REWTAP
TIX     0,0,N
```

2) Backspace logical tape N one record

```
CALL    BSRTAP
TIX     0,0,N
```

3) Rewind and unload logical tape N

```
CALL    RUNTAP
TIX     0,0,N
```

4) Backspace logical tape N one file

```
CALL    BSFTAP
TIX     0,0,N
```

5) Write end of file on logical tape N

```
CALL    WEFTAP
TIX     0,0,N
```

6) Set logical tape N to low density

```
CALL    SETLOW
TIX     0,0,N
```

7) Set logical tape N to high density

```
CALL    SETHIH
TIX     0,0,N
```

8) Skip M records on logical tape N

```
        CALL    SKPREC
        TIX     M,0,N
```

9) Skip M files on logical tape N

```
        CALL    SKPFIL
        TIX     M,0,N
```

Note: Skipping of files and records is overlapped, so computing (and I/O on channels different from the one which N is on) may go on while the skipping is done.

10) Check activity of logical tape N

```
        CALL    CHEKIO
L       TIX     T,0,N
```

If T=0 then control will be returned to L+1 only after logical tape N is inactive.

If T≠0 and logical tape N is inactive then control goes to location T.

If T≠0 and logical tape N is active then control is immediately returned to location L+1.

B - Data selects

The following routine has four entries and is used for all data transmission. An important feature of the routine is that it may be called at trap time. The calling sequence is

```
        CALL    XXXXXX
L       TIX     EOR,0,N
L+1     TIX     L(IOC),W,ETT
L+2     TIX     EOF,T,RTT
```

where XXXXXX = RDSBIN for reading binary records.
RDSDEC for reading BCD records.
WRSBIN for writing binary records.
WRSDEC for writing BCD records.

Control returns to location L+3.

N = the logical tape number of the tape to be read or written.

W≠0 if it is desired to wait until the specified I/O operation is completed before returning to the caller.

T≠0 if only one try is desired for reading even though the record may be redundant. This feature is most useful when the program must determine the mode of the information to be read.

EOR, ETT, EOF, and RTT are trap time exits to the user's routines. Any or all of these exits may be zero. A user's exit (if present) must be to a routine which carries out the desired function and then returns by means of a TRA 1,4 .

L(IOC) is the location of the first of a block of I/O commands. Up to P commands are allowed. If more than P commands are necessary then at least one of the first P+1 must be a TCH command. The I/O commands must terminate with a command which causes a channel interrupt, i. e. the last command must be a IOXT . The first P of the I/O commands are moved to storage within the select routine so the original block at L(IOC) may be modified immediately upon return from the select routine. However, modification is not allowed if there are more than P commands. IOS is distributed such that P=10. An assembly parameter

(described later) may be changed to increase or decrease P.

Calling sequence information is also moved to storage within IOS and hence the contents of L, L+1, and L+2 may be modified immediately upon return to L+3 .

Noise records as defined in IOEX will be accepted if there is at least one input output and proceed command preceding the last I/O command, i.e. the IOXT . Hence, tape may be erased by the two I/O commands:

```
IOCP    0,0,0
IORT    0,0,0
```

However, if only the second of the above two commands was used, then there would be a noise indication.

Permanent Redundancy

IOEX is trusted to write correctly. In reading, a redundant record (as read the last time) is accepted as correct and the user's RTT exit (if specified) is taken. The maximum number of read tries is determined by an IBSYS assembly parameter, or is one if T≠0 in the calling sequence.

End of Tape

If there is no end of tape user's exit specified and the end of tape condition is detected during writing then 2 end of files are added to the tape, a message is printed for the operators and the tape is rewound and unloaded. The machine then pauses for a fresh tape. It is assumed that the record being written at the time the end of tape is encountered is short enough to be written correctly and that enough tape is left to hold the 2 end of file marks. After the pause, a check for RTT and EOR exits is made. If a user's ETT exit is specified then it is taken without doing any of the above actions.

User Exits

EOR, ETT, EOF, and RTT, if non-zero specify entries to subroutines coded by the user. Each of these subroutines must carry out its desired function and return by means of a TRA 1,4 .

An entry to a user's routine is made at trap time, i.e. when an interrupt condition occurs due to channel command trap, a redundant read or write, detection of an end of file in reading, or detection of the end of tape in writing.

On entry to a user's routine the following information is available.

a) The address of the accumulator contains the number of words read or written by the channel command just completed or in use at the time of the interrupt.

b) The decrement of the accumulator contains the logical number of the unit in use at time of interrupt.

Restrictions on the User's Routine

1) The user's routine must exit by means of a TRA 1,4 .

2) For efficient I/O, the user's routine should not be overly time consuming.

3) Only one user's routine is entered for a single interrupt. The order of checking for an exit is as follows:

Reading	Writing
End of file exit (EOF)	End of tape exit (ETT)
Redundancy exit (RTT)	Redundancy exit (RTT)
End of record exit (EOR)	End of record exit (EOR)

4) Activity checking by calling CHEKIO is permissible only if in the sequence:

```
CALL    CHEKIO
TIX     T,0,N
```

T is non-zero. If T is zero, and logical tape N is active, then an endless wait will occur.

5) Index registers 1, 2, the AC, MQ, and indicators need not be saved by the user's routine.

6) Calls to REWTAP, BSRTAP, RUNTAP, BSFTAP, WEFTAP, SETLOW, SETHIH, SKPREC, SKPFIL, RDSBIN, RDSDEC, WRSBIN, WRSDEC may be issued by a user's routine, but only for a unit on the same channel on which the interrupt occurred.

7) Storage within IOS is allocated for 8 blocks of I/O commands and parameters. The number of blocks may be increased or decreased by an assembly parameter. One of these blocks is reserved whenever a logical unit is active. It is possible (if there is not one block per logical unit) that a block will not be available when activity is required. A data-select at non-trap time causes no trouble because an automatic wait for a free block will occur. However, at trap time there may not be more than one block available and no more than one data select should be issued without insuring that there is an available block.

Restrictions 6) and 7) above may be overcome by means of a special trap time routine which may be called by the user's routine. The calling sequence is as follows:

