NAME

chdir, fchdir - change working directory

SYNOPSIS

#include <unistd.h>

int chdir(const char *_path_);
int fchdir(int _fd_);

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

fchdir():
_BSD_SOURCE || _XOPEN_SOURCE >= 500 ||
_XOPEN_SOURCE && _XOPEN_SOURCE_EXTENDED
|| /* Since glibc 2.12: */ _POSIX_C_SOURCE >= 200809L

DESCRIPTION

chdir() changes the current working directory of the calling process to the directory specified in *path*.

fchdir() is identical to **chdir**(); the only difference is that the directory is given as an open file descriptor.

RETURN VALUE

On success, zero is returned. On error, -1 is returned, and *errno* is set appropriately.

ERRORS

Depending on the filesystem, other errors can be returned. The more general errors for **chdir**() are listed below:

EACCES
Search permission is denied for one of the components of *path*. (See also path_resolution(7).)

EFAULT
path points outside your accessible address space.

EIO An I/O error occurred.

ELOOP
Too many symbolic links were encountered in resolving *path*.

ENAMETOOLONG
path is too long.

ENOENT
The file does not exist.

ENOMEM
Insufficient kernel memory was available.

ENOTDIR
A component of *path* is not a directory.

The general errors for **fchdir**() are listed below:

EACCES
Search permission was denied on the directory open on *fd*.

EBADF
fd is not a valid file descriptor.

CONFORMING TO

SVr4, 4.4BSD, POSIX.1-2001.

NOTES

The current working directory is the starting point for interpreting relative pathnames (those not starting with /).

A child process created via fork(2) inherits its parent's current working directory. The current working directory is left unchanged by execve(2).

SEE ALSO

chroot(2), getcwd(3), path_resolution(7)

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/.