*Full credit is given to the above companies including the OS that this PDF file was generated!*

## *Rocky Enterprise Linux 9.2 Manual Pages on command 'fread.3'*

***$ man fread.3***

FREAD(3)                    Linux Programmer's Manual                    FREAD(3)

NAME

fread, fwrite - binary stream input/output

SYNOPSIS

```
#include <stdio.h>

size_t fread(void *ptr, size_t size, size_t nmemb, FILE *stream);

size_t fwrite(const void *ptr, size_t size, size_t nmemb,

              FILE *stream);
```

DESCRIPTION

The function fread() reads nmemb items of data, each size bytes long,
from the stream pointed to by stream, storing them at the location
given by ptr.

The function fwrite() writes nmemb items of data, each size bytes long,
to the stream pointed to by stream, obtaining them from the location
given by ptr.

For nonlocking counterparts, see unlocked_stdio(3).

RETURN VALUE

On success, fread() and fwrite() return the number of items read or written. This number equals the number of bytes transferred only when size is 1. If an error occurs, or the end of the file is reached, the return value is a short item count (or zero).

The file position indicator for the stream is advanced by the number of bytes successfully read or written.

fread() does not distinguish between end-of-file and error, and callers must use feof(3) and ferror(3) to determine which occurred.

## ATTRIBUTES

For an explanation of the terms used in this section, see attributes(7).

```
?????????????????????????????????????????????
?Interface        ? Attribute     ? Value   ?
?????????????????????????????????????????????
?fread(), fwrite() ? Thread safety ? MT-Safe ?
?????????????????????????????????????????????
```

## CONFORMING TO

POSIX.1-2001, POSIX.1-2008, C89.

## EXAMPLES

The program below demonstrates the use of fread() by parsing /bin/sh ELF executable in binary mode and printing its magic and class:

```
$ ./a.out
ELF magic: 0x7f454c46
Class: 0x02
```

### Program source

```
#include <stdio.h>
#include <stdlib.h>
#define ARRAY_SIZE(arr) (sizeof(arr) / sizeof((arr)[0]))
int
main(void)
{
    FILE *fp = fopen("/bin/sh", "rb");
    if (!fp) {
        perror("fopen");
        return EXIT_FAILURE;
    }
    unsigned char buffer[4];
    size_t ret = fread(buffer, ARRAY_SIZE(buffer), sizeof(*buffer), fp);
    if (ret != sizeof(*buffer)) {
        fprintf(stderr, "fread() failed: %zu\n", ret);
        exit(EXIT_FAILURE);
    }
    printf("ELF magic: %#04x%02x%02x%02x\n", buffer[0], buffer[1],
           buffer[2], buffer[3]);
    ret = fread(buffer, 1, 1, fp);
    if (ret != 1) {
        fprintf(stderr, "fread() failed: %zu\n", ret);
        exit(EXIT_FAILURE);
    }
    printf("Class: %#04x\n", buffer[0]);
    fclose(fp);
    exit(EXIT_SUCCESS);
}
```

## SEE ALSO

read(2), write(2), feof(3), ferror(3), unlocked_stdio(3)

## COLOPHON

This page is part of release 5.10 of the Linux man-pages project. A description of the project, information about reporting bugs, and the latest version of this page, can be found at https://www.kernel.org/doc/man-pages/.