NAME

truncate, **ftruncate** - truncate or extend a file to a specified length

LIBRARY

Standard C Library (libc, -lc)

SYNOPSIS

#include <unistd.h>

int
truncate(*const char *path*, *off_t length*);

int
ftruncate(*int fd*, *off_t length*);

DESCRIPTION

The **truncate**() system call causes the file named by *path* or referenced by *fd* to be truncated or extended to *length* bytes in size. If the file was larger than this size, the extra data is lost. If the file was smaller than this size, it will be extended as if by writing bytes with the value zero.

The **ftruncate**() system call causes the file or shared memory object backing the file descriptor *fd* to be truncated or extended to *length* bytes in size. The file descriptor must be a valid file descriptor open for writing. The file position pointer associated with the file descriptor *fd* will not be modified.

RETURN VALUES

Upon successful completion, the value 0 is returned; otherwise the value -1 is returned and the global variable *errno* is set to indicate the error. If the file to be modified is not a directory or a regular file, the **truncate**() call has no effect and returns the value 0.

ERRORS

The **truncate**() system call succeeds unless:

[ENOTDIR] A component of the path prefix is not a directory.

[ENAMETOOLONG] A component of a pathname exceeded 255 characters, or an entire path name exceeded 1023 characters.

[ENOENT] The named file does not exist.

[EACCES] Search permission is denied for a component of the path prefix.

[EACCES] The named file is not writable by the user.

[ELOOP] Too many symbolic links were encountered in translating the pathname.

[EPERM] The named file has its immutable or append-only flag set, see the chflags(2) manual page for more information.

[EISDIR] The named file is a directory.

[EROFS] The named file resides on a read-only file system.

[ETXTBSY] The file is a pure procedure (shared text) file that is being executed.

[EFBIG] The *length* argument was greater than the maximum file size.

[EINVAL] The *length* argument was less than 0.

[EIO] An I/O error occurred updating the inode.

[EINTEGRITY] Corrupted data was detected while reading from the file system.

[EFAULT] The *path* argument points outside the process's allocated address space.

The **ftruncate**() system call succeeds unless:

[EBADF] The *fd* argument is not a valid descriptor.

[EINVAL] The *fd* argument references a file descriptor that is not a regular file or shared memory object.

[EINVAL] The *fd* descriptor is not open for writing.

SEE ALSO

chflags(2), open(2), shm_open(2)

HISTORY

The **truncate**() and **ftruncate**() system calls appeared in 4.2BSD.

BUGS

These calls should be generalized to allow ranges of bytes in a file to be discarded.

Historically, the use of **truncate**() or **ftruncate**() to extend a file was not portable, but this behavior became required in IEEE Std 1003.1-2008 ("POSIX.1").