```
        CALL    ISITOK
L       TIX     BUSY,0,N
```

Control will return to location L+1 if it is permissible to select logical unit N. Control is returned to location BUSY if (1) logical tape unit N is on a channel different from the one for which the trap occurred (2) logical unit N is busy or (3) there is no available storage block for I/O command and parameter storage.

On entry to ISITOK, it is assumed that the accumulator contains what it had at the time the user's routine was entered, since the logical unit number in the decrement of the accumulator is used in determining if the channel which N is on is the same as the one for which the trap occurred.

If ISITOK is to be entered more than once, then the second or greater entry may be made to ISIT11 rather than ISITOK and the accumulator as saved on the first call will be used.

It is always permissible to re-select the logical unit for which the trap occurs, and for this type of use there is no requirement to call ISITOK .

Assembly Parameters

There are several parameters which should be checked (and if necessary changed) by the user of IOS.

1) IOUTAB: This parameter (defined by EQU) should be set to 0 to define the first 8 logical tape units to be the same as defined in the distributed FORTRAN II system under IBSYS, and set to 1 to define the first 7 logical tape units to be the same as defined in the distributed IBJOB system.

2) CALL24: Set this parameter to 0 or 1 depending upon the way the subroutines of IOS are to be entered. Use CALL24 EQU 0 if the FAP 'CALL' instruction is desired. Use CALL24 EQU 1 if the IBMAP 'CALL' instruction is desired.

3) ATONCE: This parameter (distributed as 8) is the number of I/O command and parameter storage blocks allocated in the IOS package.

4) IOCSIZ: This parameter (distributed as 10) is the maximum number of I/O commands which are moved. The total number of cells reserved within IOS will be ATONCE* (IOCSIZ+4). This total number should not be less than 60. It is desirable that ATONCE be at least as great as the maximum number of tape units in use at any one time.

The logical tape unit table as distributed is as follows.

Logical Unit	SYSUNI (IOUTAB=1)	SYSUNI (IOUTAB=0)
1	SYSUT1	SYSLB1
2	SYSUT2	SYSUT3
3	SYSUT3	SYSUT4
4	SYSUT4	SYSUT1
5	SYSIN1	SYSIN1
6	SYSOU1	SYSOU1
7	SYSPP1	SYSPP1
8	SYSCK1	SYSUT2
9	SYSCK2	SYSCK1
10	SYSLB1	SYSCK2

The table may be modified or extended by the user.

Error Exits

The only error exit in IOS is when the user specifies an illegal logical tape number. A logical tape number is considered illegal if it falls outside the range of the assembled table or if the word in the table of logical units is zero. Also if the SYSUNI function is not attached. All errors go to symbolic location E1E1E1 within IOS. From E1E1E1, control is sent to the IBSYS dump (SYSDMP) routine. If desired, the user may substitute his own error handling.

Use of IOS With Other I/O Routines

Since IOS uses IOEX to accomplish input-output, traps must not be disabled, and machine instructions for input-output must not be issued, until all pending interrupts have a chance to be processed.

FORTRAN II library routines for input-output do not use IOEX. Therefore, before using the FORTRAN II library routines for I/O, the user must delay until IOS has completed all of its operations. This may be accomplished by giving the sequence

```
CALL  CHEKIO
TIX   0,0,N
```

for N=the logical tape number of each unit which has been used by IOS.

The above restriction does not apply to IBJOB.

Example 1 -- Single Record Buffered BCD Output

Write a MAP subroutine to write BCD records of length N on logical tape T. The maximum record size is M words, and the records are to be single buffered.

The subroutine to accomplish this appears below, and is used by giving the calling sequence:

```
        CALL  WRITE(DUM)
        ORG   *-1              OVERLAY WITH PARAMETERS
DUM     PZE   A,0,N            A=ORIGIN OF N BCD WORDS
        Returns here
```

The sequence could also be specified as follows:

```
        CALL  WRITE
        TIX   A,0,N
        Returns here
```

The second sequence causes the second word of the code generated by the CALL pseudo-operation to be wrong, but the return will still be handled correctly since TIX with a tag of zero is a no operation. It is assumed that the subroutine uses an IOS assembled to handle the MAP 'CALL'.

Assembly parameters (T and M) for the WRITE subroutine are defined to write the regular output tape (SYSOU1). It is assumed that N≥3 in order to avoid writing short records which may be treated as noise.

For illustrative purposes the I/O commands are written so that short records (less than 3 words) could be written without giving a short record error message, and even to accept an N=0 in which case a short piece of tape would be erased. The subroutine assumes that IOS and hence IOEX will write each record correctly, and also that IOS will handle end of tape conditions.

