NAME

acl **-** virtual file system access control lists

SYNOPSIS

#include <sys/param.h>
#include <sys/vnode.h>
#include <sys/acl.h>

In the kernel configuration file:
options UFS_ACL

DESCRIPTION

Access control lists, or ACLs, allow fine-grained specification of rights for vnodes representing files and directories. However, as there are a plethora of file systems with differing ACL semantics, the vnode interface is aware only of the syntax of ACLs, relying on the underlying file system to implement the details. Depending on the underlying file system, each file or directory may have zero or more ACLs associated with it, named using the *type* field of the appropriate vnode ACL calls: VOP_ACLCHECK(9), VOP_GETACL(9), and VOP_SETACL(9).

Currently, each ACL is represented in-kernel by a fixed-size *acl* structure, defined as follows:

```
struct acl {
        unsigned int        acl_maxcnt;
        unsigned int        acl_cnt;
        int                 acl_spare[4];
        struct acl_entry    acl_entry[ACL_MAX_ENTRIES];
};
```

An ACL is constructed from a fixed size array of ACL entries, each of which consists of a set of permissions, principal namespace, and principal identifier. In this implementation, the *acl_maxcnt* field is always set to ACL_MAX_ENTRIES.

Each individual ACL entry is of the type *acl_entry_t*, which is a structure with the following members:

acl_tag_t ae_tag

The following is a list of definitions of ACL types to be set in *ae_tag*:

ACL_UNDEFINED_FIELD	Undefined ACL type.
ACL_USER_OBJ	Discretionary access rights for processes whose effective user ID matches the user ID of the file’s owner.

ACL_USER Discretionary access rights for processes whose effective user ID matches the ACL entry qualifier.

ACL_GROUP_OBJ Discretionary access rights for processes whose effective group ID or any supplemental groups match the group ID of the file’s owner.

ACL_GROUP Discretionary access rights for processes whose effective group ID or any supplemental groups match the ACL entry qualifier.

ACL_MASK The maximum discretionary access rights that can be granted to a process in the file group class. This is only valid for POSIX.1e ACLs.

ACL_OTHER Discretionary access rights for processes not covered by any other ACL entry. This is only valid for POSIX.1e ACLs.

ACL_OTHER_OBJ Same as ACL_OTHER.

ACL_EVERYONE Discretionary access rights for all users. This is only valid for NFSv4 ACLs.

Each POSIX.1e ACL must contain exactly one ACL_USER_OBJ, one ACL_GROUP_OBJ, and one ACL_OTHER. If any of ACL_USER, ACL_GROUP, or ACL_OTHER are present, then exactly one ACL_MASK entry should be present.

uid_t ae_id

The ID of user for whom this ACL describes access permissions. For entries other than ACL_USER and ACL_GROUP, this field should be set to ACL_UNDEFINED_ID.

acl_perm_t ae_perm

This field defines what kind of access the process matching this ACL has for accessing the associated file. For POSIX.1e ACLs, the following are valid:

ACL_EXECUTE The process may execute the associated file.

ACL_WRITE The process may write to the associated file.

ACL_READ The process may read from the associated file.

ACL_PERM_NONE The process has no read, write or execute permissions to the associated file.

For NFSv4 ACLs, the following are valid:

ACL_READ_DATA The process may read from the associated file.

ACL_LIST_DIRECTORY	Same as ACL_READ_DATA.
ACL_WRITE_DATA	The process may write to the associated file.
ACL_ADD_FILE	Same as ACL_ACL_WRITE_DATA.
ACL_APPEND_DATA	
ACL_ADD_SUBDIRECTORY	Same as ACL_APPEND_DATA.
ACL_READ_NAMED_ATTRS	Ignored.
ACL_WRITE_NAMED_ATTRS	Ignored.
ACL_EXECUTE	The process may execute the associated file.
ACL_DELETE_CHILD	
ACL_READ_ATTRIBUTES	
ACL_WRITE_ATTRIBUTES	
ACL_DELETE	
ACL_READ_ACL	
ACL_WRITE_ACL	
ACL_WRITE_OWNER	
ACL_SYNCHRONIZE	Ignored.

acl_entry_type_t ae_entry_type

This field defines the type of NFSv4 ACL entry. It is not used with POSIX.1e ACLs. The following values are valid:

ACL_ENTRY_TYPE_ALLOW

ACL_ENTRY_TYPE_DENY

acl_flag_t ae_flags

This field defines the inheritance flags of NFSv4 ACL entry. It is not used with POSIX.1e ACLs. The following values are valid:

ACL_ENTRY_FILE_INHERIT

ACL_ENTRY_DIRECTORY_INHERIT

ACL_ENTRY_NO_PROPAGATE_INHERIT

ACL_ENTRY_INHERIT_ONLY

ACL_ENTRY_INHERITED

The ACL_ENTRY_INHERITED flag is set on an ACE that has been inherited from its parent. It may also be set programmatically, and is valid on both files and directories.

SEE ALSO

acl(3), vaccess(9), vaccess_acl_nfs4(9), vaccess_acl_posix1e(9), VFS(9), VOP_ACLCHECK(9), VOP_GETACL(9), VOP_SETACL(9)

AUTHORS

This manual page was written by Robert Watson.