Author(s) : J. Zoll
Library: KERNLIB, UNIX and VMS
Submitter :
Submitted: 19.09.1991
Language : Fortran + C
Revised:

Handle Fixed-length Records on Unix Streams

The routines of this package are an interface to the C library functions open, read, write, lseek, close, to permit a Fortran program to handle an unstructured Unix file as a string of fixed-length binary records. Both sequential and direct-access READ / WRITE can be simulated.

These routines are simple little interface routines, there is no book-keeping done of the files which have been opened, the properties of the file have to be specified on each call, and the user is responsible for the consistency of all his calls for a particular file.

Processing has to be different for a disk file or for a tape file; therefore the medium must be indicated in the calls. Also, a user could take the source of these routines and modify them to add other branches for special processing.

New files are opened with the default permissions 644; one may set different permissions by calling `CFPERM` just before calling `CFOPEN`, which resets to the default after every call.

Three parameters are common to almost all routines :

```
LUNDES  is the file-descriptor of C to identify the file;
        with CFOPEN this is an output parameter,
        for all other routines it is an input parameter.

MEDIUM  = 0  for disk file, normal
          1      tape file, normal
          2      disk file, user coded I/O
          3      tape file, user coded I/O

NWREC   is the number of machine words for each one
        of the fixed-length records.
```

In the examples below it is assumed that for a given file these three parameters are available in something like `COMMON` storage.

Structure:

`SUBROUTINE` subprograms
User Entry Names: `CFOPEN`, `CFGET`, `CFPUT`, `CFSIZE`, `CFTELL`, `CFSEEK`, `CFREW`, `CFCLOS`, `CFPERM`
Files Referenced: Parameter

Usage:

Note: the symbol `*` designates output parameters.

Open a file

```
CALL CFOPEN(LUNDES,MEDIUM,NWREC, CHMODE, NBUF, CHNAME, ISTAT)

    LUNDES* file-descriptor returned

    CHMODE  CHARACTER string selecting the IO mode :

            = 'r'   open for reading
              'r+'  open for read/write
              'w'   create or truncate for writing
              'w+'  open for write/read, create or truncate
              'a'   append
              'a+'  open for append/read

            [ add the letter "l" if labeled tape,
              action on this is not yet implemented ]

    NBUF    not used for the time being, always give zero
    CHNAME  name of the file, CHARACTER variable
    ISTAT*  status, =zero if success
```

For example, create a new file in the current directory :

```
    MEDIUM = 0
    NWREC  = 900
    CALL CFOPEN(LUNDES,MEDIUM,NWREC, 'w', 0, 'run201.dat', ISTAT)
    IF(ISTAT .NE. 0)          GO TO trouble
```

Read next record

```
CALL CFGET(LUNDES,MEDIUM,NWREC, NWTAK, MBUF, ISTAT)

   *NWTAK*  input:  number of words to be read
            output: number of words actually read

    MBUF*   vector to be read into
    ISTAT*  status, = zero if success,
            = -1   if end-of-file
```

To simulate direct-access reading one has to call CFSEEK first.
For example:

```
 << if the 7th record of the file is to be read:
    CALL CFSEEK(LUNDES,MEDIUM,NWREC, 6, ISTAT)
    IF(ISTAT .NE. 0)              GO TO trouble   >>

    NWTAK = NWREC
    CALL CFGET(LUNDES,MEDIUM,NWREC, NWTAK, MBUF, ISTAT)
    IF(ISTAT .EQ.-1)         GO TO eof
    IF(ISTAT .NE. 0)         GO TO trouble
```

Write next record

```
CALL CFPUT(LUNDES,MEDIUM,NWREC, MBUF, ISTAT)

    MBUF    vector to be written, NWREC words
    ISTAT*  status, =zero if success
```

Get the size of the file

```
CALL CFSIZE(LUNDES,MEDIUM,NWREC, NRECT, ISTAT)

    NRECT*  number of records on the file
    ISTAT*  status, =zero if success

Careful : this will position the file to the end.
```

Get the current file position

```
CALL CFTELL(LUNDES,MEDIUM,NWREC, NRECC, ISTAT)

    NRECC*  number of records before current
    ISTAT*  status, =zero if success
```

Set the current file position

```
CALL CFSEEK(LUNDES,MEDIUM,NWREC, NRECC, ISTAT)

    NRECC   number of records before current
    ISTAT*  status, =zero if success
```

For example :

```
CALL CFSEEK(..., 0, ISTAT)    position to start-of-file
CALL CFSEEK(..., 6, ISTAT)    position to 7th record
use CFSIZE to position to end-of-file
```

Rewind the file

```
CALL CFREW(LUNDES,MEDIUM)
```

Close the file

```
CALL CFCLOS(LUNDES,MEDIUM)
```

Set the permissions for the next open

```
CALL CFPERM(IPERM)

    IPERM   the permissions as a decimal integer,
            as returned by STATF (Z265) for example
```

For example (using `NCOCTI` of `M432`) :

```
CALL CFPERM(NCOCTI('660'))

      set read and write for owner and group only.
```

•