NAME

chroot, fchroot – change root directory

SYNOPSIS

#include <unistd.h>

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

int fchroot(int *fildes***);**

DESCRIPTION

chroot( ) and **fchroot( )** cause a directory to become the root directory, the starting point for path searches for path names beginning with **/**. The user's working directory is unaffected by the **chroot( )** and **fchroot( )** functions.

path points to a path name naming a directory. The *fildes* argument to **fchroot( )** is the open file descriptor of the directory which is to become the root.

The effective user ID of the process must be super-user to change the root directory. **fchroot( )** is further restricted in that while it is always possible to change to the system root using this call, it is not guaranteed to succeed in any other case, even should *fildes* be valid in all respects.

The ''**..**'' entry in the root directory is interpreted to mean the root directory itself. Thus, ''**..**'' cannot be used to access files outside the subtree rooted at the root directory. Instead, **fchroot( )** can be used to set the root back to a directory which was opened before the root directory was changed.

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

chroot( ) will fail and the root directory will remain unchanged if one or more of the following are true:

EACCES	Search permission is denied for a component of the path prefix of *dirname*. Search permission is denied for the directory referred to by *dirname*.
EBADF	The descriptor is not valid.
EFAULT	*path* points to an illegal address.
EINVAL	**fchroot( )** attempted to change to a directory which is not the system root and external circumstances do not allow this.
EINTR	A signal was caught during the **chroot( )** function.
EIO	An I/O error occurred while reading from or writing to the file system.
ELOOP	Too many symbolic links were encountered in translating *path*.
EMULTIHOP	Components of *path* require hopping to multiple remote machines and file system type 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 a null pathname.

ENOLINK *path* points to a remote machine and the link to that machine is no longer active.

ENOTDIR Any component of the path name is not a directory.

EPERM The effective user of the calling process is not super-user.

SEE ALSO

chroot(1M), **chdir**(2)

WARNINGS

The only use of **fchroot( )** that is appropriate is to change back to the system root.