NAME

rmdir – remove a directory

SYNOPSIS

#include <unistd.h>

int rmdir(const char **path****);**

MT-LEVEL

Async-Signal-Safe

DESCRIPTION

rmdir( ) removes the directory named by the path name pointed to by *path*. The directory must not have any entries other than "**.**" and "**..**".

If the directory's link count becomes zero and no process has the directory open, the space occupied by the directory is freed and the directory is no longer accessible. If one or more processes have the directory open when the last link is removed, the "**.**" and "**..**" entries, if present, are removed before **rmdir( )** returns and no new entries may be created in the directory, but the directory is not removed until all references to the directory have been closed.

Upon successful completion **rmdir( )** marks for update the **st_ctime** and **st_mtime** fields of the parent directory.

RETURN VALUES

Upon successful completion, a value of 0 is returned. Otherwise, a value of –1 is returned and **errno** is set to indicate the error.

ERRORS

The named directory is removed unless one or more of the following are true:

EACCES	Search permission is denied for a component of the path prefix.
EACCES	Write permission is denied on the directory containing the directory to be removed.
EACCES	The parent directory has the **S_ISVTX** variable set and is not owned by the user; the directory is not owned by the user and is not writable by the user; the user is not a super-user.
EBUSY	The directory to be removed is the mount point for a mounted file system.
EEXIST	The directory contains entries other than those for "**.**" and "**..**".
EFAULT	*path* points to an illegal address.
EINVAL	The directory to be removed is the current directory.
EINVAL	The final component of *path* is ''.''.
EIO	An I/O error occurred while accessing the file system.
ELOOP	Too many symbolic links were encountered in translating *path*.
EMULTIHOP	Components of *path* require hopping to multiple remote machines and the file system does not allow it.
ENAMETOOLONG	The length of the *path* argument exceeds {**PATH_MAX**}, or the length of a *path* component exceeds {**NAME_MAX**} while {**_POSIX_NO_TRUNC**} is in effect.

ENOENT	The named directory does not exist or is the null pathname.
ENOLINK	*path* points to a remote machine, and the connection to that machine is no longer active.
ENOTDIR	A component of the path prefix is not a directory.
EROFS	The directory entry to be removed is part of a read-only file system.

SEE ALSO **mkdir**(1), **rm**(1), **mkdir**(2)