NAME

acl_set_file — set an ACL by filename

LIBRARY

Linux Access Control Lists library (libacl, −lacl).

SYNOPSIS

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

int
acl_set_file(const char *path_p, acl_type_t type, acl_t acl);
```

DESCRIPTION

The **acl_set_file**() function associates an access ACL with a file or directory, or associates a default ACL with a directory. The pathname for the file or directory is pointed to by the argument *path_p*.

The effective user ID of the process must match the owner of the file or directory or the process must have the CAP_FOWNER capability for the request to succeed.

The value of the argument *type* is used to indicate whether the access ACL or the default ACL associated with *path_p* is being set. If the *type* parameter is ACL_TYPE_ACCESS, the access ACL of *path_p* shall be set. If the *type* parameter is ACL_TYPE_DEFAULT, the default ACL of *path_p* shall be set. If the argument *type* specifies a type of ACL that cannot be associated with *path_p*, then the function fails.

The *acl* parameter must reference a valid ACL according to the rules described on the *acl_valid*(3) manual page if the *type* parameter is ACL_TYPE_ACCESS, and must either reference a valid ACL or an ACL with zero ACL entries if the *type* parameter is ACL_TYPE_DEFAULT. If the *acl* parameter references an empty ACL, then the **acl_set_file**() function removes any default ACL associated with the directory referred to by the *path_p* parameter.

RETURN VALUE

The **acl_set_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

If any of the following conditions occur, the **acl_set_file**() function returns `-1` and sets *errno* to the corresponding value:

[EACCES]	Search permission is denied for a component of the path prefix or the object exists and the process does not have appropriate access rights. Argument *type* specifies a type of ACL that cannot be associated with *path_p*.
[EINVAL]	The argument *acl* does not point to a valid ACL. The ACL has more entries than the file referred to by *path_p* can obtain. The *type* parameter is not ACL_TYPE_ACCESS or ACL_TYPE_DEFAULT. The *type* parameter is ACL_TYPE_DEFAULT, but the file referred to by *path_p* is not a directory.
[ENAMETOOLONG]	The length of the argument *path_p* is too long.
[ENOENT]	The named object does not exist or the argument *path_p* points to an empty string.
[ENOSPC]	The directory or file system that would contain the new ACL cannot be extended or the file system is out of file allocation resources.
[ENOTDIR]	A component of the path prefix is not a directory.
[ENOTSUP]	The file identified by *path_p* cannot be associated with the ACL because the file system on which the file is located does not support this.

[EPERM] The process does not have appropriate privilege to perform the operation to set the ACL.

[EROFS] This function requires modification of a file system which is currently read-only.

STANDARDS

IEEE Std 1003.1e draft 17 (“POSIX.1e”, abandoned)

The behavior of **acl_set_file**() when the *acl* parameter refers to an empty ACL and the *type* parameter is ACL_TYPE_DEFAULT is an extension in the Linux implementation, in order that all values returned by **acl_get_file**() can be passed to **acl_set_file**(). The POSIX.1e function for removing a default ACL is **acl_delete_def_file**().

SEE ALSO

acl_delete_def_file(3), *acl_get_file*(3), *acl_set_fd*(3), *acl_valid*(3), *acl*(5)

AUTHOR

Derived from the FreeBSD manual pages written by Robert N M Watson <rwatson@FreeBSD.org>, and adapted for Linux by Andreas Gruenbacher <andreas.gruenbacher@gmail.com>.