Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'rmdir.3p' command

$ man rmdir.3p

RMDIR(3P) POSIX Programmer's Manual RMDIR(3P)

PROLOG

This manual page is part of the POSIX Programmer's Manual. The Linux implementation of this interface may differ (consult the corresponding Linux manual page for details of Linux behavior), or the interface may not be implemented on Linux.

NAME

rmdir ? remove a directory

SYNOPSIS

#include <unistd.h>

int rmdir(const char *path);

DESCRIPTION

The rmdir() function shall remove a directory whose name is given by path. The directory shall be removed only if it is an empty directory.

If the directory is the root directory or the current working directory of any process, it is unspecified whether the function succeeds, or whether it shall fail and set errno to [EBUSY].

If path names a symbolic link, then rmdir() shall fail and set errno to [ENOTDIR].

If the path argument refers to a path whose final component is either dot or dot-dot, rmdir() shall fail.

If the directory's link count becomes 0 and no process has the direc? tory open, the space occupied by the directory shall be freed and the

directory shall no longer be accessible. If one or more processes have the directory open when the last link is removed, the dot and dot-dot entries, if present, shall be removed before rmdir() returns and no new entries may be created in the directory, but the directory shall not be removed until all references to the directory are closed.

If the directory is not an empty directory, rmdir() shall fail and set errno to [EEXIST] or [ENOTEMPTY].

Upon successful completion, rmdir() shall mark for update the last data modification and last file status change timestamps of the parent di? rectory.

RETURN VALUE

Upon successful completion, the function rmdir() shall return 0. Other? wise, -1 shall be returned, and errno set to indicate the error. If -1 is returned, the named directory shall not be changed.

ERRORS

The rmdir() function shall fail if:

EACCES Search permission is denied on a component of the path prefix, or write permission is denied on the parent directory of the di? rectory to be removed.

EBUSY The directory to be removed is currently in use by the system or some process and the implementation considers this to be an er? ror.

[EEXIST] or [ENOTEMPTY]

The path argument names a directory that is not an empty directory, or there are hard links to the directory other than dot or a single entry in dot-dot.

EINVAL The path argument contains a last component that is dot.

EIO A physical I/O error has occurred.

ELOOP A loop exists in symbolic links encountered during resolu? tion of the path argument.

ENAMETOOLONG

The length of a component of a pathname is longer than {NAME_MAX}.

ENOENT A component of path does not name an existing file, or the path argument names a nonexistent directory or points to an empty string.

ENOTDIR A component of path names an existing file that is neither a directory nor a symbolic link to a directory.

[EPERM] or [EACCES]

The S_ISVTX flag is set on the directory containing the file referred to by the path argument and the process does not satisfy the criteria specified in the Base Definitions volume of POSIX.1?2017, Section 4.3, Directory Protection.

EROFS The directory entry to be removed resides on a read-only file system.

The rmdir() function may fail if:

ELOOP More than {SYMLOOP_MAX} symbolic links were encountered during resolution of the path argument.

ENAMETOOLONG

The length of a pathname exceeds {PATH_MAX}, or pathname resolu? tion of a symbolic link produced an intermediate result with a length that exceeds {PATH_MAX}.

The following sections are informative.

EXAMPLES

Removing a Directory

The following example shows how to remove a directory named /home/cnd/mod1.

```
#include <unistd.h>
int status;
...
status = rmdir("/home/cnd/mod1");
```

APPLICATION USAGE

None.

RATIONALE

The rmdir() and rename() functions originated in 4.2 BSD, and they used [ENOTEMPTY] for the condition when the directory to be removed does not

exist or new already exists. When the 1984 /usr/group standard was pub?
lished, it contained [EEXIST] instead. When these functions were
adopted into System V, the 1984 /usr/group standard was used as a ref?
erence. Therefore, several existing applications and implementations
support/use both forms, and no agreement could be reached on either
value. All implementations are required to supply both [EEXIST] and
[ENOTEMPTY] in <errno.h> with distinct values, so that applications can
use both values in C-language case statements.

The meaning of deleting pathname/dot is unclear, because the name of
the file (directory) in the parent directory to be removed is not
clear, particularly in the presence of multiple links to a directory.

The POSIX.1?1990 standard was silent with regard to the behavior of
rmdir() when there are multiple hard links to the directory being re?
moved. The requirement to set errno to [EEXIST] or [ENOTEMPTY] clari?
fies the behavior in this case.

If the current working directory of the process is being removed, that
should be an allowed error.

Virtually all existing implementations detect [ENOTEMPTY] or the case
of dot-dot. The text in Section 2.3, Error Numbers about returning any
one of the possible errors permits that behavior to continue. The
[ELOOP] error may be returned if more than {SYMLOOP_MAX} symbolic links
are encountered during resolution of the path argument.

FUTURE DIRECTIONS

None.

SEE ALSO

Section 2.3, Error Numbers, mkdir(), remove(), rename(), unlink()

The Base Definitions volume of POSIX.1?2017, Section 4.3, Directory
Protection, <unistd.h>

COPYRIGHT

Electrical and Electronics Engineers, Inc and The Open Group. In the event of any discrepancy between this version and the original IEEE and The Open Group Standard, the original IEEE and The Open Group Standard is the referee document. The original Standard can be obtained online at http://www.opengroup.org/unix/online.html .

Any typographical or formatting errors that appear in this page are most likely to have been introduced during the conversion of the source files to man page format. To report such errors, see https://www.ker?nel.org/doc/man-pages/reporting_bugs.html .