

NAME

`attr_set`, `attr_setf` – set the value of a user attribute of a filesystem object

C SYNOPSIS

```
#include <attr/attributes.h>
```

```
int attr_set (const char *path, const char *attrname,
              const char *attrvalue, const int valuelength,
              int flags);
```

```
int attr_setf (int fd, const char *attrname,
               const char *attrvalue, const int valuelength,
               int flags);
```

DESCRIPTION

The `attr_set` and `attr_setf` functions provide a way to create attributes and set/change their values.

Path points to a path name for a filesystem object, and *fd* refers to the file descriptor associated with a file. If the attribute *attrname* does not exist, an attribute with the given name and value will be created and associated with that indicated filesystem object. If an attribute with that name already exists on that filesystem object, the existing value is replaced with the new value given in this call. The new attribute value is copied from the *attrvalue* buffer for a total of *valuelength* bytes. The *flags* argument can contain the following symbols bitwise OR'ed together:

ATTR_ROOT

Look for *attrname* in the **root** address space, not in the **user** address space. (limited to use by super-user only)

ATTR_DONTFOLLOW

Do not follow symbolic links when resolving a *path* on an `attr_set` function call. The default is to follow symbolic links.

ATTR_CREATE

Return an error (EEXIST) if an attribute of the given name already exists on the indicated filesystem object, otherwise create an attribute with the given name and value. This flag is used to implement a pure create operation, without this flag `attr_set` will create the attribute if it does not already exist. An error (EINVAL) will be returned if both ATTR_CREATE and ATTR_REPLACE are set in the same call.

ATTR_REPLACE

Return an error (ENOATTR) if an attribute of the given name does not already exist on the indicated filesystem object, otherwise replace the existing attribute's value with the given value. This flag is used to implement a pure replacement operation, without this flag `attr_set` will create the attribute if it does not already exist. An error (EINVAL) will be returned if both ATTR_CREATE and ATTR_REPLACE are set in the same call.

`attr_set` will fail if one or more of the following are true:

[ENOATTR]	The attribute name given is not associated with the indicated filesystem object and the ATTR_REPLACE flag bit was set.
[E2BIG]	The value of the given attribute is too large, it exceeds the maximum allowable size of an attribute value.
[EEXIST]	The attribute name given is already associated with the indicated filesystem object and the ATTR_CREATE flag bit was set.
[ENOENT]	The named file does not exist.
[EPERM]	The effective user ID does not match the owner of the file and the effective user ID is not super-user.

[ENOTDIR]	A component of the path prefix is not a directory.
[EACCES]	Search permission is denied on a component of the path prefix.
[EINVAL]	A bit was set in the <i>flag</i> argument that is not defined for this system call, or both the ATTR_CREATE and ATTR_REPLACE flags bits were set.
[EFAULT]	<i>Path</i> , <i>attrname</i> , or <i>attrvalue</i> points outside the allocated address space of the process.
[ELOOP]	A path name lookup involved too many symbolic links.
[ENAMETOOLONG]	The length of <i>path</i> exceeds {MAXPATHLEN}, or a pathname component is longer than {MAXNAMELEN}.

attr_setf will fail if:

[ENOATTR]	The attribute name given is not associated with the indicated filesystem object and the ATTR_REPLACE flag bit was set.
[E2BIG]	The value of the given attribute is too large, it exceeds the maximum allowable size of an attribute value.
[EEXIST]	The attribute name given is already associated with the indicated filesystem object and the ATTR_CREATE flag bit was set.
[EINVAL]	A bit was set in the <i>flag</i> argument that is not defined for this system call, or both the ATTR_CREATE and ATTR_REPLACE flags bits were set, or <i>fd</i> refers to a socket, not a file.
[EFAULT]	<i>Attrname</i> , or <i>attrvalue</i> points outside the allocated address space of the process.
[EBADF]	<i>Fd</i> does not refer to a valid descriptor.

DIAGNOSTICS

On success, zero is returned. On error, -1 is returned, and *errno* is set appropriately.

SEE ALSO

attr(1), attr_get(3), attr_list(3), attr_multi(3), attr_remove(3)