

**NAME**

**grantpt**, **ptsname**, **ptsname\_r**, **unlockpt** - pseudo-terminal access functions

**LIBRARY**

Standard C Library (libc, -lc)

**SYNOPSIS**

```
#include <stdlib.h>
```

*int*

```
grantpt(int fildes);
```

*char \**

```
ptsname(int fildes);
```

*int*

```
ptsname_r(int fildes, char *buffer, size_t buflen);
```

*int*

```
unlockpt(int fildes);
```

**DESCRIPTION**

The **grantpt**(), **ptsname**(), and **unlockpt**() functions allow access to pseudo-terminal devices. These three functions accept a file descriptor that references the master half of a pseudo-terminal pair. This file descriptor is created with **posix\_openpt**(2).

The **grantpt**() function is used to establish ownership and permissions of the slave device counterpart to the master device specified with *fildes*. The slave device's ownership is set to the real user ID of the calling process, and the permissions are set to user readable-writable and group writable. The group owner of the slave device is also set to the group "tty".

The **ptsname**() function returns the full pathname of the slave device counterpart to the master device specified with *fildes*. This value can be used to subsequently open the appropriate slave after **posix\_openpt**(2) and **grantpt**() have been called.

The **ptsname\_r**() function is the thread-safe version of **ptsname**(). The caller must provide storage for the results of the full pathname of the slave device in the *buffer* and *bufsize* arguments.

The **unlockpt**() function clears the lock held on the pseudo-terminal pair for the master device specified with *fildes*.

## RETURN VALUES

The **grantpt()**, **ptsname\_r()**, and **unlockpt()** functions return the value 0 if successful; otherwise the value -1 is returned and the global variable *errno* is set to indicate the error.

The **ptsname()** function returns a pointer to the name of the slave device on success; otherwise a NULL pointer is returned.

## ERRORS

The **grantpt()**, **ptsname()**, **ptsname\_r()** and **unlockpt()** functions may fail and set *errno* to:

[EBADF]                *fildev* is not a valid open file descriptor.

[EINVAL]              *fildev* is not a master pseudo-terminal device.

In addition, the **ptsname\_r()** function may set *errno* to:

[ERANGE]              The buffer was too small.

In addition, the **grantpt()** function may set *errno* to:

[EACCES]              The slave pseudo-terminal device could not be accessed.

## SEE ALSO

posix\_openpt(2), pts(4), tty(4)

## STANDARDS

The **ptsname()** function conforms to IEEE Std 1003.1-2008 ("POSIX.1").

This implementation of **grantpt()** and **unlockpt()** does not conform to IEEE Std 1003.1-2008 ("POSIX.1"), because it depends on `posix_openpt(2)` to create the pseudo-terminal device with proper permissions in place. It only validates whether *fildev* is a valid pseudo-terminal master device. Future revisions of the specification will likely allow this behaviour, as stated by the Austin Group.

## HISTORY

The **grantpt()**, **ptsname()** and **unlockpt()** functions appeared in FreeBSD 5.0.