

1 SUMMARY

This subroutine provides 'get/put' facilities for one or more direct access files through a single in-core buffer, so that actual input-output operations are often avoided. The buffer is divided into fixed-length 'pages' and all actual input-output is performed by transferring a page to or from a record of the file.

ATTRIBUTES — **Version:** 1.0.0. **Types:** OF01A. **Original date:** February 1983. **Origin:** J.K.Reid, Harwell.

2 HOW TO USE THE PACKAGE

2.1 The argument list and calling sequence (initialization)

This call must be made once in any run that uses OF01 and it must be made before any other calls.

```
CALL OF01A
```

2.2 The argument list and calling sequence (file declaration)

This call must be made for any file that is to be accessed through OF01 before any such access.

```
CALL OF01B( IFILE, LEN, LENW, IFLAG)
```

IFILE is an INTEGER variable which must be set to the unit number of the file. **IFILE** is not altered. If **IFILE** specifies a file that has already been declared then no action is taken. **Restriction:** **IFILE**>0.

LEN is an INTEGER variable that the user must set to the length of the actual file in pages (the default page length is 1024 Fortran storage units, see §2.5). **LEN** is not altered. **Restriction:** **LEN**>0.

LENW is an INTEGER variable that the user must set to the length in pages of the part of the file that has been written and is not to be regarded as having been overwritten by zeros. OF01 will behave as if pages beyond this contain integer zeros. **LENW** is not altered. If **LENW**<0 it will act as if **LENW**=0.

IFLAG is an INTEGER variable that the user need not set. On return it has one of the values:

0. Successful entry.
1. Error. The maximum possible number of files have already been declared (see **MFILE** in §2.5).
2. Error. **IFILE** or **LEN** is not positive.

2.3 The argument list and calling sequence (file closure)

This call declares to OF01 that no further access is to be made through OF01 for a particular file. Any information that belongs to the file but is in the buffer is transferred to the file itself. One line of output documenting the action is produced on stream LP (see §2.5).

```
CALL OF01C( IFILE, LENW)
```

IFILE is an INTEGER variable which must be set to the unit number of the file. **IFILE** is not altered. If **IFILE** does not specify a file that has been opened by OF01B then no action is taken.

LENW is an INTEGER variable that the user need not set. The subroutine will set it to the length in pages of the part of the file that has been written.

2.4 The argument list and calling sequence (file access)

This call provides actual access to a file.

```
CALL OF01D( IFILE, LOC, IARY, N, IOD, IFLAG)
```

IFILE is an INTEGER variable which must be set to the unit number of the file. It must have been declared by calling OF01B. IFILE is not altered.

LOC is an INTEGER variable which must be set to the address within the file of the data to be read or written. The file is addressed by Fortran storage units in the same way as an integer array. LOC is not altered. **Restriction:** LOC>0.

IARY is an INTEGER array of length at least N to or from which data transfer is made. It is altered only on a get call. It may also be an array of another type, though such use breaks the Fortran standard.

N is an INTEGER variable that the user must set to the number of words to be transferred. If N≤0 then no action is taken. N is not altered.

IOD is an INTEGER variable that the user must set to one to get data from the file and to any other value to put data onto the file. IOD is not altered.

IFLAG is an INTEGER variable that the user need not set. On return it has one of the values:

0. Successful entry.
3. Error. IFILE has not been declared to OF01 by an OF01B call.
4. Error. LOC is not positive.
5. Error. Attempt to access data beyond the end of the file.

2.5 Common

OF01 contains common areas of the form

```
COMMON/OF01E/LP, NPAGE, LPAGE, MFILE,
*          NCALL( 2 ), NIO( 2 ), NWD( 2 ),
*          IPRVT( 3 )
INTEGER PAGE
COMMON/OF01F/PAGE( 9, 20 )
INTEGER BUF
COMMON/OF01G/BUF( 20480 )
INTEGER FILE
COMMON/OF01H/FILE( 3, 6 )
```

LP is an INTEGER variable that holds the stream number for diagnostic messages. If LP≤0 then messages are suppressed. The default value is 6.

NPAGE is an INTEGER variable holding the number of pages in the in-core buffer. The default value is 20. It must not be altered after OF01A has been called.

