

NAME

posix_fallocate - pre-allocate storage for a range in a file

LIBRARY

Standard C Library (libc, -lc)

SYNOPSIS

```
#include <fcntl.h>
```

int

```
posix_fallocate(int fd, off_t offset, off_t len);
```

DESCRIPTION

Required storage for the range *offset* to *offset + len* in the file referenced by *fd* is guaranteed to be allocated upon successful return. That is, if **posix_fallocate()** returns successfully, subsequent writes to the specified file data will not fail due to lack of free space on the file system storage media. Any existing file data in the specified range is unmodified. If *offset + len* is beyond the current file size, then **posix_fallocate()** will adjust the file size to *offset + len*. Otherwise, the file size will not be changed.

Space allocated by **posix_fallocate()** will be freed by a successful call to `creat(2)` or `open(2)` that truncates the size of the file. Space allocated via **posix_fallocate()** may be freed by a successful call to `ftruncate(2)` that reduces the file size to a size smaller than *offset + len*.

RETURN VALUES

If successful, **posix_fallocate()** returns zero. It returns an error on failure, without setting *errno*.

ERRORS

Possible failure conditions:

- | | |
|----------|--|
| [EBADF] | The <i>fd</i> argument is not a valid file descriptor. |
| [EBADF] | The <i>fd</i> argument references a file that was opened without write permission. |
| [EFBIG] | The value of <i>offset + len</i> is greater than the maximum file size. |
| [EINTR] | A signal was caught during execution. |
| [EINVAL] | The <i>len</i> argument was less than or equal to zero, the <i>offset</i> argument was less than zero, or the operation is not supported by the file system. |

- [EIO] An I/O error occurred while reading from or writing to a file system.
- [EINTEGRITY] Corrupted data was detected while reading from the file system.
- [ENODEV] The *fd* argument does not refer to a file that supports **posix_fallocate**.
- [ENOSPC] There is insufficient free space remaining on the file system storage media.
- [ENOTCAPABLE] The file descriptor *fd* has insufficient rights.
- [ESPIPE] The *fd* argument is associated with a pipe or FIFO.

SEE ALSO

creat(2), ftruncate(2), open(2), unlink(2)

STANDARDS

The **posix_fallocate()** system call conforms to IEEE Std 1003.1-2004 ("POSIX.1").

HISTORY

The **posix_fallocate()** function appeared in FreeBSD 9.0.

AUTHORS

posix_fallocate() and this manual page were initially written by Matthew Fleming <mdf@FreeBSD.org>.