NAME

fpathconf, pathconf – get configurable pathname variables

SYNOPSIS

#include <unistd.h>

long fpathconf(int *fildes*, **int** *name***);**

long pathconf(const char **path*, **int** *name***);**

MT-LEVEL

pathconf( ) is Async-Signal-Safe

DESCRIPTION

The functions **fpathconf( )** and **pathconf( )** return the current value of a configurable limit or option associated with a file or directory. The *path* argument points to the pathname of a file or directory; *fildes* is an open file descriptor; and *name* is the symbolic constant (defined in **<unistd.h>**) representing the configurable system limit or option to be returned.

The values returned by **pathconf( )** and **fpathconf( )** depend on the type of file specified by *path* or *fildes*. The following table contains the symbolic constants supported by **pathconf( )** and **fpathconf( )** along with the POSIX defined return value. The return value is based on the type of file specified by *path* or *fildes*.

Value of *name*	See Note
_PC_LINK_MAX	1
_PC_MAX_CANNON	2
_PC_MAX_INPUT	2
_PC_NAME_MAX	3,4
_PC_PATH_MAX	4,5
_PC_PIPE_BUF	6
_PC_CHOWN_RESTRICTED	7
_PC_NO_TRUNC	3,4
_PC_VDISABLE	2
_PC_ASYNC_IO	2
_PC_PRIO_IO	2
_PC_SYNC_IO	1

Notes:

1 If *path* or *fildes* refers to a directory, the value returned applies to the directory itself.

2 The behavior is undefined if *path* or *fildes* does not refer to a terminal file.

3 If *path* or *fildes* refers to a directory, the value returned applies to the filenames within the directory.

4 The behavior is undefined if *path* or *fildes* does not refer to a directory.

5 If *path* or *fildes* refers to a directory, the value returned is the maximum length of a relative pathname when the specified directory is the working directory.

6 If *path* or *fildes* refers to a pipe or FIFO, the value returned applies to the pipe or FIFO. If *path* or *fildes* refers to a directory, the value returned applies to any FIFOs that exist or can be created within the directory. If *path* or *fildes* refer to any other type of file, the behavior is undefined.

7 If *path* or *fildes* refers to a directory, the value returned applies to any files, other than directories, that exist or can be created within the directory.

The value of the configurable system limit or option specified by *name* does not change during the lifetime of the calling process.

RETURN VALUES

If **fpathconf** or **pathconf** are invoked with an invalid symbolic constant or the symbolic constant corresponds to a configurable system limit or option not supported on the system, a value of –1 is returned to the invoking process. If the function fails because the configurable system limit or option corresponding to *name* is not supported on the system the value of **errno** is not changed.

ERRORS

fpathconf( ) fails if the following is true:

EBADF *fildes* is not a valid file descriptor.

pathconf( ) fails if one or more of the following are true:

EACCES search permission is denied for a component of the path prefix.

ELOOP too many symbolic links are encountered while translating *path*.

EMULTIHOP components of *path* require hopping to multiple remote machines and file system type does not allow it.

ENAMETOOLONG the length of a pathname exceeds **{PATH_MAX}**, or a pathname component is longer than **{NAME_MAX}** while {**_POSIX_NO_TRUNC**} is in effect.

ENOENT *path* is needed for the command specified and the named file does not exist or if the *path* argument points to an empty string.

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

ENOTDIR a component of the path prefix is not a directory.

Both **fpathconf( )** and **pathconf( )** fail if the following is true:

EINVAL if *name* is an invalid value.

SEE ALSO

sysconf(3C), **limits**(4)