

**NAME**

fallocate – preallocate or deallocate space to a file

**SYNOPSIS**

**fallocate** [-c|-p|-z] [-o *offset*] -l *length* [-n] *filename*

**fallocate** -d [-o *offset*] [-l *length*] *filename*

**fallocate** -x [-o *offset*] -l *length* *filename*

**DESCRIPTION**

**fallocate** is used to manipulate the allocated disk space for a file, either to deallocate or preallocate it. For filesystems which support the **fallocate**(2) system call, preallocation is done quickly by allocating blocks and marking them as uninitialized, requiring no IO to the data blocks. This is much faster than creating a file by filling it with zeroes.

The exit status returned by **fallocate** is 0 on success and 1 on failure.

**OPTIONS**

The *length* and *offset* arguments may be followed by the multiplicative suffixes KiB (=1024), MiB (=1024\*1024), and so on for GiB, TiB, PiB, EiB, ZiB, and YiB (the "iB" is optional, e.g., "K" has the same meaning as "KiB") or the suffixes KB (=1000), MB (=1000\*1000), and so on for GB, TB, PB, EB, ZB, and YB.

The options **---collapse-range**, **---dig-holes**, **---punch-hole**, and **---zero-range** are mutually exclusive.

**-c, ---collapse-range**

Removes a byte range from a file, without leaving a hole. The byte range to be collapsed starts at *offset* and continues for *length* bytes. At the completion of the operation, the contents of the file starting at the location *offset+length* will be appended at the location *offset*, and the file will be *length* bytes smaller. The option **---keep-size** may not be specified for the collapse-range operation.

Available since Linux 3.15 for ext4 (only for extent-based files) and XFS.

A filesystem may place limitations on the granularity of the operation, in order to ensure efficient implementation. Typically, *offset* and *length* must be a multiple of the filesystem logical block size, which varies according to the filesystem type and configuration. If a filesystem has such a requirement, the operation will fail with the error **EINVAL** if this requirement is violated.

**-d, ---dig-holes**

Detect and dig holes. This makes the file sparse in-place, without using extra disk space. The minimum size of the hole depends on filesystem I/O block size (usually 4096 bytes). Also, when using this option, **---keep-size** is implied. If no range is specified by **---offset** and **---length**, then the entire file is analyzed for holes.

You can think of this option as doing a "**cp ---sparse**" and then renaming the destination file to the original, without the need for extra disk space.

See **---punch-hole** for a list of supported filesystems.

**-i, ---insert-range**

Insert a hole of *length* bytes from *offset*, shifting existing data.

**-l, ---length *length***

Specifies the length of the range, in bytes.

**-n, --keep-size**

Do not modify the apparent length of the file. This may effectively allocate blocks past EOF, which can be removed with a truncate.

**-o, --offset *offset***

Specifies the beginning offset of the range, in bytes.

**-p, --punch-hole**

Deallocates space (i.e., creates a hole) in the byte range starting at *offset* and continuing for *length* bytes. Within the specified range, partial filesystem blocks are zeroed, and whole filesystem blocks are removed from the file. After a successful call, subsequent reads from this range will return zeroes. This option may not be specified at the same time as the **--zero-range** option. Also, when using this option, **--keep-size** is implied.

Supported for XFS (since Linux 2.6.38), ext4 (since Linux 3.0), Btrfs (since Linux 3.7), tmpfs (since Linux 3.5) and gfs2 (since Linux 4.16).

**-v, --verbose**

Enable verbose mode.

**-x, --posix**

Enable POSIX operation mode. In that mode allocation operation always completes, but it may take longer time when fast allocation is not supported by the underlying filesystem.

**-z, --zero-range**

Zeroes space in the byte range starting at *offset* and continuing for *length* bytes. Within the specified range, blocks are preallocated for the regions that span the holes in the file. After a successful call, subsequent reads from this range will return zeroes.

Zeroing is done within the filesystem preferably by converting the range into unwritten extents. This approach means that the specified range will not be physically zeroed out on the device (except for partial blocks at the either end of the range), and I/O is (otherwise) required only to update metadata.

Option **--keep-size** can be specified to prevent file length modification.

Available since Linux 3.14 for ext4 (only for extent-based files) and XFS.

**-h, --help**

Display help text and exit.

**-V, --version**

Print version and exit.

**AUTHORS**

[Eric Sandeen](mailto:sandeen@redhat.com) <sandeen@redhat.com>, [Karel Zak](mailto:kzak@redhat.com) <kzak@redhat.com>

**SEE ALSO**

**truncate(1)**, **fallocate(2)**, **posix\_fallocate(3)**

**REPORTING BUGS**

For bug reports, use the issue tracker at <https://github.com/util-linux/util-linux/issues>.

**AVAILABILITY**

The **fallocate** command is part of the util-linux package which can be downloaded from [Linux Kernel Archive](https://www.kernel.org/pub/linux/utils/util-linux/) <<https://www.kernel.org/pub/linux/utils/util-linux/>>.