```
$IBMAP OUTPUT
* SINGLE RECORD BUFFERED OUTPUT ROUTINE--BCD MODE
        ENTRY WRITE
* ASSEMBLY PARAMETERS-DEFINED HERE FOR SYSOU1
T       EQU   6                    LOGICAL TAPE TO BE WRITTEN
M       EQU   22                   MAXIMUM RECORD SIZE
* WRITE SUBROUTINE
WRITE   SXA   SAV4,4
        CLA   3,4                  AC=PZE  L,0,N
        STD   TXI                  STORE N=NO. WORDS TO WRITE
        PAX     0,4                I4=LOCATION OF N WORD REC.
TXI     TXI   *+1,4,**             COMPUTE L+N
        SXA   GET,4                STORE FOR PICKUP
        CALL  CHEKIO(TP1)          WAIT TILL LAST RECORD OUT
        ORG   *-1                  OVERLAY CALL GENERATED WD.
TP1     TIX   0,0,T                SPECIFIES THE TAPE NUMBER
        LXD     TXI,4              I4=N
        TXI     *+1,4,BUFFER       COMPUTE BUFFER+N
        SXA   PUT,4                STORE FOR FILLING BUFFER
        LXD   TXI,4                I4=N
        TXL   *+2,4,M              OK IF .LE. M
        AXT   M,4                  WRITE NO MORE THAN M WORDS
        SXD   IOC,4                SET THE I/O COMMAND
GET     CLA   **,4                 MOVE THE LINE TO
PUT     STO   **,4                 THE BUFFER
        TIX   GET,4,1
        CALL  WRSDEC(TIX1,TIX2,TIX3)  GO TO IOS TO WRITE
        ORG   *-3                  OVERLAY CALL GENERATED WDS
TIX1    TIX   0,0,T                T=LOGICAL TAPE
TIX2    TIX   IOC,0,0              IOC=LOC. OF I/O COMMAND
TIX3    TIX   0,0,0
SAV4    AXT   **,4                 RESTORE I4
        TRA   1,4                  RETURN TO CALLER
* I/O COMMANDS AND BUFFER STORAGE
IOC     IOCP  BUFFER,0,**          **=LENGTH OF RECORD
        IORT  0,0,0                CAUSES INTERRUPT
BUFFER  BSS   M                    BUFFER STORAGE
        END                        END OF WRITE SUBROUTINE
```

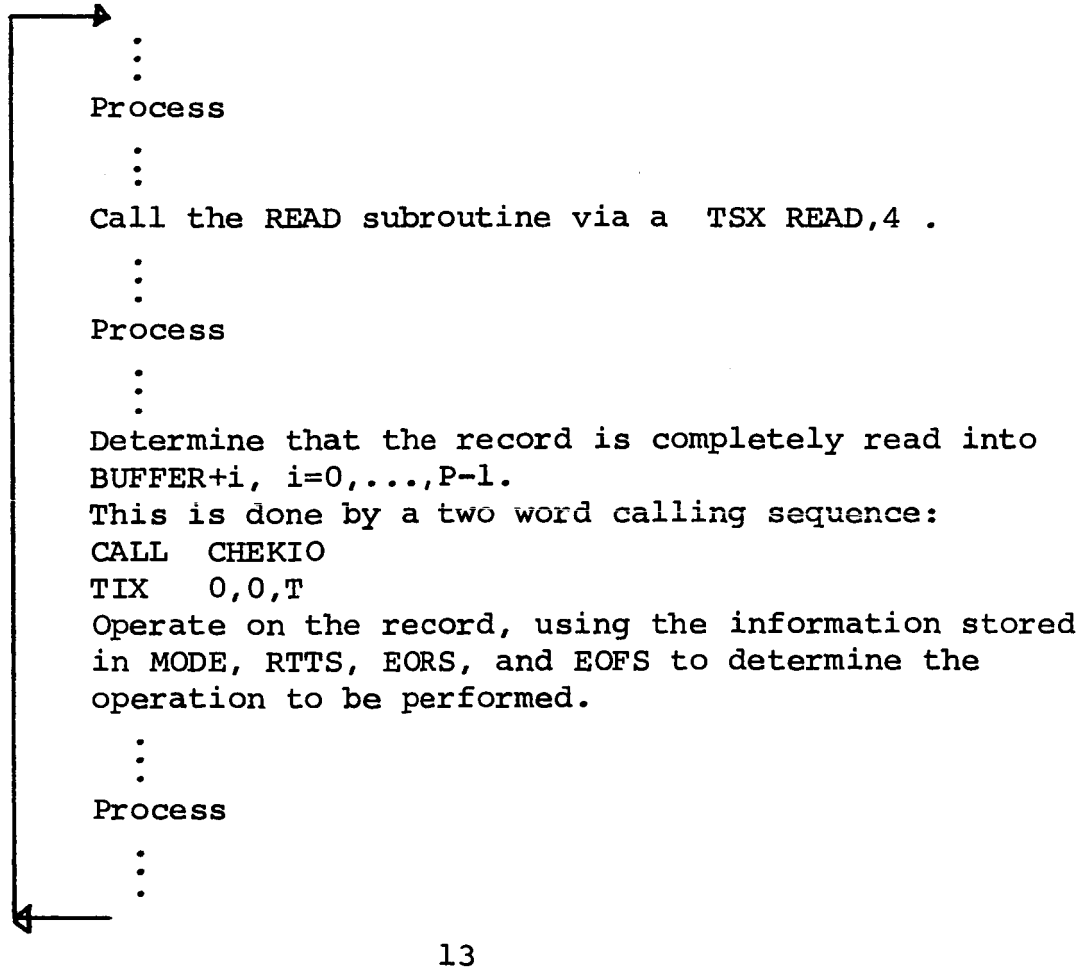
Example 2 -- Mixed Mode Input

Suppose a tape must be processed which contains both binary and BCD records, and also that the records vary in length. The maximum record length is known to not be greater than N. The logical tape number is T, and it is desired to read a record K times before calling it permanently redundant.

A subroutine (using IOS) is written to read one record into storage locations BUFFER+i, i=0,1,...P-1. When the record is finally read, the following information is available:

P, the length of the record is in the address of location EORS .
The mode of the record is BCD if the contents of location MODE is zero.
The record is permanently redundant if the contents of location RTTS is not zero.
The record is an end of file if the contents of location EOFS is not zero.

The name of the read subroutine is READ . A typical use of READ is described by the following diagram.



"Process" in the above diagram could be any type of operations desired. The reading of a record is overlapped with computing and other I/O (on channels other than that one which T is on) operations may be performed.

One application of the above diagram would be a tape copy routine. Another application is a routine to read the system input tape (SYSIN1).

The following READ subroutine has the parameters (T and N) set for reading SYSIN1 which consists of unblocked binary and BCD card images. K is defined as 100 and a bad record would be read 50 times in each mode before being called permanently redundant. If the record is accepted as read the last time, then it is in the opposite mode of that one which was used for the first try. If the mode is known and it is desired to accept bad records, then it would be better to define K as an odd integer.

The first try at reading a record is always in the same mode as the previous record was read the final time. MODE is assembled for the first read to be in BCD mode.

Note that the trap time user routine for handling redundancy calls READ, the same routine which is called in the above diagram. This illustrates the select routine of IOS being called at both interrupt and non-interrupt times.

The code for READ was written to use an IOS which was assembled to use the FAP type CALL pseudo-instruction.

