

NAME

chattr - change file attributes on a Linux file system

SYNOPSIS

chattr [**-RVf**] [**-v** *version*] [*mode*] *files...*

DESCRIPTION

chattr changes the file attributes on a Linux file system.

The format of a symbolic mode is +=[aAcCdDeijsStTu].

The operator '+' causes the selected attributes to be added to the existing attributes of the files; '-' causes them to be removed; and '=' causes them to be the only attributes that the files have.

The letters 'aAcCdDeijsStTu' select the new attributes for the files: append only (a), no atime updates (A), compressed (c), no copy on write (C), no dump (d), synchronous directory updates (D), extent format (e), immutable (i), data journalling (j), secure deletion (s), synchronous updates (S), no tail-merging (t), top of directory hierarchy (T), and undeletable (u).

The following attributes are read-only, and may be listed by [lsattr\(1\)](#) but not modified by **chattr**: compression error (E), huge file (h), indexed directory (I), inline data (N), compression raw access (X), and compressed dirty file (Z).

Not all flags are supported or utilized by all filesystems; refer to filesystem-specific man pages such as [btrfs\(5\)](#), [ext4\(5\)](#), and [xfs\(5\)](#) for more filesystem-specific details.

OPTIONS

- R** Recursively change attributes of directories and their contents.
- V** Be verbose with **chattr**'s output and print the program version.
- f** Suppress most error messages.
- v** *version*
Set the file's version/generation number.

ATTRIBUTES

A file with the 'a' attribute set can only be open in append mode for writing. Only the superuser or a process possessing the CAP_LINUX_IMMUTABLE capability can set or clear this attribute.

When a file with the 'A' attribute set is accessed, its atime record is not modified. This avoids a certain amount of disk I/O for laptop systems.

A file with the 'c' attribute set is automatically compressed on the disk by the kernel. A read from this file returns uncompressed data. A write to this file compresses data before storing them on the disk. Note: please make sure to read the bugs and limitations section at the end of this document.

A file with the 'C' attribute set will not be subject to copy-on-write updates. This flag is only supported on file systems which perform copy-on-write. (Note: For btrfs, the 'C' flag should be set on new or empty files. If it is set on a file which already has data blocks, it is undefined when the blocks assigned to the file will be fully stable. If the 'C' flag is set on a directory, it will have no effect on the directory, but new files created in that directory will the No_COW attribute.)

A file with the 'd' attribute set is not candidate for backup when the **dump(8)** program is run.

When a directory with the 'D' attribute set is modified, the changes are written synchronously on the disk; this is equivalent to the 'dirsync' mount option applied to a subset of the files.

The 'e' attribute indicates that the file is using extents for mapping the blocks on disk. It may not be removed using [chattr\(1\)](#).

The 'E' attribute is used by the experimental compression patches to indicate that a compressed file has a compression error. It may not be set or reset using [chattr\(1\)](#), although it can be displayed by [lsattr\(1\)](#).

The 'h' attribute indicates the file is storing its blocks in units of the filesystem blocksize instead of in units of sectors, and means that the file is (or at one time was) larger than 2TB. It may not be set or reset using

`chattr(1)`, although it can be displayed by `lsattr(1)`.

A file with the 'i' attribute cannot be modified: it cannot be deleted or renamed, no link can be created to this file and no data can be written to the file. Only the superuser or a process possessing the CAP_LINUX_IMMUTABLE capability can set or clear this attribute.

The 'I' attribute is used by the htree code to indicate that a directory is being indexed using hashed trees. It may not be set or reset using `chattr(1)`, although it can be displayed by `lsattr(1)`.

A file with the 'j' attribute has all of its data written to the ext3 or ext4 journal before being written to the file itself, if the filesystem is mounted with the "data=ordered" or "data=writeback" options. When the filesystem is mounted with the "data=journal" option all file data is already journalled and this attribute has no effect. Only the superuser or a process possessing the CAP_SYS_RESOURCE capability can set or clear this attribute.

A file with the 'N' attribute set indicates that the file has data stored inline, within the inode itself. It may not be set or reset using `chattr(1)`, although it can be displayed by `lsattr(1)`.

When a file with the 's' attribute set is deleted, its blocks are zeroed and written back to the disk. Note: please make sure to read the bugs and limitations section at the end of this document.

When a file with the 'S' attribute set is modified, the changes are written synchronously on the disk; this is equivalent to the 'sync' mount option applied to a subset of the files.

A file with the 't' attribute will not have a partial block fragment at the end of the file merged with other files (for those filesystems which support tail-merging). This is necessary for applications such as LILO which read the filesystem directly, and which don't understand tail-merged files. Note: As of this writing, the ext2 or ext3 filesystems do not (yet, except in very experimental patches) support tail-merging.

A directory with the 'T' attribute will be deemed to be the top of directory hierarchies for the purposes of the Orlov block allocator. This is a hint to the block allocator used by ext3 and ext4 that the subdirectories under this directory are not related, and thus should be spread apart for allocation purposes. For example it is a very good idea to set the 'T' attribute on the /home directory, so that /home/john and /home/mary are placed into separate block groups. For directories where this attribute is not set, the Orlov block allocator will try to group subdirectories closer together where possible.

When a file with the 'u' attribute set is deleted, its contents are saved. This allows the user to ask for its undeletion. Note: please make sure to read the bugs and limitations section at the end of this document.

The 'X' attribute is used by the experimental compression patches to indicate that the raw contents of a compressed file can be accessed directly. It currently may not be set or reset using `chattr(1)`, although it can be displayed by `lsattr(1)`.

The 'Z' attribute is used by the experimental compression patches to indicate a compressed file is dirty. It may not be set or reset using `chattr(1)`, although it can be displayed by `lsattr(1)`.

AUTHOR

`chattr` was written by Remy Card <Remy.Card@linux.org>. It is currently being maintained by Theodore Ts'o <tytso@alum.mit.edu>.

BUGS AND LIMITATIONS

The 'c', 's', and 'u' attributes are not honored by the ext2, ext3, and ext4 filesystems as implemented in the current mainline Linux kernels.

The 'j' option is only useful if the filesystem is mounted as ext3 or ext4.

The 'D' option is only useful on Linux kernel 2.5.19 and later.

AVAILABILITY

`chattr` is part of the e2fsprogs package and is available from <http://e2fsprogs.sourceforge.net>.

SEE ALSO

`lsattr(1)`, `btrfs(5)`, `ext4(5)`, `xfs(5)`.