

Application Note

BUFFERED INPUT/OUTPUT PACKAGE IN THE REAL TIME DISK OPERATING SYSTEM OR REAL TIME OPERATING SYSTEM

017-000003-02

NOTICE

Data General Corporation (DGC) has prepared this manual for use by DGC personnel, licensees and customers. The information contained herein is the property of DGC and shall neither be reproduced in whole or in part without DGC prior written approval.

DGC reserves the right to make changes without notice in the specifications and materials contained herein and shall not be responsible for any damages (including consequential) caused by reliance on the materials presented, including but not limited to typographical, arithmetic, or listing errors.

Original Release - October 1972
First Revision - April 1973
Second Revision - January 1975

This revision of Application Note 017-000003 is a major revision and supersedes Application Note 017-000003-01. Areas of substantive change are indicated by a vertical bar on the outer margin of each page where they occur.

TABLE OF CONTENTS

Introduction	1
Buffer Package Entry Points	1
Buffer Organization	1
Buffer Package Subroutine Usage.....	3
Open a File for Buffered I/O	3
Access a File Opened for Buffered I/O (BFACS).....	4
Close a File Opened for Buffered I/O (BFCLS).....	5
Example of Buffer Package Usage	6
BFPKG Program Listing.....	8

BUFFERED INPUT/OUTPUT PACKAGE, BFPKG

INTRODUCTION

Efficient I/O handling is the most important single factor in the effective utilization of CPU time. Input/output devices, always slow compared to the internal speed of the CPU, must be programmed to overlap their operations with main line computations whenever possible.

The RDOS and RTOS system libraries provide a module named BFPKG (MBFPKG in mapped systems) which permits faster line and sequential I/O transfers than is possible using the system I/O calls. This package is approximately 670⁸ words in length. It utilizes tasking concepts to fill (or empty) two or more core buffers asynchronously, providing a constant supply of data for program (or device) processing.

BUFFER PACKAGE ENTRY POINTS

BFPKG provides four OPEN commands, one command for accessing buffers, and one command to close files and release their buffers. These commands closely resemble system commands, and in fact the BFPKG module issues .SYSTEM open, line, sequential and close commands. The following list summarizes the entry points in BFPKG:

BFRSO	Open a file for a sequential read
BFRLO	Open a file for a line read
BFWSO	Open a file for a sequential write
BFWLO	Open a file for a line write
BFACS	Access the buffers
BFCLS	Close a file and release its buffers

Every BFPKG entry point that a user wishes to access must be referenced by him in an external normal statement, .EXTN .

BUFFER ORGANIZATION

The user must reserve a dedicated core buffer area for each type of I/O operation that is to occur asynchronously. He must also issue the appropriate open command, the buffer access command, and finally the buffer close command to release the buffers for other use. Moreover, having chosen to use buffered I/O, the user is cautioned not to attempt opening or closing the desired file by means of conventional system open and close commands. Only the special BFPKG open and close commands can be used with the buffer access command.

Each buffer area that is reserved for use with BFPKG has certain fixed portions which are used by BFPKG for housekeeping and temporary storage. Thus, the total buffer area reserved by the user must indicate sufficient storage for these fixed blocks. Each buffer area is preceded by a 21 octal word header called the Buffer Environment Table (BET). The user need be concerned with only the first three words.

BUFFER ORGANIZATION (Continued)

in the BET. The first word in the BET area (called FWA) contains the RDOS channel number to be used in the transfer, right justified. The priority at which the task filling (or emptying) the buffers is to operate must be specified in the left byte of this same word. The task priority should be higher than that of the user program desiring the buffered I/O. The contents of word FWA+1 must also be written by the user before issuing any file open calls. This word contains the integer number of segments into which the user wishes the usable (net) buffer space to be subdivided. If any value smaller than 2 is specified, BFPKG will use a default value of 2 for the number of segments.

The user may subdivide the net buffer space into as many segments as he wants provided adequate buffer area is allocated. If read or write line operations are to be performed, the buffer space will be divided into 66 decimal word buffer segments; an error return will result, with error code ERSPC, if sufficient buffer area is available for two 66 word buffers. If read or write sequential I/O is to be performed, there must be at least one word (2 bytes) per buffer segment or error code ERSPC will be issued when an attempt is made to open the file.

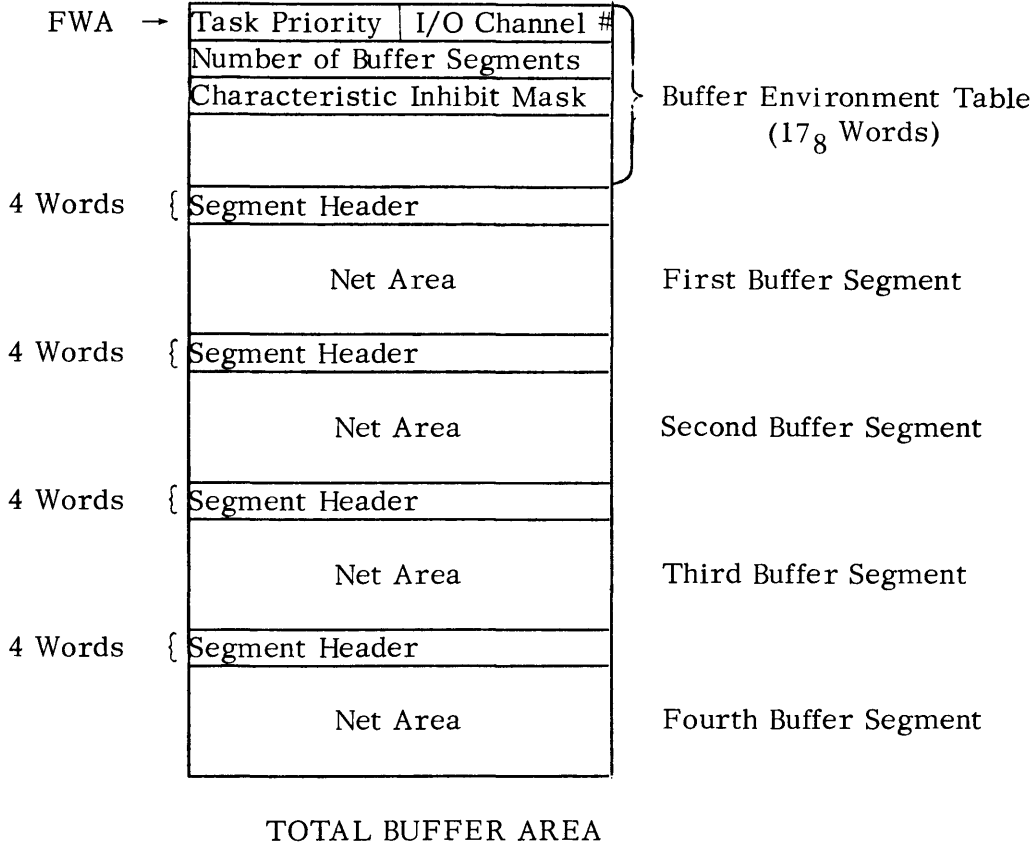
Word FWA+2, also constructed by the user, contains a characteristic inhibit mask. This mask word is used to inhibit device characteristics indicated by bit positions in a mask word. The bit definitions in this mask word are as follows:

<u>Bit</u>	<u>Mnemonic</u>	<u>Meaning</u>
1B0	DCSPL	Spooling control (RDOS only).
1B1	DCC80	Card input (80-column) device.
1B2	DCLTU	Device changing lower case ASCII to upper case.
1B3	DCFFO	Device requiring form feeds on opening.
1B4	DCFWD	Full word device (reads/writes more than one byte).
1B5	DCSPO	Spoolable device (RDOS only).
1B6	DCLAC	Output device requiring line feeds after carriage returns.
1B7	DCPCK	Input device requiring even parity check; output device requiring even parity computation.
1B8	DCRAT	Output device requiring a rubout after every tab.
1B9	DCNAF	Output device requiring nulls after every form feed.
1B10	DCKEY	A keyboard input device.
1B11	DCTO	A teletypewriter output device.
1B12	DCCNF	Output device without form feed hardware.
1B13	DCIDI	Device requiring operator intervention (RDOS only).
1B14	DCCGN	Output device without tabbing hardware.
1B15	DCCPO	Output device requiring leader/trailer.