```
* ASSEMBLY PARAMETERS--DEFINED HERE FOR UNBLOCKED SYSIN1
K       EQU   100                MAX. NUMBER OF READ TRIES
T       EQU   5                  LOGICAL TAPE TO BE READ
N       EQU   28                 MAXIMUM RECORD LENGTH
* MAIN LINE READ ROUTINE
READ    SXA   SAVE4,4
        AXC   GOIOS,4            SET I4 FOR ENTRY TO IOS
        ZET   MODE               CHECK MODE OF READING
        TRA   RDSBIN             READ IN BINARY MODE
GOIOS   TRA   RDSDEC             READ IN BCD MODE
        TIX   TRPEOR,0,T         TRPEOR=END OF RECORD EXIT
        TIX   IOCOM,0,0          IOCOM=LOCATION OF I/O COM.
        TIX   TRPEOF,7,TRPRTT    EOF EXIT,1 READ,RTT EXIT
SAVE4   AXT   **,4               IOS RETURNS CONTROL HERE
        TRA   1,4                EXIT
* TRAP TIME USER ROUTINES--CALLED BY SELECT MINUS OF IOS
TRPEOR  STZ   RTTS   END OF RECORD---IT WAS READ O.K.
        STA   EORS               STORE RECORD LENGTH
TRPEX1  STZ   EOFS               CLEAR END OF FILE WORD
TRPEX2  AXT   K,1                RESTORE THE MAXIMUM NUMBER
        SXA   TRPCNT,1           OF READ TRIES
        TRA   1,4                RETURN TO IOS
TRPEOF  STL   EOFS   END OF FILE--SET THE EOF SWITCH
        TRA   TRPEX2
TRPRTT  STA   EORS   READING ERROR--SAVE THE LENGTH
        SXA   TRPSV4,4           SAVE I4 FOR RETURN TO IOS
TRPCNT  AXT   K,4                I4=COUNTER FOR MAX. READS
        TNX   BADREC,4,1         TO BADREC IF CANT READ IT
        SXA   TRPCNT,4           SAVE THE REDUCED COUNTER
        TSX   BSRTAP,4           BACKSPACE THE TAPE
        TIX   0,0,T              T=LOGICAL TAPE NUMBER
        CLA   MODE               FLIP THE MODE SWITCH FOR
        COM                      THE NEXT READ TRY.  MODE=0
        STO   MODE               FOR BCD READ.
        TSX   READ,4             RE-READ IN DIFFERENT MODE
TRPSV4  AXT   **,4               RESTORE AND RETURN TO
        TRA   1,4                SELECT- ROUTINE OF IOS.
BADREC  STL   RTTS   PERMANENTLY REDUNDANT--SET SWITCH
        LXA   TRPSV4,4           AND RETURN TO SELECT-. THE
        TRA   TRPEX1             RECORD IS ACCEPTED THO BAD
* STORAGE AND CONSTANTS
IOCOM   IORT  BUFFER,0,N         COM. TO READ UP TO N WORDS
BUFFER  BSS   N                  RECORD IS READ INTO HERE
MODE    PZE   **                 MODE SWITCH(0 FOR BCD)
RTTS    PZE   **                 SET NOT 0 IF PERM. ERROR
EORS    PZE   **                 SET TO RECORD LENGTH.
EOFS    PZE   **                 SET NOT 0 IF END OF FILE.
* END OF READ ROUTINE
```

Symbolic List of IOS

A symbolic listing of the IOS package follows. The listing appears as a FAP subroutine. To obtain an IBMAP version, replace the * FAP card with a $IBMAP card and redefine IOUTAB and CALL24 to be one rather than zero.

The code CLRXIN through CHEKX overlays the storage for I/O command and parameter storage. This block of code is only entered once to set up the logical unit table and other initialization. The storage allocated by ATONCE and IOCSIZ must be at least as great as this block of code.

The SKPFIL and SKPREC subroutines of IOS provide a third example of a good use of the program.

```
*      FAP
       LBL     IOS00001
*          A. BEAM--- I/O ROUTINES FOR IBJOB OR F II MONITOR SYSTEMS.
*
*
* IDENTIFICATION
*
*      A. 7090/7094 I/O SUBROUTINE PACKAGE FOR IBSYS      UOM IOS
*      B. A. BEAM, JANUARY, 1966
*      C. COMPUTER SCIENCE CENTER, UNIV. OF MARYLAND, COLLEGE PARK, MD.
*
*
       ENTRY   RDSDEC
       ENTRY   RDSBIN
       ENTRY   WRSDEC
       ENTRY   WRSBIN
       ENTRY   CHEKIO
       ENTRY   SETLOW
       ENTRY   SETHIH
       ENTRY   REWTAP
       ENTRY   RUNTAP
       ENTRY   BSRTAP
       ENTRY   BSFTAP
       ENTRY   WEFTAP
       ENTRY   ISITOK
       ENTRY   ISIT11
       ENTRY   SKPREC
       ENTRY   SKPFIL
*************************************************************************
*                                                                       *
*      A S S E M B L Y  P A R A M E T E R S      **********************   *
*                                                                       *
*************************************************************************
*      MAKE    IOUTAB = 0  FOR FORTRAN II (FAP) ASSEMBLY
*      MAKE    IOUTAB = 1  FOR IBJOB (IBMAP) ASSEMBLY
*      MAKE    CALL24 = 0  IF FAP  CALL IS DESIRED TO BE USED
*      MAKE    CALL24 = 1  IF MAP  CALL IS DESIRED TO BE USED
IOUTAB EQU     0
CALL24 EQU     0
*
*
 K     EQU     CALL24
 T     EQU     IOUTAB
*
*MAX. NO. UNITS WHICH CAN BE HANDLED AT ONE TIME
ATONCE EQU     8
*
*MAX. NO. OF I/O COMMANDS WHICH ARE MOVED
IOCSIZ EQU     10
*
```

```
*                I/O UNIT TABLE
*
 IOUTB BSS      0   MAKE ADDITIONS AFTER THIS CARD
*                 IBJOB  F II    F II          LOGICAL NO.
        PZE     T*SYSLB1+SYSCK2-T*SYSCK2                10
        PZE     T*SYSCK2+SYSCK1-T*SYSCK1                 9
        PZE     T*SYSCK1+SYSUT2-T*SYSUT2                 8
        PZE     T*SYSPP1+SYSPP1-T*SYSPP1                 7
        PZE     T*SYSOU1+SYSOU1-T*SYSOU1                 6
        PZE     T*SYSIN1+SYSIN1-T*SYSIN1                 5
        PZE     T*SYSUT4+SYSUT1-T*SYSUT1                 4
        PZE     T*SYSUT3+SYSUT4-T*SYSUT4                 3
        PZE     T*SYSUT2+SYSUT3-T*SYSUT3                 2
        PZE     T*SYSUT1+SYSLB1-T*SYSLB1                 1
 IOPUT  PZE     IOPUT-IOUTB
*********************************************************************
*                I B S Y S  S Y M B O L  D E F I N I T I O N S
*********************************************************************
SYSACC  BOOL    122
SYSCK1  BOOL    155
SYSCK2  BOOL    156
SYSCUR  BOOL    102
SYSDMP  BOOL    115
SYSIDR  BOOL    117
SYSIN1  BOOL    151
SYSLB1  BOOL    140
SYSLB2  BOOL    141
SYSOU1  BOOL    147
SYSPOS  BOOL    106
SYSPP1  BOOL    153
SYSTRA  BOOL    100
SYSUT1  BOOL    157
SYSUT2  BOOL    160
SYSUT3  BOOL    161
SYSUT4  BOOL    162
.CHEXI  BOOL    134
.MODSW  BOOL    135
(ACTIV  BOOL    702
(BCD5R  BOOL    720
(CHXAC  BOOL    724
(COMMM  BOOL    736
(CVPRT  BOOL    722
(NDATA  BOOL    704
(PAUSE  BOOL    712
(PROUT  BOOL    706
(RCHXI  BOOL    727
(SYMUN  BOOL    714
(TRAPX  BOOL    734
(TRPSW  BOOL    742
(URRXI  BOOL    725
*********************************************************************
*
* ALL ERRORS COME HERE AND HENCE TO SYSDMP
*
E1E1E1  STL     0
        TRA     SYSDMP
*
*********************************************************************
```

