



Linux Ubuntu 26.04 Manual Pages on command 'posix_fallocate.3'

\$ man posix_fallocate.3

posix_fallocate(3) Library Functions Manual posix_fallocate(3)

NAME

posix_fallocate - allocate file space

LIBRARY

Standard C library (libc, -lc)

SYNOPSIS

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

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

Feature Test Macro Requirements for glibc (see feature_test_macros(7)):

```
posix_fallocate():
```

```
    _POSIX_C_SOURCE >= 200112L
```

DESCRIPTION

The function `posix_fallocate()` ensures that disk space is allocated for the file referred to by the file descriptor `fd` for the bytes in the range starting at `offset` and continuing for `size` bytes. After a successful call to `posix_fallocate()`, subsequent writes to bytes in the specified range are guaranteed not to fail because of lack of disk space.

If the size of the file is less than `offset+size`, then the file is increased to this size; otherwise the file size is left unchanged.

RETURN VALUE

`posix_fallocate()` returns zero on success, or an error number on failure. Note that `errno` is not set.

ERRORS

EBADF `fd` is not a valid file descriptor, or is not opened for writing.

? There is a race condition where concurrent file size increases by another thread or process could result in a file whose size is smaller than expected.

? If fd has been opened with the O_APPEND or O_WRONLY flags, the function fails with the error EBADF.

In general, the emulation is not MT-safe. On Linux, applications may use fallocate(2) if they cannot tolerate the emulation caveats. In general, this is only recommended if the application plans to terminate the operation if EOPNOTSUPP is returned, otherwise the application itself will need to implement a fallback with all the same problems as the emulation provided by glibc.

SEE ALSO

fallocate(1), fallocate(2), lseek(2), posix_fadvise(2)

Linux man-pages 6.17

2026-02-08

posix_fallocate(3)