Each buffer subdivision or segment will have its first four words reserved for a segment header, also used by BFPKG. Thus, the net buffer space available for data transfers is calculated as follows:

$$\text{Net Buffer Space} = \text{Total Buffer Area} - 218 - 4 * \text{number of buffer segments}$$

The following illustration shows a typical buffer area structure:



Thus, if the total buffer area consists of 117 decimal words and four segments are to be used, the net buffer space would be equal to 84 words, 21 words per buffer segment.

BUFFER PACKAGE SUBROUTINE USAGE

Open a File for Buffered I/O

Before a file can participate in a buffered I/O transfer, the file must be opened. There are four file opening routines in BFPKG, one for each type of I/O transfer. The four symbolic entries to the open routines are:

- | | |
|-------|--|
| BFRSO | Open a file to be read sequentially |
| BFRLO | Open a file to be read line by line |
| BFWSO | Open a file to be written sequentially |
| BFWLO | Open a file to be written line by line |

Open a File for Buffered I/O (Continued)

AC0 must contain a byte pointer to the file name. AC1 contains the integer word size of the total buffer area. AC2 contains the FWA of the buffer area, i. e., the starting address of the total buffer area. The format of the open command is:

```
.EXTN BFRSO (BFRLO, BFWSO, or BFWLO)
.BFRSO: BFRSO (.BFRLO: BFRLO, etc.)
```

(AC0) = byte pointer to file name

(AC1) = word size of buffer

(AC2) = FWA of the buffer area

JSR @.BFRSO (BFRLO, etc.)

error return

normal return

Accumulators are restored upon a normal return.

Possible errors resulting from an attempted open are identical to those which can occur from issuing a .SYSTEM .OPEN command:

<u>AC2</u>	<u>Mnemonic</u>	<u>Meaning</u>
0	ERFNO	Illegal channel number.
1	ERFNM	Illegal file name.
12	ERDLE	File does not exist.
21	ERUFT	Attempt to use a channel which is already in use.

Having decided to use buffered I/O calls for processing a file, the user is cautioned not to attempt opening or closing that file by means of conventional system open and close commands. BFPKG calls and their system counterparts may not be mixed in processing a given file.

Access a File Opened for Buffered I/O (BFACS)

Having opened a file for buffered I/O transfers, the specified type of transfer is initiated by means of the buffer access command, BFACS. Before issuing the call, AC0 must contain a byte pointer to the user data area to or from which the transfers will ultimately occur. If the file is open for sequential transfers, AC1 must contain the number of bytes which are to be transferred. AC2 contains the FWA of that buffer area which will be used during the transfer. This must be the same address as was input to the open command for this transfer. The format of the buffer access command is as follows:

Access a File Opened for Buffered I/O (BFACS) (Continued)

.EXTN BFACS
.BFACS: BFACS

(AC0) = byte pointer to user buffer area
(AC1) = number of bytes (for sequential transfers only)
(AC2) = FWA of the buffer area
JSR @ .BFACS
error return
normal return

Accumulators are restored upon a normal return.

All the errors which could result from the use of system calls to read or write lines or sequential bytes can occur when this command is invoked. A complete list of these errors follows.

<u>AC2</u>	<u>Mnemonic</u>	<u>Meaning</u>
0	ERFNO	Illegal channel number.
3	ERICD	Illegal command for the device.
5	ERWRO	Attempt to write an existent file.
6	EREOF	End of file.
7	ERRPR	Attempt to read a read protected file.
10	ERWPR	Attempt to write a write protected file.
15	ERFOP	Attempt to reference an unopened file.
22	ERLLI	Line limit (133 characters) exceeded.
24	ERPAR	Parity error on read line.
26	ERMEM	Available memory exhausted.
27	ERSPC	Not sufficient buffer space given.
30	ERFIL	File read error.
47	ERSIM	Attempt to perform multiple reads or writes on a QTY line simultaneously.

Close a File Opened for Buffered I/O (BFCLS)

Upon completion of a specified buffered transfer, either more of the same type of transfer can be requested by another call to BFACS or the file may be closed. Closing the file by a call to BFCLS kills the buffer management task and releases the Buffer Environment Table and the buffer area, so all the area may be reused for other transfers, released to the system, etc. Before calling BFCLS, AC2 must be set to the FWA for this file. The format of this call is:

Close a File Opened for Buffered I/O (BFCLS) (Continued)

```
.EXTN BFCLS
.BFCLS:  BFCLS

          (AC2) = address of file FWA
          JSR @ .BFCLS
          error return
          normal return
```

Accumulators are restored upon a normal return.

Possible errors resulting from this call are:

<u>AC2</u>	<u>Mnemonic</u>	<u>Meaning</u>
0	ERFNO	Illegal channel number.
15	ERFOP	Attempt to reference a channel not in use.

EXAMPLE OF BUFFER PACKAGE USAGE

One example of advantageous use of the buffer package is the processing of a large data file a few bytes at a time. Rather than making a call to the operating system every time a few bytes of data is required, it is more efficient to buffer the data in the user's program area by making a call to the system to fill the buffer.

The following program illustration shows the portions of such a program which use the buffer package.

EXAMPLE OF BUFFER PACKAGE USAGE

.TITL EG
 .EXTN BFRSO,BFACS,BFCLS
 .NREL

```

START:  .           ;START OF PROGRAM
        .
        .
        LDA 0 NAME   ;GET FILE NAME
        LDA 1 BSIZE  ;GET BUFFER SIZE
        LDA 2 BFWA1  ;GET START OF BUFFER
        JSR 0.BFRSO  ;OPEN FOR READ SEQUENTIAL
        JSR ERROR    ;ERROR RETURN
        .
        .
        .
AGAIN:  LDA 0 BPTR   ;GET USER AREA BYTE POINTER
        LDA 1 C4     ;GET START OF BUFFER
        LDA 2 BFWA1  ;GET DATA FROM BUFFER
        JSR 0.BFACS  ;CHECK ERROR CODE
        JSR ERR1
        .
        .
        .
        JMP AGAIN   ;MORE TO COME
FINIS:  .
        .
        .
        LDA 2 BFWA1  ;GET START OF BUFFER
        JSR 0.BFCLS  ;NORMAL RETURN
        JSR ERROR
        .SYSTEM
        .RTN
ERR1:   LDA 1 EOF    ;GET END-OF-FILE CODE
        SUB# 1 2 SNR
        JMP 0 3      ;EOF ERROR IS OKAY
ERROR:  .SYSTEM     ;ERROR RETURN
        .ERTN
CHL=    2           ;I/O CHANNEL FOR INPUT

BPTR:   .+1*2
        .BLK 2
C4:     4
        .TXM 1
NAME:   .+1*2
        .TXT /DATAFILE/

BFWA1:  FWA1        ;START OF BUFFER AREA
FWA1:   50*400+CHL ;PRIORITY=50, CHANNEL # =CHL
        2          ;2 BUFFER SEGMENTS
        0          ;CHARACTERISTIC MASK
        .BLK 21*3  ;REST OF BUFFER ENVIRONMENT TABLE
        .BLK SIZE/2+4 ;2 BUFFER SEGMENTS "SIZE" BYTES LONG
        .BLK SIZE/2+4
        .END START
    
```

BFPKG PROGRAM LISTING

Following is the program listing of BFPKG as released with revision 4.00 of the Real Time Disk Operating System and revision 4.00 of the Real Time Operating System.

