## NAME

getxattr, lgetxattr, fgetxattr – retrieve an extended attribute value

## SYNOPSIS

```
#include <sys/types.h>
#include <sys/xattr.h>

ssize_t getxattr(const char *path, const char *name,
                 void *value, size_t size);
ssize_t lgetxattr(const char *path, const char *name,
                 void *value, size_t size);
ssize_t fgetxattr(int fd, const char *name,
                 void *value, size_t size);
```

## DESCRIPTION

Extended attributes are *name*:*value* pairs associated with inodes (files, directories, symbolic links, etc.). They are extensions to the normal attributes which are associated with all inodes in the system (i.e., the **stat**(2) data). A complete overview of extended attributes concepts can be found in **xattr**(7).

**getxattr**() retrieves the value of the extended attribute identified by *name* and associated with the given *path* in the filesystem. The attribute value is placed in the buffer pointed to by *value*; *size* specifies the size of that buffer. The return value of the call is the number of bytes placed in *value*.

**lgetxattr**() is identical to **getxattr**(), except in the case of a symbolic link, where the link itself is interrogated, not the file that it refers to.

**fgetxattr**() is identical to **getxattr**(), only the open file referred to by *fd* (as returned by **open**(2)) is interrogated in place of *path*.

An extended attribute *name* is a null-terminated string. The name includes a namespace prefix; there may be several, disjoint namespaces associated with an individual inode. The value of an extended attribute is a chunk of arbitrary textual or binary data that was assigned using **setxattr**(2).

If *size* is specified as zero, these calls return the current size of the named extended attribute (and leave *value* unchanged). This can be used to determine the size of the buffer that should be supplied in a subsequent call. (But, bear in mind that there is a possibility that the attribute value may change between the two calls, so that it is still necessary to check the return status from the second call.)

## RETURN VALUE

On success, these calls return a nonnegative value which is the size (in bytes) of the extended attribute value. On failure, −1 is returned and *errno* is set appropriately.

## ERRORS

**E2BIG** The size of the attribute value is larger than the maximum size allowed; the attribute cannot be retrieved. This can happen on filesystems that support very large attribute values such as NFSv4, for example.

**ENODATA**
The named attribute does not exist, or the process has no access to this attribute.

**ENOTSUP**
Extended attributes are not supported by the filesystem, or are disabled.

**ERANGE**
The *size* of the *value* buffer is too small to hold the result.

In addition, the errors documented in **stat**(2) can also occur.

## VERSIONS

These system calls have been available on Linux since kernel 2.4; glibc support is provided since version 2.3.

## CONFORMING TO

These system calls are Linux-specific.

## EXAMPLE

See **listxattr**(2).

## SEE ALSO

**getfattr**(1), **setfattr**(1), **listxattr**(2), **open**(2), **removexattr**(2), **setxattr**(2), **stat**(2), **symlink**(7), **xattr**(7)

## COLOPHON

This page is part of release 5.05 of the Linux *man-pages* project. A description of the project, information about reporting bugs, and the latest version of this page, can be found at https://www.kernel.org/doc/man−pages/.