NAME

acl_calc_mask — calculate the file group class mask

LIBRARY

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

SYNOPSIS

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

int
acl_calc_mask(acl_t *acl_p);
```

DESCRIPTION

The **acl_calc_mask**() function calculates and sets the permissions associated with the ACL_MASK ACL entry of the ACL referred to by *acl_p*. The value of the new permissions is the union of the permissions granted by all entries of tag type ACL_GROUP, ACL_GROUP_OBJ, or ACL_USER. If the ACL referred to by *acl_p* already contains an ACL_MASK entry, its permissions are overwritten; if it does not contain an ACL_MASK entry, one is added.

If the ACL referred to by *acl_p* does not contain enough space for the new ACL entry, then additional working storage may be allocated. If the working storage cannot be increased in the current location, then it may be relocated and the previous working storage is released and a pointer to the new working storage is returned via *acl_p*.

The order of existing entries in the ACL is undefined after this function.

Any existing ACL entry descriptors that refer to entries in the ACL continue to refer to those entries. Any existing ACL pointers that refer to the ACL referred to by *acl_p* continue to refer to the ACL.

RETURN VALUE

The **acl_calc_mask**() 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_calc_mask**() function returns `-1` and sets *errno* to the corresponding value:

[EINVAL]	The argument *acl* is not a valid pointer to an ACL.
[ENOMEM]	The **acl_calc_mask**() function is unable to allocate the memory required for an ACL_MASK ACL entry.

STANDARDS

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

SEE ALSO

acl_check(3), *acl_get_entry*(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>.