LICENSED MATERIAL - PROPERTY OF DATA GENERAL CORPORATION

0001 BFPKG MACRO REV 03.00

11:29:20 01/07/75

01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28

```
*****  
;  
; NAME: BFPKG.SR                PART NUMBER: 090-001297  
;  
; DESCRIPTION: BUFFER PACKAGE  
;  
; REVISION HISTORY:  
;      REV.          DATE  
;      00            01/09/73  
;      01            05/18/73  
;      02            11/30/73  
;  
; COPYRIGHT (C) DATA GENERAL CORPORATION, 1973  
; ALL RIGHTS RESERVED.  
; LICENSED MATERIAL - PROPERTY OF DATA GENERAL CORPORATION.  
*****
```

LEGENSED MATERIAL - PROPERTY OF DATA GENERAL CORPORATION

10002 BFPKG

```

01
02      000001      .IFE      MSW
03
04
05      .TITL      BFPKG      ; BUFFERED I/O PACKAGE
06
07      .RB        BFPKG.RB
08
09      .ENDC
10      000000      .IFN      MSW
11
12      .TITL      MBFPKG      ; MAPPED BUFFERED I/O PACKAGE
13
14      .RB        MBFPKG.RB
15
16      .ENDC
17
18      .ENT      BFRSO      ; OPEN FOR READ SEQ
19
20      .ENT      BFRLO      ; OPEN FOR READ LINE
21
22      .ENT      BFWSO      ; OPEN FOR WRITE SEQ
23
24      .ENT      BFWLO      ; OPEN FOR WRITE LINE
25
26      .ENT      BFACS      ; ACCESS THE BUFFERS
27
28      .ENT      BFCLS      ; CLOSE
29
30      .EXTN      .TASK      ,.KILL
31
32      .EXTN      .XMT      ,.XMTW
33
34      .EXTN      .REC
35
36      000000      .IFN      MSW
37
38      .EXTD      TMN1
39
40      .ENDC
41
42      .NREL

```

LICENSED MATERIAL - PROPERTY OF DATA GENERAL CORPORATION

!0003 BFPKG

```

01
02           ; DEFINE SOME DISPLACEMENTS
03
04           ; BUFFER ENVIRONMENT TABLE
05
06 000000 .DUSR  PRCHN=0           ; PRIORITY & CHANNEL #
07 000001 .DUSR  NBFSG=1          ; NUMBER OF BUFFER SEGMENTS
08 000002 .DUSR  CHRIM=2          ; CHARACTERISTIC INHIBIT MASK
09 000003 .DUSR  CHNBR=3          ; CHANNEL # & FLAG BITS
10                                     ; MODE FLAGS:- LINE=1B0
11                                     ;                      SEQ=0B0
12                                     ;                      READ=0B1
13                                     ;                      WRITE=1B1
14 000004 .DUSR  BFAC0=4           ; AC0 STORAGE
15 000005 .DUSR  BFAC1=5           ; AC1  "
16 000006 .DUSR  BFAC2=6           ; AC2  "
17 000007 .DUSR  PCCRY=7           ; PROGRAM COUNTER + CARRY
18 000010 .DUSR  BFUSP=10          ; STORAGE FOR USER'S USP
19 000011 .DUSR  UBP=11            ; USER'S BYTE POINTER
20 000012 .DUSR  BYCT=12           ; BYTE COUNTER
21 000013 .DUSR  SEGSZ=13          ; SEGMENT SIZE
22 000014 .DUSR  ERFLG=14          ; ERROR FLAG
23 000015 .DUSR  IOAQ=15           ; I/O ACTIVE QUEUE
24 000016 .DUSR  IODQ=16           ; I/O DORMANT QUEUE
25 000017 .DUSR  BFTP0=17          ; COMMUNICATION CHANNEL 0
26 000020 .DUSR  BFTP1=20          ; TEMPORARY & CHANNEL 1
27
28           ; SEGMENT HEADER
29
30 000000 .DUSR  LINK=0             ; LINK WORD
31 000001 .DUSR  FBA=1             ; FIRST BYTE ADDRESS
32 000002 .DUSR  BP=2              ; BYTE POINTER
33 000003 .DUSR  LBA=3             ; LIMIT BYTE ADDRESS
34                                     ; (FIRST BYTE ADDRESS BEYOND SEGMENT)
35 000004 .DUSR  HDRLT=LBA-LINK+1  ; HEADER LENGTH

```

LICENSED MATERIAL - PROPERTY OF DATA GENERAL CORPORATION

10004 BFPKG

```

01
02           ; OPEN ROUTINES
03           ;
04           ; 4 OPENS          1. BFRSO - OPEN FOR READ SEQUENTIAL
05           ;                               2. BFRLO - OPEN FOR READ LINE
06           ;                               3. BFWSO - OPEN FOR WRITE SEQUENTIAL
07           ;                               4. BFWLO - OPEN FOR WRITE LINE
08
09           ; INPUTS:          AC0      - BYTE POINTER TO FILENAME
10           ;                               AC1      - SIZE OF BUFFER AREA IN WORDS
11           ;                               AC2      - FWA OF BUFFER AREA
12           ;
13           ; CHANNEL NUMBER IS RIGHT JUSTIFIED IN FWA OF BUFFER AREA
14           ; PRIORITY IS RIGHT JUSTIFIED IN LEFT BYTE OF THE SAME WORD
15           ;
16           ; NUMBER OF SEGMENTS (IF GREATER THAN 2) IS RIGHT
17           ; JUSTIFIED IN FWA+1. IF LINE MODE IS REQUESTED THE USEABLE
18           ; AREA WILL BE DIVIDED INTO THE MAXIMUM NUMBER OF
19           ; 66 DECIMAL WORD SEGMENTS POSSIBLE
20           ;
21           ; MINIMUM BUFFER SIZE IS 27 DECIMAL WORDS
22           ;
23           ; RETURNS:          CALL+1  - ERROR RETURN
24           ;                               CALL+2  - SUCCESSFUL OPEN
25           ;
26           ; IN THE EVENT OF AN ERROR, THE ERROR CODE WILL BE RETURNED
27           ; IN AC2
28
29
30 00000'175100 BFRSO:  MOVL    3,3           ; READ SEQ OPEN
31 00001'055007      STA    3,PCCRY,2
32 00002'006557      JSR    @.SAVX          ; SAVE STATUS -- CLEAR AC0
33 00003'000421      JMP    SEQOP
34
35 00004'175100 BFRLO:  MOVL    3,3           ; READ LINE OPEN
36 00005'055007      STA    3,PCCRY,2
37 00006'006553      JSR    @.SAVX          ; SAVE STATUS -- CLEAR AC0
38 00007'000413      JMP    LINOP
39
40 00010'175100 BFWSO:  MOVL    3,3           ; WRITE SEQ OPEN
41 00011'055007      STA    3,PCCRY,2
42 00012'006547      JSR    @.SAVX          ; SAVE STATUS -- CLEAR AC0
43 00013'101240      MOVOR   0,0
44 00014'101220      MOVZR   0,0
45 00015'000407      JMP    SEQOP
46
47 00016'175100 BFWLO:  MOVL    3,3           ; WRITE LINE OPEN
48 00017'055007      STA    3,PCCRY,2
49 00020'006541      JSR    @.SAVX          ; SAVE STATUS -- CLEAR AC0
50 00021'101240      MOVOR   0,0

```


LICENSED MATERIAL - PROPERTY OF DATA GENERAL CORPORATION

10005 BFPKG

```

01
02 00022'101240 LINOP:  MOVOR  0 0           ; LINE MODE ENTRY POINT
03 00023'126001          ADC    1 1 SKP
04 00024'025001 SEQOP:  LDA    1 NBFSQ 2       ; SEQUENTIAL MODE ENTRY POINT
05 00025'045020          STA    1 BFTP1 2       ; SET MODE FLAG INITIALLY
06 00026'025000 COMOP:  LDA    1,PRCHN,2
07 00027'034525          LDA    3,CHMSK
08 00030'167400          AND    3,1           ; ISOLATE CHANNEL #
09 00031'123000          ADD    1,0           ; ADD THE FLAG BITS
10 00032'041003          STA    0,CHNBR,2       ; RETURN IT TO THE BET
11 00033'021005          LDA    0,BFAC1,2       ; GET AVAILABLE SIZE
12 00034'024522          LDA    1,BETSZ
13 00035'122400          SUB    1,0           ; REDUCE BY SIZE OF BET
14 00036'025020          LDA    1,BFTP1,2
15 00037'124015          COM#   1,1,SNR         ; LINE MODE?
16 00040'000421          JMP    LINMD           ; YES
17 00041'176120          ADCZL  3,3
18 00042'174400          NEG    3,3           ; +2
19 00043'166433          SUBZ#  3,1,SNC        ; IS SEG COUNT => 2?
20 00044'165000          MOV    3,1           ; NO, MAKE EQUAL TO 2
21 00045'045020          STA    1,BFTP1,2       ; SAVE
22 00046'127120          ADDZL  1,1           ; SEGMENT COUNT * 4
23 00047'122400          SUB    1,0           ; REDUCE BY SEG HEADER SIZE
24 00050'101113          MOVL#  0,0,SNC        ; USEABLE AREA -VE ?
25 00051'162433          SUBZ#  3,0,SNC        ; USEABLE AREA => 2 WORDS ?
26 00052'000513          JMP    NES           ; NOPE
27 00053'025020          LDA    1,BFTP1,2       ; YES, GET SEGMENT COUNT
28 00054'004470          JSR    DIVI          ; DIVIDE THE USEABLE AREA
29 00055'175125          MOVZL  3,3,SNR        ; SEG SIZE IN BYTES (IS THERE AN
30 00056'000507          JMP    NES           ; NO
31 00057'055013          STA    3,SEGSZ,2       ; YES, SAVE THE SIZE
32 00060'000411          JMP    INIT          ; INIT THE SEGMENTS
33 00061'024502 LINMD:  LDA    1,LPHSZ       ; SIZE OF LINE + HEADER
34 00062'004462          JSR    DIVI          ; DIVIDE USEABLE AREA
35 00063'102120          ADCZL  0,0
36 00064'117112          ADDL#  0,3,SZC        ; SEGMENT COUNT => 2?
37 00065'000500          JMP    NES           ; NOPE
38 00066'055020          STA    3,BFTP1,2       ; YES, SAVE THE COUNT
39 00067'034475          LDA    3,MXLSZ       ; LINE SIZE IN BYTES
40 00070'055013          STA    3,SEGSZ,2       ; THAT'S THE SEGMENT SIZE

