



Red Hat Enterprise Linux Release 9.2 Manual Pages on 'posix_fallocate.3p' command

\$ man posix_fallocate.3p

POSIX_FALLOCATE(3P) POSIX Programmer's Manual POSIX_FALLOCATE(3P)

PROLOG

This manual page is part of the POSIX Programmer's Manual. The Linux implementation of this interface may differ (consult the corresponding Linux manual page for details of Linux behavior), or the interface may not be implemented on Linux.

NAME

posix_fallocate ? file space control (ADVANCED REALTIME)

SYNOPSIS

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

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

DESCRIPTION

The `posix_fallocate()` function shall ensure that any required storage for regular file data starting at `offset` and continuing for `len` bytes is allocated on the file system storage media. If `posix_fallocate()` returns successfully, subsequent writes to the specified file data shall not fail due to the lack of free space on the file system storage media.

If the `offset+len` is beyond the current file size, then `posix_fallocate()` shall adjust the file size to `offset+len`. Otherwise, the file size shall not be changed.

It is implementation-defined whether a previous `posix_fadvise()` call influences allocation strategy.

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

RETURN VALUE

Upon successful completion, `posix_fallocate()` shall return zero; otherwise, an error number shall be returned to indicate the error.

ERRORS

The `posix_fallocate()` function shall fail if:

EBADF The `fd` argument is not a valid file descriptor.

EBADF The `fd` argument references a file that was opened without write permission.

EFBIG The value of `offset+length` is greater than the maximum file size.

EINTR A signal was caught during execution.

EINVAL The `length` argument is less than zero, or the `offset` argument is less than zero, or the underlying file system does not support this operation.

EIO An I/O error occurred while reading from or writing to a file system.

ENODEV The `fd` argument does not refer to a regular file.

ENOSPC There is insufficient free space remaining on the file system storage media.

ESPIPE The `fd` argument is associated with a pipe or FIFO.

The `posix_fallocate()` function may fail if:

EINVAL The `length` argument is zero.

The following sections are informative.

EXAMPLES

None.

APPLICATION USAGE

The `posix_fallocate()` function is part of the Advisory Information option and need not be provided on all implementations.

RATIONALE

None.

FUTURE DIRECTIONS

None.

SEE ALSO

`creat()`, `ftruncate()`, `open()`, `unlink()`

The Base Definitions volume of POSIX.1-2017, `<fcntl.h>`

COPYRIGHT

Portions of this text are reprinted and reproduced in electronic form from IEEE Std 1003.1-2017, Standard for Information Technology -- Portable Operating System Interface (POSIX), The Open Group Base Specifications Issue 7, 2018 Edition, Copyright (C) 2018 by the Institute of Electrical and Electronics Engineers, Inc and The Open Group. In the event of any discrepancy between this version and the original IEEE and The Open Group Standard, the original IEEE and The Open Group Standard is the referee document. The original Standard can be obtained online at <http://www.opengroup.org/unix/online.html>.

Any typographical or formatting errors that appear in this page are most likely to have been introduced during the conversion of the source files to man page format. To report such errors, see https://www.kernel.org/doc/man-pages/reporting_bugs.html.

IEEE/The Open Group 2017 POSIX_FALLOCATE(3P)