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 'fdetach.3p' command

$ man fdetach.3p

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

fdetach ? detach a name from a STREAMS-based file descriptor (STREAMS)

SYNOPSIS

#include <stropts.h>

int fdetach(const char *path);

DESCRIPTION

The fdetach() function shall detach a STREAMS-based file from the file to which it was attached by a previous call to fattach(). The path ar? gument points to the pathname of the attached STREAMS file. The process shall have appropriate privileges or be the owner of the file. A suc? cessful call to fdetach() shall cause all pathnames that named the at? tached STREAMS file to again name the file to which the STREAMS file was attached. All subsequent operations on path shall operate on the underlying file and not on the STREAMS file.

All open file descriptions established while the STREAMS file was at? tached to the file referenced by path shall still refer to the STREAMS file after the fdetach() has taken effect.

If there are no open file descriptors or other references to the STREAMS file, then a successful call to fdetach() shall be equivalent to performing the last close() on the attached file.

RETURN VALUE

Upon successful completion, fdetach() shall return 0; otherwise, it shall return -1 and set errno to indicate the error.

ERRORS

The fdetach() function shall fail if:

EACCES Search permission is denied on a component of the path prefix.

EINVAL The path argument names a file that is not currently attached.

ELOOP A loop exists in symbolic links encountered during resolution 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 path is an empty string.

ENOTDIR

A component of the path prefix names an existing file that is neither a directory nor a symbolic link to a directory, or the path argument contains at least one non-<slash> character and ends with one or more trailing <slash> characters and the last pathname component names an existing file that is neither a di? rectory nor a symbolic link to a directory.

EPERM The effective user ID is not the owner of path and the process does not have appropriate privileges.

The fdetach() 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

Detaching a File

The following example detaches the STREAMS-based file /tmp/named-STREAM from the file to which it was attached by a previous, successful call to fattach(). Subsequent calls to open this file refer to the underly? ing file, not to the STREAMS file.

```
#include <stropts.h>
...
    char *pathname = "/tmp/named-STREAM";
    int ret;
    ret = fdetach(pathname);
```

APPLICATION USAGE

None.

RATIONALE

None.

FUTURE DIRECTIONS

The fdetach() function may be removed in a future version.

SEE ALSO

fattach()

The Base Definitions volume of POSIX.1?2017, <stropts.h>

COPYRIGHT

files to man page format. To report such errors, see https://www.ker?

nel.org/doc/man-pages/reporting_bugs.html .