```

LICENSED MATERIAL - PROPERTY OF DATA GENERAL CORPORATION

10006 BFPKG

```

01
02
03 00071'024465 INIT: LDA 1,BETSZ
04 00072'147000 ADD 2,1 ; LINK WORD ADDRESS OF FIRST SEG
05 00073'045012 STA 1,BYCT,2 ; INTO THE BET (TEMPORARY)
06 00074'102400 SUB 0,0 ; UNARY ZERO
07 00075'041015 STA 0,IOAQ,2 ; CLEAR THE QUEUES
08 00076'041016 STA 0,IODQ,2
09 00077'041014 STA 0,ERFLG,2 ; CLEAR THE ERROR FLAG
10 00100'135000 INTLP: MOV 1,3
11 00101'024456 LDA 1,C4
12 00102'167120 ADDZL 3,1 ; FIRST BYTE ADDRESS
13 00103'045401 STA 1,FBA,3
14 00104'045402 STA 1,BP,3 ; BYTE POINTER
15 00105'021013 LDA 0,SEGSZ,2 ; SEGMENT SIZE
16 00106'107000 ADD 0,1 ; LIMIT BYTE ADDRESS
17 00107'045403 STA 1,LBA,3 ; INTO HEADER
18 00110'125220 MOVZR 1,1 ; WORD ADDRESS
19 00111'045400 STA 1,LINK,3 ; SET THE LINK
20 00112'015020 DSZ BFTP1,2 ; ANYMORE SEGMENTS?
21 00113'000765 JMP INTLP ; YES
22 00114'126400 SUB 1,1 ; NO
23 00115'045400 STA 1,LINK,3 ; CLEAR THE LINK
24
25
26
27
28 00116'021004 LDA 0,BFAC0,2 ; FILENAME BYTE POINTER
29 00117'025002 LDA 1,CHRIM,2 ; CHARACTERISTIC MASK
30 00120'035003 LDA 3,CHNBR,2 ; CHANNEL #
31 00121'030433 LDA 2,CHMSK ; CHANNEL # MASK
32 00122'173400 AND 3,2
33 00123'006017 .SYSTEM
34 00124'014077 .OPEN CPU
35 00125'000441 JMP NES+1
36 00126'171000 MOV 3,2
37 00127'021000 LDA 0,PRCHN,2 ; RECOVER PRIORITY
38 00130'024423 LDA 1,PRIMK
39 00131'123700 ANDS 1,0 ; RETAIN IN RIGHT BYTE
40 00132'035003 LDA 3,CHNBR,2 ; FLAG WORD
41 00133'025012 LDA 1,BYCT,2 ; FIRST QUEUE ENTRY
42 00134'177112 ADDL# 3,3,SZC ; READ OR WRITE ?
43 00135'045015 STA 1,IOAQ,2 ; WRITE, PUT ON PROGRESS Q
44 00136'177113 ADDL# 3,3,SNC ; TEST READ OR WRITE ?
45 00137'045016 STA 1,IODQ,2 ; READ, PUT ON DORMANT Q
46 00140'024420 LDA 1,TASK ; FWA OF TASK
47 00141'077777 .TASK
48 00142'000424 JMP NES+1
49 00143'002417 JMP @RTN1
50
51

```

LICENSED MATERIAL - PROPERTY OF DATA GENERAL CORPORATION

10007 BFPKG

01

02

03 00144'055017 DIVI: STA 3,BFTP0,2

04 00145'176401 SUB 3,3,SKP

05 00146'175400 INC 3,3

06 00147'122422 SUBZ 1,0,SZC

07 00150'000776 JMP .-2

08 00151'123000 ADD 1,0

09 00152'003017 JMP @BFTP0,2

10

11

12 00153'177400 PRIMK: 177400

13 00154'000077 CHMSK: 77

14 00155'000027 SPERR: ERSPC

15 00156'000021 BETSZ: BFTP1+1

16 00157'000004 C4: 4

17 00160'000476'TASK: .TSK

18 00161'000645'.SAVX: .SAVE

19 00162'000377'RTN1: .RTRN

20 00163'000107 LPHSZ: SCLLG/2+HDRLT+1

21 00164'000205 MXLSZ: SCLLG+1

22

23

24 00165'030770 NES: LDA 2,SPERR

25 00166'126000 ADC 1,1

26 00167'034016 .ERR: LDA 3,USP

27 00170'124015 COM# 1,1,SNR

28 00171'045403 STA 1,CHNBR,3

29 00172'051406 STA 2,BFAC2,3

30 00173'021407 LDA 0,PCCRY,3

31 00174'126120 ADCZL 1,1

32 00175'123000 ADD 1,0

33 00176'041407 STA 0,PCCRY,3

34 00177'002763 JMP @RTN1

35

36 00200'000200'TRMTB: .

37 00201'000015 15

38 00202'000014 14

39 00203'000000 0

40

; CARRIAGE RETURN
; FORM FEED
; NULL AND END OF TABLE

LICENSED MATERIAL - PROPERTY OF DATA GENERAL CORPORATION

10008 BFPKG

```

01          ; BUFFER ACCESS ROUTINES
02
03 00204'175100 BFACS:  MOVL    3,3          ; PC+CARRY
04 00205'055007          STA    3,PCCRY,2
05 00206'035003          LDA    3,CHNBR,2
06 00207'175112          MOVL#   3,3,SZC          ; LINE OR SEQUENTIAL?
07 00210'024754          LDA    1,MXLSZ          ; LINE
08 00211'006750          JSR    @.SAVX
09
10 00212'021004          LDA    0,BFAC0 2
11 00213'041011          STA    0,UBP,2
12 00214'045012          STA    .1,BYCT,2
13 00215'035015 TRLP:   LDA    3,IOAQ,2          ; IOAQ ENTRY
14 00216'175004          MOV    3,3,SZR          ; IS THERE A SEGMENT?
15 00217'000405          JMP    GO              ; YES
16 00220'020545          LDA    0,SGAD0        ; NO
17 00221'143000          ADD    2,0            ; FORM SIGNAL ADDRESS
18 00222'077777          .REC              ; KICK THE TASK AND
19 00223'000772          JMP    TRLP           ; WAIT FOR FREE SEGMENT
20
21          ; SEGMENT AVAILABLE FOR PROCESSING ON USER SIDE
22
23 00224'035402 GO:     LDA    3,BP,3          ; GET THE BYTE POINTER
24 00225'021003          LDA    0,CHNBR,2        ; CHANNEL #
25 00226'103112          ADDL#  0,0,SZC          ; READ OR WRITE?
26 00227'035011          LDA    3,UBP,2          ; WRITE, GET USER BYTE POINTER
27 00230'175200          MOVR   3,3            ; ADDRESS + CARRY SWITCH
28 00231'021400          LDA    0,0,3          ; SOURCE WORD
29 00232'024534          LDA    1,BTMSK         ; BYTE MASK ( RIGHT )
30 00233'101003          MOV    0,0,SNC         ; WHICH BYTE?
31 00234'101300          MOVS   0,0            ; LEFT
32 00235'123400          AND    1,0            ; ISOLATE THE BYTE (RIGHT JUSTIFIED)
33 00236'035015          LDA    3,IOAQ,2        ; GET THE Q ENTRY
34 00237'025003          LDA    1,CHNBR,2
35 00240'127113          ADDL#  1,1,SNC         ; READ OR WRITE?
36 00241'011402          ISZ   BP,3            ; READ, BUMP SYSTEM POINTER
37 00242'127112          ADDL#  1,1,SZC         ; JUST CHECKING
38 00243'011011          ISZ   UBP,2          ; WRITE, BUMP USER POINTER
39
40 00244'035015 STBYT:  LDA    3,IOAQ,2
41 00245'035402          LDA    3,BP,3          ; GET THE BYTE POINTER
42 00246'127113          ADDL#  1,1,SNC         ; READ OR WRITE
43 00247'035011          LDA    3,UBP,2          ; READ, USER POINTER IS DESTINATION
44 00250'175200          MOVR   3,3
45 00251'025400          LDA    1,0,3          ; DESTINATION WORD
46 00252'101003          MOV    0,0,SNC         ; WHICH BYTE
47 00253'105301          MOVS   0,1,SKP        ; LEFT
48 00254'107000          ADD    0,1            ; RIGHT
49 00255'045400          STA    1,0,3          ; PUT IT BACK
50 00256'035015          LDA    3,IOAQ,2
51 00257'025003          LDA    1,CHNBR,2
52 00260'127113          ADDL#  1,1,SNC         ; READ OR WRITE
53 00261'011011          ISZ   UBP,2          ; READ, BUMP USER POINTER
54 00262'127112          ADDL#  1,1,SZC         ; WRITE, BUMP SYSTEM POINTER
55 00263'011402          ISZ   BP,3            ; DEC THE COUNT
56 00264'015012          DSZ   BYCT,2          ; NO CATASTROPHE
57 00265'000401          JMP    .+1