```
*              NON-DATA SELECTS      **********************
*
*       CALL    YYYYYY
*       TIX     0,0,N
*
* YYYYYY = SETLOW, SETHIH, REWTAP, RUNTAP, BSRTAP, BSFTAP, OR WEFTAP
*
*****************************************************************************
*
SETLOW LDQ      SDL
       TRA      NDATAX
SETHIH LDQ      SDH
       TRA      NDATAX
REWTAP LDQ      REW
       TRA      NDATAX
RUNTAP LDQ      RUN
       TRA      NDATAX
BSRTAP LDQ      BSR
       TRA      NDATAX
BSFTAP LDQ      BSF
       TRA      NDATAX
WEFTAP LDQ      WEF
NDATAX TRA      NDAT  WILL BE REPLACED BY SXD  NDATAS,4
       NZT      (TRPSW
       SXA      NDATAS,4                NON-TRAP
       CLA      1+2*K,4
       PDX      0,4             LOGICAL UNIT
       TXL      E1E1E1,4,0
       TXH      E1E1E1,4,IOPUT-IOUTB
       NZT      IOPUT,4
       TRA      E1E1E1
*  INSERT OTHER CHECKS IF DESIRED.....................................
       CAL      IOPUT,4
       LGR      18
       RQL      18
       ENB      PZE0
       STQ      NDATA1
       TSX      (NDATA,4
NDATA1 PZE      **,0,**
NDATAS TIX      **,0,**
       LXD      NDATAS,4           TRAP TIME EXIT
       ZET      (TRPSW
       TRA      2+2*K,4
       LXA      NDATAS,4             NOT TRAP TIME
       ENB*     (TRAPX            RESTORE TRAPS
       TRA      2+2*K,4
*         END OF NDATA
*
*****************************************************************************
 SDL    PZE      0,0,1
 SDH    PZE      0,0,2
 REW    PZE      0,0,3
 RUN    PZE      0,0,4
 BSR    PZE      0,0,5
 BSF    PZE      0,0,6
 WEF    PZE      0,0,7
```

```
**********************************************************************
*
*
*ROUTINE TO BE USED ONLY AT TRAP TIME---CALLED BY THE USERS TRAP TIME
*ROUTINE....ON ENTRY THE AC MUST HAVE WHAT IT HAD WHEN THE USERS
*ROUTINE WAS ENTERED.....
*
*        C A L L I N G  S E Q U E N C E
*
*       CALL    ISITOK
*       TIX     BUSY,0,N        GOES TO BUSY IF N IS BUSY OR WRONG CHAN.
*       RETURN HERE IF O.K. TO RESELECT UNIT N
*
* AFTER THE FIRST CALL  ENTRY CAN BE TO ISIT11  IN WHICH CASE THE AC
* NEED NOT BE SUPPLIED AND THE VALUE IT HAD ON FIRST ENTRY WILL BE USED.
*
ISITOK STO      ISITAC            SAVE  AC
ISIT11 SXA      ISNOOK,1
       CLA      ISITAC             RESTORE THE AC
       PDX      0,1              I1 = TRAP UNIT
       CLA      IOPUT,1
       ANA      CHNLMS
       SLW      ISITCH               TRAPPED CHANNEL
       CLA      1+2*K,4
       PDX      0,1          I1 = UNIT IN QUESTION
       TXL      E1E1E1,1,0
       TXH      E1E1E1,1,IOPUT-IOUTB
       NZT      IOPUT,1
       TRA      E1E1E1                    ILLEGAL UNIT
       STD      ISOK02              SET UNIT FOR CHEKIO
       CLA      IOPUT,1
       ANA      CHNLMS          CHANNEL IN QUESTION
       ERA      ISITCH              DO THEY MATCH
       TZE      ISOK01            YES----IF ZERO
ISNOOK AXT      **,1
       CLA      ISITAC            RESTORE AC
       TRA*     1+2*K,4          GO TO BUSY EXIT
ISITAC PZE      **
ISITCH PZE      **
ISOK01 SXA      ISOK03,4                SAVE  I4
       TSX      CHEKIO,4         CHECK IF UNIT FREE
       TXI      *+3,0,1
       PZE      0
       BES      2*K-2
ISOK02 TIX      ISOK04,0,**           LOGICAL UNIT
ISOK03 AXT      **,4           ACTIVE-----DONT SELECT
       TRA      ISNOOK             GO TO BUSY EXIT
ISOK04 AXT      IOTSIZ,1
       NZT      IOCTAB+IOTSIZ,1     CHECK IF I/O BLOCK IS AVAILABLE
       TRA      ITISOK             YES
       TIX      *-2,1,IOCDEC          NOT YET
       TRA      ISOK03              NO
ITISOK LXA      ISOK03,4
       LXA      ISNOOK,1
       CLA      ISITAC
       TRA      2+2*K,4
*
```

```
**************************************************************************
* ROUTINE TO DETERMINE IF LOGICAL UNIT N IS ACTIVE
*
*       C A L L I N G    S E Q U E N C E
*      CALL    CHEKIO
*      TIX     T,0,N
CHEKIO TRA     CHEK   WILL BE REPLACED BY  PXA  0,1
       XCA
       CAL     1+2*K,4
       PDX     0,1
       TXL     E1E1E1,1,0
       TXH     E1E1E1,1,IOPUT-IOUTB
       NZT     IOPUT,1
       TRA     E1E1E1                    ILLEGAL UNIT
       CLA*    IOPUT,1        T NOT 0 THEN TO  2,4 IF N ACTIVE
       PAC     0,1
       CAL     1+2*K,4
       ANA     CHEK77         T NOT 0 THEN TO T IF N INACTIVE
       TNZ     CHEKIN
       ZET     1,1            T = 0 THEN WAIT UNTIL INACTIVE AND TO 2,4
       TRA     *-1
CHEKIX XCA
       PAX     0,1
       TRA     2+2*K,4
CHEKIN ZET     1,1
       TRA     CHEKIX
       PAC     0,4
       TXI     CHEKIX,4,2+2*K
CHEK77 PZE     -1          END OF CHEKIO
*
**************************************************************************
*
```

