

**NAME**

**funopen, fropen, fwopen** - open a stream

**LIBRARY**

Standard C Library (libc, -lc)

**SYNOPSIS**

**#include <stdio.h>**

*FILE \**

```
funopen(const void *cookie, int (*readfn)(void *, char *, int), int (*writefn)(void *, const char *, int),  
        fpos_t (*seekfn)(void *, fpos_t, int), int (*closefn)(void *));
```

*FILE \**

```
fropen(void *cookie, int (*readfn)(void *, char *, int));
```

*FILE \**

```
fwopen(void *cookie, int (*writefn)(void *, const char *, int));
```

**DESCRIPTION**

The **funopen()** function associates a stream with up to four "I/O functions". Either *readfn* or *writefn* must be specified; the others can be given as an appropriately-typed NULL pointer. These I/O functions will be used to read, write, seek and close the new stream.

In general, omitting a function means that any attempt to perform the associated operation on the resulting stream will fail. If the close function is omitted, closing the stream will flush any buffered output and then succeed.

The calling conventions of *readfn*, *writefn*, *seekfn* and *closefn* must match those, respectively, of `read(2)`, `write(2)`, `lseek(2)`, and `close(2)` with the single exception that they are passed the *cookie* argument specified to **funopen()** in place of the traditional file descriptor argument.

Read and write I/O functions are allowed to change the underlying buffer on fully buffered or line buffered streams by calling `setvbuf(3)`. They are also not required to completely fill or empty the buffer. They are not, however, allowed to change streams from unbuffered to buffered or to change the state of the line buffering flag. They must also be prepared to have read or write calls occur on buffers other than the one most recently specified.

All user I/O functions can report an error by returning -1. Additionally, all of the functions should set the external variable *errno* appropriately if an error occurs.

An error on **closefn()** does not keep the stream open.

As a convenience, the include file `<stdio.h>` defines the macros **fropen()** and **fwopen()** as calls to **funopen()** with only a read or write function specified.

## RETURN VALUES

Upon successful completion, **funopen()** returns a FILE pointer. Otherwise, NULL is returned and the global variable *errno* is set to indicate the error.

## ERRORS

[EINVAL]           The **funopen()** function was called without either a read or write function. The **funopen()** function may also fail and set *errno* for any of the errors specified for the routine `malloc(3)`.

## SEE ALSO

`fcntl(2)`, `open(2)`, `fclose(3)`, `fopen(3)`, `fopencookie(3)`, `fseek(3)`, `setbuf(3)`

## HISTORY

The **funopen()** functions first appeared in 4.4BSD.

## BUGS

The **funopen()** function may not be portable to systems other than BSD.

The **funopen()** interface erroneously assumes that *fpas\_t* is an integral type; see `fseek(3)` for a discussion of this issue.