NAME

getpeereid **-** get the effective credentials of a UNIX-domain peer

LIBRARY

Standard C Library (libc, -lc)

SYNOPSIS

#include <unistd.h>

int
getpeereid(*int s*, *uid_t *euid*, *gid_t *egid*);

DESCRIPTION

The **getpeereid**() function returns the effective user and group IDs of the peer connected to a UNIX-domain socket. The argument *s* must be a connected UNIX-domain socket (unix(4)) of type SOCK_STREAM on which either connect(2) or listen(2) has been called. The effective user ID is placed in *euid*, and the effective group ID in *egid*.

The credentials returned to the listen(2) caller are those of its peer at the time it called connect(2); the credentials returned to the connect(2) caller are those of its peer at the time it called listen(2). This mechanism is reliable; there is no way for either side to influence the credentials returned to its peer except by calling the appropriate system call (i.e., either connect(2) or listen(2)) under different effective credentials.

One common use of this routine is for a UNIX-domain server to verify the credentials of its client. Likewise, the client can verify the credentials of the server.

IMPLEMENTATION NOTES

On FreeBSD, **getpeereid**() is implemented in terms of the LOCAL_PEERCRED unix(4) socket option.

RETURN VALUES

The **getpeereid**() 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 **getpeereid**() function fails if:

[EBADF]	The argument *s* is not a valid descriptor.
[ENOTSOCK]	The argument *s* is a file, not a socket.

[ENOTCONN] The argument *s* does not refer to a socket on which connect(2) or listen(2) have been called.

[EINVAL] The argument *s* does not refer to a socket of type SOCK_STREAM, or the kernel returned invalid data.

SEE ALSO

connect(2), getpeername(2), getsockname(2), getsockopt(2), listen(2), unix(4)

HISTORY

The **getpeereid**() function appeared in FreeBSD 4.6.