

**NAME**

chroot - change root directory

**SYNOPSIS**

```
#include <unistd.h>
```

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

Feature Test Macro Requirements for glibc (see [feature\\_test\\_macros\(7\)](#)):

**chroot()**:

Since glibc 2.2.2:

```
BSD_SOURCE ||  
(_XOPEN_SOURCE >= 500 ||  
_XOPEN_SOURCE && _XOPEN_SOURCE_EXTENDED) &&  
!( _POSIX_C_SOURCE >= 200112L || _XOPEN_SOURCE >= 600)
```

Before glibc 2.2.2: none

**DESCRIPTION**

**chroot()** changes the root directory of the calling process to that specified in *path*. This directory will be used for pathnames beginning with */*. The root directory is inherited by all children of the calling process.

Only a privileged process (Linux: one with the **CAP\_SYS\_CHROOT** capability) may call **chroot()**.

This call changes an ingredient in the pathname resolution process and does nothing else.

This call does not change the current working directory, so that after the call *.* can be outside the tree rooted at */*. In particular, the superuser can escape from a chroot jail by doing:

```
mkdir foo; chroot foo; cd ..
```

This call does not close open file descriptors, and such file descriptors may allow access to files outside the chroot tree.

**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 are listed below:

**EACCES**

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

**EPERM**

The caller has insufficient privilege.

**CONFORMING TO**

SVr4, 4.4BSD, SUSv2 (marked LEGACY). This function is not part of POSIX.1-2001.

**NOTES**

A child process created via [fork\(2\)](#) inherits its parent's root directory. The root directory is left unchanged by [execve\(2\)](#).

FreeBSD has a stronger [jail\(\)](#) system call.

**SEE ALSO**

[chdir\(2\)](#), [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/>.