NAME

fexecve - execute program specified via file descriptor

SYNOPSIS

#include <unistd.h>

int fexecve(int *fd*, **char *const** *argv*[], **char *const** *envp*[]);

Feature Test Macro Requirements for glibc (see feature_test_macros(7)):

fexecve():
    Since glibc 2.10:
        _XOPEN_SOURCE >= 700 || _POSIX_C_SOURCE >= 200809L
    Before glibc 2.10:
        _GNU_SOURCE

DESCRIPTION

fexecve() performs the same task as execve(2), with the difference that the file to be executed is specified via a file descriptor, *fd*, rather than via a pathname. The file descriptor *fd* must be opened read-only, and the caller must have permission to execute the file that it refers to.

RETURN VALUE

A successful call to **fexecve**() never returns. On error, the function does return, with a result value of -1, and *errno* is set appropriately.

ERRORS

Errors are as for execve(2), with the following additions:

EINVAL
fd is not a valid file descriptor, or *argv* is NULL, or *envp* is NULL.

ENOSYS
The */proc* filesystem could not be accessed.

VERSIONS

fexecve() is implemented since glibc 2.3.2.

CONFORMING TO

POSIX.1-2008. This function is not specified in POSIX.1-2001, and is not widely available on other systems. It is specified in POSIX.1-2008.

NOTES

On Linux, **fexecve**() is implemented using the proc(5) filesystem, so */proc* needs to be mounted and available at the time of the call.

If *fd* is a file descriptor that refers to an interpreter script and has been marked as close-on-exec (see the discussion of the **FD_CLOEXEC** in fcntl(2)), **fexecve**() will fail to execute the script, since, by the time the script interpreter tries to access the script file, *fd* has already been closed.

The idea behind **fexecve**() is to allow the caller to verify (checksum) the contents of an executable before executing it. Simply opening the file, checksumming the contents, and then doing an execve(2) would not suffice, since, between the two steps, the filename, or a directory prefix of the pathname, could have been exchanged (by, for example, modifying the target of a symbolic link). **fexecve**() does not mitigate the problem that the *contents* of a file could be changed between the checksumming and the call to **fexecve**(); for that, the solution is to ensure that the permissions on the file prevent it from being modified by malicious users.

SEE ALSO

execve(2)

COLOPHON

This page is part of release 3.74 of the Linux *man-pages* project. A description of the project, information about reporting bugs, and the latest version of this page, can be found at http://www.kernel.org/doc/man-pages/.