```
* SELECT ROUTINE FOR IBSYS (WITH DC IBSYS CAPABILITY)
IOCDEC EQU     IOCSIZ+1+3          I/O COM. BLOCK SIZE
IOTSIZ EQU     ATONCE*IOCSIZ+ATONCE*4
*
DCMODE SYN     .CHEXI
 MODSW SYN     .MODSW
*
*
RDSDEC LDQ     RTDFLG              CALL XXXXXX,4
       TRA     RDSWRS              TIX  EOR,0,N
RDSBIN LDQ     RTBFLG              TIX  L(IOC),W,ETT
       TRA     RDSWRS              TIX  EOF,T,RTT
WRSDEC LDQ     WTDFLG              RETURN
       TRA     RDSWRS
WRSBIN LDQ     WTBFLG
RDSWRS TRA     DATA   WILL BE REPLACED BY  PXA  0,1
       LGR     18                  SAVE I1
****** FREE THE UNIT
       CLA     1+2*K,4
       PDX     0,1                 I1 = UNIT
       TXL     E1E1E1,1,0
       TXH     E1E1E1,1,IOPUT-IOUTB
       NZT     IOPUT,1
       TRA     E1E1E1              ILLEGAL UNIT
*              INSERT OTHER CHECKS IF DESIRED.
       CLA*    IOPUT,1             PRF L(UCB1) TO AC.
       PAC     0,1                 I1 = -L(UCB1)
*   CHECK IF TRAP TIME AND BUSY
       NZT     1,1
       TRA     *+7            NOT BUSY
       NZT     (TRPSW
       TRA     *+5            NOT TRAP TIME
 RWIGN XCL                     TRAP TIME AND BUSY
       PDX     0,1            RESTORE  I1
       PXD     0,0            CLEAR    AC
       TRA     4+2*K,4
       ZET     1,1                 WAIT FOR WORD 2
       TRA     *-1                 TO BE ZERO.
*              SAVE  L(UCB)
       LGR     15
       RQL     33
****** FIND A FREE BLOCK FOR IOC STORAGE
 RW1   AXC     IOCTAB,1
 RW2   NZT     0,1
       TRA     RW3                 HAVE ONE
       TXI     *+1,1,-IOCDEC
 RWT1  TXH     RW2,1,**   = -IOCTAB-IOTSIZ
       NZT     (TRPSW
       TRA     RW1                 KEEP LOOKING
       RQL     18
       TRA     RWIGN  NO BLOCK AVAIL, AND ITS TRAP TIME... IGNORE CALL.
******         DISABLE TRAPS
 RW3   ENB     PZEO                DISABLE
****** SETUP I/O STORAGE BLOCK AND (ACTIV CALL
       SXA     UCB2,1              -L(I/O STORAGE)
```

```
        SXA     PDNI4S,1
        CLA     2+2*K,4
        STA     RW11                    L(IOC)
        STC     1,1                     L(IOC),WAIT,ETT
        CLA     3+2*K,4
        STT     RTTSWT
        STO     2,1                     EOF,ONETIME,RTT
        CLA     1+2*K,4
        STC     0,1                     EOR,IGNORE,UNIT
        SXD     PDNI4S,4                SAVE I4
        SXA     RWI2,2                  SAVE I2
        PDX     0,4                  ** I4 = UNIT
        XCL                             PRF I1,0,L(UCB)
        STA     RWI1                    SAVE I1 (OF CALLER
        PDC     0,2                  ** I2 = -L(UCB)
        STP     UCB2                    TYPE OF SELECT
        CLA     IOPUT,4
        STA     RW33                    SET SYSUNI
        CLA     PDNI4S
        STO     PDNI4+1
        NZT     (TRPSW
        STC     PDNI4
        PXD     0,0
        STP     RW33
        CAL     WTDFLG                  MZE
        ZET     (TRPSW                  IF TRAP TIME
        ORS     RW33                    GIVE PRIORITY
        ARS     1
        ZET     RTTSWT
        ORS     UCB2                    SUPPRESS RTT MESSAGE
        CAL     UCB2
        SLW     1,2                     PLANT UCB2
******  MOVE THE COMMANDS
        LAC     UCB2,4                  GET NOISE IGNORE
        SXA     RW44,4                  SWITCH ADDRESS
        AXT     0,4
 RW11   CAL     **,4                    ** = LOC OF USER I/O
        SLW     3,1
        ARS     18+15
        LBT
        TRA     *+4                     ITS IOXP
        ARS     1                       ODD
        TZE     *+4                     ITS TCH
        TRA     *+6                     ITS IOXT
        COM
 RW44   STT     **                      IGNORE NOISE
        TXI     *+1,4,-1
        TXI     *+1,1,-1
        TXH     RW11,4,-IOCSIZ-1
*
 RWI1   AXT     **,1
 RWI2   AXT     **,2
        TSX     (ACTIV,4
 RW33   PZE     **                      ** = SYSUNI
```

```
        ENB     PZEO                    DISABLE
        CLA     PDNI4+1
        NZT     (TRPSW
        CLA     PDNI4
        PAX     0,4                     I4=-L(I/O STORAGE)
        XCA                             SAVE IN MQ
        CLA     1,4
        ANA     TAGMSK                  DO WE WAIT
        ZET     (TRPSW
        TRA     RW99            NO IF TRAP TIME
        TZE     RW99                    NO IF ZERO
*               WAIT UNTIL COMPLETE
        ENB*    (TRAPX
        ZET     0,4                     WAIT
        TRA     *-1
 RW99   XCA
        PDX     0,4                     RESTORE I4
        ENB     PZEO
        NZT     (TRPSW
        ENB*    (TRAPX
        TRA     4+2*K,4
*
*       END OF  SETUP AND CALL (ACTIV PART
*
* T R A P  T I M E  S E L E C T  * * * * * * *
*
*ON ENTRY TO MADSEL---
*UCB WORD 2 =  PRF -L(I/O BLOCK),0,MADSEL
*PRF=1X1 FOR WRITE BINARY
*    1X0 FOR WRITE BCD
*    0X1 FOR READ  BINARY
*    0X0 FOR READ  BCD
*LET L(I/O BLOCK)=A ,THEN THE I/O BLOCK IS AS FOLLOWS.
*
* A+0   TIX     EOR,IGNORENOISE,N
* A+1   TIX     L(IOC),W,ETT            L(IOC) NOT USED BY MADSEL
* A+2   TIX     EOF,T,RTT
* A+3   IOXY
*       ...
*       ...
*       ...
* A+12  IOXY
* A+13  TCH
MADSEL  SXA     SELI14,4                SAVE I4
        PAX     0,4                     I4= L(UCB)
        SXA     SELWD1,4                INDIRECT UCB WORD1
        TXI     *+1,4,1
        SXA     SELWD2,4                INDIRECT UCB WORD2
        XCA
        CLA*    SELWD2
        PAC     0,2                     L(I/O BLOCK)
        SXA     SELIOB,2
        PAX     0,2                     I2=-L(I/O BLOCK)
        TQP     SLPLUS                  SELECT +
        TRA     SELMN                   SELECT -
```

