NAME

extattr_delete_fd, **extattr_delete_file**, **extattr_delete_link**, **extattr_get_fd**, **extattr_get_file**, **extattr_get_link**, **extattr_list_fd**, **extattr_list_file**, **extattr_list_link**, **extattr_set_fd**, **extattr_set_file**, **extattr_set_link** - system calls to manipulate VFS extended attributes

LIBRARY

Standard C Library (libc, -lc)

SYNOPSIS

#include <sys/types.h>
#include <sys/extattr.h>

int
extattr_delete_fd(*int fd*, *int attrnamespace*, *const char *attrname*);

int
extattr_delete_file(*const char *path*, *int attrnamespace*, *const char *attrname*);

int
extattr_delete_link(*const char *path*, *int attrnamespace*, *const char *attrname*);

ssize_t
extattr_get_fd(*int fd*, *int attrnamespace*, *const char *attrname*, *void *data*, *size_t nbytes*);

ssize_t
extattr_get_file(*const char *path*, *int attrnamespace*, *const char *attrname*, *void *data*, *size_t nbytes*);

ssize_t
extattr_get_link(*const char *path*, *int attrnamespace*, *const char *attrname*, *void *data*, *size_t nbytes*);

ssize_t
extattr_list_fd(*int fd*, *int attrnamespace*, *void *data*, *size_t nbytes*);

ssize_t
extattr_list_file(*const char *path*, *int attrnamespace*, *void *data*, *size_t nbytes*);

ssize_t
extattr_list_link(*const char *path*, *int attrnamespace*, *void *data*, *size_t nbytes*);

ssize_t

extattr_set_fd(*int fd, int attrnamespace, const char *attrname, const void *data, size_t nbytes*);

ssize_t
extattr_set_file(*const char *path, int attrnamespace, const char *attrname, const void *data, size_t nbytes*);

ssize_t
extattr_set_link(*const char *path, int attrnamespace, const char *attrname, const void *data, size_t nbytes*);

DESCRIPTION

Named extended attributes are meta-data associated with vnodes representing files and directories. They exist as "name=value" pairs within a set of namespaces.

The **extattr_get_file**() system call retrieves the value of the specified extended attribute into a buffer pointed to by *data* of size *nbytes*. The **extattr_set_file**() system call sets the value of the specified extended attribute to the data described by *data*. The **extattr_delete_file**() system call deletes the extended attribute specified. The **extattr_list_file**() returns a list of attributes present in the requested namespace. Each list entry consists of a single byte containing the length of the attribute name, followed by the attribute name. The attribute name is not terminated by ASCII 0 (nul). The **extattr_get_file**() and **extattr_list_file**() calls consume the *data* and *nbytes* arguments in the style of read(2); **extattr_set_file**() consumes these arguments in the style of write(2).

If *data* is NULL in a call to **extattr_get_file**() and **extattr_list_file**() then the size of defined extended attribute data will be returned, rather than the quantity read, permitting applications to test the size of the data without performing a read. The **extattr_delete_link**(), **extattr_get_link**(), and **extattr_set_link**() system calls behave in the same way as their _file counterparts, except that they do not follow symlinks.

The **extattr_get_fd**(), **extattr_delete_fd**(), **extattr_list_fd**(), and **extattr_set_fd**() calls are identical to their "_file" counterparts except for the first argument. The "_fd" functions take a file descriptor, while the "_file" functions take a path. Both arguments describe a file associated with the extended attribute that should be manipulated. The "_fd" functions can be used with file descriptors opened with the O_PATH flag.

The following arguments are common to all the system calls described here:

attrnamespace the namespace in which the extended attribute resides; see extattr(9)

attrname the name of the extended attribute

Named extended attribute semantics vary by file system implementing the call. Not all operations may be supported for a particular attribute. Additionally, the format of the data in *data* is attribute-specific.

For more information on named extended attributes, please see extattr(9).

RETURN VALUES

If successful, the **extattr_get_fd**(), **extattr_get_file**(), **extattr_get_link**(), **extattr_list_fd**(), **extattr_list_file**(), **extattr_list_link**(), **extattr_set_fd**(), **extattr_set_file**(), and **extattr_set_link**() calls return the number of bytes that were read or written from the *data*, respectively. If *data* was NULL, then **extattr_get_fd**(), **extattr_get_file**(), **extattr_get_link**(), **extattr_list_fd**(), **extattr_list_file**(), and **extattr_list_link**() return the number of bytes available to read. If any of the calls are unsuccessful, the value -1 is returned and the global variable *errno* is set to indicate the error.

The **extattr_delete_file**() function returns the value 0 if successful; otherwise the value -1 is returned and the global variable *errno* is set to indicate the error.

ERRORS

The following errors may be returned by the system calls themselves. Additionally, the file system implementing the call may return any other errors it desires.

[EFAULT] The *attrnamespace* and *attrname* arguments, or the memory range defined by *data* and *nbytes* point outside the process's allocated address space.

[ENAMETOOLONG]
The attribute name was longer than EXTATTR_MAXNAMELEN.

The **extattr_get_fd**(), **extattr_set_fd**(), **extattr_delete_fd**(), and **extattr_list_fd**() system calls may also fail if:

[EBADF] The file descriptor referenced by *fd* was invalid.

Additionally, the **extattr_get_file**(), **extattr_set_file**(), and **extattr_delete_file**() calls may also fail due to the following errors:

[ENOATTR] The requested attribute was not defined for this file.

[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] A component of the path name that must exist does not exist.

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

SEE ALSO

extattr(3), getextattr(8), setextattr(8), extattr(9), VOP_GETEXTATTR(9), VOP_SETEXTATTR(9)

HISTORY

Extended attribute support was developed as part of the TrustedBSD Project, and introduced in FreeBSD 5.0. It was developed to support security extensions requiring additional labels to be associated with each file or directory.

CAVEATS

This interface is under active development, and as such is subject to change as applications are adapted to use it. Developers are discouraged from relying on its stability.

BUGS

In earlier versions of this API, passing an empty string for the attribute name to **extattr_get_fd**(), **extattr_get_file**(), or **extattr_get_link**() would return the list of attributes defined for the target object. This interface has been deprecated in preference to using the explicit list API, and should not be used.