```

LICENSED MATERIAL - PROPERTY OF DATA GENERAL CORPORATION

10009 BFPKG

```

01
02 00266'025003      LDA      1,CHNBR,2
03 00267'125113      MOVL#   1,1,SNC      ; LINE OR SEQUENTIAL?
04 00270'000414      JMP     SEGTST      ; SEQ, TEST SEGMENT
05 00271'034707      LDA      3,TRMTB    ; LINE, TEST FOR TERMINATOR
06 00272'175400      TRMLP:  INC      3,3
07 00273'025400      LDA      1,0,3
08 00274'122415      SUB#    1,0,SNR      ; IS CHAR A TERMINATOR
09 00275'000404      JMP     TRMFD       ; YES, KILL THE SEGMENT
10 00276'125004      MOV     1,1,SZR      ; NO, END OF TABLE?
11 00277'000773      JMP     TRMLP       ; NO, KEEP GOING
12 00300'000404      JMP     SEGTST
13
14 00301'035015      TRMFD:  LDA      3,IOAQ,2
15 00302'021402      LDA      0,BP,3
16 00303'041403      STA      0,LBA,3      ; SEGMENT IS DEAD
17
18 00304'035015      SEGTST: LDA      3,IOAQ,2
19 00305'021402      LDA      0,BP,3
20 00306'025403      LDA      1,LBA,3
21 00307'106032      ADCZ#   0,1,SZC      ; IS SEGMENT ALIVE?
22 00310'000430      JMP     BYTST       ; YES, TEST THE COUNT
23 00311'025400      LDA      1,LINK,3    ; NO
24 00312'125112      MOVL#   1,1,SZC      ; IS SEGMENT IN ERROR?
25 00313'000436      JMP     COMCT       ; YES, COMPUTE THE BYTE COUNT
26 00314'024514      LDA      1,IOAOF
27 00315'004544      JSR     DEQ         ; REMOVE FROM ACTIVE Q
28 00316'025014      LDA      1,ERFLG,2   ; GET ERROR FLAG
29 00317'125004      MOV     1,1,SZR      ; ANY ERRORS?
30 00320'000410      JMP     NOQ         ; YES, DROP SEGMENT ON FLOOR
31 00321'024510      LDA      1,IODOF
32 00322'004511      JSR     ENQ         ; PUT ON DORMANT Q
33 00323'020504      LDA      0,SGAD1
34 00324'143000      ADD     2,0
35 00325'126000      ADC     1,1
36 00326'077777      .XMT
37 00327'030016      LDA      2,USP      ; ERROR RETURN IS A NO-NO
38 00330'021003      NOQ:   LDA      0,CHNBR,2
39 00331'101113      MOVL#   0,0,SNC      ; LINE OR SEQ?
40 00332'000406      JMP     BYTST       ; SEQ
41 00333'025005      LDA      1,BFAC1,2   ; REQUESTED BYTE COUNT
42 00334'021012      LDA      0,BYCT,2    ; CURRENT BYTE COUNT
43 00335'106400      SUB     0,1          ; NUMBER OF BYTES TRANSFERRED
44 00336'045005      STA      1,BFAC1,2   ; RETURN IN AC1
45 00337'000440      JMP     .RTRN       ; RETURN
46
47 00340'021012      BYTST:  LDA      0,BYCT,2
48 00341'101004      MOV     0,0,SZR      ; BYTE COUNTER SATISFIED?
49 00342'000653      JMP     TRLP        ; NO
50 00343'021003      LDA      0,CHNBR,2   ; YES, DONE
51 00344'035015      LDA      3,IOAQ,2
52 00345'101113      MOVL#   0,0,SNC      ; LINE OR SEQUENTIAL
53 00346'000403      JMP     COMCT       ; SEQ, COMPUTE THE COUNT
54 00347'020463      LDA      0,LINER     ; LINE, GET LINE TOO LONG ERROR
55 00350'041400      STA      0,LINK,3    ; FLAG SEGMENT IN ERROR

```

LICENSED MATERIAL - PROPERTY OF DATA GENERAL CORPORATION

10010 BFPKG

```

01
02
03 00351'025005 COMCT: LDA 1,BFAC1,2 ; REQUESTED BYTE COUNT
04 00352'021012 LDA 0,BYCT,2 ; CURRENT BYTE COUNT
05 00353'106400 SUB 0,1 ; NUMBER OF BYTES TRANSFERRED
06 00354'045005 STA 1,BFAC1,2
07 00355'021402 LDA 0,BP,3
08 00356'025403 LDA 1,LBA,3
09 00357'122423 SUBZ 1,0,SNC ; END OF SEGMENT?
10 00360'000417 JMP .RTRN ; NO
11 00361'021400 LDA 0,LINK,3 ; YES, TEST FOR ERROR
12 00362'101102 MOVL 0,0,SZC ; ANY ERRORS?
13 00363'000404 JMP .RTNT ; YES
14 00364'000413 JMP .RTRN ; NO
15
16
17 00365'000017 SGAD0: BFTP0
18 00366'000377 BTMSK: 377
19
20
21 00367'023015 .RTNT: LDA 0,@IOAQ,2
22 00370'101100 MOVL 0,0
23 00371'101220 MOVZR 0,0 ; CLEAR BIT 0
24 00372'041006 STA 0,BFAC2,2 ; RETURN ERROR CODE
25 00373'025007 LDA 1,PCCRY,2 ; RETURN ADDRESS & CARRY
26 00374'102120 ADCZL 0,0 ; -2
27 00375'107000 ADD 0,1 ; RETURN TO CALL+1 ON ERROR
28 00376'045007 STA 1,PCCRY,2
29
30 000000 .RTRN: .IFN MSW
31
32 ISZ 1 ; SET SCHEDULER MODE IN MAPPED WORLD
33
34 .ENDC
35
36 00377'030016 LDA 2,USP
37 00400'034012 LDA 3,USTP ; USER STATUS TABLE POINTER
38 00401'035414 LDA 3,USTCT,3 ; CURRENT TCB
39 00402'021010 LDA 0,BFUSP,2 ; USP FROM BET
40 00403'040016 STA 0,USP
41 00404'041404 STA 0,TAC3,3 ; SET INTO TCB
42 00405'021007 LDA 0,PCCRY,2 ; PC + CARRY FROM BET
43 00406'101400 INC 0,0 ; BUMP RETURN
44 00407'101400 INC 0,0
45 00410'041400 STA 0,TPC,3 ; & SET INTO TCB
46 00411'021004 LDA 0,BFAC0,2 ; AC0
47 00412'041401 STA 0,TAC0,3
48 00413'025005 LDA 1,BFAC1,2 ; & AC1
49 00414'045402 STA 1,TAC1,3
50 00415'031006 LDA 2,BFAC2,2 ; & AC2
51 00416'051403 STA 2,TAC2,3
52
53 000000 .IFN MSW
54
55 JMP @.TMN1 ; RETURN VIA SCHEDULER
56
57 .ENDC
58
59 000001 .IFE MSW
60