LPAGE is an INTEGER variable holding the number of words (Fortran storage units) in each page of the buffer. The default value is 1024. It must not be altered after OF01A has been called. **Restriction** (Cray only): LPAGE must be a multiple of 512.

MFILE is an INTEGER variable holding the maximum number of files that can be treated at once. The default value is 6. It must not be altered after OF01A has been called.

NCALL is an INTEGER array of length 2 that records the total number of get/put calls of OF01D.

NIO is an INTEGER array of length 2 that records the number of get/put transfers of pages from/to a file (i.e. actual input/output operations).

NWD is an INTEGER array of length 2 that records the number of words got/put by calls of OF01D.

IPVT is an INTEGER array of length 3 used for internal communication by OF01. It must not be altered by the user.

PAGE is an INTEGER array of size (9, NPAGE) used to hold index information about the data in the buffer pages. Note that this must be redeclared if NPAGE is increased from its default value. It must not be altered by the user.

BUF is an INTEGER array of length NPAGE*LPAGE used for the in-core buffer. Note that this must be redeclared if NPAGE or LPAGE is increased from its default value. It must not be altered by the user.

FILE is an INTEGER array of size (3, MFILE) used to hold data on the files being handled. Note that this must be redeclared if MFILE is increased from its default value. It must not be altered by the user.

3 GENERAL INFORMATION

Use of common: uses common areas OF01E/F/G/H, see §2.5.

Workspace: provided in the common areas, see §2.5.

Other routines called directly: Uses the private subroutines OF01I/J. The Cray version calls SETPOS.

Input/output: diagnostic messages on unit LP (see §2.5) and actual I/O (by OF01J only) on random-access files whose unit numbers are passed in argument IFILE.

Restrictions: *IFILE*>0, *LOC*>0, *LEN*>0; (Cray only) LPAGE must be a multiple of 512.

4 METHOD

The most recently accessed pages of the files are held in the buffer. For each buffer page there is a record of the file unit number and of the page number within the file. Wanted pages are found quickly with the help of a simple hashing function, and hash clashes are resolved by holding doubly-linked lists of pages having identical hash codes. Once the buffer is full and another page is wanted, the least recently accessed buffer page is freed. It is identified quickly with the aid of a doubly-linked list of pages in order of last access (whenever a page is accessed it is removed from its old position and placed at the front). A flag is kept for each page to indicate whether it has changed since its entry into the buffer so that only pages which have been changed need be written to file when they are freed. On each call of OF01D all wanted pages that are in the buffer are accessed before those that are not in order to avoid freeing a page that may be needed.

The efficiency is application dependent. If this is important the user may try several values of NPAGE and LPAGE, monitoring the number of actual input/output operations recorded in array NIO.

5 EXAMPLE OF USE

The following simple example opens a file, puts 1,2,3,...,20 into positions 10,11,..., gets data from positions 1,2,...,40 and then closes it. The code is as follows:

```
INTEGER IARY(40)
C DEFINE FILE 1 TO HAVE 4 RECORDS OF LENGTH 1024 WORDS
  OPEN(1, STATUS=OLD, ACCESS=DIRECT, RECL=1024)
C INITIALIZE OF01
  CALL OF01A
C OPEN FILE 1 TO OF01
  CALL OF01B(1, 4, 0, IFLAG)
  DO 10 I=1, 20
    IARY(I)=I
10  CONTINUE
C WRITE 1,2,3,...,20 TO POSITIONS 10,11,... IN THE FILE
```

```
      CALL OF01D(1,10,IARY,20,2,IFLAG)
C GET FIRST 40 INTEGERS FROM FILE AND PRINT THEM
      CALL OF01D(1,1,IARY,40,1,IFLAG)
      WRITE(6,20)IARY
20    FORMAT(' CONTENTS OF IARY ARE'/(10I5))
C CLOSE FILE 1 TO OF01
      CALL OF01C(1,LENW)
      STOP
      END
```

The output produced is as follows:

CONTENTS OF IARY ARE

0	0	0	0	0	0	0	0	0	1
2	3	4	5	6	7	8	9	10	11
12	13	14	15	16	17	18	19	20	0
0	0	0	0	0	0	0	0	0	0

OF01C HAS CLOSED FILE 1 OF LENGTH 4 PAGES. 1 PAGES WERE USED.