NAME

pathfind – search for named file in named directories

SYNOPSIS

cc [*flag* ...] *file* ... **–lgen** [*library* ...]

#include <libgen.h>

char ∗**pathfind(const char** ∗*path*, **const char** ∗*name*, **const char** ∗*mode***);**

MT-LEVEL

MT-Safe

DESCRIPTION

pathfind( ) searches the directories named in *path* for the file *name*. The directories named in *path* are separated by semicolons. *mode* is a string of option letters chosen from the set [**rwxfbcdpugks**]:

Letter	**Meaning**
r	readable
w	writable
x	executable
f	normal file
b	block special
c	character special
d	directory
p	FIFO (pipe)
u	set user ID bit
g	set group ID bit
k	sticky bit
s	size nonzero

Options read, write, and execute are checked relative to the real (not the effective) user ID and group ID of the current process.

If the file *name*, with all the characteristics specified by *mode*, is found in any of the directories specified by *path*, then **pathfind( )** returns a pointer to a string containing the member of *path*, followed by a slash character (**/**), followed by *name*.

If *name* begins with a slash, it is treated as an absolute path name, and *path* is ignored.

An empty *path* member is treated as the current directory. **/** is not prepended at the occurrence of the first match; rather, the unadorned *name* is returned.

EXAMPLES

To find the **ls** command using the **PATH** environment variable:

pathfind (getenv ("PATH"), "ls", "rx")

RETURN VALUES

If no match is found, **pathname** returns a null pointer, **((char *) 0)**.

SEE ALSO

sh(1), **test**(1), **access**(2), **mknod**(2), **stat**(2), **getenv**(3C)

NOTES

The string pointed to by the returned pointer is stored in an area that is reused on subsequent calls to **pathfind( )**. The string should not be deallocated by the caller.

When compiling multi-thread applications, the **_REENTRANT** flag must be defined on the compile line. This flag should only be used in multi-thread applications.