```

LICENSED MATERIAL - PROPERTY OF DATA GENERAL CORPORATION

```

0011 BFPKG
01 00417'035400      LDA      3,TPC,3      ; RECOVER PC FROM TCB
02 00420'175220      MOVZR   3,3           ; RESTORE CARRY - FORM RETURN ADDRESS
03 00421'060277      INTDS                    ; MUST DISABLE
04 00422'054404      STA      3,RRLOC     ; RETURN ADDRESS IN SYSTEM PAGE ZERO
05 00423'034016      LDA      3,USP      ; GET USER STACK POINTER
06 00424'060177      INTEN                    ; LET 'EM GO, IGOR
07 00425'002401      JMP      @RRLOC      ; THIS JUMP IN MEMORY OF MR. PEABODY
08 00426'000000 RRLOC: 0
09
10                      .ENDC
11
12
13 00427'000020 SGAD1: BFTP1
14 00430'000015 IOAOF: IOAQ
15 00431'000016 IODOF: IODQ
16 00432'100022 LINER: 180+ERLLI
17          000000      .IFN      MSW
18
19          .TMN1:      TMN1
20
21                      .ENDC

```

LICENSED MATERIAL - PROPERTY OF DATA GENERAL CORPORATION

```

10012 BFPKG
01
02
03
04      ; ENQUEUE AN ENTRY
05      ;
06      ; AC0 - ADDRESS OF LINK WORD OF ENTRY
07      ;
08      ; AC1 - RELATIVE Q WORD ADDRESS
09
10
11 00433'133000 ENQ:   ADD      1,2      ; INDEX ON Q WORD
12 00434'025000      LDA      1,0,2      ; GET CURRENT Q ENTRY
13 00435'125005      MOV      1,1,SNR    ; IS Q EMPTY?
14 00436'000411      JMP      NOLNK      ; YES
15 00437'131000      MOV      1,2      ; NO, INDEX ON ENTRY
16 00440'025000      LDA      1,LINK,2    ; GET LINK WORD
17 00441'125005      MOV      1,1,SNR    ; IS THERE A LINK?
18 00442'000405      JMP      NOLNK      ; NO
19 00443'125112      MOVL#   1,1,SZC      ; ERROR SEGMENT?
20 00444'001400      JMP      0,3      ; YES, DON'T ENQUEUE
21 00445'131000      MOV      1,2      ; INDEX ON LINK
22 00446'000772      JMP      -6      ; GET NEXT ENTRY
23 00447'041000 NOLNK: STA      0,LINK,2    ; PUT NEW ENTRY ON THE Q
24 00450'111000      MOV      0,2      ; INDEX ON THE NEW ENTRY
25 00451'025001      LDA      1,FBA,2
26 00452'045002      STA      1,BP,2      ; INIT THE BYTE POINTER
27 00453'102400      SUB      0,0
28 00454'025000      LDA      1,LINK,2    ; GET LINK WORD
29 00455'125113      MOVL#   1,1,SNC      ; ERROR SEGMENT?
30 00456'041000      STA      0,LINK,2    ; NO, CLEAR ITS LINK
31 00457'030016      LDA      2,USP
32 00460'001400      JMP      0,3      ; RETURN
33
34
35      ; DEQUEUE AN ENTRY
36      ;
37      ; AC0 - ADDRESS OF ENTRY'S LINK WORD (RETURNED)
38      ;
39      ; AC1 - RELATIVE ADDRESS OF Q WORD
40
41 00461'133000 DEG:   ADD      1,2      ; INDEX ON Q WORD
42 00462'145000      MOV      2,1
43 00463'031000      LDA      2,0,2      ; GET CURRENT ENTRY
44 00464'141005      MOV      2,0,SNR    ; IS THERE AN ENTRY?
45 00465'000407      JMP      DEQRT      ; NO
46 00466'045002      STA      1,BP,2      ; YES, SAVE Q WORD ADDRESS IN THE
47 00467'025000      LDA      1,LINK,2    ; GET LINK WORD
48 00470'125112      MOVL#   1,1,SZC      ; SEGMENT IN ERROR?
49 00471'126400      SUB      1,1      ; YES, CLEAR THIS Q
50 00472'031002      LDA      2,BP,2      ; GET Q WORD ADDRESS
51 00473'045000      STA      1,0,2      ; MAKE LINE THE CURRENT ENTRY
52 00474'030016 DEGRT: LDA      2,USP
53 00475'001400      JMP      0,3
54

```


LICENSED MATERIAL - PROPERTY OF DATA GENERAL CORPORATION

10013 BFPKG

01

02 ; I/O PROCESSING TASK OF BUFERR PACKAGE

03

```

04 00476'035016 .TSK: LDA 3,IODQ,2 ; DORMANT Q ENTRY
05 00477'175004 MOV 3,3,SZR ; ANY DORMANT SEGMENTS
06 00500'000407 JMP SYSA ; YES
07 00501'020726 LDA 0,SGAD1
08 00502'143000 ADD 2,0 ; SIGNAL WORD ADDRESS
09 00503'000222' .REC
10 00504'124014 COM# 1 1 SZR ; -1 MEANS ANOTHER REQUEST
11 00505'077777 .KILL ; OTHERWISE CLOSE FUNCTION
12 00506'000770 JMP .TSK
13 00507'050016 SYSA: STA 2,USP
14 00510'021401 LDA 0,FBA,3 ; FIRST BYTE ADDRESS
15 00511'025013 LDA 1,SEGSZ,2 ; SEGMENT SIZE
16 00512'035003 LDA 3,CHNBR,2
17 00513'030526 LDA 2,.CMSK
18 00514'173400 AND 3,2
19 00515'177112 ADDL# 3,3,SZC ; READ OR WRITE?
20 00516'000414 JMP WRT ; WRITE
21 00517'175113 MOVL# 3,3,SNC ; READ--LINE OR SEQ?
22 00520'000405 JMP RDSQ
23 00521'006017 .SYSTEM ; LINE READ
24 00522'015477 .RDL CPU
25 00523'000404 JMP .TER0
26 00524'000404 JMP .TER0+1
27 00525'006017 RDSQ: .SYSTEM ; SEQUENTIAL READ
28 00526'015077 .RDS CPU
29 00527'004526 .TER0: JSR ERROR
30 00530'030016 LDA 2,USP
31 00531'000414 JMP COMTK
32
33 00532'175113 WRT: MOVL# 3,3,SNC ; LINE OR SEQ?
34 00533'000405 JMP WRSQ
35 00534'006017 .SYSTEM ; LINE WRITE
36 00535'017077 .WRL CPU
37 00536'000404 JMP .TER1
38 00537'000404 JMP .TER1+1
39 00540'006017 WRSQ: .SYSTEM ; SEQUENTIAL WRITE
40 00541'016477 .WRS CPU
41 00542'004513 .TER1: JSR ERROR
42 00543'030016 LDA 2,USP
43 00544'025013 LDA 1,SEGSZ,2
44 00545'035016 COMTK: LDA 3,IODQ,2
45 00546'123000 ADD 1,0
46 00547'041403 STA 0,LBA,3 ; INIT LIMIT BYTE ADDRESS
47 00550'024661 LDA 1,IODOF
48 00551'004710 JSR DEQ ; REMOVE FROM DORMANT Q
49 00552'024656 LDA 1,IOAOF
50 00553'004660 JSR ENQ ; PUT ON ACTIVE Q
51 00554'020611 LDA 0,SGAD0 ; GET THE OFFSET
52 00555'143000 ADD 2,0 ; FORM THE ADDRESS
53 00556'126000 ADC 1,1
54 00557'000326' .XMT ; SIGNAL THE PACKAGE
55 00560'030016 LDA 2,USP ; ERROR RETURNS ARE A NO-NO
56 00561'000715 JMP .TSK