```
*SELECT++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
SLPLUS TSX     GETPAR,4            RTT SWITCH
       TIX     7,0,15
       TZE     *+2
       STL*    (URRXI              ONE TRY ONLY
       CLA     SELIOB              SET FOR
       ADD     SEL333              RESET AND
       STA*    (RCHXI              LOAD CHANNEL
       CLA*    SELWD1
       PDX     0,4                 I4= UNIT ADDRESS
       LDQ*    SELWD2
       RQL     2
       STZ     MODSW          SET   BCD  MODE FOR  DC
       TQP     SLPSET
       TXI     *+1,4,16            MAKE BINARY
       STL     MODSW          SET BINARY MODE FOR  DC
SLPSET SXA     SELWOP,4
       SXA     SELROP,4
       LDQ*    SELWD2
       TQP     SELRD1              READ IF PLUS
       ZET*    DCMODE              CHECK D-C
SELWOP WRS     **                  NOT D-C
       TRA     SELRCH
SELRD1 ZET*    DCMODE              CHECK D-C
SELROP RDS     **                  NOT D-C
SELRCH XEC*    (RCHXI
SELPME LXA     SELI14,4            EXIT
       TRA     1,4
*******END OF SELECT++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
*      A       TSX GETPAR,4
*      A+1     NUM MASK,0,SHIFT
GETPAR CAL     1,4                 NUM = WORD NUMBER =0,1,...
       STA     SELMSK              MASK = 15 BIT ADDRESS MASK
       ARS     18                  DOES ARS SHIFT BEFORE MASKING
       STA     SELARS              EXITS WITH RESULT IN AC.
       ARS     15
       ACL     SELIOB
       STA     *+1
       CAL     **
SELARS ARS     **
       ANA     SELMSK
       TRA     2,4
*              END OF GET PAR
*SELECT----------------------------------------------------------------------
 SELMN SXD     SELI14,1            SAVE I1
       LFT     400000              CHECK NOISE
       TRA     NNOISE              NO NOISE
*  CHECK IF NOISE IS TO BE IGNORED.
       TSX     GETPAR,4
       PZE     7,0,15
       TNZ     NNOISE              NOISE OK IF NOT ZERO
*              IGNORE THE NOISE
       LXA     SELI14,4
       TRA     2,4
NNOISE LDQ*    SELWD2              SAVE WORD 2
```

```
        STZ*    SELWD2              RELEASE UNIT
        LNT     200000
        TRA     SELER               NO EOT OR EOF
        TQP     SELEF               ITS END OF FILE
        TRA     SELET               ITS END OF TAPE
SELER   LNT     100000
        TRA     SELOK               NO REDUNDANCY
        TSX     GETPAR,4            REDUNDANCY---ACCEPT
        TIX     -1,0,18             OR GO TO USER
        TRA     SELEX
SELOK   TSX     GETPAR,4            NO ERROR DETECTED
        PZE     -1,0,0
        TRA     SELEX
SELET   TSX     GETPAR,4            END OF TAPE
        TXI     -1,0,18
        TNZ     SELEX               GO TO USER EXIT
        TSX     GETPAR,4            GET THE LOGICAL
        PZE     -1,0,18             UNIT
        PAX     0,4
        CAL     IOPUT,4
        STA     SELU1               SYSUNI
        STA     SELU2
        AXT     2,2
        TSX     (NDATA,4
SELU1   PZE     **,0,7              END FILE THE TAPE
        NOP
        TIX     *-3,2,1
        TSX     (NDATA,4
SELU2   PZE     **,0,4              REWIND-UNLOAD
        LDQ*    SELWD1              UCB WORD 1
        TSX     (CVPRT,4            PRINT UNIT AND
        PZE     SELUB,0,4           MESSAGE
        TSX     (PAUSE,4            PAUSE FOR NEW TAPE
SELXT1  STZ*    SELIOB              RELEASE THE BLOCK
        TRA     SELER                EXIT
SELUB   BCI     4, FULL TAPE---REFRESH.
SELEF   TSX     GETPAR,4            END OF FILE DETECTED
        TIX     -1,0,0
        TRA     SELEX
SELXTT  STZ*    SELIOB
        TRA     SELPME
SELEX   TZE     SELXTT          CHECK FOR USER EXIT
        STA     SELXT                   THERE IS A USER EXIT.
*               SAVE I1 AND I4
        CLA     SELI14
        STO     SELSVT
        PXD     0,1
        PDC     0,2
        LDC     (COMMM,4
        CAL     (COMMM
        ORA     WTDFLG              WTDFLG=MZE 0
        SBM     -1,4                GET NO. OF WORDS
        STA     SELWD1              TRANSMITTED
        CAL*    SELIOB
```