```

LICENSED MATERIAL - PROPERTY OF DATA GENERAL CORPORATION

10014 BFPKG

01

02

03

; BUFFER CLOSE FUNCTION PROCESSING

04

```

05 00562'175100 BFCLS:  MOVL    3,3
06 00563'055007          STA    3,PCCRY,2
07 00564'004461          JSR    .SAVE
08 00565'021003          LDA    0,CHNBR,2
09 00566'100015          COM#   0,0,SNR          ; CHECK IF CHANNEL HAS BEEN OPENED
10 00567'000441          JMP    CHCL1          ; NO--JUST EXIT
11 00570'103113          ADDL#  0,0,SNR          ; READ OR WRITE?
12 00571'000432          JMP    CHCLS          ; READ, JUST CLOSE THE CHANNEL
13 00572'102400          SUB    0,0
14 00573'041020          STA    0,BFTP1,2
15 00574'020633          LDA    0,SGAD1          ; SIGNAL ADDRESS OFFSET
16 00575'143000          ADD    2,0              ; SIGNAL WORD ADDRESS
17 00576'126000          ADC    1,1              ; SIGNAL
18 00577'077777          .XMTW
19 00600'030016          LDA    2,USP              ; VERY BAD
20 00601'035015          LDA    3,IOAQ,2          ; NO
21 00602'025400          LDA    1,LINK,3
22 00603'125102          MOVL   1,1,SZC          ; ANY ERRORS
23 00604'002440          JMP    @RTN3              ; YES
24 00605'025401          LDA    1,FBA,3
25 00606'035402          LDA    3,BP,3
26 00607'136405          SUB    1,3,SNR          ; ACTIVE DATA COUNT
27 00610'000413          JMP    CHCLS          ; NO DATA IN SEGMENT
28 00611'055013          STA    3,SEGSZ,2        ; SAVE IN BET
29 00612'024616          LDA    1,IOAOF
30 00613'004646          JSR    DEQ              ; REMOVE CURRENT ENTRY
31 00614'024615          LDA    1,IODOF
32 00615'004616          JSR    ENG              ; AND PUT ON DORMANT Q
33 00616'020611          LDA    0,SGAD1
34 00617'143000          ADD    2,0
35 00620'126000          ADC    1,1
36 00621'000577'        .XMTW
37 00622'030016          LDA    2,USP

```

38

; CLOSE I/O CHANNEL--KILL I/O HANDLING TASK

39

40

```

41 00623'020604 CHCLS:  LDA    0,SGAD1
42 00624'143000          ADD    2,0              ; SETUP SIGNAL WORD ADDRESS
43 00625'126520          SUBZL  1 1              ; SETUP TO KILL BUFFER TASK
44 00626'000621'        .XMTW
45 00627'030016          LDA    2,USP
46 00630'021003 CHCL1:  LDA    0,CHNBR,2
47 00631'126000          ADC    1,1
48 00632'045003          STA    1,CHNBR,2
49 00633'030406          LDA    2,.CMSK
50 00634'113400          AND    0,2              ; CHANNEL NUMBER ONLY
51 00635'006017          .SYSTEM
52 00636'014477          .CLOS  CPU
53 00637'002403          JMP    @.ER1
54 00640'002403          JMP    @RTN2
55 00641'000077 .CMSK:  77
56 00642'000167'.ER1:  .ERR
57 00643'000377'RTN2:  .RTRN
58 00644'000367'RTN3:  .RTNT

```

LICENSED MATERIAL - PROPERTY OF DATA GENERAL CORPORATION

10015 BFPKG

01

02

03 ; ROUTINE TO SAVE STATUS OF CALLING TASK IN BUFFER TABLE

04

05 ;ON RETURN:

06 ; AC0=0

07 ; AC1=ORIGINAL AC1 IN CALL

08 ; AC2=ADDRESS OF FWA OF BUFFER TABLE

09

10 00645'041004 .SAVE: STA 0,BFAC0,2

11 00646'045005 STA 1,BFAC1,2

12 00647'051006 STA 2,BFAC2,2

13 00650'020016 LDA 0,USP

14 00651'041010 STA 0,BFUSP,2

15 00652'050016 STA 2,USP

16 00653'102400 SUB 0,0

17 00654'001400 JMP 0,3

18

19 ;ERROR ROUTINE--SETS ERROR FLAG IN BUFFER TABLE

20

21 00655'141100 ERROR: MOVL 2,0

22 00656'101240 MOVOR 0,0

23 00657'030016 LDA 2,USP

24 00660'015014 DSZ ERFLG,2

25 00661'031016 LDA 2,IODQ,2

26 00662'041000 STA 0,LINK,2

27 00663'021001 LDA 0,FBA,2

28 00664'001400 JMP 0,3

29

30 .END

LICENSED MATERIAL - PROPERTY OF DATA GENERAL CORPORATION
 0016 BFPKG

ADC	102000		5/03	7/25	9/35	13/53	14/17	14/35	14/47
ADD	103000		5/09	6/04	6/16	7/08	7/32	8/17	8/48
			9/34	10/27	12/11	12/41	13/08	13/45	13/52
			14/16	14/34	14/42				
AND	103400		5/08	6/32	8/32	13/18	14/50		
BETSZ	000156'		5/12	6/03	7/15				
BFAC0	000004		3/14	6/28	8/10	10/46	15/10		
BFAC1	000005		3/15	5/11	9/41	9/44	10/03	10/06	10/48
			15/11						
BFAC2	000006		3/16	7/29	10/24	10/50	15/12		
BFACS	000204'	EN	2/26	8/03					
BFCLS	000562'	EN	2/28	14/05					
BFRL0	000004'	EN	2/20	4/35					
BFRSO	000000'	EN	2/18	4/30					
BFTP0	000017		3/25	7/03	7/09	10/17			
BFTP1	000020		3/26	5/05	5/14	5/21	5/27	5/38	6/20
			7/15	11/13	14/14				
BFUSP	000010		3/18	10/39	15/14				
BFWLO	000016'	EN	2/24	4/47					
BFWSO	000010'	EN	2/22	4/40					
BP	000002		3/32	6/14	8/23	8/36	8/41	8/55	9/15
			9/19	10/07	12/26	12/46	12/50	14/25	
BTMSK	000366'		8/29	10/18					
BYCT	000012		3/20	6/05	6/41	8/12	8/56	9/42	9/47
			10/04						
BYTST	000340'		9/22	9/40	9/47				
C4	000157'		6/11	7/16					
CHCL1	000630'		14/10	14/46					
CHCLS	000623'		14/12	14/27	14/41				
CHMSK	000154'		5/07	6/31	7/13				
CHNBR	000003		3/09	5/10	6/30	6/40	7/28	8/05	8/24
			8/34	8/51	9/02	9/38	9/50	13/16	14/08
			14/46	14/48					
CHRIM	000002		3/08	6/29					
COM	100000		5/15	7/27	13/10	14/09			
COMCT	000351'		9/25	9/53	10/03				
COMOP	000026'		5/06						
COMTK	000545'		13/31	13/44					
CPU	000077		6/34	13/24	13/28	13/36	13/40	14/52	
DEQ	000461'		9/27	12/41	13/48	14/30			
DEQRT	000474'		12/45	12/52					
DIVI	000144'		5/28	5/34	7/03				
DSZ	014000		6/20	8/56	15/24				
ENQ	000433'		9/32	12/11	13/50	14/32			
ERFLG	000014		3/22	6/09	9/28	15/24			
ERLLI	000022		11/16						
ERROR	000655'		13/29	13/41	15/21				
ERSPC	000027		7/14						
FBA	000001		3/31	6/13	12/25	13/14	14/24	15/27	
GO	000224'		8/15	8/23					
HDRLT	000004		3/35	7/20					
INC	101400		7/05	9/06	10/43	10/44			
INIT	000071'		5/32	6/03					
INTDS	060277		11/03						
INTEN	060177		11/06						
INTLP	000100'		6/10	6/21					
IOAOF	000430'		9/26	11/14	13/49	14/29			
IOAQ	000015		3/23	6/07	6/43	8/13	8/33	8/40	8/50
			9/14	9/18	9/51	10/21	11/14	14/20	