```
        PDX     0,4
        PXD     0,4
        ORA     SELWD1              PZE NO.WDS.,0,N
        AXC     (COMMM,1
        STZ*    SELIOB              FREE THE BLOCK
        AXC     **+1,4
 SELXT  TXI     **,2,1              GO TO USER
*                                   USER RETURNS HERE
 SELUR  ENB     PZEO
        CLA     SELSVT
        STO     SELI14
        PDX     0,1                 RESTORE I1
        TRA     SELPME              EXIT
*               END OF SELECT----------------------------------------
*
*********************************************************************
* CONSTANTS AND STORAGE
*
 UCB2   PZE     **,0,MADSEL         **=-L(I/O STORAGE)
RTDFLG  PZE     0
RTBFLG  PON     0
WTDFLG  MZE     0
WTBFLG  MON     0
 PZEO   SYN     RTDFLG
PONI4S  PZE     **
RTTSWT  PZE     0,**,0
CHNLMS  OCT     017000000000         CHANNEL MASK
TAGMSK  PZE     0,7,0
SELI14  PZE     **,0,0*0            **=I4 , 0*0 = I1
SELWD1  PZE     **                  ** = L(UCB WORD 1)
SELWD2  PZE     **                  ** = L(UCB WORD 2)
SELIOB  PZE     **                  ** = L(I/O BLOCK)
SELMSK  PZE     **
SEL333  PZE     3
IOCTAB  BSS     IOTSIZ
 PONI4  BSS     2
SELSVT  BSS     1
*
*********************************************************************
*
SELEND  SYN     *
        BES     IOCTAB-*
        PZE     0
CLRXIN  STZ     SELEND,4
        TIX     *-1,4,1
        AXT     **,4
        TRA     **
INCLRX  SXA     *-2,4
        STO     *-2
        AXC     IOCTAB+IOTSIZ,4
        SXD     RWT1,4
```

```
* PUT ACTUAL CHANNEL INTO LOGICAL UNIT WORD
        AXT     IOPUT-IOUTB,4           I4 = NUMBER OF UNITS
CLRLP   CLA*    IOPUT,4
        NZT*    IOPUT,4
        STZ     IOPUT,4
        STA     *+1
        CLA     **
        ANA     CHNLMS
        ZET     IOPUT,4
        STD     IOPUT,4
        TIX     CLRLP,4,1
        AXT     SELEND-INCLRX,4
        CLA     SKIPX
        STO     SKPI4T
        CLA     NDATX
        STO     NDATAX
        CLA     DATAX
        STO     RDSWRS
        CLA     CHEKX
        STO     CHEKIO
        TRA     CLRXIN
SKIP    CLA     *+2
        TRA     INCLRX
        TRA     SKPI4T
NDAT    CLA     *+2
        TRA     INCLRX
        TRA     NDATAX
DATA    CLA     *+2
        TRA     INCLRX
        TRA     RDSWRS
CHEK    CLA     *+2
        TRA     INCLRX
        TRA     CHEKIO
SKIPX   PXA     **,4
NDATX   SXD     NDATAS,4
DATAX   PXA     0,1
CHEKX   PXA     0,1
        BES     SELEND-*
*
*       END OF  SELECT
*
```

```
*SUBROUTINE TO SKIP M FILES/RECORDS ON LOGICAL N
*       CALL    XXXXXX          XXXXXX = SKPREC OR
*       TIX     M,0,N               XXXXXX = SKPFIL
*       RETURN
SKPREC  LDQ     SKPRDS              PZE 0,0,L(IOC)
        TRA     *+2
SKPFIL  LDQ     SKPFLS              PZE 0,0,L(IOC)
SKPI4T  TRA     SKIP    WILL BE REPLACED BY  PXA  **,4
        LGR     18                  SAVE I4
        CAL     1+2*K,4
        PDX     0,4                 LOGICAL UNIT
        TXL     E1E1E1,4,0
        TXH     E1E1E1,4,IOPUT-IOUTB
        NZT     IOPUT,4
        TRA     E1E1E1
        CLA*    IOPUT,4
        PAC     0,4                 -L(UCB)
        NZT     1,4                 IS UNIT BUSY
        TRA     SKPTIP              NOT BUSY
        NZT     (TRPSW              IT IS BUSY
        TRA     *-3                 NOT TRAP TIME SO HOLD
        XCL                         IT IS TRAP TIME
        PDX     0,4                 SO IGNORE REQUEST
        PXD     0,0                 IF UNIT IS BUSY.
        TRA     2+2*K,4
SKPTIP  LGL     18
        PAX     0,4                 RESTORE
        SXA     SKPI4T,4            SAVE FOR TRAP TIME
        NZT     (TRPSW
        SXA     SKPI4S,4            NOT TRAP TIME
        CLA     1+2*K,4
        ANA     CHEK77              GET SKIP COUNT
        TNZ     *+3
        CLA     1+2*K,4
        TRA     2-K,4
        ENB     PZE0
        CLA     1+2*K,4
        PDC     0,4                 I4=-UNIT
        STD     SKPLGN              PUT UNIT INTO CALL
        LGL     18                  LOCATION OF IOC
        STA     SKPIOC              PUT L(IO) INTO CALL
        STA     SKPCON,4            SAVE FOR TRAP TIME
        STC     SKPCON,4            SAVE COUNT FOR TRAP TIME
        NZT     (TRPSW
        ENB*    (TRAPX              RESTORE TRAPS
        TSX     RDSBIN,4
        TXI     *+5,0,1
        PZE     0
        BES     2*K-2
SKPLGN  TIX     SKPEOR,0,**         ** = LOGICAL UNIT
SKPIOC  TIX     **,0,0
```

```
        TIX     SKPEOR,7,SKPEOR
        ENB     PZEO
SKPI4S  AXT     **,4
        ZET     (TRPSW
        LXA     SKPI4T,4            ITS TRAP TIME
        NZT     (TRPSW
        ENB*    (TRAPX
        TRA     2+2*K,4
*
SKPRDS  PZE     0,0,SKPRIO
SKPFLS  PZE     0,0,SKPFIO
SKPRIO  IORTN   0,0,0
SKPFIO  IORPN   0,0,-1
SKPCON  TCH     *-1
        BSS     IOPUT-IOUTB         STORAGE FOR SKIPPING
*
*               AT TRAP TIME----------------------------------------------
SKPEOR  SXA     SKPTI4,4            SAVE I4
        PDC     0,4                 I4=-LOGICAL UNIT
        STD     SKPTUN              PUT UNIT INTO CALL
        CLA     SKPCON,4            PRF L(IO),0,NTOSKP
        STA     SKPTIO              L(IO) INTO CALL
        PDX     0,1                 REDUCE THE
        TXI     *+1,1,-1            SKIP COUNT
        TXL     SKPTI4,1,0          FINISHED IF ZERO
        PXD     0,1
        STD     SKPCON,4            PUT COUNT BACK
        TSX     RDSBIN,4
        TXI     *+5,0,1
        PZE     0
        BES     2*K-2
SKPTUN  TIX     SKPEOR,0,**
SKPTIO  TIX     **,0,0
        TIX     SKPEOR,7,SKPEOR
SKPTI4  AXT     **,4
        TRA     1,4
* E N D   O F   S K P F I L  - -  S K P R E C
        END
```