LICENSED MATERIAL - PROPERTY OF DATA GENERAL CORPORATION
0017 BFPKG

IODOF 000431'	9/31	11/15	13/47	14/31				
IODQ 000016	3/24	6/08	6/45	11/15	13/04	13/44	15/25	
ISZ 010000	8/36	8/38	8/53	8/55	10/32			
JMP 000000	4/33	4/38	4/45	5/16	5/26	5/30	5/32	
	5/37	6/21	6/35	6/48	6/49	7/07	7/09	
	7/34	8/15	8/19	8/57	9/04	9/09	9/11	
	9/12	9/22	9/25	9/30	9/40	9/45	9/49	
	9/53	10/10	10/13	10/14	10/55	11/07	12/14	
	12/18	12/20	12/22	12/32	12/45	12/53	13/06	
	13/12	13/20	13/22	13/25	13/26	13/31	13/34	
	13/37	13/38	13/56	14/10	14/12	14/23	14/27	
	14/53	14/54	15/17	15/28				
JSR 004000	4/32	4/37	4/42	4/49	5/28	5/34	8/08	
	9/27	9/32	13/29	13/41	13/48	13/50	14/07	
	14/30	14/32						
LBA 000003	3/33	3/35	6/17	9/16	9/20	10/08	13/46	
LDA 020000	5/04	5/06	5/07	5/11	5/12	5/14	5/27	
	5/33	5/39	6/03	6/11	6/15	6/28	6/29	
	6/30	6/31	6/37	6/38	6/40	6/41	6/46	
	7/24	7/26	7/30	8/05	8/07	8/10	8/13	
	8/16	8/23	8/24	8/26	8/28	8/29	8/33	
	8/34	8/40	8/41	8/43	8/45	8/50	8/51	
	9/02	9/05	9/07	9/14	9/15	9/18	9/19	
	9/20	9/23	9/26	9/28	9/31	9/33	9/37	
	9/38	9/41	9/42	9/47	9/50	9/51	9/54	
	10/03	10/04	10/07	10/08	10/11	10/21	10/25	
	10/36	10/37	10/38	10/39	10/42	10/46	10/48	
	10/50	11/01	11/05	12/12	12/16	12/25	12/28	
	12/31	12/43	12/47	12/50	12/52	13/04	13/07	
	13/14	13/15	13/16	13/17	13/30	13/42	13/43	
	13/44	13/47	13/49	13/51	13/55	14/08	14/15	
	14/19	14/20	14/21	14/24	14/25	14/29	14/31	
	14/33	14/37	14/41	14/45	14/46	14/49	15/13	
	15/23	15/25	15/27					
LINER 000432'	9/54	11/16						
LINK 000000	3/30	3/35	6/19	6/23	9/23	9/55	10/11	
	12/16	12/23	12/28	12/30	12/47	14/21	15/26	
LINMD 000061'	5/16	5/33						
LINOP 000022'	4/38	5/02						
LPHSZ 000163'	5/33	7/20						
MOV 101000	5/20	6/10	6/36	8/14	8/30	8/46	9/10	
	9/29	9/48	12/13	12/15	12/17	12/21	12/24	
	12/42	12/44	13/05					
MSW 000000	2/02	2/10	2/36	10/30	10/53	10/59	11/17	
MXLSZ 000164'	5/39	7/21	8/07					
NBFSQ 00000.1	3/07	5/04						
NEG 100400	5/18							
NES 000165'	5/26	5/30	5/37	6/35	6/48	7/24		
NOLNK 000447'	12/14	12/18	12/23					
NOQ 000330'	9/30	9/38						
PCCRY 000007	3/17	4/31	4/36	4/41	4/48	7/30	7/33	
	8/04	10/25	10/28	10/42	14/06			
PRCHN 000000	3/06	5/06	6/37					
PRIMK 000153'	6/38	7/12						
RDSQ 000525'	13/22	13/27						
RRLOC 000426'	11/04	11/07	11/08					
RTN1 000162'	6/49	7/19	7/34					
RTN2 000643'	14/54	14/57						

LICENSED MATERIAL - PROPERTY OF DATA GENERAL CORPORATION
 0018 BFPKG

RTN3	000644'	14/23	14/58					
SCLLG	000204	7/20	7/21					
SEGSZ	000013	3/21	5/31	5/40	6/15	13/15	13/43	14/28
SEGTS	000304'	9/04	9/12	9/18				
SEQOP	000024'	4/33	4/45	5/04				
SGAD0	000365'	8/16	10/17	13/51				
SGAD1	000427'	9/33	11/13	13/07	14/15	14/33	14/41	
SKP	000001	5/03	7/04	8/47				
SNC	000003	5/19	5/24	5/25	6/44	8/30	8/35	8/42
		8/46	8/52	9/03	9/39	9/52	10/09	12/29
		13/21	13/33	14/11				
SNR	000005	5/15	5/29	7/27	9/08	12/13	12/17	12/44
		14/09	14/26					
SPERR	000155'	7/14	7/24					
STA	040000	4/31	4/36	4/41	4/48	5/05	5/10	5/21
		5/31	5/38	5/40	6/05	6/07	6/08	6/09
		6/13	6/14	6/17	6/19	6/23	6/43	6/45
		7/03	7/28	7/29	7/33	8/04	8/11	8/12
		8/49	9/16	9/44	9/55	10/06	10/24	10/28
		10/40	10/41	10/45	10/47	10/49	10/51	11/04
		12/23	12/26	12/30	12/46	12/51	13/13	13/46
		14/06	14/14	14/28	14/48	15/10	15/11	15/12
		15/14	15/15	15/26				
STBYT	000244'	8/40						
SUB	102400	5/13	5/23	6/06	6/22	7/04	9/08	9/43
		10/05	12/27	12/49	14/13	14/26	15/16	
SYSA	000507'	13/06	13/13					
SZC	000002	5/36	6/42	7/06	8/06	8/25	8/37	8/54
		9/21	9/24	10/12	12/19	12/48	13/19	14/22
SZR	000004	8/14	9/10	9/29	9/48	13/05	13/10	
TAC0	000001	10/47						
TAC1	000002	10/49						
TAC2	000003	10/51						
TAC3	000004	10/41						
TASK	000160'	6/46	7/17					
TPC	000000	10/45	11/01					
TRLP	000215'	8/13	8/19	9/49				
TRMFD	000301'	9/09	9/14					
TRMLP	000272'	9/06	9/11					
TRMTB	000200'	7/36	9/05					
UBP	000011	3/19	8/11	8/26	8/38	8/43	8/53	
USP	000016	7/26	9/37	10/36	10/40	11/05	12/31	12/52
		13/13	13/30	13/42	13/55	14/19	14/37	14/45
		15/13	15/15	15/23				
USTCT	000014	10/38						
USTP	000012	10/37						
WRSQ	000540'	13/34	13/39					
WRT	000532'	13/20	13/33					
.CLOS	014400	14/52						
.CMSK	000641'	13/17	14/49	14/55				
.ER1	000642'	14/53	14/56					
.ERR	000167'	7/26	14/56					
.KILL	000505' XN	2/30	13/11					
.OPEN	014000	6/34						
.RDL	015400	13/24						
.RDS	015000	13/28						
.REC	000503' XN	2/34	8/18	13/09				
.RTNT	000367'	10/13	10/21	14/58				

LICENSED MATERIAL - PROPERTY OF DATA GENERAL CORPORATION
 0019 BFPKG

.RTRN	000377'	7/19	9/45	10/10	10/14	10/29	14/57
.SAVE	000645'	7/18	14/07	15/10			
.SAVX	000161'	4/32	4/37	4/42	4/49	7/18	8/08
.SYST	006017	6/33	13/23	13/27	13/35	13/39	14/51
.TASK	000141' XN	2/30	6/47				
.TER0	000527'	13/25	13/26	13/29			
.TER1	000542'	13/37	13/38	13/41			
.TSK	000476'	7/17	13/04	13/12	13/56		
.WRL	017000	13/36					
.WRS	016400	13/40					
.XMT	000557' XN	2/32	9/36	13/54			
.XMTW	000626' XN	2/32	14/18	14/36	14/44		



How Do You Like This Manual?

Title _____ No. _____

We wrote the book for you, and naturally we had to make certain assumptions about who you are and how you would use it. Your comments will help us correct our assumptions and improve our manuals. Please take a few minutes to respond.

If you have any comments on the software itself, please contact your Data General representative. If you wish to order manuals, consult the Publications Catalog (012-330).

Who Are You?

- EDP Manager
- Senior System Analyst
- Analyst/Programmer
- Operator
- Other _____

What programming language(s) do you use? _____

How Do You Use This Manual?

(List in order: 1 = Primary use)

_____ Introduction to the product

_____ Reference

_____ Tutorial Text

_____ Operating Guide

Do You Like The Manual?

Yes	Somewhat	No	
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Is the manual easy to read?
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Is it easy to understand?
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Is the topic order easy to follow?
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Is the technical information accurate?
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Can you easily find what you want?
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Do the illustrations help you?
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Does the manual tell you everything you need to know?

Comments?

(Please note page number and paragraph where applicable.)

Large empty box for comments.

From:

Name _____ Title _____ Company _____

Address _____ Date _____

FOLD DOWN

FIRST

FOLD DOWN

FIRST
CLASS
PERMIT
No. 26
Southboro
Mass. 01772

BUSINESS REPLY MAIL

No Postage Necessary if Mailed in the United States

Postage will be paid by:

Data General Corporation

Southboro, Massachusetts 01772

ATTENTION: Software Documentation

FOLD UP

SECOND

FOLD UP